Journal of *Marine Science and Engineering*

MDPI

Article

A COLREGs-Compliant Collision Avoidance Decision Approach Based on Deep Reinforcement Learning

Weiqiang Wang [1], Liwen Huang [1,2,3], Kezhong Liu [1,2,3,*], Xiaolie Wu [1] and Jingyao Wang [1]

[1] School of Navigation, Wuhan University of Technology, Wuhan 430063, China; weiqiangwang@whut.edu.cn (W.W.); lwhuang@whut.edu.cn (L.H.); xlwu@whut.edu.cn (X.W.); jywang@whut.edu.cn (J.W.)
[2] Hubei Key Laboratory of Inland Shipping Technology, Wuhan University of Technology, Wuhan 430063, China
[3] National Engineering Research Center for Water Transport Safety, Wuhan University of Technology, Wuhan 430063, China
* Correspondence: kzliu@whut.edu.cn

Abstract: It is crucial to develop a COLREGs-compliant intelligent collision avoidance system for the safety of unmanned ships during navigation. This paper proposes a collision avoidance decision approach based on the deep reinforcement learning method. A modified collision avoidance framework is developed that takes into consideration the characteristics of different encounter scenarios. Hierarchical reward functions are established to assign reward values to constrain the behavior of the agent. The collision avoidance actions of the agent under different encounter situations are evaluated on the basis of the COLREGs to ensure ship safety and compliance during navigation. The deep Q network algorithm is introduced to train the proposed collision avoidance decision framework, while various simulation experiments are performed to validate the developed collision avoidance model. Results indicate that the proposed method can effectively perform tasks that help ships avoid collisions in different encounter scenarios. The proposed approach is a novel attempt for intelligent collision avoidance decisions of unmanned ships.

Keywords: collision avoidance; unmanned ships; deep reinforcement learning; COLREGs; reward functions

check for updates

Citation: Wang, W.; Huang, L.; Liu, K.; Wu, X.; Wang, J. A COLREGs-Compliant Collision Avoidance Decision Approach Based on Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 944. https://doi.org/10.3390/jmse10070944

Academic Editor: Spyros Hirdaris

Received: 17 June 2022
Accepted: 7 July 2022
Published: 9 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, great theoretical and technical achievements have been made in the field of unmanned ships [1,2], and a series of advanced unmanned vehicles have been used in many marine missions, for example, in environmental monitoring, marine transportation, coastal investigating, and remote sensing [3–5]. While ensuring navigation safety and avoiding maritime accidents have always been some of the most essential elements of unmanned ships, according to the official maritime accident reports, ship collision is the most frequent type of all maritime accidents [6,7], which might cause serious human casualties, massive damage to property, and environmental pollution. Therefore, it is important to develop an autonomous collision avoidance approach for unmanned ships that can work in various navigation scenarios [8].

To this end, many efficient methods have been developed. For example, the artificial potential field method [9,10], the velocity obstacle algorithm [11,12], the dynamic window method [13], and the heuristic algorithm [14,15] have been widely used in research on how ships can avoid any collision. In the meantime, the advancement of artificial intelligence technology, particularly reinforcement learning, provides a new possibility for ships to avoid a collision due to its obvious superiority in problems that can be solved via decision making [16–18]. In the past few years, some typical ship intelligent collision avoidance models based on reinforcement learning methods have been proposed [7,19,20].

To ensure the coordination of collision avoidance operations between ships, it is imperative that all ships participating in collision avoidance comply with the Convention on the International Regulations for Preventing Collisions at Sea (COLREGs) and good seamanship [21–23], which are also two significant factors that must be considered when designing intelligent collision avoidance models in real navigation situations. However, a vast number of existing collision avoidance studies with reinforcement learning methods focus on the optimization of models, lacking a comprehensive interpretation of the COLREGs and good seamanship [7,24,25]. More specifically, these studies have adopted a set of uniform reward functions to evaluate action in various encounter situations, without considering that different scenarios may require considering different aspects of COLREGs. In addition, since there is a lack of sufficient analysis of characteristics in different encounter scenes, the input states of these models have a lot of redundant information [7,20], which not only affects the efficiency of the decision making but also hinders the differential training of the network models. Furthermore, unmanned ships are generally under-actuated systems with huge inertia and relatively weak power, making unmanned ships extremely slow to respond to the maneuver. Therefore, hydrodynamic characteristics of unmanned ships are another vital element that must be considered when designing intelligent collision avoidance models.

In this paper, an intelligent collision avoidance model based on a deep reinforcement learning method considering both the constraints according to the COLREGs and the hydrodynamic characteristics of ships is developed. Particularly, an efficient algorithm called deep Q network (DQN) is adopted due to its superiority in rapid convergence and stability. In the proposed model, ship encounter scenarios are divided into different types and every encounter type is matched with a specific combination of reward functions to evaluate collision avoidance actions in terms of performance. Moreover, every encounter type has a corresponding definition method for state space and network structure, which reduces the redundant information and lays down a foundation for differential decision making in various ship encounter scenarios. The proposed model is capable of making collision avoidance schemes for different encounter scenarios taking into consideration the corresponding rules of COLREGs.

The remaining part of this paper is organized as follows. Section 2 provides a brief overview of the existing collision avoidance methods and the development of reinforcement learning. In Section 3, the deep-reinforcement-learning-based collision avoidance approach is introduced in detail, including the definition of state space and action space, the design of reward functions, and the training process of the algorithm. Several simulation experiments and results analyses are presented in Section 4. Section 5 summarizes the conclusion and lays down the future path of research on this topic.

2. Literature Review

2.1. Intelligent Collision Avoidance Methods

Recently, numerous studies have been carried out on detecting imminent collisions and decision support using various approaches. One of the most commonly used methods is the traditional approach, based on geometric models and mathematical calculations, such as the artificial potential field (APF) method and the velocity obstacle (VO) method. The main principle of the APF method is that a strong repulsive force will be applied to the target ship when it enters potential fields of other vessels so that the target ship would be forced away and a collision will be avoided [26]. For instance, [27] proposed a real-time collision avoidance method for complex encounter situations, which combines a modified repulsion potential field function with the corresponding virtual forces. In [28], a collision cone with a risk detection function in the control model is introduced and a dynamic collision avoidance algorithm based on the layered artificial potential field is presented. The VO method is another typical traditional approach for avoiding collisions, which avoids multiple obstacles by calculating a set of velocities that may lead to a collision at a certain time in the future [6]. This method was first used in robot control, and recently researchers applied it to ships

for avoiding collisions. In [11,29], the VO algorithm is used to make collision avoidance decisions when the velocities of ships are non-linear and predictable. By incorporating the danger degree of the approaching vessels and the avoidance ability of a vessel, [30] proposed a time-varying collision risk measurement for precaution against collisions. The graphic method is also one of the typical traditional methods. In order to consider the maneuverability and hydrometeorological conditions of the ship, [31] proposed a path planning method based on the interpolation of the ship's state vector according to the data from measurements conducted during the sea trials of the ship. On the basis of geometric analysis, [32,33] proposed a distributed anti-collision decision support formulation for multi-ship-encounter situations. This distributed collision avoidance idea has also been widely used in the field of robot collision avoidance. For example, [34] formulate the robot control as multi-step matrix game and optimized the collision avoidance task by dual linear programming. In terms of collision risk analysis, [35,36] proposed a real-time collision risk assessment method. In [37,38], collision risk factors and traffic complexity in some special navigational waters, such as polar regions and inland rivers, was analyzed. However, due to the dependence on complicated mathematical formulas, these traditional methods are sensitive to minor environmental disturbances [39]. Thus, a slight change in parameters may lead to a failure to avoid a collision.

The heuristic algorithm is another representative method that may help ships to avoid a collision. It formulates the collision avoidance problem as a multi-objective optimization problem, while the solution represents the feasible avoidance operation. For example, [14] adopted the ant colony algorithm to construct a collision avoidance decision model that combines the COLREGs, good seamanship, and real-time dynamic data from AIS. In [40], the authors explored the application of the genetic algorithm in the field of ship-collision avoidance and proposed a path planning method that enables providing the theoretical shortest route taking into account both safety and economy. For multiple unmanned surface vehicles, [41] developed a cooperative collision avoidance method based on an improved genetic algorithm, in which retention, deletion, and replacement were applied and a fitness construction method based on an analytic hierarchy process was proposed. By combining fuzzy logic and genetic algorithm, [42] designed a collision avoidance decision support system. In addition, due to fewer model parameters, a simple structure, and a fast convergence speed, particle swarm optimization has always been commonly used to solve multi-objective optimization problems. In [43], the authors proposed a COLREGs-compliant path planning method based on particle swarm optimization. Then, [15] introduced a hierarchical sorting rule and presented a hierarchical multi-objective particle swarm optimization algorithm to avoid collisions [15]. However, the heuristic algorithm generally relies on a large number of iterations to provide a solution, greatly increasing the calculation cost and decision latencies. As a result, in practical encounter scenarios, it is a challenge to guarantee the real-time performance of collision avoidance decisions based on these methods.

2.2. Deep Reinforcement Learning

In recent years, artificial intelligence technology has made great progress and the development of reinforcement learning methods provides a new means for intelligent collision avoidance. Reinforcement learning (RL) is a trial-and-error algorithm that uses the mechanism of rewards and punishments to complete both the agent's learning from the environment and behavior mapping. In 1957, the Markov decision process was presented, widely regarded as the foundation of reinforcement learning. In 1989, Watkins constructed and updated the Q table and proposed the Q-learning algorithm, which has been one of the most frequently used reinforcement learning algorithms until now. However, due to space limitation in the Q table, the Q-learning algorithm is only suitable for dealing with the problem with discontinuous states. Therefore, in [44], DeepMind proposed a deep reinforcement learning (DRL) method based on function approximation. In this research, the neural network replaces the Q table and the problem of space explosion is

completely solved. Owing to the excellent self-learning ability, the reinforcement learning method was widely used to solve complex sequential optimization decision problems [7]. Moreover, the reinforcement learning method can understand and interpret the unknown environment, endowing it with great potential to address the problem of avoiding collisions in complex encounter scenarios. Compared with the existing collision avoidance methods summarized above, the reinforcement learning method is significantly superior in terms of the anti-interference capability and the speed of decision making.

Several studies have been conducted using reinforcement learning methods to create strategies on making decisions that will help avoid collisions. For example, [45,46] adopted the Q-learning algorithm for collision avoidance in a scenario of multi-static obstacles, where the dynamic characteristics of a cargo ship and the water restriction were considered. However, these studies have only focused on static obstacles, not taking into consideration uncertain environmental disturbances. More recently, [19] presented an intelligent method for multi-ship-collision avoidance by combining the deep Q-learning algorithm with expert knowledge. By considering the COLREGs and various encounter scenarios, a series of improved anti-collision decision formulations have been developed based on the DRL algorithm [7,20,47]. In [48], the authors extended the DRL method for shape maintenance and collision avoidance for USV formations in complex navigation conditions. Based on the encounter situations classified by the COLREGs, [49] constructed 19 single-vessel collision avoidance scenarios as training sets for the agent, and a double deep Q network (DDQN) algorithm was introduced to complete the training of the decision model. In addition, since the image tends to contain a lot of information that cannot be simply described by parameters, [24,25] proposed a novel collision avoidance approach by introducing a convolutional neural network in which, instead of a few parameterized indexes, real-time encounter images were the input state of the collision avoidance model. However, most of these studies have adopted a unified set of reward functions to train a single decision network for collision avoidance and lack a comprehensive consideration of the different aspects of the COLREGs for specific encounter situations when designing the reward functions. Meanwhile, it is difficult to ensure the compliance with the COLREGs of the collision avoidance schemes formulated by a single decision network for different encounter scenarios. Thus, the anti-collision decision models based on the DRL method still have a lot of room for improvement.

From the above research analysis, it can be concluded that due to its strong self-learning ability, the deep reinforcement learning method has become one of the new choices for collision avoidance research. To address the existing issues in relevant studies, we develop a novel decision model based on the deep reinforcement learning method by categorizing different encounter types. Meanwhile, a hierarchical combination of reward functions is designed that combines navigation safety, the COLREGs, and good seamanship. In addition, a new network construction, training, and decision-making framework for collision avoidance schemes is proposed. This study will lay a solid foundation for the practical application of the reinforcement learning method in the maneuvering of unmanned ships.

3. Method for Ship-Collision Avoidance Based on DRL

3.1. Motion Model of the Ship

3.1.1. Coordinate System

To describe the movement of a ship, a ship motion coordinate system is established, as shown in Figure 1. In this figure, the coordinate system **XOY** is fixed to the earth, while the system **xoy** is fixed to the ship. The origin of the coordinate system **xoy** is **o**, which is also the center of gravity of the ship; **O** is the origin of the coordinate system **XOY**; X_0 and Y_0 are the projections of the center of gravity of the ship on the **X** and **Y** axes, respectively; ψ indicates the course of the ship; and δ represents the rudder angle of the ship.

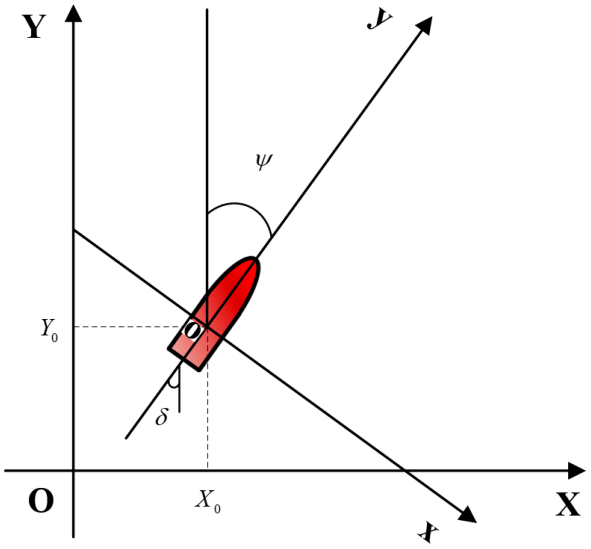

Figure 1. Coordinate system.

The position and speed of the ship in **XOY** and **xoy** can be converted into each other by Equations (1) and (2):

$$\begin{cases} [x, y] = [X - X_0,\ Y - Y_0] \times \boldsymbol{A} \\ [\dot{x}, \dot{y}] = [\dot{X}, \dot{Y}] \times \boldsymbol{A} \end{cases} \tag{1}$$

$$\begin{cases} [X, Y] = [x, y] \times \boldsymbol{A^{-1}} + [X_0, Y_0] \\ [\dot{X}, \dot{Y}] = [\dot{x}, \dot{y}] \times \boldsymbol{A^{-1}} \end{cases} \tag{2}$$

$$\boldsymbol{A} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \tag{3}$$

where $\boldsymbol{A}$ is the conversion matrix, $[X, Y]$ and $[x, y]$ are the coordinates of the ship in systems **XOY** and **xoy**, respectively, and $[X_0, Y_0]$ is the coordinate of the origin **o** of the coordinate system **xoy** in system **XOY**.

3.1.2. Motion Model

During sailing, a ship is affected by huge hydrodynamics. To obtain the accurate trajectory of the ship, it is necessary to consider the maneuverability of the ship. Because the vertical movement (heave, roll, and pitch) of the ship is negligible [50], this paper mainly studies the movement in the horizontal dimension (surge, sway, and yaw). In the **xoy** system, the surge velocity is $\boldsymbol{v}$, the sway velocity is $\boldsymbol{u}$, and the yaw velocity is $\boldsymbol{r}$; the motion model can be expressed as Equation (4) by using the MMG model:

$$\begin{cases} (m + m_y)\dot{v} - (m + m_x)ur = Y_H + Y_P + Y_R \\ (m + m_x)\dot{u} - (m + m_y)vr = X_H + X_P + X_R \\ (I_{ZZ} + J_{ZZ})\dot{r} = N_H + N_R \end{cases} \tag{4}$$

where m is the mass of the hull and m_x and m_y are the added mass along the x-axis and the added mass along the y-axis, respectively. X_H, Y_H, X_P, Y_P, X_R, and Y_R are the external forces along the x-axis and the y-axis of the hull, the propeller, and the rudder. I_{ZZ} and J_{ZZ} are the moments of inertia around the z-axis. N_H and N_R are the yaw moment around the z-axis of the hull and the rudder.

The model expresses the correspondence between the rudder angle and various motion characteristics of the ship. The input of the model is the rudder angle, and the output is the real-time ship motion parameters (surge velocity $\boldsymbol{v}$, sway velocity $\boldsymbol{u}$, and yaw velocity $\boldsymbol{r}$).

Therefore, the position in the **xoy** coordinate system and the course of the ship at any moment can be calculated using Equation (5), where $x(0)$, $y(0)$, and $\psi(0)$ are the initial position in the **xoy** coordinate system and course of the ship. Furthermore, the real-time position, course, and velocity of the ship in the **XOY** coordinate system can be calculated by combining Equations (2), (4) and (5):

$$\begin{cases} x(t) = x(0) + \int u dt \\ y(t) = y(0) + \int v dt \\ \psi(t) = \psi(0) + \int r dt \end{cases} \tag{5}$$

It should be noted that some hydrodynamic calculation modules in the MMG model are still in the process of continuous research and improvement. The accuracy of the model is also lower than that of the integral model (such as the Abkowitz model). However, considering that the research focus of this paper is the collision avoidance framework based on deep reinforcement learning, we choose a relatively simple model to simulate the ship motion. If higher trajectory prediction accuracy is needed, a more accurate ship motion model should be selected.

3.2. DRL Method for Ship-Collision Avoidance

Through the motion model, the current state of the ship can be linked to its next state in real time. Therefore, the ship-collision avoidance problem can be defined as a sequential decision-making problem, which can be modeled by the Markov Decision Process (MDP).

As illustrated in Figure 2, the ship (agent) departs from the initial state s_0 and then selects an action $a_0 \in A(s_0)$ that can maximize the future return $G = \sum_{k=0}^{\infty} \gamma^k r_{k+1}$ following the policy $\pi_\theta(a|s)$. $A(s_0)$ is the set of actions available in the state s_0, $\pi_\theta(a|s)$ represents the probability that $a_t = a$ if $s_t = s$, and $\gamma \in [0, 1]$ is the discount rate. The policy used here is an ε-greedy policy, which balances "exploitation" and "exploration." Exploitation means to select an action with the maximal value function, and exploration refers to attempting a possible action randomly, which can prevent the algorithm from falling into a local optimum. The agent will first select and then perform the action to reach the next state s_1 and obtain a reward r_1 from the environment. The parameter θ in the policy $\pi_\theta(a|s)$ will be updated according to the reward value. The agent will continue to perform the above process until it reaches the end state s_n. Through extensive interaction with the environment, the agent will obtain a target policy, which only selects the action with the maximal value function and does not explore other actions.

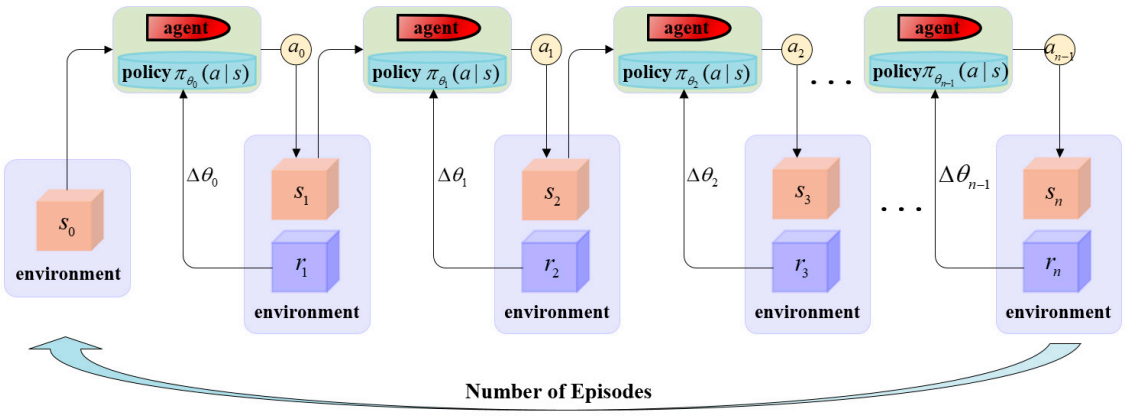

Figure 2. Schematic of the DRL method for ship-collision avoidance.

3.2.1. State Space

The state space is the observation of the environment, on the basis of which the agent selects its actions. According to the different encounter scenarios, three ways of defining the state space are proposed.

(1) Multi-ship encounter

According to the COLREGs, considering the different bearings of the target ship, two ships can face any of three types of encounter situations: overtaking, head-on, and crossing. For target ships from different directions, the responsibility of the own ship (the agent in this paper) can be specified by the illustration shown in Figure 3 [14,40,51]. It should be noted that the coordinate system used here is the xoy system, which is fixed to the agent ship, and the origin is the center of gravity of the agent ship; the detection range is set to 6 NM. Because the agent has different responsibilities in terms of avoiding target ships in different regions, when there are multiple target ships around the agent, we adopt the method proposed in [7] and select the ship closest to the agent in each region as the target for state input. The state S_{TM} can be expressed as Equation (6):

$$S_{TM} = [d_{Ti}, \beta_{Ti}, \psi_{Ti}, v_{Ti}] \tag{6}$$

where d_{Ti} are the distances between the agent and the target ships, β_{Ti} are the relative bearings of the target ships to the agent, ψ_{Ti} are the courses of the target ships, v_{Ti} are the velocities of the target ships, and i represents the index of the target ships. It is noteworthy that d_{Ti} and β_{Ti} are relative to the frame fixed to the agent, while ψ_{Ti} and v_{Ti} are relative to the frame fixed to the earth. Furthermore, when there is no target ship in one of the above four regions, the four elements in the state space of this region will be assigned a value of 0 so that this region will not affect the formulation of the final avoidance action.

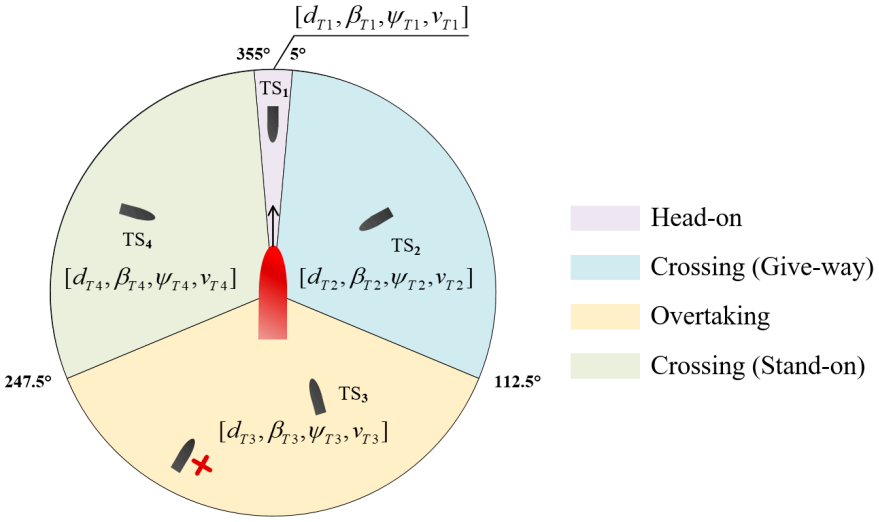

Figure 3. Target ship classification and its state space determination on the basis of the COLREGs.

Apart from dynamic target ships, the agent might also encounter static obstacles. There is no clear responsibility and requirement for the agent to avoid the static obstacles in different directions. Therefore, among all the obstacles that are at risk of colliding with the agent, we select the one closest to the agent to observe its state S_{OM}, as per Equation (7):

$$S_{OM} = [d_O, \beta_O] \tag{7}$$

where d_O is the relative distance between the agent and the obstacle and β_O is the relative bearing of the obstacle to the agent.

In addition to the above, the state of the agent itself S_{AM} and the state of the destination S_{DM} will also affect the choice of action. Therefore, S_{AM} and S_{DM} are also observed:

$$S_{AM} = [\psi_A, v_A] \tag{8}$$

$$S_{DM} = [d_D, \beta_D] \tag{9}$$

where ψ_A is the course of the agent, v_A is the velocity of the agent, d_D is the relative distance between the agent and the destination, and β_D is the relative bearing of the destination to the agent.

Consequently, the state space S_M of the multi-ship encounter scenario is a combination of S_{TM}, S_{OM}, S_{AM}, and S_{DM}, and it contains a total of 22 elements.

(2) Two-ship encounter

The state space in the two-ship encounter scenario is defined similarly to that in the multi-ship scenario. However, since static obstacles might affect the effectiveness of rules in the COLREGs, we will not consider static obstacles in the state space definition of the two-ship encounter scenario. Thus, the state space S_T of the two-ship encounter scenario is composed of three parts: the state of the target ship S_{TT}, the state of the agent S_{AT}, and the state of the destination S_{DT}. S_{TT} is defined according to Equation (10), while S_{AT} and S_{DT} are defined similarly to S_{AM} and S_{DM} in the multi-ship encounter scenario. The state space of this encounter scenario consists of eight elements. Note that, when there are both static obstacles and target ships in the encounter situation, the definition of state space should refer to the multi-ship encounter scenario:

$$S_{TT} = [d_T, \beta_T, \psi_T, v_T] \tag{10}$$

where d_T is the distance between the agent and the target ship, β_T is the relative bearing of the target ship to the agent, ψ_T is the course of the target ship, and v_T is the velocity of the target ship.

(3) Avoiding Static obstacles

In this scenario, the COLREGs do not put any constraint on the movements of the ship and the ship only needs to take effective avoidance action according to the information about the obstacle and the destination. Therefore, the state space S_S of this scenario is composed of three parts: the state of the obstacle S_{OS}, the state of the agent S_{AS}, and the state of the destination S_{DS}. These states are defined in the same way as introduced in the multi-ship encounter scenarios, and there are six elements in the state space.

(4) State space

In summary, the state space of the multi-ship encounter scene is composed of four parts, which includes 22 elements in total. The state spaces of the two-ship encounter scenario and the scenario involving only avoiding static obstacles are both composed of three parts, with eight and six elements, respectively. The definition of the state space is shown in Figure 4. Note that the courses and velocities involved in the state space are values relative to the coordinate system fixed to the earth, while the distances and bearings are values relative to the coordinate system fixed to the agent. Meanwhile, we assume that all states can be observed, and these states will be regarded as the input to the decision networks.

3.2.2. Action Space

When avoiding a collision, the operator can avoid obstacles by changing the course or speed. However, the ship is affected by huge inertia, thus changing the speed may not achieve instant results. Therefore, operators prefer to maintain the speed and only change the course to avoid a collision [50]. In this paper, we take the rudder angles as the action space since the ship can maintain or change its course under rudder control, and a discrete action space A that considers manipulation experiences is defined as Equation (11):

$$A = [-35^\circ, -20^\circ, -15^\circ, -10^\circ, -5^\circ, 0^\circ, 5^\circ, 10^\circ, 15^\circ, 20^\circ, 35^\circ] \tag{11}$$

where A is a vector containing 11 elements, ranging from -35° to 35°, each element representing a rudder angle that the agent can select. This design is largely in compliance with the navigation experience because operators tend to choose a hard port or a hard

starboard to avoid obstacles only when the situation is urgent but are more willing to choose a smaller and appropriate rudder angle in general encounter scenarios.

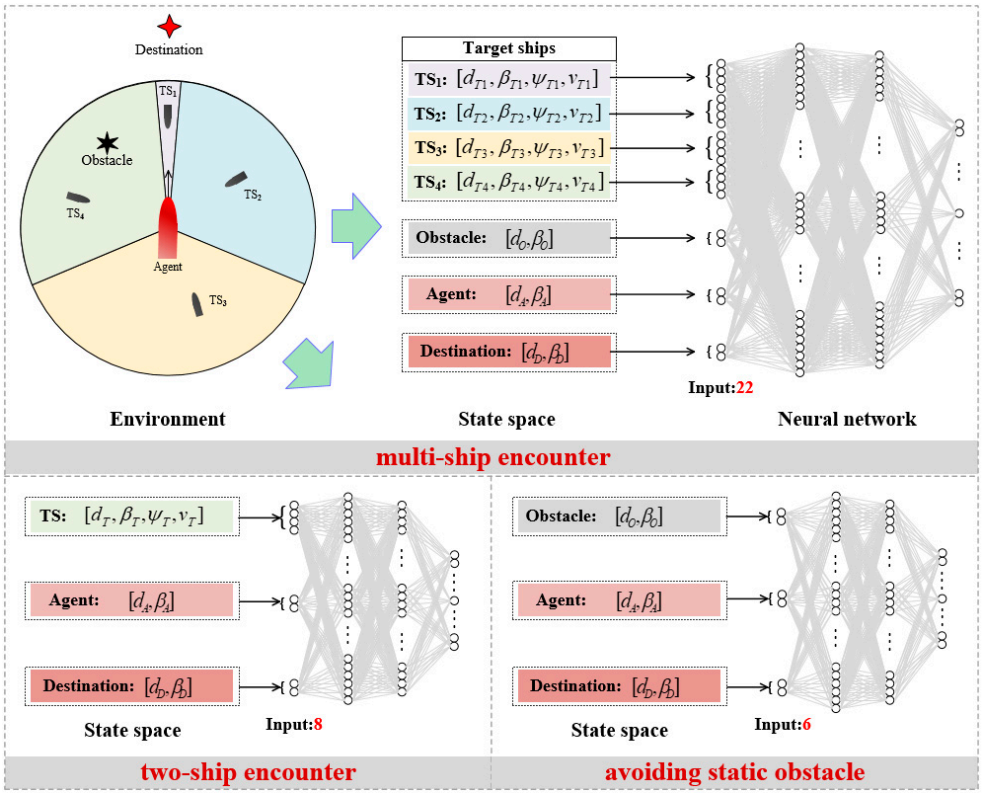

Figure 4. Schematic diagram of state space determination method in different scenarios.

The research aims to construct a state-to-action mapping that enables the agent to perform an optimal action when observing a particular state. Therefore, the input of the network is the state S observed by the agent. However, it is noteworthy that the output of the neural network is not the actions that the agent shall perform but the future rewards Q for each action. The agent will choose to execute the action with the highest future reward, as shown in Figure 5. Therefore, the number of output neurons of the neural network constructed in this paper equals the number of elements in the action space.

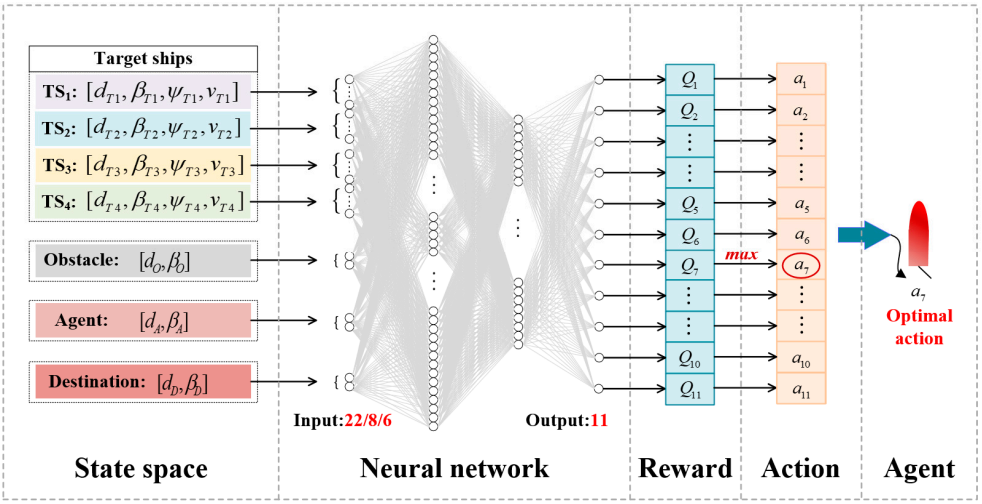

Figure 5. The action selection process of the agent.

3.2.3. Reward Functions

The reward is an evaluation of the action quality. The agent will continuously optimize its actions according to the reward feedback by the environment and finally complete the task as expected. This research aims to enable the agent to not only avoid colliding with obstacles but also deviate as little as possible from the original course by manipulating its rudder angle according to the COLREGs.

According to the COLREGs, the agents have different responsibilities in terms of target ships from different directions and the selection criteria of avoidance actions are also different. Therefore, we built a set of hierarchical reward functions. The first layer contains the reward functions that need to be implemented in all encounter scenarios, and we defined it as the base layer. The reward functions involved in the second layer need to be selectively executed according to the encounter situation, and this layer is defined as the COLREGs layer. The final value of the reward obtained by the agent is the sum of all the reward functions performed by the agent at the first and second layers.

(1) The base layer

The main purpose of the reward functions defined in the base layer is to drive the agent to find a collision avoidance path that satisfies the requirements of both safety and economy. It mainly includes five reward functions: the goal reward function R_{goal}, the advance reward function $R_{advance}$, the collision reward function $R_{collision}$, the rudder angle reward function R_{rudder}, and the yaw reward function R_{yaw}. The meanings and expressions of these functions are as follows:

The goal reward R_{goal} is defined to guide the agent to approach the destination. It can be calculated using Equation (12):

$$R_{goal} = \begin{cases} r_{goal} & if\ distance_{goal_t} < \gamma_0 \\ \lambda_{goal}\left(distance_{goal_t-1} - distance_{goal_t}\right) & otherwise \end{cases} \tag{12}$$

where $distance_{goal_t}$ is the distance between the agent and the destination at time t and γ_0, r_{goal}, and λ_{goal} are constants. As the agent approaches the destination, the reward value R_{goal} is positive. Otherwise, it is negative. When the $distance_{goal_t}$ is less than γ_0, it is considered that the agent has reached the destination and thus receives the largest reward r_{goal}.

Furthermore, to guide the agent sailing in a positive direction, the velocity projection in the forward direction must be positive. Consequently, the advance reward function can be designed as Equation (13):

$$R_{advance} = \begin{cases} -r_{advance} & if\ v_{advance_t} < 0 \\ \int_{t-1}^{t} v_{advance_t} dt & otherwise \end{cases} \tag{13}$$

where $R_{advance}$ represents the advance reward value, $r_{advance}$ is a constant, and $v_{advance_t}$ is the velocity projection in the forward direction at time t. The method for calculating the advance reward function is provided in Appendix A. When $v_{advance_t}$ is positive, the agent will receive a positive reward value. Otherwise, the agent will receive a large negative reward.

The collision reward is critical to encourage the agent to avoid obstacles (static obstacles and target ships). The reward functions can be expressed as Equation (14):

$$R_{collision} = \begin{cases} -r_{collision} & if\ distance_{obstacle_t} < \gamma_1 \\ 0 & otherwise \end{cases} \tag{14}$$

where $distance_{obstacle_t}$ is the shortest distance between the agent and the surrounding obstacles (static obstacles and target ships) and γ_1 and $r_{collision}$ are constants. When $distance_{obstacle_t}$ is less than γ_1, the agent will receive a negative reward $-r_{collision}$. Otherwise, the agent receives no reward.

In addition, to avoid large yaw and maintain a satisfactory rudder efficiency, the operator tends to select a moderate rudder angle to avoid obstacles. Therefore, the rudder angle reward function and the yaw reward function are defined as Equations (15) and (16), respectively:

$$R_{rudder} = \begin{cases} -r_{rudder} & if\ rudder\ angle_t \notin [\mu_1, \mu_2] \\ 0 & otherwise \end{cases} \quad (15)$$

$$R_{yaw} = \frac{S_{yaw_max} - S_{yaw_t}}{S_{yaw_max}} \quad (16)$$

where *rudder angle_t* is the action implemented by the agent at time t, r_{rudder}, μ_1 and μ_2 are constants; S_{yaw_t} is the yaw distance of the agent at time t, and it can be calculated by Equation (17); and S_{yaw_max} is the maximum allowable yaw distance:

$$S_{yaw} = \left| \int v \times \sin(\theta_0 - \theta_1) dt \right| \quad (17)$$

where v is the real-time velocity of the agent and the method for calculating θ_0 and θ_1 is provided in Appendix A.

(2) The COLREGs layer

The COLREGs explain the responsibilities of each ship in different encounter scenarios and make detailed provisions for the actions that ships should take to avoid a collision in different encounter situations. According to these constraints, ships should implement significantly different avoidance schemes for different encounter situations. However, when designing reward functions for the COLREGs, previous studies did not sufficiently consider the differences in the avoidance actions of agents when facing different avoidance responsibilities and encounter situations. One of the most significant differences between ship-collision avoidance and, for example, robot-collision avoidance and vehicle-collision avoidance is that the strategy for ships to avoid a collision must comply with the COLREGs and the implementation of the COLREGs will directly affect the practicability of the collision avoidance algorithm.

This paper constructs a set of reward functions for the COLREGs in the COLREGs layer, which take both different collision avoidance responsibilities and different encounter situations into consideration. These reward functions have five dimensions: stand-on, give-way, head-on, crossing, and overtaking. Note that the agent does not need to execute all the reward functions in this layer. Instead, the agent will implement the reward functions specific to its encounter situation, and normalizes their values based on their units. The five dimensions of the reward functions are defined as follows:

Stand-on

The provision by the COLREGs for actions by a stand-on ship is related to Rule 17, which states as follows:

"(a) (i) Where one of two vessels is to keep out of the way the other shall keep her course and speed.

"(a) (ii) The latter vessel may, however, take action to avoid collision by her maneuver alone, as soon as it becomes apparent to her that the vessel required to keep out of the way is not taking appropriate action in compliance with these Rules.

"(b) When, from any cause, the vessel required to keep her course and speed finds herself so close that collision cannot be avoided by the action of the give-way vessel alone, she shall take such action as will best aid to avoid collision.

"(c) A power-driven vessel which takes action in a crossing situation in accordance with subparagraph (a) (ii) of this Rule to avoid collision with another power-driven vessel shall, if the circumstances of the case admit, not alter course to port for a vessel on her own port side."

It can be seen that the responsibilities of the stand-on ship will change as the situation develops. At the initial moment of collision risk formation, the stand-on ship does not need to take any action. If the incoming ship does not take action in time, the stand-on ship should take an effective action. Since the COLREGs do not provide clear requirements for the stand-on ship's actions in the action stage, we use the reward functions in the base layer to constrain its actions in this stage. However, according to the provisions mentioned above, the stand-on ship cannot turn left in the crossing situation. Thus, we specifically design the reward function $R_{stand-on-crossing}$ for the stand-on ship to execute in this scenario (Equation (18)):

$$R_{stand-on-crossing} = \begin{cases} -r_{turn\ left} & if\ turn\ left \\ 0 & otherwise \end{cases} \tag{18}$$

In the encounter scenario where the stand-on ship does not need to act, it shall just maintain its speed and heading. It does not need to judge the rudder angle to be performed, so we do not define the reward function specially.

Give-way

Rule 16 in the COLREGs makes the following statement about the action of the give-way ship:

> "Every vessel which is directed to keep out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear."

Therefore, in the COLREGs, the constraints for the avoidance action of the give-way ship mainly include three aspects: early, substantial, and clear. Here, "early" means that the give-way ship should take actions as soon as possible, "substantial" means that the action taken by the give-way ship must be obvious, and "clear" means that the give-way ship must maintain sufficient distance from the target ship during the process of avoiding a collision. According to these requirements, we constructed three corresponding reward functions $R_{give-way1}$, $R_{give-way2}$, and $R_{give-way3}$, respectively, as Equations (19)–(21):

$$R_{give-way1} = -r_{early} \cdot \Delta t \tag{19}$$

$$R_{give-way2} = \begin{cases} 0 & \Delta\varphi \le \Delta\varphi_{min} \\ r_{substantial} & \Delta\varphi_{min} < \Delta\varphi < \Delta\varphi_{max} \\ 0 & \Delta\varphi \ge \Delta\varphi_{max} \end{cases} \tag{20}$$

$$R_{give-way3} = \begin{cases} 0 & D_{pre} \le D_{CPA} \\ -\frac{D_{pre}-D_{CPA}}{D_{pre}-D_{col}} \cdot r_{clear} & D_{col} \le D_{CPA} < D_{pre} \end{cases} \tag{21}$$

where r_{early}, $r_{substantial}$, and r_{clear} are three reward factors; Δt is the time interval between the agent detecting the risk of collision and starting to take avoidance action; $\Delta\varphi$ is the course change of the agent; $\Delta\varphi_{min}$ is the value of the course change that is perceptible to the crew; $\Delta\varphi_{max}$ is the threshold of acceptable course change; D_{CPA} is the closest distance between the agent and the target ship; D_{col} is the minimum safe distance between the agent and the target ship; and D_{pre} is the pre-warning distance between the agent and the target ship. D_{pre} is greater than D_{col}, and it is obtained according to sailing experience.

Head-on

The determination of the head-on situation and the corresponding method of avoiding collision is provided for in Rule 14 in the COLREGs:

> "When two power-driven vessels are meeting on reciprocal or nearly reciprocal courses so as to involve risk of collision each shall alter her course to starboard so that each shall pass on the port side of the other."

From the above description, we can see that the COLREGs require the ship to take two actions in a head-on situation: "alter the course to starboard" and "pass on the port side of the other". Therefore, we define the turn right reward function $R_{head-on1}$ and

the port side passing reward function $R_{head-on2}$ for the head-on situation, respectively, as Equations (22) and (23):

$$R_{head-on1} = \begin{cases} r_{starboard} & if\ turn\ right \\ 0 & otherwise \end{cases} \tag{22}$$

$$R_{head-on2} = \begin{cases} r_{pass} & 180^\circ < \beta_{CPA} < 360^\circ \\ 0 & otherwise \end{cases} \tag{23}$$

where $r_{starboard}$ and r_{pass} are constants and β_{CPA} is the bearing of the target ship to the agent at the closest point of approach.

Crossing

In the COLREGs, Rule 15 describes the 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."

It can be seen that in the crossing situation, the main restriction put on the movement of the give-way ship is avoiding crossing ahead of the other vessel. On the basis of this constraint, we define the reward function $R_{crossing}$ to prevent the give-way ship from crossing the bow of the stand-on ship (Equation (24)).

$$R_{crossing} = \begin{cases} r_{crossing} & 90^\circ < \alpha_{CPA} < 270^\circ \\ 0 & otherwise \end{cases} \tag{24}$$

where $r_{crossing}$ is a constant and α_{CPA} is the bearing of the agent in relation to the target ship at the closest point of approach.

Overtaking

In regard to overtaking, Rule 13 in the COLREGs states that "any vessel overtaking any other shall keep out of the way of the vessel being overtaken."

The above description stipulates that it is the responsibility of the overtaking ship to avoid a collision but there is no clarity as to what action said ship should take to avoid a collision. Therefore, in the overtaking situation, we do not define a specific reward function but directly use the reward functions in the base layer to evaluate the avoidance actions of the agent.

(3) The combination of reward functions

As mentioned earlier, the agent does not need to perform all of the reward functions defined above during the training process of the decision-making network. Instead, it will implement the reward functions corresponding to its responsibility and encounter situation.

According to the responsibility and the encounter situation, we divide the collision avoidance scenarios into seven categories: avoiding a static obstacle, multi-ship encounter, head-on, overtaking, crossing, a general scenario in which a stand-on ship needs to take actions, and a crossing encounter scenario where a stand-on ship needs to take actions. The combination of reward functions for each scene is shown in Figure 6. It is important to note that we only categorized scenarios in which an agent needs to take actions but did not consider the scenario where the agent is a stand-on ship and does not need to take actions. In that case, the agent only needs to maintain the current speed and course but does not need to make decisions about the actions it needs to perform. In addition, the quantitative judgment criteria of different encounter situations and the exact time when the stand-on ship must begin taking action are provided in [50].

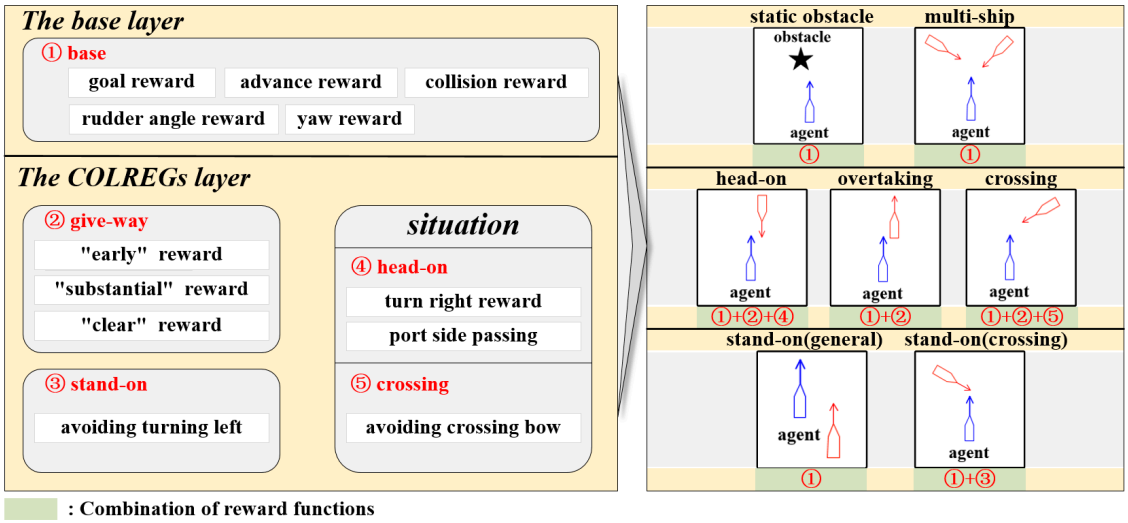

Figure 6. Definition of the reward function and the combination of the reward function in different encounter situations.

3.3. DRL Algorithm Design and Implementation

The designed collision avoidance method is trained by using the deep Q network (DQN). The DQN algorithm was proposed in 2013 [44], and based on its initial algorithm, the concept of the target network was introduced in [52], which improves the training stability. As shown in Figure 7, the DQN algorithm includes five parts: the environment, loss function, the evaluate network, the target network, and replay memory. Note that the two neural networks have the same structure. However, the parameters θ in the evaluate network are updated in each time step, while the parameters θ' in the target network are updated only when the agent accomplishes a complete episode. Compared with traditional RL methods, the DQN algorithm can be applied to the problem of the continuous state space. Furthermore, the introduction of replay memory increases the efficiency of data utilization and learning speed. To consider the hydrodynamic characteristics of the ship, an MMG module is added to the framework, which can accurately calculate the state of the agent at every moment.

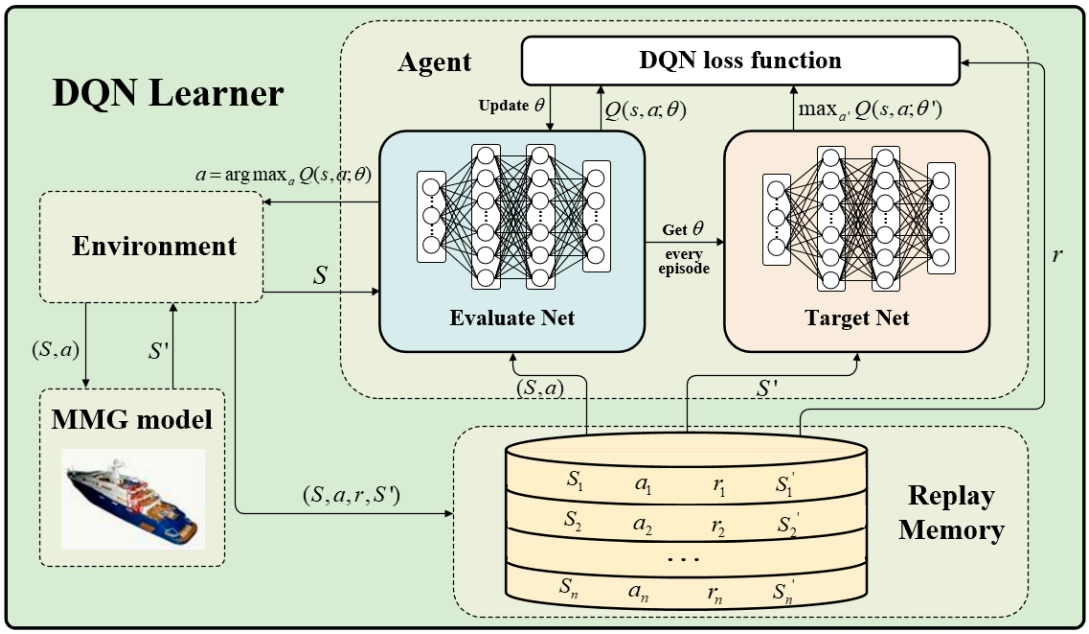

Figure 7. Schematic of the collision avoidance algorithm based on the DQN.

At each training step, the evaluate network outputs approximate reward values $Q(S, a; \theta)$ for each action based on the current state S of the agent. Then an appropriate action a is selected following the ε-greedy policy shown in Equation (25):

$$a = \begin{cases} \arg max_a Q(S, a; \theta) & with\ probability\ \varepsilon \\ random\ action\ a \in A & otherwise \end{cases} \tag{25}$$

Next, the agent performs the action a and calculates the state S' in the next time step through the MMG module while observing the reward r obtained from the environment. Subsequently, the experience (S, a, r, S') is stored in the replay memory module. An experience set $(S_i, a_i, r_i, S'_i)\ i \in \xi$ is randomly sampled from the replay memory module, which is used as the data set for network parameter update, and ξ is the number of the experience in the replay memory module. After that, the target network outputs an action value vector based on the previous experience set, which is regarded as the real action value, as shown in Equation (26):

$$U_i = r_i + \gamma max_a Q\left(S'_i, a; \theta'\right) \tag{26}$$

To update the network parameters, a loss function that calculates the difference between the real action value and approximate action value is designed, as shown in Equation (27):

$$L(\theta) = \frac{1}{\xi} \sum_{i \in \xi} [U_i - Q(S_i, a_i; \theta)]^2 \tag{27}$$

The loss function can be continuously optimized via the stochastic gradient descent strategy shown in Equation (28):

$$\theta = \theta + \alpha \frac{1}{\xi} \sum_{i \in \xi} [U_i - Q(S_i, a_i; \theta)] \nabla Q(S_i, a_i; \theta) \tag{28}$$

Through the above process, the parameters of the evaluate network are updated once and the state of the agent is transformed into S'.

When the agent has completed an entire episode, that is, it has reached the end state, the parameters of the target network will be covered by the evaluate network parameters, thus completing a parameter update of the target network, as shown in Equation (29).

$$\theta' = \theta \tag{29}$$

It should be noted that the model training process based on the DQN algorithm generally takes the number of training episodes as the termination rule, and this number will increase as the complexity of the training scene increases. When the reward value obtained by the model stabilizes at a high value, it proves that the model has converged.

The termination conditions of each training episode of the agent are defined as the following four states: (1) reaching the target point, (2) sailing out of the test area, (3) sailing to the opposite direction of the target point, and (4) colliding with the obstacles. Algorithm 1 provides a detailed description of the proposed collision avoidance algorithm based on the DQN.

Algorithm 1: The pseudocode of the DQN algorithm: Collision avoidance based on the DQN.

Initialize the replay memory with capacity C
Initialize evaluate network $Q(\cdot,\cdot,\theta)$ and target network $Q(\cdot,\cdot,\theta')$ with random parameters θ_0; $\theta = \theta_0$, $\theta' = \theta$
Set the maximum training episodes N
for episode = 1 to N **do**
    Initialize the state S
    while $S \neq end$ **do**
        Use ε-greedy policy to choose action a

$$a = \begin{cases} \arg max_a Q(S,a;\theta) & with\ probability\ \varepsilon \\ random\ action\ a \in A & otherwise \end{cases}$$

        Execute a, use the MMG model to calculate the next state S', observe the reward r
        Store experience (S,a,r,S') into replay memory
        Sample a batch of experience (S_i,a_i,r_i,S'_i) $i \in \xi$ from replay memory
        Use the target network $Q(\cdot,\cdot,\theta')$ to calculate the real action value U
        $U_i = r_i + \gamma max_a Q(S'_i,a;\theta')$
        Update evaluate network parameters θ via stochastic gradient descent strategy
        $\theta = \theta + \alpha \frac{1}{\xi} \sum_{i \in \xi} [U_i - Q(S_i,a_i;\theta)] \nabla Q(S_i,a_i;\theta)$
        Update state $S = S'$
    Update evaluate network parameters $\theta' = \theta$
end for

3.4. The Superiority of the Method

Compared to the previous method, one superiority of the approach proposed in this paper is that the characteristics of different encounter scenarios are fully considered in the definition of the state space. In addition, a novel state space definition method is proposed. This method improves the applicability of the algorithm and lays a foundation for the construction and training of different network structures for various encounter scenarios.

Another superiority of the proposed method is that the collision avoidance framework divides all encounter scenarios into seven types according to the avoidance constraints of the COLREGs for different encountered scenes. In the meantime, the corresponding combination of reward functions for each encounter type is built to train the agent. For the seven encounter types, seven corresponding neural networks are constructed, and each network is trained by using a specific combination of reward functions designed for each encounter type. This collision avoidance framework fully considers the characteristics of state observation in different scenes. Moreover, by constructing different training framework, it solves the problem of insufficient consideration of the different stipulations of the COLREGs in different scenes. Figure 8 displays the frame diagram of the collision avoidance framework.

Once the training process is completed, a set of networks suitable for collision avoidance decisions in different encounter types will be obtained. To make a collision avoidance decision, the agent will first judge the encounter type and then select the corresponding neural network. Particularly, each neural network in the proposed approach is trained by using a specific combination of reward functions for the different encounter types. Thus, compared with using a single neural network to make decisions for all encounter scenarios, the avoidance scheme obtained by the proposed algorithm for different scenarios will comply much better with the avoidance constraints of the COLREGs. The decision-making process the agent follows to avoid a collision is shown in Figure 9.

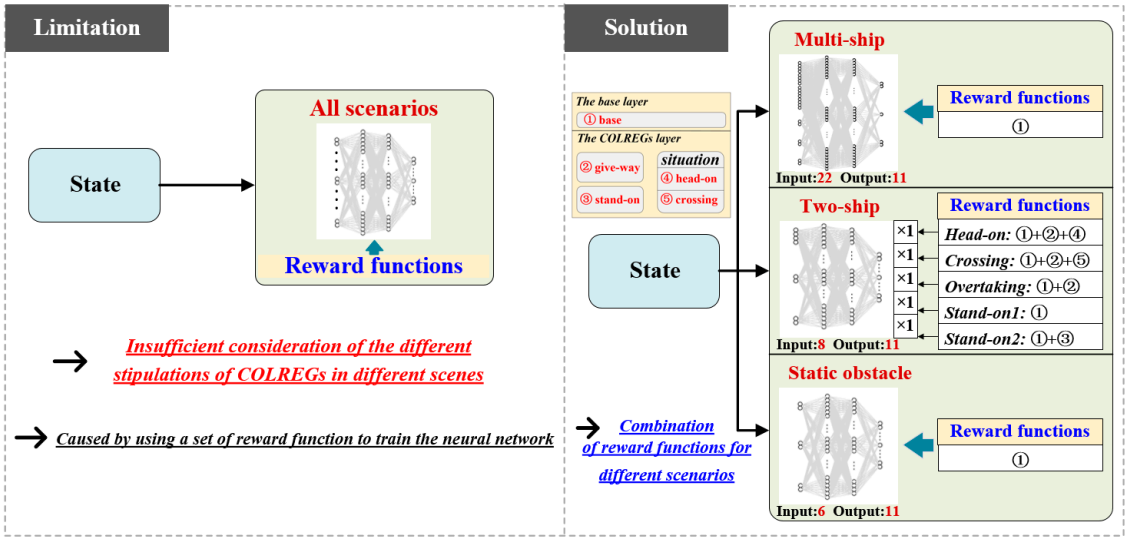

Figure 8. Network construction and combination of reward functions in different scenarios.

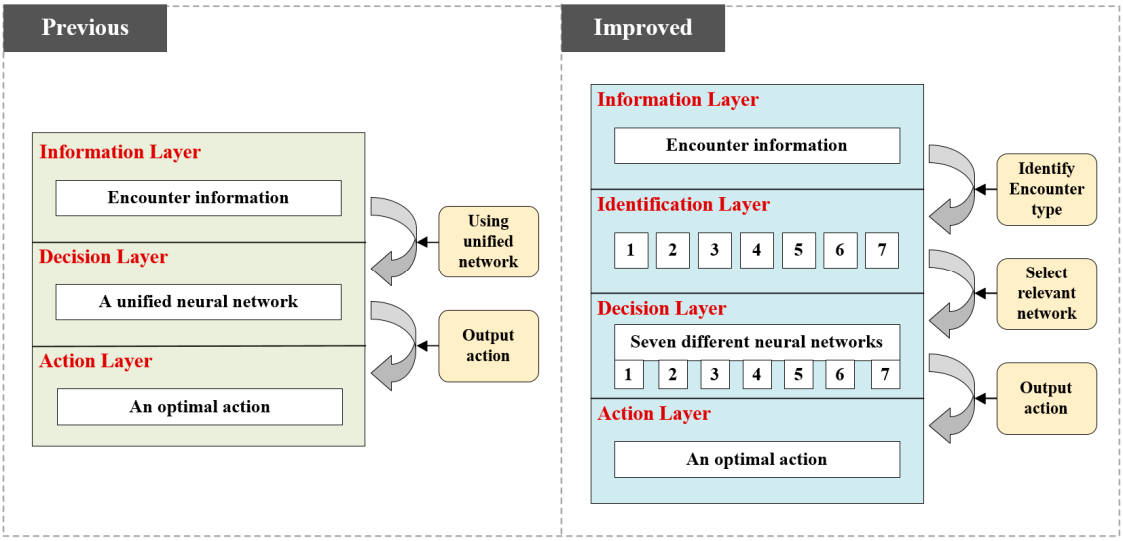

Figure 9. The decision-making process followed by the agent to avoid a collision.

4. Simulation Experiments

4.1. Preparation for the Simulation

The purpose of a simulation is to train an agent to perform collision avoidance operations compliant with the COLREGs in various encounter scenarios. The deep neural networks used in the experiments were built and trained using Tensorflow v2.2.0. These networks have a similar structure, and all of them are fully connected networks with two hidden layers. The number of input and output neurons of the network is equal to the number of elements contained in the state space and the action space, respectively. Relu activation functions and Adam optimizer are also applied to support the smooth training of the algorithm. The network setting parameters are displayed in Table 1.

Table 1. Hyperparameters of the algorithm.

Learning Rate	0.01	**Discount Rate**	0.99
Batch size	128	**Capacity**	1×10^4
Hidden layer 1	32	**Hidden layer 2**	32

During the training, the agent will obtain its real-time position via the MMG model, while gaining a relative reward from the environment. The parameters of the MMG model used in the experiment are taken from a real ship, and some principal data are displayed in Table 2. Note that, to reduce the computation, the state of the agent is updated every 20 s.

Table 2. Principal parameters of the ship.

Length	225 (m)	**Breadth**	32.5 (m)
Draft	14.5 (m)	**Displacement**	90,000 × 10^3 (kg)
$\mathbf{C_b}$	0.8715	**Revolutions**	90 (r/min)
Acreage of rudder	56.88 (m^2)	**Propeller advance**	4.738 (m)

To train the proposed collision avoidance model effectively, each network is heavily trained by the corresponding encounter scenario. Figure 10 describes the training results for the complex multi-ship encounter scenario. In Figure 10, the curve represents the average episodic reward with a moving windows size of 50 episodes, and the number of total training episodes is 5000. On the basis of the learning, the average episodic reward increases with the training and converges to a stable range after about 2500 episodes, while the policy converges to the final policy.

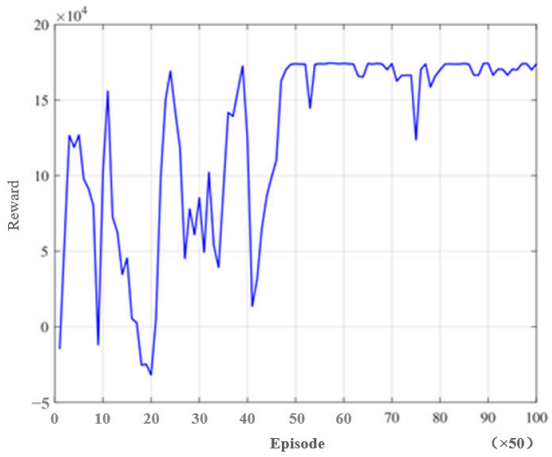

Figure 10. Episodic reward during the training process of the DRL.

4.2. Application Examples

To validate the effectiveness of the proposed collision avoidance method, various complex simulation cases were carried out via the PyCharm platform. Each case is in charge of (1) a static obstacle, (2) a two-ship scenario, (3) a multi-ship scenario, and (4) a scenario in which a stand-on ship should perform the maneuver to avoid the collision. Note that the situations are designed from the point of view of the agent. Moreover, to improve the speed, the original position and destination are predefined within an acceptable range.

4.2.1. Case 1

In this scenario, an isolated obstacle is set in the center of the test area. The size of this area is set to 6 × 6 NM, and the midpoint of the lower boundary is taken as the origin. The starting point of the navigation task is the origin, and the endpoint, marked in red, is the midpoint of the upper boundary. When the agent reaches the yellow area, with a radius of 1000 m, it can be considered to have completed the task. The initial course of the agent is set to zero, so the agent needs to go through the area where the obstacle is located. The minimum distance that must be maintained between the agent and the obstacle is set to 1000 m.

As shown in Figure 11, the agent will find a trajectory leading to the target area, while safely avoiding the obstacle. This trajectory is relatively smooth, and no sharp course change is required. In the simulation result, an obvious avoidance maneuver was

performed at the starting position so that the course of the agent no longer intersects with the obstacle. After sailing along the new course for a while, the agent performs a port-side avoidance maneuver to reduce the yaw distance and gradually returns to the initial course. When the destination is approached, another port-side avoidance maneuver is executed, and the agent eventually travels to the target area. It can be seen that the distance between the agent and the destination decreases linearly with passage of time and the distance between the agent and the obstacle first reduces and then increases. During the whole navigation process, the agent and the obstacle are the nearest to each other, i.e., 2025 m apart, at 44 × 20 s. According to a detailed analysis, this distance is larger than the limit value, thus increasing the yaw cost of the agent. This may be explained by the fact that the punishment for collision would be much more severe than that for the yawing. As a result, the agent tends to select a much safer distance from the obstacle to avoid it. This phenomenon also exists in later simulation scenarios.

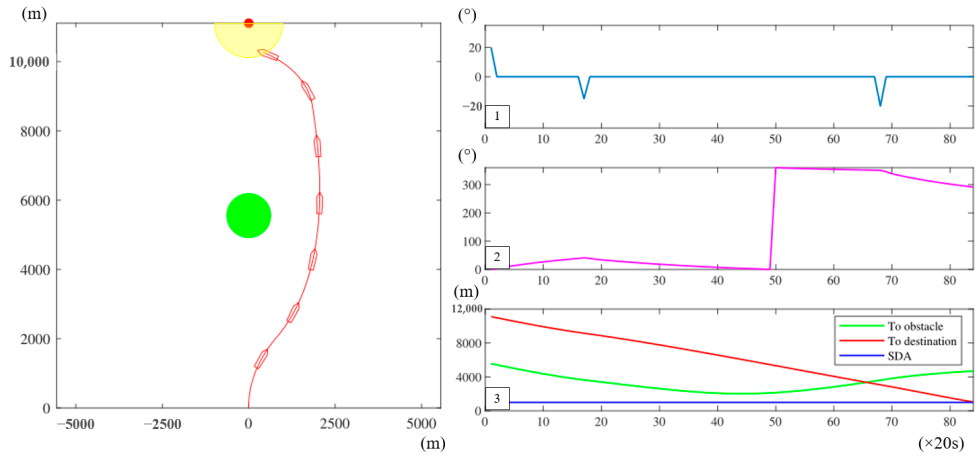

Figure 11. Simulation result for avoiding a single obstacle (1: rudder angle; 2: course; 3: distance).

4.2.2. Case 2

Unlike avoiding static obstacles, when avoiding dynamic ships, the avoidance maneuver performed by the agent should ensure not only safety but also compliance with the COLREGs. In this section, three typical two-ship encounter situations are designed: head-on, overtaking, and crossing. The agent and the target ship set off from their starting positions and once collision has been avoided, the test mission is also terminated. The minimum distance that must be maintained between two ships is set to 1 NM. The simulation results of different encounter scenarios are shown in Figure 12.

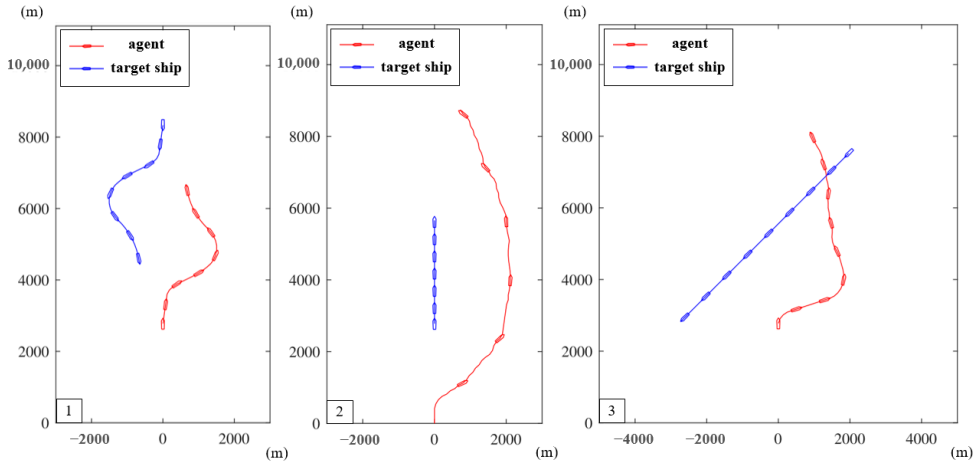

Figure 12. Simulation results for two-ship-collision avoidance (1: head-on; 2: overtaking; 3: crossing).

In the head-on situation, a target ship approaching the agent with a reciprocal course is designed, and the speed of the target ship is the same as that of the agent. According to the COLREGs, both ships are give-way ships in this scenario. Therefore, they change course to the starboard at the same time when the collision risk between the two ships is detected. When the collision risk disappears, they return to their target course eventually. In the experimental result, both ships perform a large movement to ensure that the intentions can be detected by the other ship clearly, which complies with the constraints in the COLREGs for the maneuver of a give-way ship. Meanwhile, a sufficient safety distance is maintained continuously between the agent and the target ship throughout the whole avoidance process.

In the overtaking situation, a target ship is set to head north, which is the same as the direction the agent is heading. The agent overtakes the target ship from behind. In compliance with the COLREGs, the agent can avoid collision with the target ship on both sides. In this case, the agent chooses the starboard side, and a predefined safety distance of 1 NM is guaranteed throughout the entire avoidance process. Due to the low relative velocity, the avoidance maneuver takes a long time to perform, but eventually, the agent overtakes the target ship and returns to the initial course. Meanwhile, the course of the agent changes smoothly without drastic manipulation, which is consistent with the dynamic characteristics of the ship.

In the crossing situation, a target ship approaches the agent from 45° forward of the beam and the speed of the target ship is the same as that of the agent. As described in the COLREGs, in the case of the crossing scenario, the ship that has the other ship on its starboard side is the give-way ship, and if the circumstances of the case admit, the give-way ship shall avoid crossing ahead of the other ship. Therefore, the agent changes course to the starboard and avoids the target ship from its stern. A sufficient safety distance is observed, and when the task of avoiding a collision is complete, the agent alters the course to the port to revert to its original course.

4.2.3. Case 3

In this case, a more complex simulation experiment involving three ships is designed. The agent is designed to avoid collisions with two target ships simultaneously, while taking into account the provisions of the COLREGs and good seamanship. The initial positions of these ships are set in advance, and if any avoidance action is not performed, collision will be inevitable. The minimum allowable distance between the agent and the other two ships is set to 1 NM.

Figure 13 illustrates the simulation result for the multiple-ship-encounter scenario. In this situation, two target ships are defined on the starboard side of the agent. Since there are no specialized rules for multi-ship-encounter scenarios in the COLREGs, the agent shall perform the maneuver in compliance with good seamanship. According to the trajectories described in Figure 13, the agent initially maintains its original course, but as its distance from target ship1 decreases, the agent performs a rudder angle to the starboard. Since the agent detects a collision risk from target ship2, it does not revert to its original course immediately but chooses to continue sailing along the new course. When the collision risk has disappeared, the agent selects a rudder angle to the port and returns to its original path. The two avoidance maneuvers performed by the agent are moderate and do not employ the extreme operation of a full rudder. Moreover, the maximum course angle change was measured as 73°, which is large enough for the target ships to identify the intentions of the agent. In the meantime, the trajectory of the agent is extraordinarily smooth, and the reason is that the maneuverability of the ship is fully considered. In addition, the maneuver that changes the agent's course to the starboard and allows it to pass the target ships on the port side is also compliant with good seamanship. The minimum distance between the agent and target ship1 is 3216 m, and that between the agent and target ship2 is 3878 m; both are within an acceptable range.

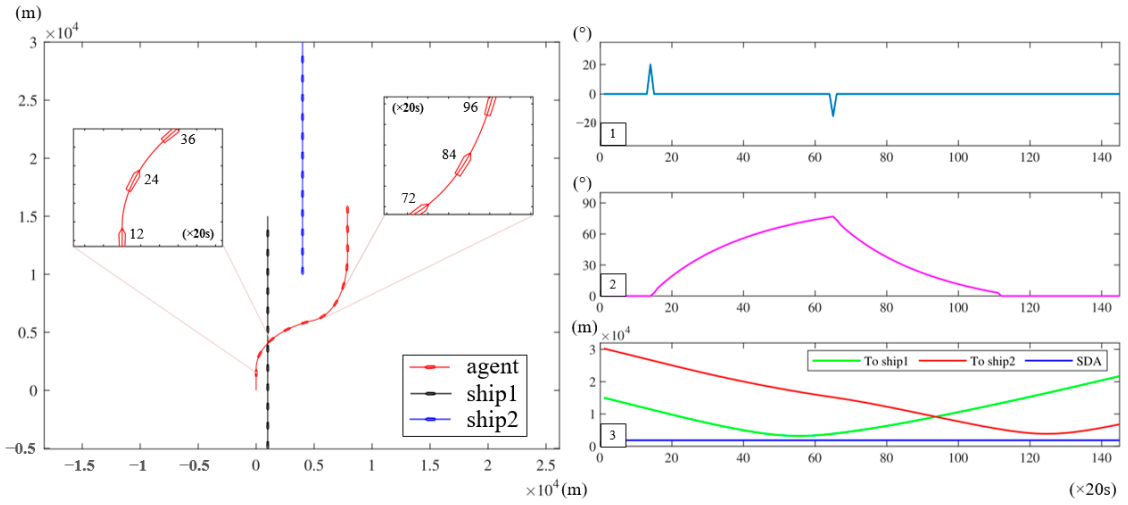

Figure 13. Simulation result for multi-ship-collision avoidance (1: rudder angle; 2: course; 3: distance).

4.2.4. Case 4

There is always some degree of environmental uncertainty that unmanned ships might encounter when sailing at sea. Sometimes, due to some uncontrollable cause, the give-way ship may not be immediately able perform an avoidance action, intensifying the collision risk between two ships. In this case, the COLREGs require the stand-on ship to take such action as will best help avoid a collision. To verify the effectiveness of the proposed method for this scene, a crossing situation is set as shown in Figure 14.

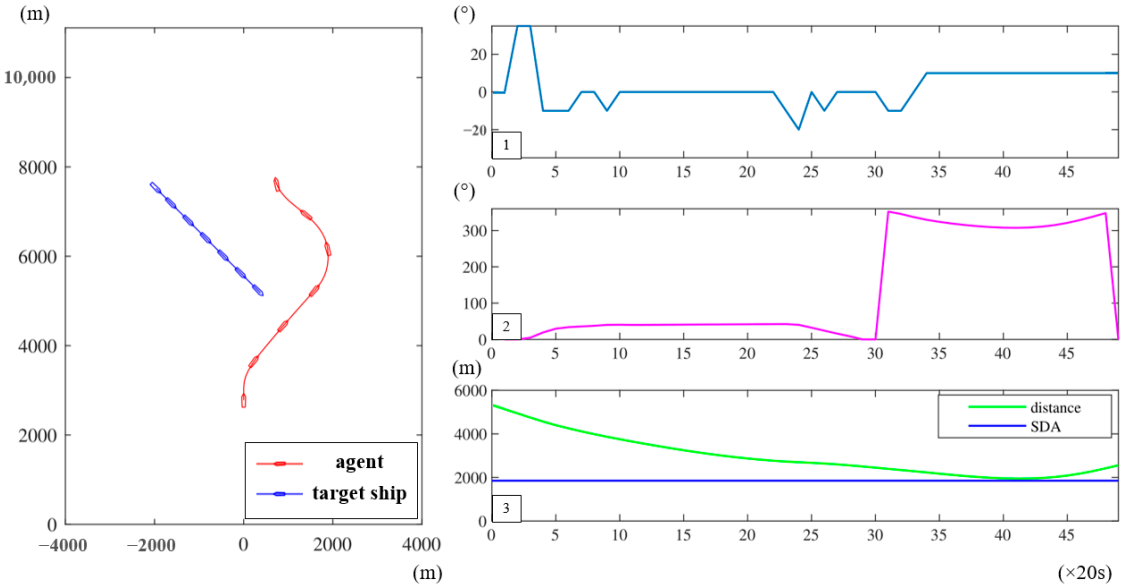

Figure 14. Simulation result for two-ship-collision avoidance in the scenario where the stand-on ship needs to take actions (1: rudder angle; 2: course; 3: distance).

Figure 14 represents the simulation result for collision avoidance during the designed crossing encounter scenario. According to the experimental trajectories, there is a target ship coming from the port side of the agent and the target ship is the give-way ship, while the agent should maintain its initial course and speed. However, as the distance between the two ships reduces, the target ship does not take appropriate action in compliance with the COLREGs. Thus, the agent decides to take effective unilateral actions to avoid collision. In this case, the agent performs a hard starboard so that the collision risk can be mitigated rapidly. Meanwhile, the target ship can easily discern such a large maneuver, which ensures

the coordination of operation between the two ships. Moreover, according to the COLREGs, in such a crossing situation, the stand-on ship should take actions to avoid a collision, if circumstances permit, the stand-on ship shall not alter course to port for a ship on its port side. It can conclude that the actions taken by the agent have taken full account of the COLREGs. In addition, the course of the agent changes smoothly, without sudden fluctuations, and eventually returns to the target heading. The agent and the target ship are the closest to each other (i.e., 1949 m apart), at approximately 41 × 20 s, which is larger than the limit value (1 NM).

5. Discussion

The traditional collision avoidance method based on geometric analysis and PID (Proportional Integral Derivative) control has been widely used in the construction of some collision avoidance platforms due to its high decision accuracy. In order to verify the effectiveness of the proposed model, we compared the proposed model with this traditional collision avoidance model.

Figure 15 illustrates the avoidance process of the agent driven by a traditional collision avoidance method based on the PID algorithm in the same multi-ship encounter scene as Case 3. Some significant collision avoidance parameters planned by the two methods are compared respectively, and the results are shown in Table 3.

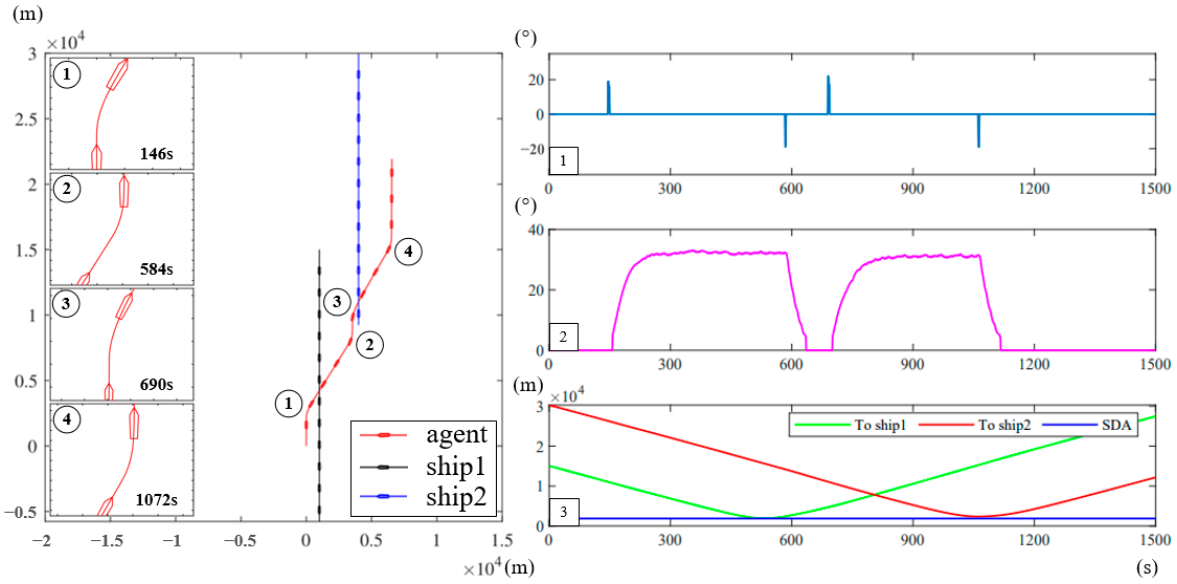

Figure 15. Simulation result for multi-ship collision avoidance based on PID (1: rudder angle; 2: course; 3: distance).

Table 3. Comparisons between different methods.

Methods	Turning Times	Drift Distance	DCPA	Course Change Value	Maximum Rudder Angle
PID	4	6575 m	1883 m	33°	22°
DRL	2	7890 m	3216 m	77°	20°

According to the trajectory of the agent and the statistical results in Table 3, it can be found that compared with the traditional method, the new model proposed in this paper can complete the collision avoidance task with fewer turning times, and the course change value of the agent is more obvious, which makes it easier for the target ship to understand the agent's intentions. In addition, during the whole collision avoidance process, the agent driven by the DRL model can maintain a more adequate safety distance from the target ship with a smaller and appropriate rudder angle.

However, compared with the traditional method, the yaw distance of the agent is longer in the model based on DRL. The reason is that the traditional collision avoidance model is a precise analytical model, which can find the most economical collision avoidance scheme according to the real-time navigation state. However, this decision-making mode also increases the complexity of collision avoidance. For example, in this scene, the agent completes the avoidance task in four turns, while the method based on the DRL model only takes two. This is because the traditional method does not have the same global awareness ability of the surrounding environment as the DRL model, which is also one of the most typical characteristics of the DRL method. Although this characteristic of the DRL method causes an increase in yaw distance, it is believed that this problem can be solved by optimizing the setting of the reward functions.

In addition, it should be noted that the data of a ready-made ship model were used in the experimental verification section of the paper. However, the motion characteristics of different ships are not the same. Therefore, in the actual situation, before using the collision avoidance method proposed in this paper, it is necessary to build a motion model according to the motion parameters of the ship (the agent) and train the collision avoidance algorithm to adapt to the motion characteristics of the ship (the agent) by using this motion model.

6. Conclusions and Future Work

In this paper, a deep-reinforcement-learning-based collision avoidance method is developed for unmanned ships. To consider the manipulative characteristics of the ship, the MMG model is introduced, by which real-time navigation information can be inferred. Then, the state and action spaces that correspond to the navigation experience are designed and a new framework for collision avoidance decision-making network construction and training is proposed. Moreover, to take full account of the COLREGs, a set of hierarchical reward functions is developed, which is used in the training of the decision-making network. Subsequently, by introducing the DQN algorithm, a collision avoidance decision model is built. Finally, to validate the applicability of the proposed method, a variety of simulated scenarios are designed with comprehensive performance evaluation. The simulation results show that the proposed method enables the agent to avoid collision safely in a complex environment, while ensuring its compliance with the COLREGs and good seamanship. This method could provide a novel attempt at collision avoidance for unmanned ships.

In terms of future work, there are mainly two aspects. On the one hand, in the experiment of the multi-ship scenario, only the agent performs action to avoid a collision, while in a practical situation, the multi-ship-collision-avoidance task generally requires the cooperation of ships. Therefore, a cooperative multi-ship-collision-avoidance method based on the DRL is one of the focuses of future research. On the other hand, intelligent navigation is a complex task that includes path planning, path following, collision avoidance, etc. The collision avoidance algorithm will be activated only when the collision risk occurs between the agent and obstacles. Designing an efficient algorithm that integrates multi-tasking and enables support decisions for the entire process of intelligent navigation is another focus of future research.

Author Contributions: Conceptualization, L.H. and K.L.; methodology, W.W.; software, X.W.; validation, J.W., X.W. and W.W.; formal analysis, L.H. and K.L.; investigation, L.H.; resources, K.L.; data curation, K.L.; writing—original draft prepa-ration, W.W.; writing—review and editing, W.W.; visualization, X.W.; supervision, L.H.; project administration, K.L.; funding acquisition, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 52031009), China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was funded by the National Natural Science Foundation of China (Grant No. 52031009), China.

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

Appendix A Method for Calculating $v_{advance}$

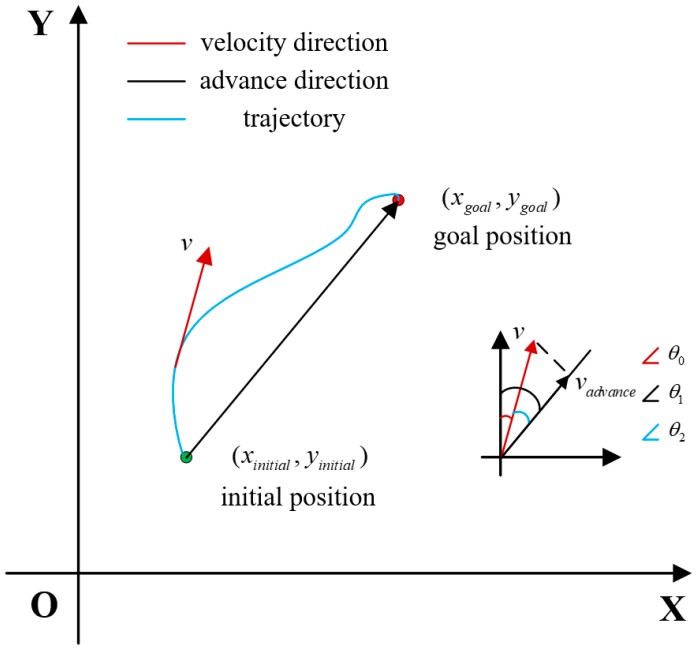

Figure A1. The projection of the velocity in the forward direction.

Table A1. The calculation of $v_{advance}$.

X-axis	Y-axis	θ_0	θ_1	θ_2	$v_{advance}$
$x_{goal} - x_{initial} > 0$	$y_{goal} - y_{initial} > 0$	C	$arctan\frac{x_{goal}-x_{initial}}{y_{goal}-y_{initial}}$	$\lvert\theta_0 - \theta_1\rvert$	$v \times cos\theta_2$
$x_{goal} - x_{initial} > 0$	$y_{goal} - y_{initial} < 0$		$arctan\frac{x_{goal}-x_{initial}}{y_{goal}-y_{initial}} + \pi$		
$x_{goal} - x_{initial} < 0$	$y_{goal} - y_{initial} < 0$				
$x_{goal} - x_{initial} < 0$	$y_{goal} - y_{initial} > 0$		$arctan\frac{x_{goal}-x_{initial}}{y_{goal}-y_{initial}} + 2\pi$		

Here, θ_0 is the course of the ship. It can be obtained on the basis of real-time information. Next, θ_2 is the angle between the velocity direction and the advance direction. Its value is positive.

References

1. Campbell, S.; Naeem, W.; Irwin, G.W. A Review on Improving the Autonomy of Unmanned Surface Vehicles through Intelligent Collision Avoidance Manoeuvres. *Annu. Rev. Control* **2012**, *36*, 267–283. [CrossRef]
2. Liu, Z.; Zhang, Y.; Yu, X.; Yuan, C. Unmanned Surface Vehicles: An Overview of Developments and Challenges. *Annu. Rev. Control* **2016**, *41*, 71–93. [CrossRef]
3. Naeem, W.; Irwin, G.W.; Yang, A. COLREGs-Based Collision Avoidance Strategies for Unmanned Surface Vehicles. *Mechatronics* **2012**, *22*, 669–678. [CrossRef]
4. Tang, P.; Zhang, R.; Liu, D.; Huang, L.; Liu, G.; Deng, T. Local Reactive Obstacle Avoidance Approach for High-Speed Unmanned Surface Vehicle. *Ocean Eng.* **2015**, *106*, 128–140. [CrossRef]
5. Wang, W.; Mateos, L.A.; Park, S.; Leoni, P.; Gheneti, B.; Duarte, F.; Ratti, C.; Rus, D. Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 6189–6196. [CrossRef]
6. Cho, Y.; Han, J.; Kim, J. Efficient COLREG-Compliant Collision Avoidance in Multi-Ship Encounter Situations. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 1899–1911. [CrossRef]

7. Zhao, L.; Roh, M.-I. COLREGs-Compliant Multiship Collision Avoidance Based on Deep Reinforcement Learning. *Ocean Eng.* **2019**, *191*, 106436. [CrossRef]
8. Park, B.S.; Yoo, S.J. Adaptive-Observer-Based Formation Tracking of Networked Uncertain Underactuated Surface Vessels with Connectivity Preservation and Collision Avoidance. *J. Frankl. Inst.* **2019**, *356*, 7947–7966. [CrossRef]
9. Song, A.L.; Su, B.Y.; Dong, C.Z.; Shen, D.W.; Xiang, E.Z.; Mao, F.P. A Two-Level Dynamic Obstacle Avoidance Algorithm for Unmanned Surface Vehicles. *Ocean Eng.* **2018**, *170*, 351–360. [CrossRef]
10. 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]
11. Huang, Y.; Van Gelder, P.; Wen, Y. Velocity Obstacle Algorithms for Collision Prevention at Sea. *Ocean Eng.* **2018**, *151*, 308–321. [CrossRef]
12. 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]
13. Molinos, E.J.; Llamazares, Á.; Ocaña, M. Dynamic Window Based Approaches for Avoiding Obstacles in Moving. *Robot. Auton. Syst.* **2019**, *118*, 112–130. [CrossRef]
14. Tsou, M.C.; Hsueh, C.K. The Study of Ship Collision Avoidance Route Planning by Ant Colony Algorithm. *J. Mar. Sci. Technol.* **2010**, *18*, 746–756. [CrossRef]
15. Hu, L.; Naeem, W.; Rajabally, E.; Watson, G.; Mills, T.; Bhuiyan, Z.; Raeburn, C.; Salter, I.; Pekcan, C. A Multiobjective Optimization Approach for COLREGs-Compliant Path Planning of Autonomous Surface Vehicles Verified on Networked Bridge Simulators. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1167–1179. [CrossRef]
16. Meyer, E.; Heiberg, A.; Rasheed, A.; San, O. COLREG-Compliant Collision Avoidance for Unmanned Surface Vehicle Using Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 165344–165364. [CrossRef]
17. Xie, S.; Chu, X.; Zheng, M.; Liu, C. A Composite Learning Method for Multi-Ship Collision Avoidance Based on Reinforcement Learning and Inverse Control. *Neurocomputing* **2020**, *411*, 375–392. [CrossRef]
18. Chun, D.H.; Roh, M.-I.; Lee, H.W.; Ha, J.; Yu, D. Deep Reinforcement Learning-Based Collision Avoidance for an Autonomous Ship. *Ocean Eng.* **2021**, *234*, 109216. [CrossRef]
19. 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]
20. Ma, Y.; Zhao, Y.; Wang, Y.; Gan, L.; Zheng, Y. Collision-Avoidance under COLREGS for Unmanned Surface Vehicles via Deep Reinforcement Learning. *Marit. Policy Manag.* **2020**, *47*, 665–686. [CrossRef]
21. Wang, Y.; Yu, X.; Liang, X.; Li, B. A COLREGs-Based Obstacle Avoidance Approach for Unmanned Surface Vehicles. *Ocean Eng.* **2018**, *169*, 110–124. [CrossRef]
22. Woerner, K.; Benjamin, M.R.; Novitzky, M.; Leonard, J.J. Quantifying Protocol Evaluation for Autonomous Collision Avoidance: Toward Establishing COLREGS Compliance Metrics. *Auton. Robot.* **2019**, *43*, 967–991. [CrossRef]
23. Zhou, X.Y.; Huang, J.J.; Wang, F.W.; Wu, Z.L.; Liu, Z.J. A Study of the Application Barriers to the Use of Autonomous Ships Posed by the Good Seamanship Requirement of COLREGs. *J. Navig.* **2020**, *73*, 710–725. [CrossRef]
24. Woo, J.; Kim, N. Collision Avoidance for an Unmanned Surface Vehicle Using Deep Reinforcement Learning. *Ocean Eng.* **2020**, *199*, 107001. [CrossRef]
25. Wu, X.; Chen, H.; Chen, C.; Zhong, M.; Xie, S.; Guo, Y.; Fujita, H. The Autonomous Navigation and Obstacle Avoidance for USVs with ANOA Deep Reinforcement Learning Method. *Knowl.-Based Syst.* **2020**, *196*, 105201. [CrossRef]
26. Pang, Z.H.; Zheng, C.B.; Sun, J.; Han, Q.L.; Liu, G.P. Distance- And Velocity-Based Collision Avoidance for Time-Varying Formation Control of Second-Order Multi-Agent Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 1253–1257. [CrossRef]
27. Lyu, H.; Yin, Y. COLREGS-Constrained Real-Time Path Planning for Autonomous Ships Using Modified Artificial Potential Fields. *J. Navig.* **2019**, *72*, 588–608. [CrossRef]
28. Xu, X.; Pan, W.; Huang, Y.; Zhang, W. Dynamic Collision Avoidance Algorithm for Unmanned Surface Vehicles via Layered Artificial Potential Field with Collision Cone. *J. Navig.* **2020**, *73*, 1306–1325. [CrossRef]
29. 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]
30. Huang, Y.; van Gelder, P.H.A.J.M. Time-Varying Risk Measurement for Ship Collision Prevention. *Risk Anal.* **2020**, *40*, 24–42. [CrossRef]
31. 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]
32. Liu, J.; Zhang, J.; Yan, X.; Soares, C.G. Multi-Ship Collision Avoidance Decision-Making and Coordination Mechanism in Mixed Navigation Scenarios. *Ocean Eng.* **2022**, *257*, 111666. [CrossRef]
33. Zhang, J.; Zhang, D.; Yan, X.; Haugen, S.; Soares, C.G. A Distributed Anti-Collision Decision Support Formulation in Multi-Ship Encounter Situations under COLREGs. *Ocean Eng.* **2015**, *105*, 336–348. [CrossRef]
34. Lisowski, J. Synthesis of a Path-Planning Algorithm for Autonomous Robots Moving in a Game Environment during Collision Avoidance. *Electronics* **2021**, *10*, 675. [CrossRef]

35. Zhang, M.; Montewka, J.; Manderbacka, T.; Kujala, P.; Hirdaris, S. A Big Data Analytics Method for the Evaluation of Ship-Ship Collision Risk Reflecting Hydrometeorological Conditions. *Reliab. Eng. Syst. Saf.* **2021**, *213*, 107674. [CrossRef]
36. Zhang, M.; Conti, F.; Le Sourne, H.; Vassalos, D.; Kujala, P.; Lindroth, D.; Hirdaris, S. A Method for the Direct Assessment of Ship Collision Damage and Flooding Risk in Real Conditions. *Ocean Eng.* **2021**, *237*, 109605. [CrossRef]
37. Zhang, M.; Zhang, D.; Fu, S.; Kujala, P.; Hirdaris, S. A Predictive Analytics Method for Maritime Traffic Flow Complexity Estimation in Inland Waterways. *Reliab. Eng. Syst. Saf.* **2022**, *220*, 108317. [CrossRef]
38. Zhang, M.; Zhang, D.; Goerlandt, F.; Yan, X.; Kujala, P. Use of HFACS and Fault Tree Model for Collision Risk Factors Analysis of Icebreaker Assistance in Ice-Covered Waters. *Saf. Sci.* **2019**, *111*, 128–143. [CrossRef]
39. Johansen, T.A.; Perez, T.; Cristofaro, A. Ship Collision Avoidance and COLREGS Compliance Using Simulation-Based Control Behavior Selection with Predictive Hazard Assessment. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3407–3422. [CrossRef]
40. Tsou, M.C.; Kao, S.L.; Su, C.M. Decision Support from Genetic Algorithms for Ship Collision Avoidance Route Planning and Alerts. *J. Navig.* **2010**, *63*, 167–182. [CrossRef]
41. Wang, H.; Fu, Z.; Zhou, J.; Fu, M.; Ruan, L. Cooperative Collision Avoidance for Unmanned Surface Vehicles Based on Improved Genetic Algorithm. *Ocean Eng.* **2021**, *222*, 108612. [CrossRef]
42. Fiskin, R.; Atik, O.; Kisi, H.; Nasibov, E.; Johansen, T.A. Fuzzy Domain and Meta-Heuristic Algorithm-Based Collision Avoidance Control for Ships: Experimental Validation in Virtual and Real Environment. *Ocean Eng.* **2021**, *220*, 108502. [CrossRef]
43. Hu, L.; Naeem, W.; Rajabally, E.; Watson, G.; Mills, T.; Bhuiyan, Z.; Salter, I. COLREGs-Compliant Path Planning for Autonomous Surface Vehicles: A Multiobjective Optimization Approach. *IFAC-PapersOnLine* **2017**, *50*, 13662–13667. [CrossRef]
44. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
45. Bhopale, P.; Kazi, F.; Singh, N. Reinforcement Learning Based Obstacle Avoidance for Autonomous Underwater Vehicle. *J. Mar. Sci. Appl.* **2019**, *18*, 228–238. [CrossRef]
46. 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]
47. 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]
48. Zhou, X.; Wu, P.; Zhang, H.; Guo, W.; Liu, Y. Learn to Navigate: Cooperative Path Planning for Unmanned Surface Vehicles Using Deep Reinforcement Learning. *IEEE Access* **2019**, *7*, 165262–165278. [CrossRef]
49. 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]
50. He, Y.; Jin, Y.; Huang, L.; Xiong, Y.; Chen, P.; Mou, J. Quantitative Analysis of COLREG Rules and Seamanship for Autonomous Collision Avoidance at Open Sea. *Ocean Eng.* **2017**, *140*, 281–291. [CrossRef]
51. Tam, C.; Bucknall, R. Cooperative Path Planning Algorithm for Marine Surface Vessels. *Ocean Eng.* **2013**, *57*, 25–33. [CrossRef]
52. 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]