Journal of *Marine Science and Engineering*

MDPI

Article

Optimizing Multi-Vessel Collision Avoidance Decision Making for Autonomous Surface Vessels: A COLREGs-Compliant Deep Reinforcement Learning Approach

Weidong Xie [1,*], Longhui Gang [1,*], Mingheng Zhang [2], Tong Liu [1] and Zhixun Lan [1]

[1] College of Navigation, Dalian Maritime University, Dalian 116026, China; liutong202111@163.com (T.L.); lanzhixun@dlmu.edu.cn (Z.L.)
[2] School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China; zhangmh@dlut.edu.cn
* Correspondence: xwd@dlmu.edu.cn (W.X.); ganglh@dlmu.edu.cn (L.G.); Tel.: +86-139-0252-9032 (W.X.); +86-189-0098-2692 (L.G.)

Abstract: Automatic collision avoidance decision making for vessels is a critical challenge in the development of autonomous ships and has become a central point of research in the maritime safety domain. Effective and systematic collision avoidance strategies significantly reduce the risk of vessel collisions, ensuring safe navigation. This study develops a multi-vessel automatic collision avoidance decision-making method based on deep reinforcement learning (DRL) and establishes a vessel behavior decision model. When designing the reward function for continuous action spaces, the criteria of the "Convention on the International Regulations for Preventing Collisions at Sea" (COLREGs) were adhered to, taking into account the vessel's collision risk under various encounter situations, real-world navigation practices, and navigational complexities. Furthermore, to enable the algorithm to precisely differentiate between collision avoidance and the navigation resumption phase in varied vessel encounter situations, this paper incorporated "collision avoidance decision making" and "course recovery decision making" as state parameters in the state set design, from which the respective objective functions were defined. To further enhance the algorithm's performance, techniques such as behavior cloning, residual networks, and CPU-GPU dual-core parallel processing modules were integrated. Through simulation experiments in the enhanced Imazu training environment, the practicality of the method, taking into account the effects of wind and ocean currents, was corroborated. The results demonstrate that the proposed algorithm can perform effective collision avoidance decision making in a range of vessel encounter situations, indicating its efficiency and robust generalization capabilities.

Keywords: automatic collision avoidance decision making; multi-ship encounter situations; deep reinforcement learning; COLREGs

Citation: Xie, W.; Gang, L.; Zhang, M.; Liu, T.; Lan, Z. Optimizing Multi-Vessel Collision Avoidance Decision Making for Autonomous Surface Vessels: A COLREGs-Compliant Deep Reinforcement Learning Approach. *J. Mar. Sci. Eng.* **2024**, *12*, 372. https://doi.org/10.3390/jmse12030372

Academic Editor: Carlos Guedes Soares

Received: 7 January 2024
Revised: 18 February 2024
Accepted: 19 February 2024
Published: 22 February 2024

1. Introduction

With the globalization of the world economy, the number of ships used for marine transportation has increased significantly, leading to more congested waterways. This development has highlighted the importance of safety in navigation. The primary objective of safe navigation for vessels is to enable them to perform effective collision avoidance decision making in complex navigation environments, minimizing or preventing collisions. Compliance with the COLREGs is crucial in preventing collision accidents during ship encounter situations [1]. According to statistics, over 80% of ship collisions are caused by human error, specifically, the crew's failure to adhere to the COLREGs when making crucial decisions [2]. Presently, the COLREGs are based on human-made agreements, leading to open interpretations of specific rules and resulting in difficulty in establishing a uniform standard, which makes it challenging for crew members to take avoidance actions

in compliance with the rules. To solve with these problems, it is recommended to quantify the COLREGs and establish an autonomous collision avoidance decision-making method, which is capable of dealing with various encounter situations. In order to reduce the risk of ship collision avoidance, it is essential to enhance the level of vessel automation technology, reduce human intervention in the decision-making process, and gradually achieve automation and intelligence of vessel operations.

With the development of artificial intelligence algorithms, numerous intelligent algorithms have been applied in ship collision avoidance research. For example, in 1999, Harris used neural networks to investigate ship collision avoidance [3]. In 2000, Smierzchalski et al. utilized genetic algorithms to plan a ship's navigation path, which solved the ship collision avoidance problem [4]. In 2004, Lee et al. developed a fuzzy logic collision avoidance algorithm using an improved potential field method, which satisfied the COLREGs for static and dynamic obstacles [5]. In 2008, Zhuo et al. supported ship drivers in making navigation decisions by using fuzzy neural networks [6]. In 2012, Ahn et al. developed a collision avoidance system by combining expert systems and fuzzy reasoning systems and proposed a method for calculating collision risk using neural networks [7]. In 2012, Su et al. created a database for ship collision avoidance using fuzzy logic theory on the Vessel Traffic Service (VTS) system [8]. In 2015, Szlapczynski et al. proposed a method for collision avoidance that utilized game theory hypotheses and evolution rules [9]. While the method demonstrated robust real-time performance and was effective in dealing with multi-ship collision avoidance problems in water areas, it occasionally violated the COLREGs. In 2018, Gao et al. designed an online ship real-time prediction model based on Automatic Identification System (AIS) data characteristics and bidirectional long short-term memory recurrent neural network [10]. Huang et al. proposed a generalized velocity obstacle algorithm for ship collision avoidance in 2019 and designed a ship collision avoidance system based on this algorithm [11]. In the same year, Xie et al. applied an improved beetle swarm antenna search algorithm to ship collision avoidance prediction [12].

Traditional non-learning-based collision avoidance algorithms for vessels often require extensive computations. Most of these algorithms pre-generate collision-free paths for all foreseeable scenarios based on a known global map, leading vessels to move along predetermined trajectories. Consequently, when creating collision-free paths for multi-vessel encounters, these algorithms tend to be inefficient. The chosen paths may not be optimal, posing limitations in real-time responsiveness of autonomous navigation systems. To address this constraint, reinforcement learning (RL) is emerging as a frequently utilized method in artificial intelligence and control disciplines. As a form of self-supervised learning, RL stands out due to its capacity to learn optimal strategies through interaction with its environment, without the need for explicit labeled data [13]. And its ability to adapt and improve based on feedback from the environment is making it an increasingly prominent tool in various domains. The agents trained through RL are particularly suitable for local planning and in tackling scenarios with unknown environments, given their ability to continuously adapt and derive strategies based on environmental feedback.

The fundamental principle of reinforcement learning is to continuously interact between the agent and the environment, obtain state information and reward functions fed back by the environment, guide the agent's actions, and repeatedly optimize its action policy by trial-and-error training to obtain the maximum reward return [14]. The primary problem that reinforcement learning needs to solve is the adaptive dynamic programming problem based on sequential decision making. Researchers have proposed policy search algorithms for this problem and quantitatively evaluate the quality of the policy by introducing value functions, leading to the development of classical reinforcement learning algorithms, including Q-learning and Sarsa [15–17]. These reinforcement learning methods are usually unable to handle autonomous control problems that require a significant amount of representation capability. With the rapid development of deep learning, its outstanding feature learning capability can well compensate for the shortcomings of these reinforcement learning, leading to a new direction for research into reinforcement learning

algorithms that integrate deep learning technology. Currently, the development of DRL technology has achieved remarkable results. DRL algorithms are a crucial technique in the realm of machine learning, which can effectively tackle decision-making challenges in ship collision avoidance that involve continuous state and action spaces. Ship collision avoidance is a dynamic process in which vessels continuously interact with the surrounding environment, learning and making decisions as they navigate. DRL can offer more flexible solutions for sudden changes in maritime conditions compared to classic collision avoidance algorithms, once it has undergone training and reached convergence. Thus, DRL is an effective method for solving collision avoidance problems.

Mnih et al. proposed a Deep Q Network (DQN) algorithm in 2013 [18] that combines Q-learning with neural networks, achieving a level of play on Atari games that rivals human game players. The algorithm uses neural networks to efficiently compute Q-values, circumventing the need for a computationally demanding Q-table construction. The DeepMind team improved the DQN algorithm in 2015 by proposing two neural networks [19], one for generating actions and the other for evaluating their performance, resulting in enhanced algorithm convergence and training efficiency. Dinh et al. extended the application of DRL to autonomous obstacle avoidance for robots in 2017 [20]. Their approach utilized sensor-derived position and motion information of obstacles around the robot and classic path planning algorithms to enable safe avoidance. To address issues of data diversity, the algorithm leveraged a memory pool to store sample data and randomly shuffled them during training, reducing data correlation and improving neural network convergence. In 2017, Linhai Xie et al. proposed a novel algorithm referred to as Double Deep Q-Network (Double DQN) [21], which utilizes two identical neural networks to estimate actions from images. The proposed algorithm discretizes the robot's motion environment and formulates a reward function to satisfy collision avoidance requirements, allowing the robot to effectively perform turning actions to circumvent obstacles in its current motion environment. Although the algorithm has been shown to be effective, its reliance on prior knowledge of the robot's environmental information limits its versatility in handling significant changes in the robot's motion environment, which in turn restricts its applicability.

The gradient-based methods to optimize deep neural networks has become the prevailing approach, due to the remarkable generalization capabilities of these methods in high-dimensional input spaces. A number of algorithms based on this principle have garnered significant attention, including the deep deterministic policy gradient (DDPG) algorithm proposed by the DeepMind team in 2016 [22], the asynchronous advantage actor–critic (A3C) algorithm proposed by Mnih et al. in 2016 [23], the Proximal Policy Optimization (PPO) algorithm developed by Schulman et al. in 2017 [24], and the soft actor–critic (SAC) algorithm proposed by Haarnoja et al. in 2018 [25]. In the context of continuous control tasks, Tai et al. demonstrated in 2016 that policy gradient methods are generally considered to be more effective [26]. Regarding collision avoidance issues discussed in this article, it is evident that the PPO algorithm outperforms other methods. The PPO algorithm is capable of handling continuous action and state spaces, and exhibits strong robustness in complex environments and reward functions. Zhao et al. [27] and Meyer et al. [28] have previously demonstrated the successful application of the PPO algorithm to multi-ship collision avoidance in 2019 and 2020, respectively. In 2020, L. Engstrom demonstrated that the PPO algorithm can indeed enhance the performance of deep policy gradient algorithms [29]. Additionally, Ryohei Sawada applied PPO in combination with LSTM neural networks to achieve autonomous ship collision avoidance in continuous action spaces [30]. Most recently, in 2021, Thomas Nakken Larsen compared the effectiveness of various DRL algorithms for safe navigation in challenging waterways [31].

Based on a comprehensive review of the relevant literature, it has been concluded that decision making for ship collision avoidance presents several challenges.

Firstly, the majority of research conducted in this field has discretized the ship's environment and actions, neglecting the continuous nature of both factors in actual mar-

itime scenarios. Collision avoidance decision-making algorithms have not adequately accounted for the characteristics of ship movement and lack a comprehensive and specific quantification and implementation of the COLREGs.

Secondly, in situations involving multiple ships, collision avoidance algorithms typically prioritize avoiding each ship based on its level of collision risk, from high to low. However, these algorithms do not take into account the presence of other ships when attempting to avoid the ship with the highest level of collision risk, which can create potential risks to other ships.

Thirdly, classic collision avoidance methods aimed at balancing collision avoidance, ship models, and the COLREGs are no longer feasible in terms of both accuracy and efficiency.

Under conditions where ship positions, navigation, and speed can be observed and obtained from external sensors such as radar and cameras, and focusing on encounter situations involving ships of the same type, this paper proposes a method for ship collision avoidance decision making based on DRL algorithms, which is designed for dealing with diverse encounter situations, including complex multi-ship encounters. The proposed method is intended to improve navigation safety and reduce the incidence of vessel collisions that are attributable to human error. The remainder of this paper is organized as follows: Section 2 introduces the collision avoidance process of autonomous vessels. Section 3 develops a decision-making method based on DRL algorithm. Section 4 describes a design of the simulation environment and experiment, and the algorithm's effectiveness and generalization capability are verified through simulation analysis in a test scenario. Section 5 summarizes the research work and looks forward to the future work. Figure 1 presents the technical roadmap central to this study.

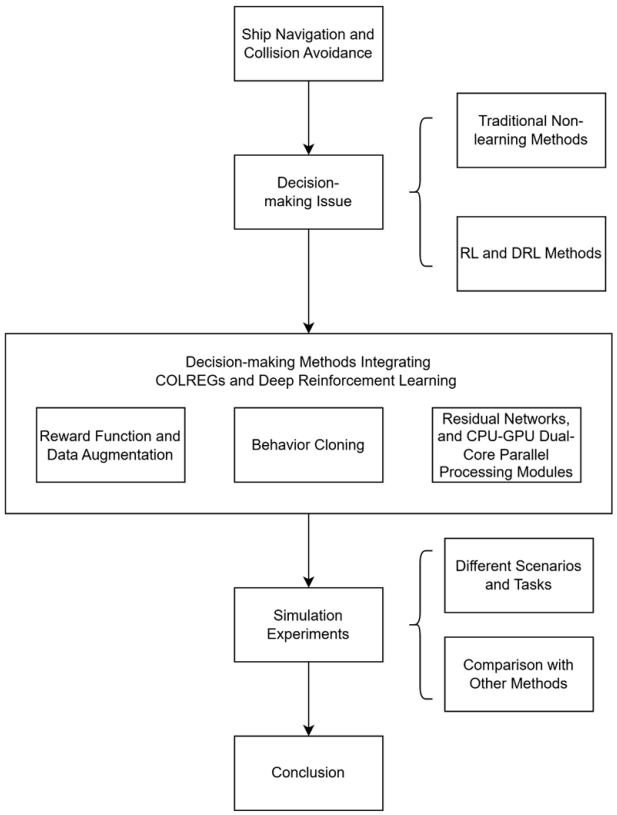


Figure 1. The technology roadmap of this paper.

2. Process of Collision Avoidance

The collision avoidance process of autonomous vessels typically involves four primary stages shown in Figure 2: environmental perception, situation recognition, collision avoidance decision making, and collision avoidance control.

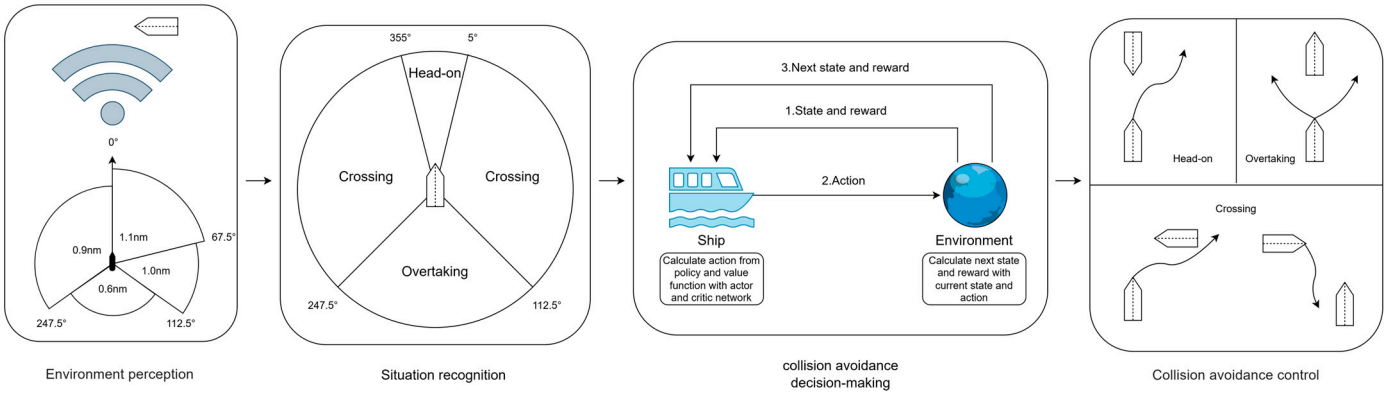


Figure 2. Collision avoidance process.

In the first stage, a range of sensors including radar, cameras, and the Automatic Identification System are utilized to collect data regarding ship positions, speeds, headings, and environmental conditions. In the second stage, based on the collected information, predictions are made regarding the future trajectories of other vessels. This process involves identifying the current encounter situation and assessing potential collision risks. In the third stage, once the current encounter situation is determined, the autonomous vessel will perform collision avoidance decision making based on scheduled rules and algorithms. These decisions may involve altering the course, adjusting the speed, or implementing other measures to avoid collision. In the fourth stage, once the collision avoidance decision making are formulated, the autonomous vessel executes these decisions through an automatic control system. This stage typically involves the utilization of advanced control systems, such as autopilots and dynamic positioning systems, to control vessel motion and ensure it avoids collisions.

The aim of this study is to tackle the problem of collision avoidance in multi-ship encounter situations, where vessels autonomously make decisions to avoid target vessels while adhering to the COLREGs. In this chapter, we considered factors such as the COLREGs, ship domain model, ship maneuvering performance, and other elements relevant to ship risk assessment.

2.1. An Analysis of the COLREGs

The COLREGs serve as the foundation for all vessels navigating on the seas, dictating that all vessels comply with them to coordinate their movements and ensure safe passage. Some related studies have been conducted based on the analysis and application of the COLREGs. M.R. Benjamin et al. focused on the operations of autonomous unmanned marine vehicles to ensure adherence to the COLREGs [32]. Chauvin and Lardjane researched the decision-making process and strategies in maritime interactions [33]. L. P. Perera et al. developed an intelligent decision-making system based on fuzzy logic, guided by the COLREGs [34]. In this study, vessels recognize themselves as the "own ship" (OS) and other vessels in the vicinity as "target ships" (TS), creating collision avoidance decision making based on their observations and the state information they get from nearby vessels. If a TS enters the detection area of the OS, and there is a danger of collision between the two vessels, the OS must react to avoid the TS while following the COLREGs before resuming its initial trajectory towards the destination once safety is guaranteed. According to the COLREGs, every vessel in various encounter situations is obligated to adhere to a specific course to prevent collision. These encounter situations are categorized based on the relative position and direction of the OS and the TS.

2.2. Ship Domain Model

According to Goodwin [35], the domain is "the effective area around a ship which a navigator would like to keep free with respect to other ships and stationary objects". The

ship domain is a generalization of the safe distance and comes from the observation that the safe distance is not the same in all directions. Currently, there are many different ship domains widely used, but each ship domain typically has a different meaning depending on the author's definition or the purpose of developing domain models. Ship domains can be roughly divided into those developed by theoretical analyses, those based on experts' knowledge and those determined empirically [36]. In collision avoidance studies, the ship domain commonly used is based on domains determined through experimental data statistics, where three classic vessel domains are defined as the Fuji ship domain, the Goodwin ship domain, and the Coldwell ship domain.

The concept of a ship domain is used to calculate collision risk areas and to define a safe domain around the OS or TS, which is an area that other vessels should not enter to avoid collision. However, due to the different definitions of ship domains, different safety standards are used for different ship domains. The first safety standard is that the domain of the own ship should not be violated by the target ship; the second is that the domain of the target ship should not be violated by the own ship; the third is that the domains of both ships should not be violated by each other; and the fourth is that the domains of both ships do not overlap and are independent of each other. The difference between these four safety standards is very important because it has a significant impact on the distance between vessels. The Fuji ship domain uses the second safety standard, and the Coldwell ship domain uses the first safety standard. The first two safety standards are asymmetric, using same standards based on different vessel is evaluated can lead to different safety evaluation results. When using the third safety standard, the determined distance between vessels is reasonable for each encounter situation, and it is safe for both ships and conforms to the typical domain definition.

The Goodwin ship domain is widely used for testing and evaluating collision avoidance methods, and the domain is applicable to multiple safety standards. Incorporating factors related to ships in various domain models, the Fuji ship domain and Coldwell ship domain are associated with the ship's length, taking into account ship-specific considerations. On the other hand, the Goodwin ship domain is unrelated to ship factors and does not consider ship-specific elements. At the same time, the Goodwin ship domain definition considers the COLREGs and can be used to simulate various encounter situations. This makes it a universal and flexible tool for evaluating navigation safety and making collision avoidance decision, which is in line with the research needs of this paper. Therefore, this paper adopts the modified Goodwin domain model [37] and the third safety standard for research.

3. Method

In this chapter, we propose a decision-making method based on a DRL algorithm for the process of multi-ship collision avoidance. The primary reason for opting to learn COLREGs through a machine learning framework rather than direct implementation is that machine learning methods are better suited to handling the complex, dynamic, and uncertain marine environments. While COLREGs provide clear guidelines for navigation, the direct application of these rules might not cover all possible encounter situations, especially in situations involving multiple vessels and variable conditions; even if feasible, it could require extensive resources. Machine learning approaches enable the flexible application of these rules across different contexts, thereby enhancing the adaptability and robustness of decision-making processes.

The method process is illustrated in Figure 3. If the OS is the stand-on vessel, it generally does not need to perform collision avoidance actions; it only needs to continue on its original course. Therefore, our main focus of study is when the OS is the give-way vessel. Randomly selecting one of the encounter situations, the OS sails towards its destination and the ship state information is calculated as input for the DRL algorithm. Using the CRI and distance between ships, the OS detects whether the current target ship is in danger of collision. In the collision avoidance phase, the OS is trained using a

specific reward function to avoid collisions. When Time to Closest Point of Approaching (TCPA) and the trend in the distance between ships indicate that the collision risk has been resolved, the OS enters the navigation resumption phase. In this phase, it is trained with a different reward function designed to guide the ship to reach its destination as quickly and efficiently as possible. The algorithm records the reward and re-initializes the environment for the next round of training if the OS reaches its destination or a collision occurs. After a certain amount of training, the OS optimal collision avoidance behavior is obtained for the encounter situations.

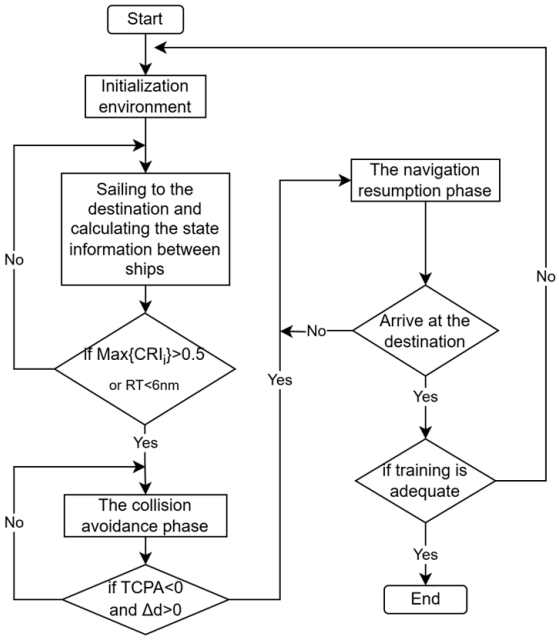


Figure 3. Method implementation process.

The method develops innovations and improvements, primarily in the design of network architecture, initialization, state and action sets, and the reward function.

3.1. The DRL Modeling

The present study develops a collision avoidance algorithm based on DRL, which integrates RL with deep neural networks for enhanced performance. According to Sutton et al. [38], reinforcement learning algorithms are a type of algorithm that involves an agent taking actions based on the environment, with the objective of maximizing the value function calculated through rewards in the present state. To aid in such decision making, RL utilizes the Markov decision process (MDP) framework, which allows agents to interact with the environment and make consecutive decisions. Recently, DRL has become a new approach to RL utilizing multi-layer neural networks as the agent's value function. Due to the outstanding performance and robust data storage capabilities of neural networks, DRL surpasses classical RL algorithms in effectively dealing with increasingly complex problems. DRL algorithms use the powerful modeling capabilities of DL to enable the accurate control of intelligent agents based on complex, high-dimensional inputs, thereby making it possible to solve complex collision avoidance problems.

The present study uses the PPO algorithm, which is an actor–critic DRL algorithm. When addressing control problems that have a particular objective, such as ship collision avoidance [39], the PPO algorithm exhibits strong performance and stable learning convergence in numerous applications. This study focuses on the problem of collision avoidance, where an agent represents its OS and the environment is composed of TS and navigation waypoints. The actor–critic DRL method involves two neural networks. In each round of the Markov decision process, the OS receives the current state $s_t \in S$ and reward r_t from

the ocean environment, where S is the set of all possible states. The actor network selects actions at $a_t \in A$ based on the policy $\pi_{\boldsymbol{\theta}}(a_t|s_t)$, which represents the decision-making process, while the critic network computes the value function $V(s)$ for state s_t to assess the selected action. Here, A is the set of all actions that the agent can choose from the current state s_t and the policy represents the probability distribution of selecting actions at A. Subsequently, the agent executes a_t in the environment and calculates the next state s_{t+1} and reward r_{t+1} based on the current state s_t and the selected action a_t. In this study, the actor and critic networks are used separately and do not share data between them. At the end of each iteration, the policy for each neural network is updated using the obtained rewards and value functions.

The PPO objective function $L^{CLIP}(\boldsymbol{\theta})$ is given by

$$L^{CLIP}(\boldsymbol{\theta}) = \hat{\mathrm{E}}_{\mathrm{t}}\left[\min\left(r_t(\boldsymbol{\theta})\hat{A}_t, \mathrm{clip}(r_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \tag{1}$$

and

$$r_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}_{old}}(a_t|s_t)} \tag{2}$$

$$\mathrm{clip}(x, l, r) = \max(\min(x, r), l) \tag{3}$$

where in Equation (1), $r_t(\boldsymbol{\theta})$ is the ratio of probabilities, which is the ratio of the probability that the policy before the update takes a specific action in a specific state and the probability that the current policy takes the same action in the same state. The clip function is applied to $r_t(\boldsymbol{\theta})$, using between $(1-\epsilon)$ and $(1+\epsilon)$, according to the clipping hyperparameter ϵ with a value of 0.1–0.2. As a result, the value of $L^{CLIP}(\boldsymbol{\theta})$ stably changes within a small range near 1, even in the presence of significant differences between the previous and current policies. The PPO framework utilizes the generalized advantage estimation (GAE) for estimating $\hat{A}_t$:

$$\hat{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+1}^V \tag{4}$$

and

$$\delta_t^V = r_t + \gamma V_\omega(s_{t+1}) - V_\omega(s_t) \tag{5}$$

At the heart of this formula lies the result of the critic network, denoted as $V(s)$, which represents a learned estimation of the state's value. The impact of future rewards is accounted for by the discount factor γ, while λ serves as a coefficient for GAE to regulate the variance and bias of the model.

Upon initialization of the network parameters, the ship's navigational state set serves as input data for the network. In each iteration, the PPO algorithm obtains a batch of Markov chains from the current policy and calculates the reward and GAE functions. The policy is then updated using the PPO objective and the Adam optimizer. Afterward, the value function is fitted using mean square error regression and updated with the gradient descent method. Through a repeated iterative process, the optimal collision avoidance strategy for ships is trained, leading to convergence.

Ships encounter a variety of situations, which require collision avoidance actions that conform to both the constraints of ship maneuverability and the COLREGs. The collision avoidance decision-making process can be effectively modeled using the Markov decision process. We construct a state set s_t that includes the encounter environment information based on the navigational information between ships to ensure that the input neural network has sufficient effective information on ship navigation. The collision avoidance action a_t takes into account ship maneuverability and mainly adopts ship turning behavior to ensure the collision avoidance decision making's implementability. The reward function r_t is the most crucial part of the ship PPO collision avoidance model, as it forms the basis for the ship's collision avoidance decision while taking into account the COLREGs and economic practicality. When there is no risk of ship collision, the ship usually navigates to its destination according to the prescribed course. However, when there is a risk of collision,

the collision avoidance decision making taken by the ship must consider the safety and effectiveness of the collision avoidance process between ships, as well as whether the ship complies with the COLREGs during the collision avoidance process. We summarize the PPO algorithm in Algorithm 1.

Algorithm 1 PPO with the Ships Collision Avoidance

A. Input: ship state s_t, ship action a_t and ship reward r_t
B. Training Procedure:
  Initialize policy network
  Repeat for each epoch:
    a. Collect batch of observations from environment.
    b. Compute action probabilities and values with policy network.
    c. Compute advantages using generalized advantage estimation.
    d. Compute old log probabilities of actions.
    e. Repeat for each update iteration:
      i. Compute new log probabilities of actions;
      ii. Compute ratio of new and old probabilities;
      iii. Compute surrogate loss;
      iv. Compute surrogate loss;
      v. Compute value loss;
      vi. Compute total loss;
      vii. Compute gradients of total loss;
      viii. Update policy network with gradients.
C. Return trained policy network.

3.2. Network Architecture and Initialization Settings

For the design of the neural network structure, we used residual networks. The purpose is to solve the problem of gradient disappearance in the training process of deep neural networks. This problem can cause useful information to not be propagated throughout the network. Our residual networks (as shown in Figure 4a) consist of four residual blocks. Except for the input and output layers, each residual block contains two convolutional layers, a tanh activation function and a skip connection, which allows information to be directly propagated through the residual block, avoiding the problem of gradient disappearance. The convolutional kernel size is set to 3 and the stride is 1. The pooling kernel for the pooling layer is 2 and the stride is 2. The number of hidden units in the linear layer is 128, and the activation function is the tanh function. The dropout parameter is set to 0.5.

This article presents a training strategy that adopts a dual-actor network. The first actor network represents the original policy network and is responsible for action selection and interaction with the environment, while the second actor network serves as the new policy network and is responsible for learning and modification. To facilitate the training process, both actor networks are trained in parallel using a combination of GPU and CPU. During training, the parameters of the two actor networks are independent of each other. The input for the actor networks is the state data set in the algorithm, and the output is a normal distribution that corresponds to the ship's action value. The actor networks are updated using importance weight processing, temporal difference error, and clip function guidance. The critic network takes the state data set as input and outputs the corresponding value function. The critic network is updated using temporal difference error guidance. Finally, the Adam optimization algorithm is utilized to train the PPO-based collision avoidance algorithm for ships.

In the network initialization phase of this study, Behavioral Cloning (BC) was utilized for the pre-training of the policy network. Using BC serves to establish a more stable initial state for the policy network, thereby facilitating the training process of PPO and enhancing the robustness of the algorithm. AIS is a vital tool in maritime navigation, providing accurate, real-time data about a ship's position, course, and speed. Some studies have

investigated ship collision avoidance behavior by decoding AIS data [40]. We decoded the AIS data from July 2021 for the Florida Straits, connecting the Atlantic waters. This process focused on extracting key navigational parameters such as location, heading, speed, and course alterations. To identify collision avoidance trajectories from the AIS data, we applied a method similar to the Sliding Window Algorithm. Informed by these real-world vessel trajectories as shown in Figure 5 and by integrating the reward function developed in this study, representative avoidance trajectories for three distinct encounter situations were filtered out from a dataset of 321 real-ship trajectories. This filtering process generated an expert strategy dataset comprising 1918 state–action pairs that exemplify efficient avoidance tactics. Utilizing this dataset, supervised learning was executed. Given a state s_t and its corresponding expert action a_{expert}, the objective of the policy network $\pi_{\boldsymbol{\theta}}(a_t|s_t)$ is to minimize the difference between its output action and the expert action using the mean squared error loss.

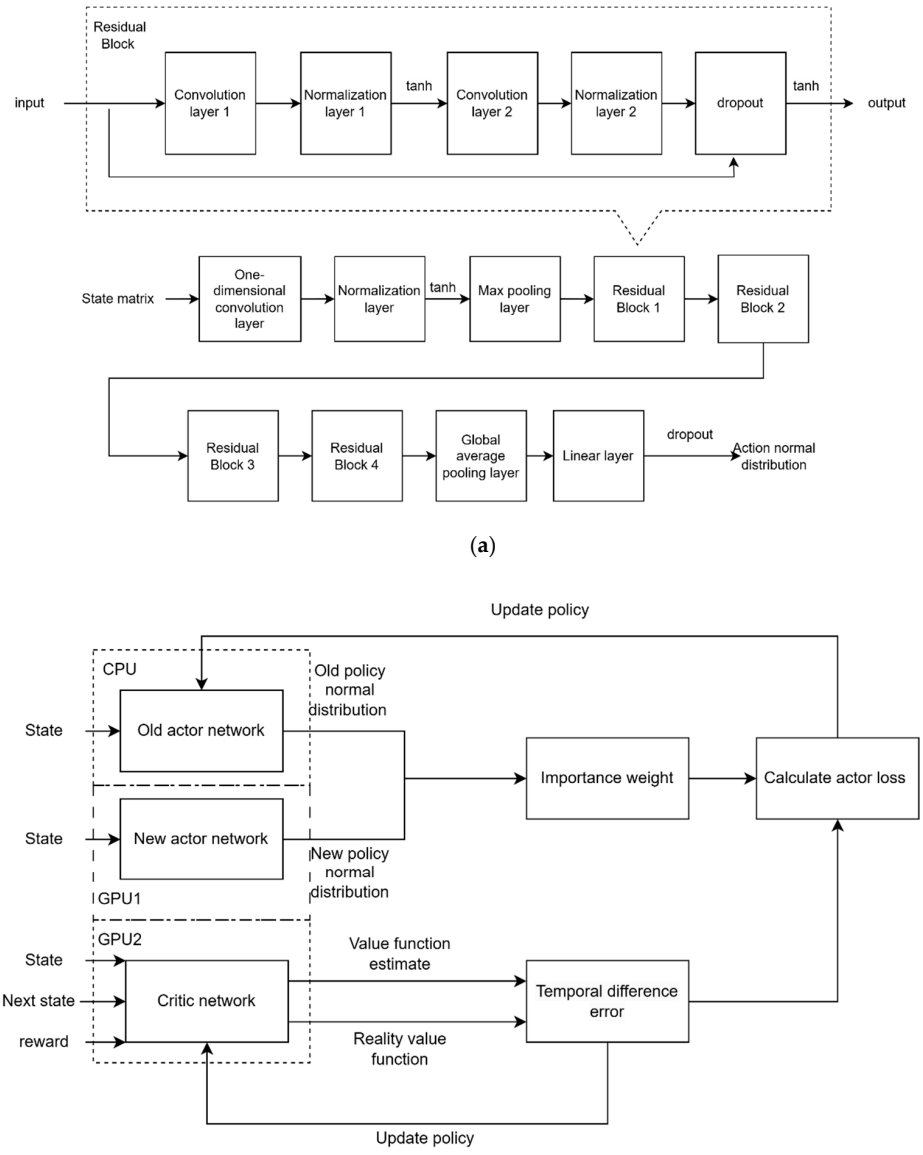
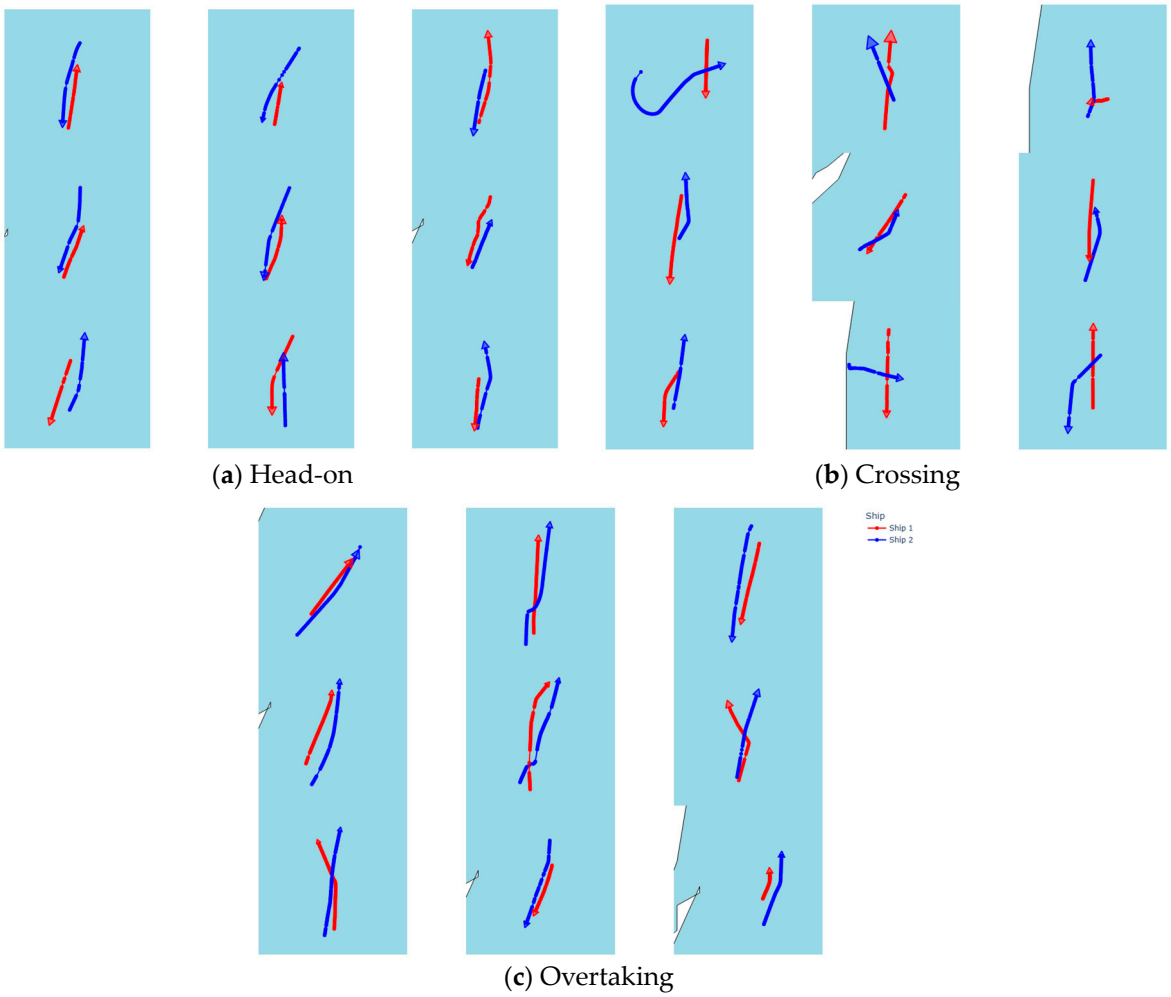


Figure 4. Improvements in algorithmic networks. (**a**) Design of residual network framework. (**b**) Policy update process with GPU and CPU dual core parallel design.

$$L = MSE\left(\pi_{\theta}(s), a_{\text{expert}}\right) \tag{6}$$



(**a**) Head-on (**b**) Crossing

(**c**) Overtaking

Figure 5. Actual vessel collision avoidance trajectories.

Using the aforementioned loss function, we undertook pre-training for 100 iterations. The pre-training utilized the Adam optimizer with a learning rate set to 0.001 and a batch size of 64. Upon completing the pre-training, the resulting policy network was transitioned into the PPO algorithm as the initial neural network setup for the subsequent main training phase.

3.3. The Design of the State and Action Sets

The state set is defined as the information received by the agent from the environment during each training round. Both the actor and critic networks of the PPO algorithm require observable states as input information, which are subsequently used to output the turning actions in the action set for interaction with the environment and updating the algorithm. To optimize the efficiency of the algorithm, the perception field of the ship is divided into 30 regions, with each region spanning 12°. Ship collision avoidance actions are typically initiated when the distance between ships is 6 nautical miles, and the ship records and perceives the surrounding environment within a radius of 6 nautical miles. Each region is bounded by an arc length of L = 2327.29 m, which guarantees that only one target ship is present within each region and is also less than the safety distance required for most ship navigation.

The three crucial ship navigation information factors selected are distance d relative to the target ship, relative bearing θ, and speed ratio k. In addition, to assess the level of urgency in ship navigation, TCPA and Distance of Closest Point of Approaching (DCPA) are selected as state factors. The current ship phase p is also selected as the sixth information element in the state set, and is determined by TCPA and the changing trend of the distance between ships, indicating the present ship phase. The collision avoidance and navigation resumption phases are represented by 1 and 0, respectively.

In conclusion, the PPO state set includes six essential ship navigation information elements and distributes them according to the TS's region during the encounter.

$$S_t = \begin{bmatrix} d_i \\ \theta_i \\ k_i \\ Tcpa_i \\ Dcpa_i \\ p_i \end{bmatrix} = \begin{bmatrix} d_1, d_2, \cdots, d_{30} \\ \theta_1, \theta_2, \cdots, \theta_{30} \\ k_1, k_2, \cdots, k_{30} \\ T_1, T_2, \cdots, T_{30} \\ D_1, D_2, \cdots, D_{30} \\ p_1, p_2, \cdots, p_{30} \end{bmatrix} \tag{7}$$

Specifically, when the TS is not detected in the immediate vicinity of the OS, the PPO state set will utilize the lowest ship danger value. The relative distance between the OS and TS will be set to 6 nautical miles, with the relative bearing and ship speed ratio both set to 0. The TCPA will be set at 2 h, and the DCPA will be set at 6 nautical miles. During the navigation resumption phase, the state set will be taken as 0.

The design of the action set must take into consideration the ship's maneuverability and fuel economy. To avoid potential damage to the main engine and unnecessary fuel consumption, the action set has been designed to prioritize maintaining speed while changing direction. Specifically, the turning angle range of the ship's avoidance behavior is defined as a continuous action set denoted by: $A = \{-\Delta\psi, \Delta\psi\}$, where a left turn is denoted by a negative value and a right turn is denoted by a positive value. Furthermore, the turning angle $\Delta\psi$ is set to a value greater than 0, which can be reasonably determined based on the ship's operational characteristics, economy, and practicality.

3.4. The Reward Function Settings

In this research, the design of the reward function for ship collision avoidance in the PPO algorithm is a critical aspect. The objective is to enable the ship to navigate to its destination when there is no risk of collision. However, when collision risk is present, the ship must make an effective decision that prioritizes safety, economic practicality, and compliance with the COLREGs.

To this end, the ships are classified into two navigation modes during the training process, and reward functions suitable for each mode are chosen based on their characteristics. In a multi-ship encounter situation, the algorithm integrates the reward functions of each target ship for network training optimization. During multi-ship simulations, the sum of the rewards is utilized as the optimization evaluation target for actor–critic network training. The OS perceives the presence of collision risk and enters the collision avoidance phase. The rewards received during this phase are based on the ship's operational decisions, which ensure compliance with the COLREGs and efficient collision avoidance actions. Figure 6 depicts the collision avoidance phase, which utilizes various reward functions, including collision risk index (CRI) reward, encounter situation reward, target and collision reward, and navigation rules reward. After successfully avoiding collision, the ship enters the navigation resumption phase, and the reward function during this phase assesses whether the ship's operational decision is efficient resumption actions. This phase uses the resumption reward function and the navigation rules reward function. The influence weights of each reward function on the experimental results are considered in this study. To normalize the effect of each reward on the network, the value of each reward is adjusted to be within -1 and 1. The range of [–1, 1] is utilized to calibrate the reward functions, taking into account the unique requirements of each stage and training objective. A significant number

of training steps are necessary for the ship to reach its destination from the starting point. After the number of training steps reaches the batch size, a policy iteration is performed, and the entire training process continues until the maximum iteration value is reached. The total reward is the sum of rewards for each iteration. In training process, the ship's collision avoidance performance can be assessed by evaluating the changes in the total reward.

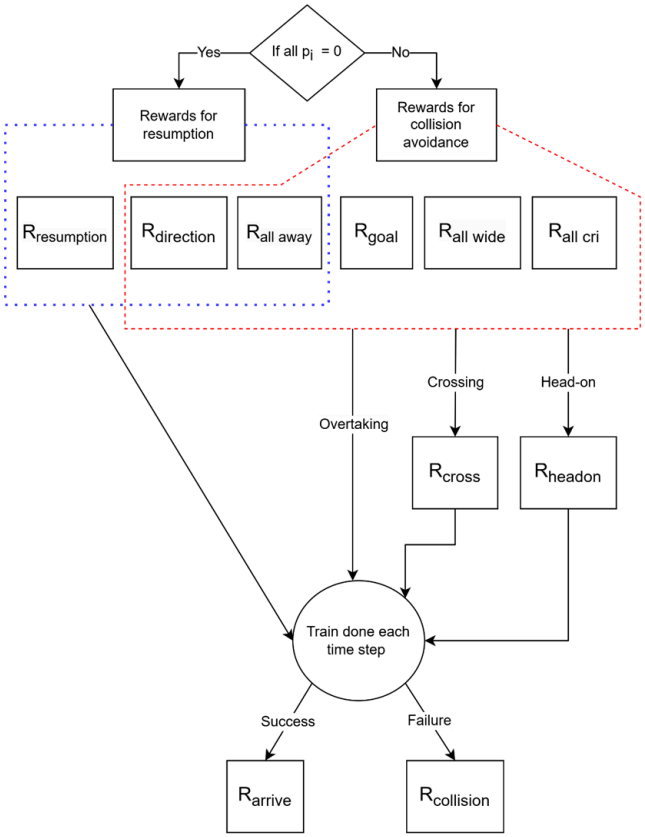


Figure 6. Diagram of reward function based on the COLREGs.

3.4.1. The Collision Risk Index Reward Function

The CRI is a fundamental concept in the field of ship collision avoidance and serves as a parameter to assess the possibility of collision between encountering ships. It is influenced by various factors, including DCPA, TCPA, ship speed ratio, distance, relative bearing, and, in some cases, visibility, water conditions, and human factors. The CRI is used to divide ship encounter situations and determine the appropriate timing for collision avoidance action. This paper proposes a PPO algorithm that uses the CRI, which is closely related to the state set elements, to design the reward function.

This study improves the revised Goodwin ship domain's membership function. This function primarily includes the DCPA, TCPA, ship speed ratio, distance, and bearing angle, and it combines the membership functions of DCPA and TCPA. This model facilitates a sophisticated risk evaluation by allowing a high *CRI* to emerge from low DCPA and TCPA values under specific conditions, thereby accounting for the interplay of additional relevant factors. This model is also designed to balance the influence of diverse factors more equitably, thereby preventing the skewed risk evaluations that might result from overly focusing on high TCPA or DCPA values in isolation. The setting of the membership functions for each parameter is guided by the principles of fuzzy set theory, aiming to better represent the gradations of these parameters.

$$CRI = \alpha_{\mathrm{cpa}} \sqrt{u_{\mathrm{Dcpa}} \cdot u_{\mathrm{Tcpa}}} + \alpha_{\mathrm{d}} u_{\mathrm{d}} + \alpha_{\theta} u_{\theta} + \alpha_{k} u_{\mathrm{k}} \tag{8}$$

The weights of the parameters are assigned such that the sum of the weights is equal to unity.

$$\alpha_{\text{cpa}} + \alpha_{\text{d}} + \alpha_{\theta} + \alpha_{k} = 1 \tag{9}$$

The *CRI* value directly reflects the level of danger in the ship encounter situation. Initially, DCPA and TCPA are given more weight. As the research progresses, greater weight is gradually applied to ship speed ratio, distance, and relative bearing. Therefore, the design of *R*cri is defined by Equations (10) and (11).

$$R_{\text{cri},i} = \lambda_{\text{correct1}}(\lambda_{\text{correct2}} - CRI_i) \tag{10}$$

$$R_{\text{all cri}} = \sum_{i=1}^{T} R_{\text{cri},i} \tag{11}$$

where $\lambda_{\text{correct1}}$ and $\lambda_{\text{correct2}}$ are correction coefficients for the reward function, dividing the reward function into positive and negative rewards with *CRI* = 0.5 as the boundary. This guides the network in identifying the current encounter situation and the urgency of collision avoidance action. T is the total number of target ships in the encounter situation, i is the target ship's number, and CRI_i is the *CRI* of the target ship i.

3.4.2. The Encounter Situation Reward Function

The focus of the study is on developing a reward function to handle specific encounter situations in compliance with the COLREGs. Therefore, we propose a reward function for collision avoidance action amplitude that takes into account the DCPA between the vessels to determine whether the avoidance behavior meets the requirements for sufficiency.

$$R_{\text{wide},i} = \begin{cases} 0 & , \text{DCPA}_{\text{safe}} \le D \\ -\frac{\text{DCPA}_{\text{safe}} - D}{\text{DCPA}_{\text{safe}} - \text{DCPA}_{\text{danger}}} & , \text{DCPA}_{\text{danger}} \le D \le \text{DCPA}_{\text{safe}} \\ -1 & , 0 \le D \le \text{DCPA}_{\text{danger}} \end{cases} \tag{12}$$

$$R_{\text{all wide}} = \sum_{i=1}^{T} R_{\text{wide},i} \tag{13}$$

where $\text{DCPA}_{\text{safe}}$ is the minimum DCPA between two vessels when they are deemed to be in a safe state, and $\text{DCPA}_{\text{danger}}$ is the minimum DCPA between two vessels when they are in a dangerous state. These two metrics are determined based on the DCPA membership function.

In a head-on situation, the OS must turn to the right and pass from the left side of the TS to avoid collision. At the closest point of approach, it is important to ensure that the OS is positioned on the left side of the TS. This is represented by the relative bearing angle $\theta \in [180^\circ, 360^\circ]$. The ideal relative bearing angle for a head-on situation is 270°, for which we have formulated a reward function using a sine function. The reward function for head-on situations is depicted in Equation (14).

$$R_{\text{headon}} = -sin\theta \tag{14}$$

In a crossing situation, the OS must alter its course and pass behind the stern of the TS to prevent crossing ahead of it. The relative bearing angle $\theta \in [90^\circ, 270^\circ]$ is used to determine if our vessel has crossed the other vessel's bow. We formulated a reward function for crossing situations, which is defined as Equation (15).

$$R_{\text{cross}} = \begin{cases} 1, & \theta \in [90^\circ, 270^\circ] \\ -\lambda_{\text{cross}} cos\theta, & \theta \notin [90^\circ, 270^\circ] \end{cases} \tag{15}$$

where λ_{cross} is the reward value adjustment coefficient for the crossing situation.

The function's visualization is provided in Figure 7.

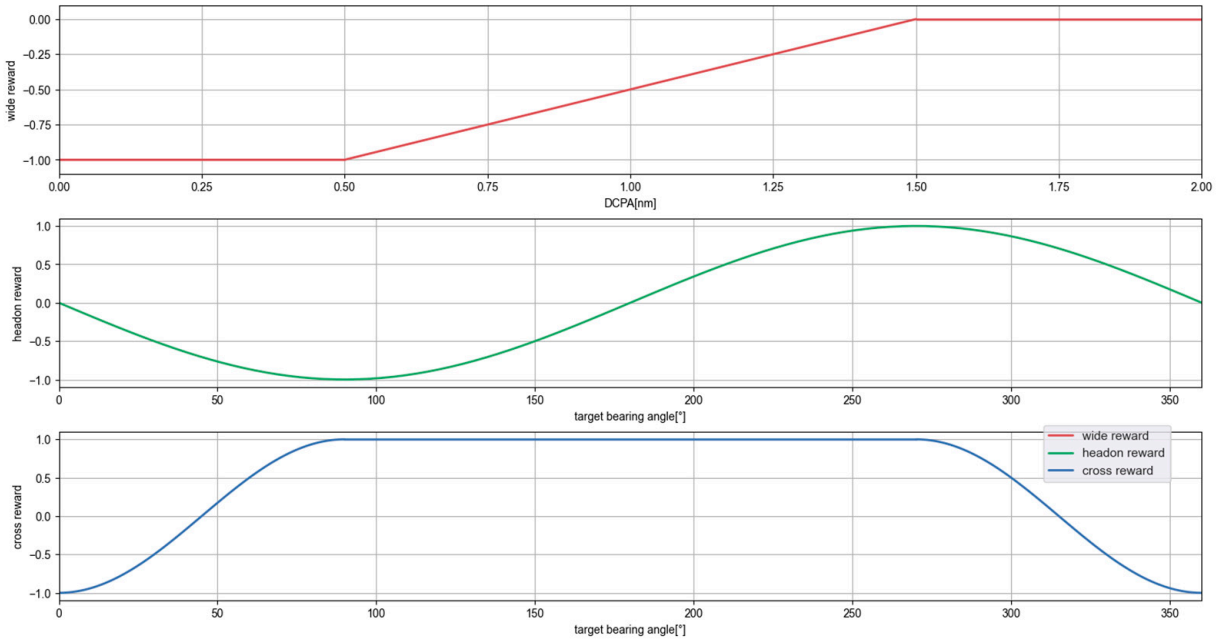


Figure 7. Encounter situation reward function curve.

3.4.3. The Target and Collision Reward Function

The target reward function for the collision avoidance phase is based on the ship's current position and its destination.

As illustrated in Figure 8, once the ship receives an avoidance command, it moves to the next position (x_{Ot+1}, y_{Ot+1}) from its current position (x_{Ot}, y_{Ot}). The ship's action area is represented by concentric circles at the current position with the center located at (x_{Ot}, y_{Ot}) and a radius equal to the product of the unit time and speed. Equation (17) is designed to calculate the target reward function for the collision avoidance phase.

$$L_t = \sqrt{\left(x_{Ot} - x_{\text{goal}}\right)^2 + \left(y_{Ot} - y_{\text{goal}}\right)^2} \tag{16}$$

$$R_{\text{goal}} = \frac{L_t - L_{t+1}}{V_O \cdot dt} \tag{17}$$

where V_O is the OS speed, and dt is the unit time step. L_t and L_{t+1} is the distance between the OS and the destination at the current and next time steps, respectively. In cases where the ship has not reached the destination, the denominator of Equation (17) represents the distance the ship can move in unit time, while the numerator indicates the difference in distance to the destination at each time step. Positive feedback is provided when the ship moves to the destination (ship 1 in Figure 8) and negative feedback is provided when it moves in the opposite direction (ship 2 in Figure 8). The magnitude of the action is also considered in determining the degree of reward, and the reward function is normalized in the range $[-1,\ 1]$.

Following the navigation resumption phase, the original target reward function is enhanced to enable quick arrival at the destination and accelerate algorithm convergence. To facilitate the agent's exploration, a potential energy function is designed by incorporating the principles of potential energy functions used in physics and the specific environment of the study. Equation (19) describes the navigation resumption reward function.

$$L_s = \sqrt{\left(x_{\text{start point}} - x_{\text{goal}}\right)^2 + \left(y_{\text{start point}} - y_{\text{goal}}\right)^2} \tag{18}$$

$$R_{\text{resumption}} = \begin{cases} \lambda_{\text{correct3}}(L_s - L_t), (L_t - L_{t+1}) > 0 \\ -\lambda_{\text{correct4}}(L_s - L_t), (L_t - L_{t+1}) \leq 0 \end{cases} \tag{19}$$

A potential energy function is utilized to encourage the agent to move from a high potential energy state to a low one, with increasing rewards as the agent approaches the goal. Conversely, the agent is penalized when transitioning from a low-potential-energy state to a high one. The potential energy function is calculated using the L_t and the distance L_s between the starting point and the goal. Correction coefficients, $\lambda_{\text{correct3}}$ and $-\lambda_{\text{correct4}}$, are used to adjust the range of the reward and punishment.

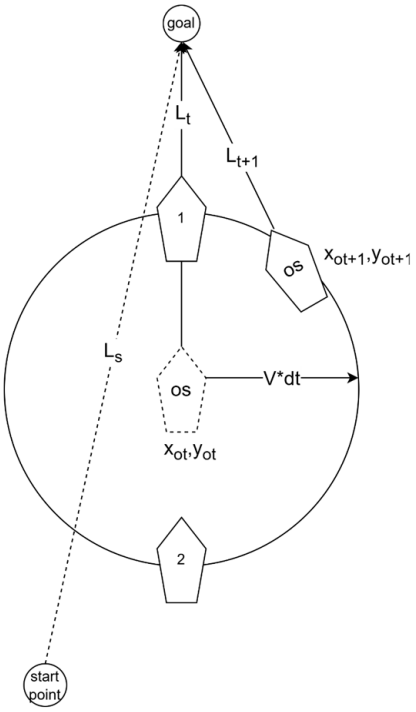


Figure 8. Diagram of the target reward function.

In the training process, if the distance between vessels is less than the Safe Distance of Approach (SDA), a collision occurs, and a penalty value $-r_{\text{collision}}$ is applied. Equation (20) describes the collision reward function.

$$R_{\text{collision}} = \begin{cases} 0, \text{ otherwise} \\ -r_{\text{collision}}, \ d \leq SDA \end{cases} \tag{20}$$

Finally, a final state reward r_{arrive} is used to motivate the OS to reach the destination.

$$R_{\text{arrive}} = \begin{cases} r_{\text{arrive}}, L_t = 0 \\ 0, L_t \neq 0 \end{cases} \tag{21}$$

This approach is effective in encouraging the OS to maintain a safe distance from the TS and arrive at the destination quickly.

3.4.4. The Navigation Rules Reward Function

This study proposes an innovative approach for regulating ship behavior, utilizing a continuous function represented by Equations (22) and (23). The function incorporates the

sparse collision reward function to guide the OS towards a better understanding of how to avoid ship collisions.

$$R_{\text{away},i} = \begin{cases} 0, & d_{\text{safe}} \le d \\ -\frac{d_{\text{safe}} - d}{d_{\text{safe}} - d_{\text{danger}}}, & d_{\text{danger}} \le d \le d_{\text{safe}} \\ -1, & d \le d_{\text{danger}} \end{cases} \tag{22}$$

$$R_{\text{allaway}} = \sum_{i=1}^{T} R_{\text{away},i} \tag{23}$$

where d_{safe} and d_{danger} are the distances between two ships during encounters that are deemed to be in a safe state and dangerous state, respectively. The values of both are determined by the distance membership function.

To address the problem of ships repeatedly circling or stopping to obtain local rewards, while ignoring their original objectives, we developed a heading keeping reward function, represented by Equation (24).

$$R_{\text{direction}} = \begin{cases} 0, & \Delta\varphi_O \le \Delta\varphi_{\text{O,max}} \\ -\frac{\Delta\varphi_O - \Delta\varphi_{\text{O,max}}}{\Delta\varphi_{\text{O,ban}} - \Delta\varphi_{\text{O,max}}}, & \Delta\varphi_{\text{O,max}} \le \Delta\varphi_O \le \Delta\varphi_{\text{O,ban}} \\ -1, & \Delta\varphi_O \le \Delta\varphi_{\text{O,ban}} \end{cases} \tag{24}$$

where $\Delta\varphi_O$ is the deviation between the ship's current heading and the target heading. The maximum allowable deviation is represented by $\Delta\varphi_{\text{O,max}}$, while $\Delta\varphi_{\text{O,ban}}$ represents the heading deviation value that must be avoided.

4. Simulation Analysis

In response to the PPO-based ship collision avoidance algorithm proposed in Section 3, this study performs simulation experiments to verify the algorithm's effectiveness across various ship encounter situations. The simulation results are analyzed to determine if the collision avoidance strategies used in different scenarios comply with the COLREGs.

4.1. Collision Avoidance Experiment Design

4.1.1. Experimental Environment and Scenarios

The environment includes OS, TS, navigational waypoints, and a target area. The target area is presumed to be a region of the sea devoid of any obstacles such as buoys or coastline. The navigational waypoints are established as the OS's intended destination, and the task of the OS is to arrive the waypoints while avoiding the TS. The Imazu problem is utilized to locate the TS in the target area, and the effectiveness of the PPO algorithm is validated by simulation of testing scenarios.

The effectiveness of the collision avoidance algorithm during the training process depends on the encounter situations used. A comprehensive scenario should consist of both simple one-on-one encounters and intricate one-on-multiple ship situations, such as the Imazu problem. Cai et al. proposed an evaluating of marine traffic simulation system for collision avoidance capability and proved the effectiveness of the Imazu problem [41]. Therefore, the Imazu problem has been selected as the training scenario for the algorithm, as it satisfies the requirements of this study's simulation experiments. As the typical Imazu problem 8 is fundamentally similar to other scenarios in terms of ship collision avoidance decision making and encounter situations, this article optimizes it. The training scenario is illustrated in Figure 9. Each box in Figure 9 depicts an encounter situation illustrated by a case of the Imazu problem, where "os" signifies the own ship, "ts" signifies the target ship, and the short bars indicate the velocity vectors of each ship. Allowing the target ship to avoid collision risks may make solving the problem easier. So this study utilizes a more precise experimental approach, where the position and heading of each ship in each scenario are set to intersect at the spatial coordinate origin. Moreover, the TS can only

moves straight ahead without altering course and relying on the OS's turning behavior to avoid collision.

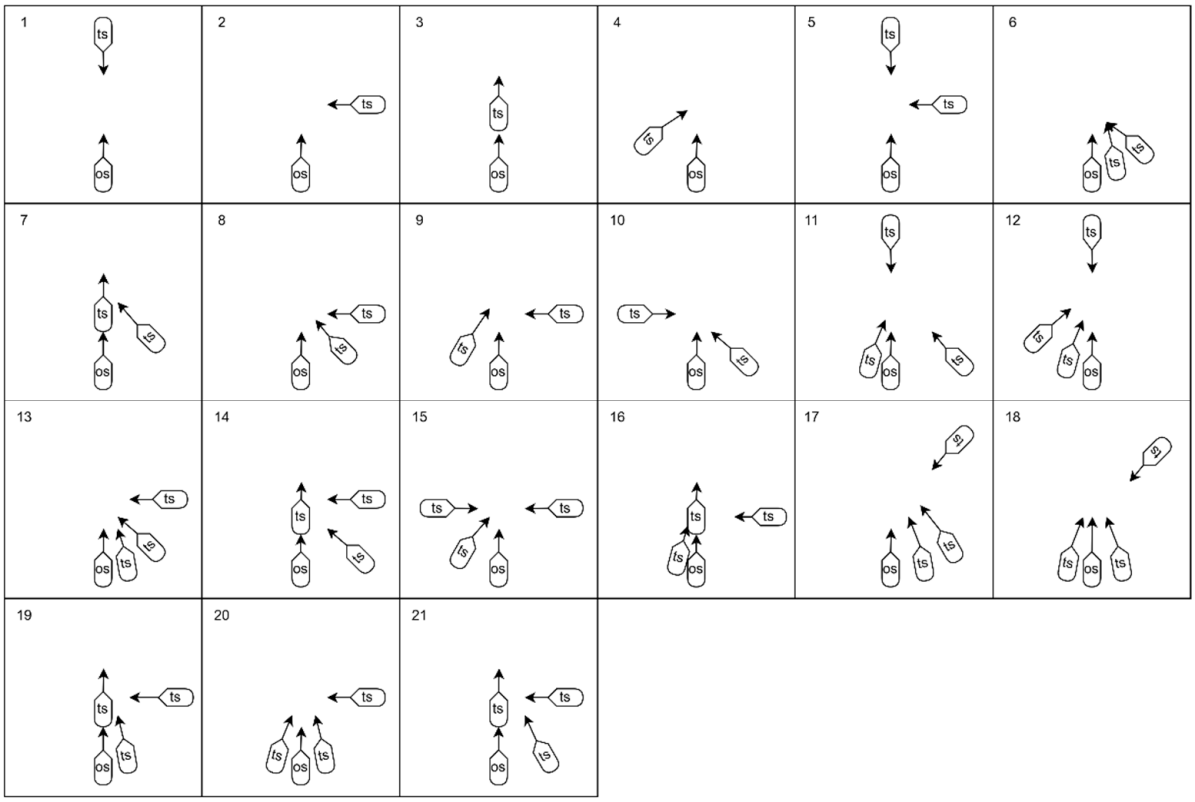


Figure 9. The optimized Imazu ship encounter problem.

During training, a randomly selected situation from the Imazu problem was used, and the target ship was placed in accordance with the specific configuration of each situation. The Imazu problem parameters are listed in the Appendix A Table A1. An OS with an initial position (X_O, Y_O), heading φ_O, speed V_O, and other parameters was established. Additionally, a certain number of target ships of the same type as the OS were placed according to the Imazu problem. For various encounter situations, the initial position (X_t, Y_t), heading φ_t, speed V_t, and other parameters of target ships were set to ensure they would collide in 30 min at a fixed coordinate. Incorporating the influence of wind, waves, and currents is paramount during the training phase. To simplify their representation, Gaussian noise was introduced to the positions, velocities, and headings of both the own ship and the target ships. Specifically, a standard deviation was established to simulate the uncertainties arising from these maritime disturbances. This noise was generated randomly in each simulation iteration. For any given parameter x, its noise can be denoted as:

$$x' = x + \sigma \cdot \mathcal{N}(n, k) \tag{25}$$

where $\mathcal{N}(n, k)$ signifies a standard normal distribution with mean n and standard deviation k, and σ represents the intensity of the Gaussian noise. In particular, for the noise associated with the positions and velocities of both the own ship and the target ship, we used $\mathcal{N}(0, 1)$. Meanwhile, for the heading noise, $\mathcal{N}(0, \pi/12)$ was used.

During training, the algorithm should reinitialize the parameters and restart the training process under three specific conditions: (1) collision between the OS and a TS, (2) successful avoidance of all target ships and reaching the destination, and (3) the completion of a preset number of training episodes or attainment of a negative reward threshold. The objective of the training is to enable the OS to solve the Imazu problem by consistently

reaching its destination without any collisions. Table 1 provides the parameters of Section 3 for each of the reward functions.

Table 1. Parameters of the reward function based on the COLREGs.

Parameter	Interpretation	Value
α_{cpa}	Weighting of CPA membership function	0.3
α_{d}	Weighting of distance membership function	0.3
α_{θ}	Weighting of target bearing angle membership function	0.25
α_{k}	Weighting of velocity ratio membership function	0.15
$\mathrm{DCPA}_{\mathrm{safe}}$	DCPA in safe state	1.5 nm
$\mathrm{DCPA}_{\mathrm{danger}}$	DCPA in danger state	0.5 nm
d_{safe}	Safe separation distance	2.5 nm
d_{danger}	Dangerous proximity distance	1.5 nm
$\Delta\varphi_{\mathrm{O,max}}$	Maximum acceptable course deviation	45°
$\Delta\varphi_{\mathrm{O,ban}}$	Prohibited course deviation	90°

4.1.2. Algorithm Parameters and Optimization Strategies

The parameters of the PPO algorithm are provided in Table 2.

Table 2. Hyperparameter of PPO algorithm.

Parameter	Value
Episodes	4000
Discounted factor	0.96
Reuse times	8
Clip epsilon	0.2
Lambda	0.98
Learning rate of actor network	2×10^{-5}
Learning rate of critic network	5×10^{-3}

Specifically, "Episodes" refers to the aggregate number of training iterations. "Discounted factor" denotes the extent to which the agent must consider future rewards when executing each step. "Reuse time" refers to the number of times each sample within the Relay Buffer should be recycled during training. "Clip epsilon" represents the coefficient used to restrict the policy update within the trust region by clipping the GAE. "Lambda" is the coefficient utilized to adjust the variance and bias of the GAE. Finally, the "Learning rates" for both the actor and critic networks correspond to the frequency of updates made to each respective network during the training process.

To enhance the performance of the PPO algorithm, some tricks were implemented:

1. The GAE calculation was optimized by normalizing the advantage and state.
2. The actor network was set to output actions using a bounded Beta distribution instead of a Gaussian distribution.
3. The Adam optimizer's eps value was set to 1×10^{-5}.
4. Orthogonal Initialization was used as the neural network initialization method.
5. Gradient clipping was incorporated to prevent the possibility of gradient explosion during the training process.
6. Reward scaling was used to rescale the reward function.
7. To improve the exploration capability of the algorithm, a policy entropy term was incorporated into the loss function of the actor network, with a coefficient set at 0.01.
8. Linear learning rate decay was utilized to improve the training process's stability in later stages and enhance training effectiveness.

The modified PPO algorithm is referred to as PPO-Max in this paper and was compared to the classic PPO algorithm in the experimental analysis.

4.2. Results

In this section, the results of an experiment on simulation effects are presented and evaluated. Specifically, the performance of three ship agents—the classic PPO algorithm, the PPO algorithm with a dual-core parallel module, and the PPO-Max algorithm with a dual-core parallel module—are compared for the Imazu problem. After training for 4000 episodes, the collision rate of the three ship agents significantly decreased. The training reward curve is shown in Figure 10.

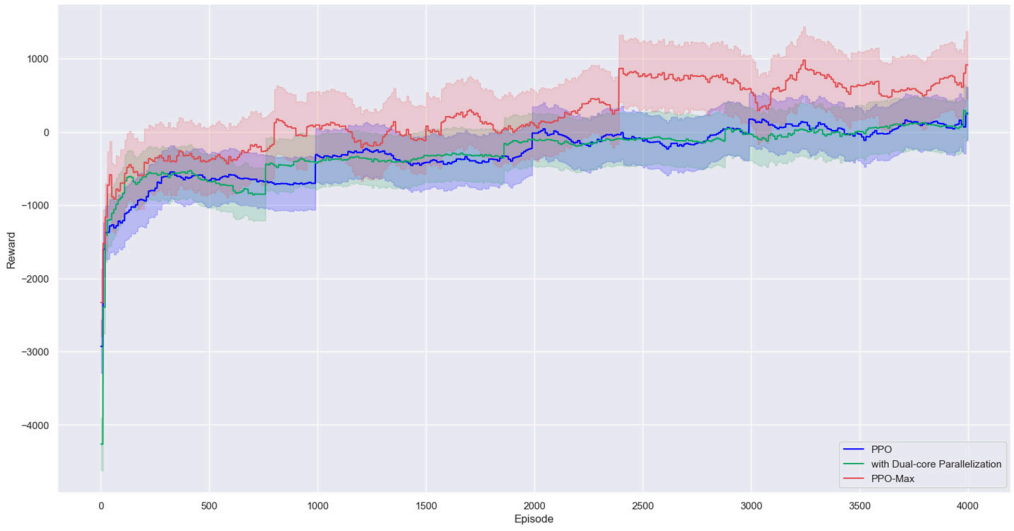


Figure 10. The reward curve of the agent during the training phase.

In Figure 10, the shadow curve and solid line represent the original and the smooth moving average rewards, respectively, for the three algorithms after 4000 episodes of training. The reward curve trend for each algorithm transitioned from a sudden increase to a gradual increase and ultimately converged. Notably, the PPO classic algorithm with the dual-core module did not exhibit a significant difference in the reward curve. Since PPO does not require data communication between the actor and critic agents during operation, there is no effect on experimental performance from setting the two modules to function on the GPU and CPU, respectively. However, it substantially improved the training speed by 48.70% (from 7.68 it/s to 11.42 it/s), resulting in considerable time savings. The PPO-Max algorithm with the dual-core module, utilized in this study, demonstrated a significantly enhanced training effect compared to the previous two algorithms. The incorporation of high-efficiency tricks, such as learning rate decay, resulted in a linear decrease in the ship's learning rate during navigation resumption phases. This ultimately allowed the OS to get the final reward of reaching its destination, while maintaining the characteristics of the ship's navigation with little change in course. As illustrated in Table 3, the fluctuations in collision avoidance success rate observed during the training process follow a similar trend to that of the reward curve. Figure 11 illustrates the ship's trajectory in the Imazu problem.

Table 3. Collision avoidance success rate during the training process.

Episode	0–1000	1000–2000	2000–3000	3000–4000
Success rate	35.7%	69.9%	88.2%	98.6%

In the Imazu problem, a total of 21 ship collision avoidance cases are examined and categorized into four groups based on the type of collision: head-on, crossing, overtaking, and other. By quantifying the COLREGs for head-on, crossing, and overtaking situations, the vessel's trajectory in these situations is found to be in compliance with applicable regulations and practical conditions, rendering it a useful reference for ship collision avoidance in real scenarios.

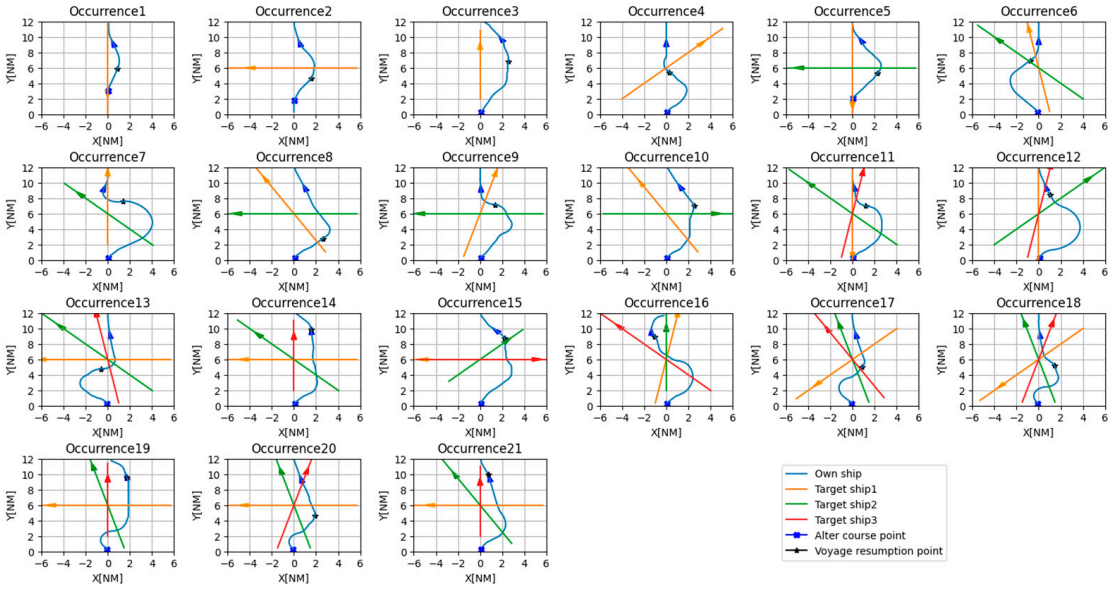
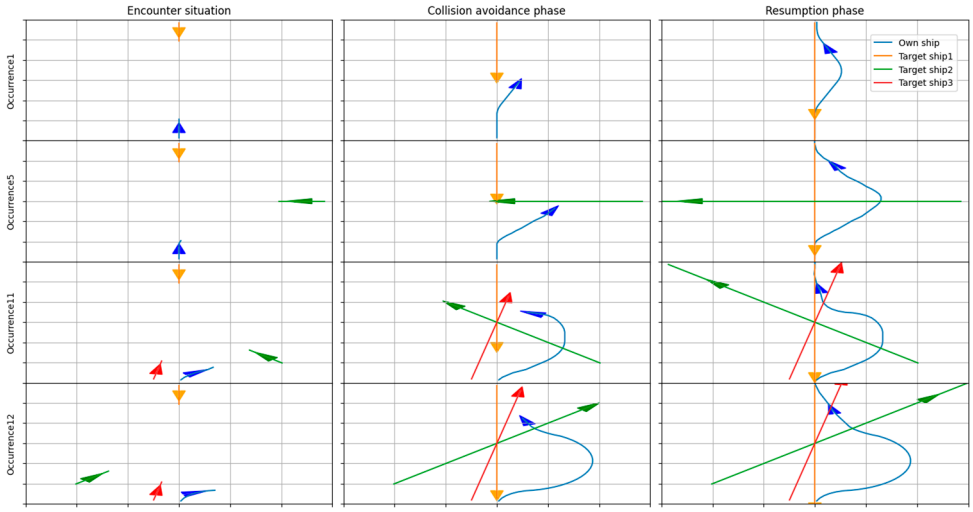


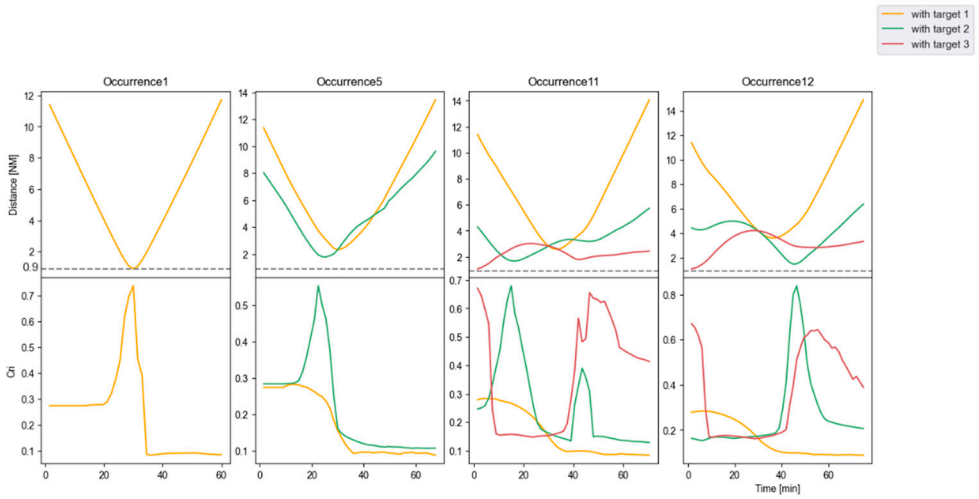
Figure 11. Trajectory of ship in Imazu problem.

The Imazu problem in the head-on situation, as presented in Figure 12, is selected for this study. To avoid collisions, the OS altered its course to starboard so that it would pass on the port side of the TS. In the event of multiple vessels approaching, the OS takes collision avoidance measures based on the CRI. Figure 12b shows the distance and CRI curves between the vessels in the head-on situation. At the start of the head-on encounter, there is a rapid decrease in the minimum distance between the two vessels, indicating a high risk of collision. Subsequently, the OS utilizes collision avoidance action based on the output from the algorithm, leading to a decrease in the distance between the vessels followed by an increase. The CRI curve displays the opposite trend, peaking in the range [0.6, 0.8] during the critical moment of the encounter. Ultimately, the collision avoidance process is accomplished, and the vessels maintain a safe encounter distance, indicating the safety and effectiveness of the collision avoidance decision making.



(**a**)

Figure 12. *Cont.*

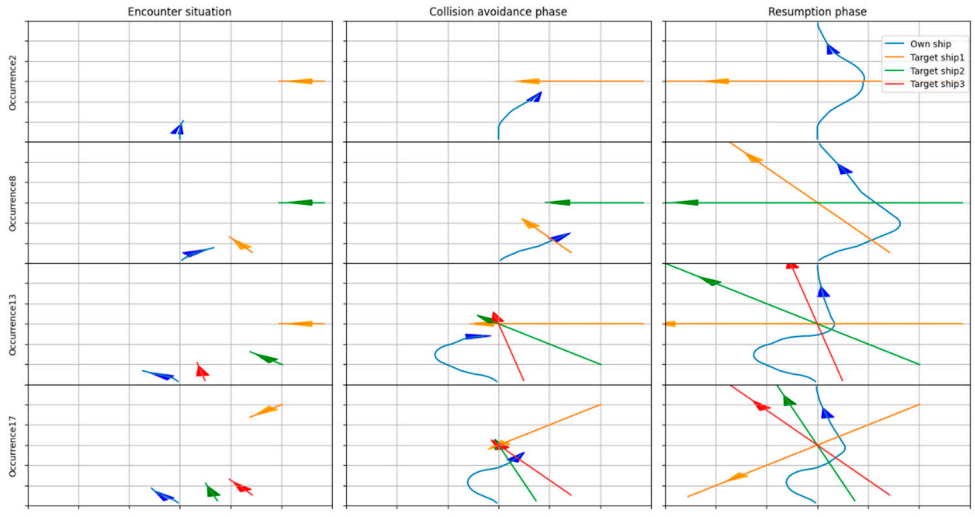


(**b**)

Figure 12. Turning behavior in head-on situations. (**a**) Trajectory of ship in head-on situations. (**b**) The distance and CRI curves in head-on situations.

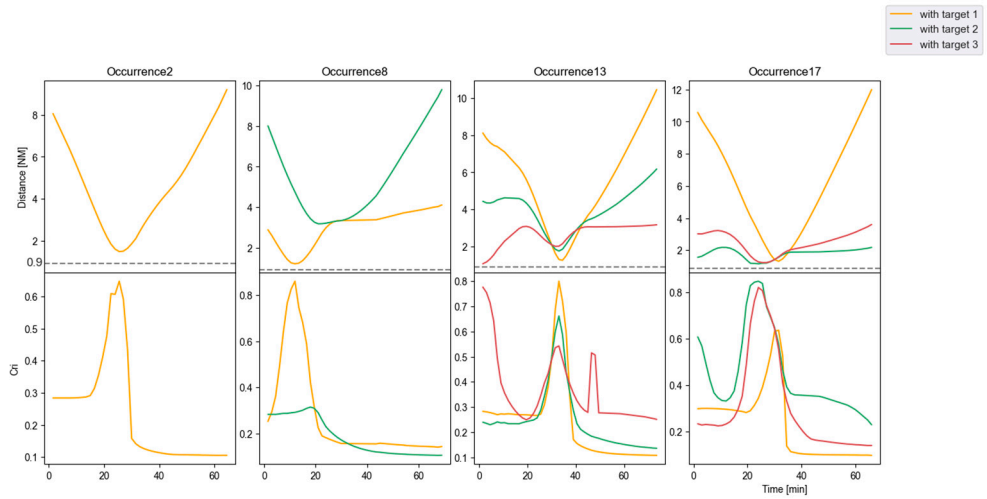
During the crossing situation in the Imazu problem, as presented in Figure 13, the OS is obligated to turn right in accordance with the COLREGs, while the target vessel maintains its speed and course. After the collision risk is eliminated, the OS passes behind the TS to avoid it. The entire collision avoidance process is safe and effective. Furthermore, the collision avoidance behavior during the multi-vessel crossing encounter is also compliant with the COLREGs, with the CRI peak fluctuating around 0.8.

In the Imazu overtaking problem, as presented in Figure 14, the OS takes a right turn at the initial stage to pass the TS's stern on the starboard side. After that, the OS maintains the new heading and proceeds forward. The TS being overtaken maintains its original course to sail. As the risk decreases, the OS returns to its original course and completes the overtaking. In occurrence 7, due to the encounter with TS2, the conditions for overtaking are no longer met, and the OS steers towards its destination.



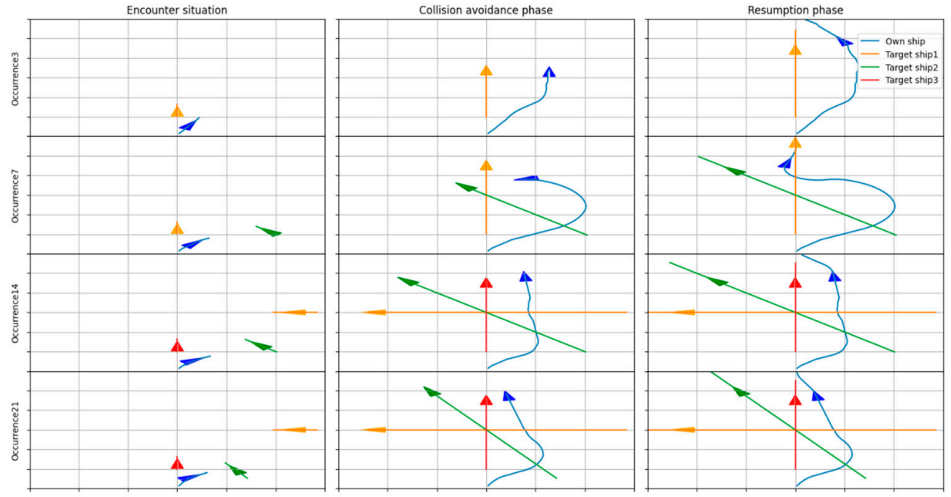
(**a**)

Figure 13. *Cont.*

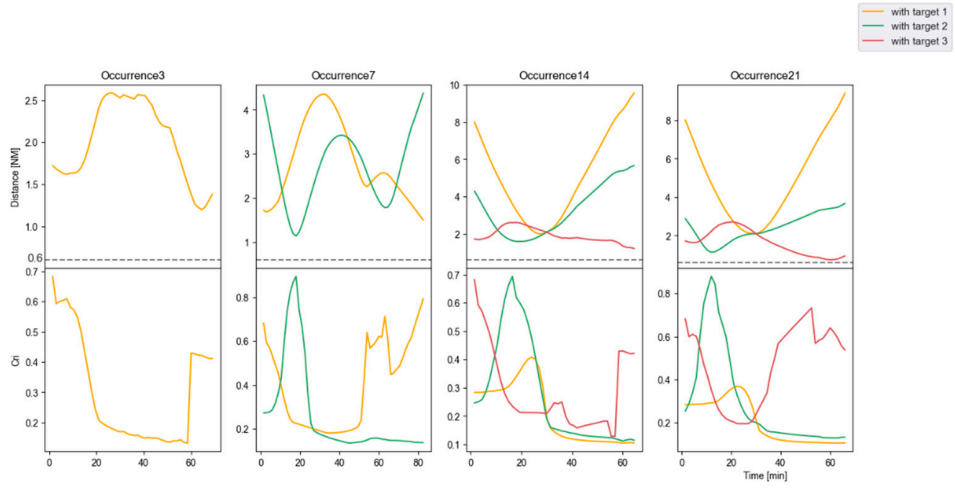


(**b**)

Figure 13. Turning behavior in crossing situations. (**a**) Trajectory of ship in crossing situations. (**b**) The distance and CRI curves in crossing situations.



(**a**)



(**b**)

Figure 14. Turning behavior in overtaking situations. (**a**) Trajectory of ship in overtaking situations. (**b**) The distance and CRI curves in overtaking situations.

In the crossing situation, when the TS approaches from the left side of the OS, the COLREGs specify that the TS as the giving-way vessel should take action to avoid collision. In this study, the TS is considered as the object for collision avoidance decision making, while allowing the OS to continue on its course. The experimental results for this scenario closely resemble those obtained in the previous section, which involved cross encounters.

To enhance the diversity of the experiments and simulate more complex situations that may arise in actual navigation, this study incorporates special scenarios from the COLREGs into the experimental design. Specifically, vessels with emergency tasks or those that are difficult to avoid are included, and the OS is required to take action to avoid a collision. The introduction of such scenarios allows for a more comprehensive evaluation of the performance and robustness of ship collision avoidance algorithms and strategies, enhancing the credibility and practicality of the research findings.

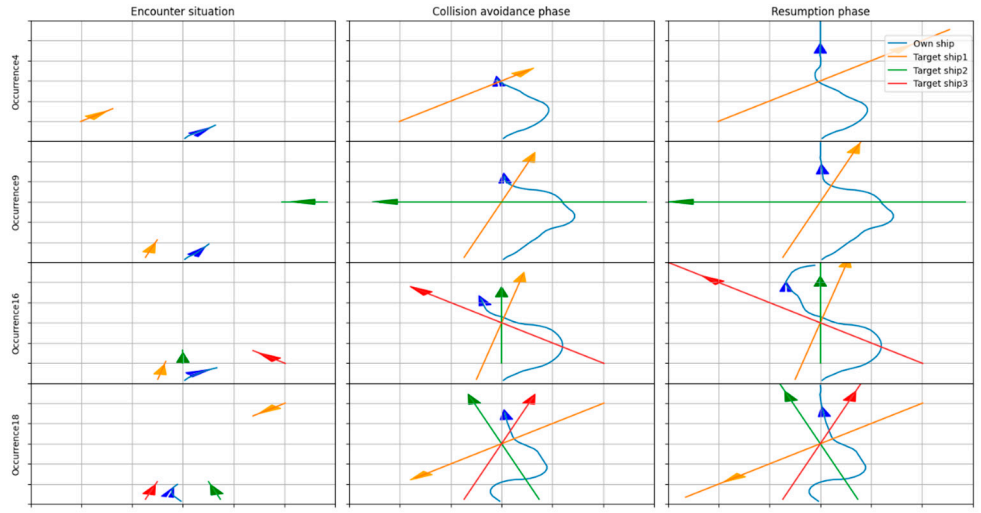
According to Rule 8 of the COLREGs, a vessel that is required to give way and must not impede the passage or safe passage of another vessel is expected to take prompt and appropriate action, as necessary under the current circumstances, to ensure that there is ample space for the other vessel to pass safely. In this situation, the OS must avoid collision based on the vessel's domain and maximum safe distance, as presented by occurrences 4, 9, 16, and 18 in Figure 15. At CPA, the OS selects the action that would result in the maximum reward, passing behind the target ship at a relative bearing of [0, 67.5°]. In occurrence 16, the OS immediately encounters another overtaking situation after avoiding the TS1. Considering the limitations of ship action and fuel economy, the OS chooses to directly initiate an overtaking behavior, which is reasonable and in compliance with ship actions. Overall, the behavior exhibited by the OS in Imazu problem is deemed appropriate for collision avoidance, with Figure 16 showing the minimum passing distance in the Imazu problem between the OS and TS.

Figure 16 classifies the ships based on the encounter situation and records the minimum passing distance between the OS and the TS. Different safe distances are established for various encounter situations: a safe distance of 0.9 nm is set for head-on and crossing situations, a safe distance of 0.6 nm is set for overtaking situations, and for other situations, a safe distance of 1.1 nm is set based on the Closest Point of Approach in the ship's domain. The minimum passing distance between the OS and the TS all exceed the established safe distance.

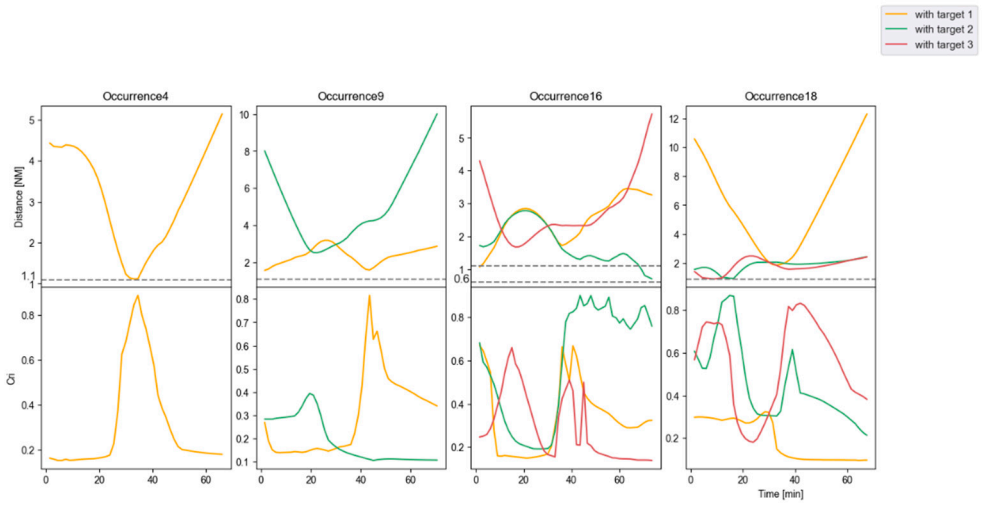
In this study, we executed a series of simulation experiments to compare and analyze the performance of various models trained with distinct algorithms in designated test scenarios. For instance, in Figure 17 occurrence 4, notable behavioral discrepancies were observed. The PPO algorithm, lacking pre-training via behavior cloning, manifested a tendency towards more conservative course correction maneuvers during decision-making processes. In contrast, the continuous action space model employed by Sawada, Ryohei favored long-distance course adjustments to circumvent potential conflicts [30]. Compared to these approaches, the PPO-max model proposed in our research not only aligns more closely with maritime collision avoidance regulations but also exhibits significant efficiency gains in path strategy formulation.

In order to assess the generalization capability of the training model, a test scenario was constructed. To execute the experiment, five target ships were selected randomly from the variety of target ships in the Imazu problem, and a collision was simulated at a fixed coordinate after 30 min. The OS was required to avoid the randomly appearing target ships and navigate towards its destination using an appropriate course. The trajectory of the OS, as depicted in Figure 18a, confirms that the trained model was able to acquire an effective method for collision avoidance, thus providing evidence of the model's generalization capability. To ensure a safe and adequate distance between the OS and the TS under various environmental, the safe distance was set to the maximum ship domain of 1.1 nm. As shown in Figure 18b, the minimum passing distance in this experiment was 1.264 nm. It is noteworthy that the trend in the distance between the OS and the TS changed from

decreasing to increasing after 43.5 min, indicating that the OS had effectively avoided the five target ships and entered the navigation resumption phase.



(**a**)



(**b**)

Figure 15. Turning behavior in other situations. (**a**) Trajectory of ship in other situations. (**b**) The distance and CRI curves in other situations.

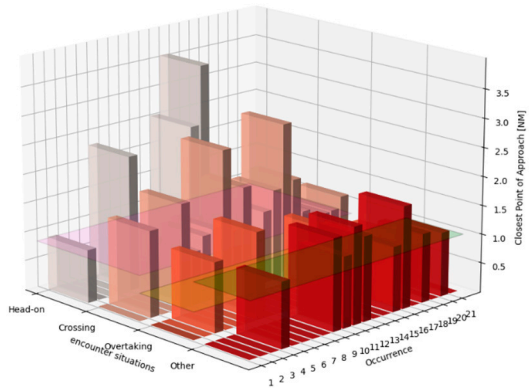


Figure 16. Minimum passing distance of Imazu problem.

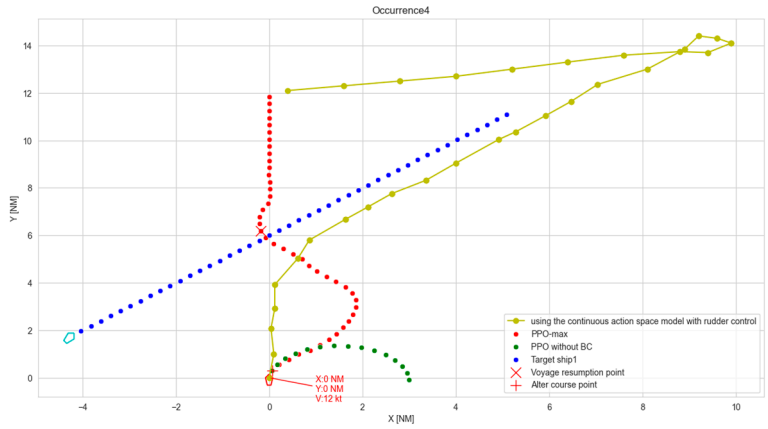
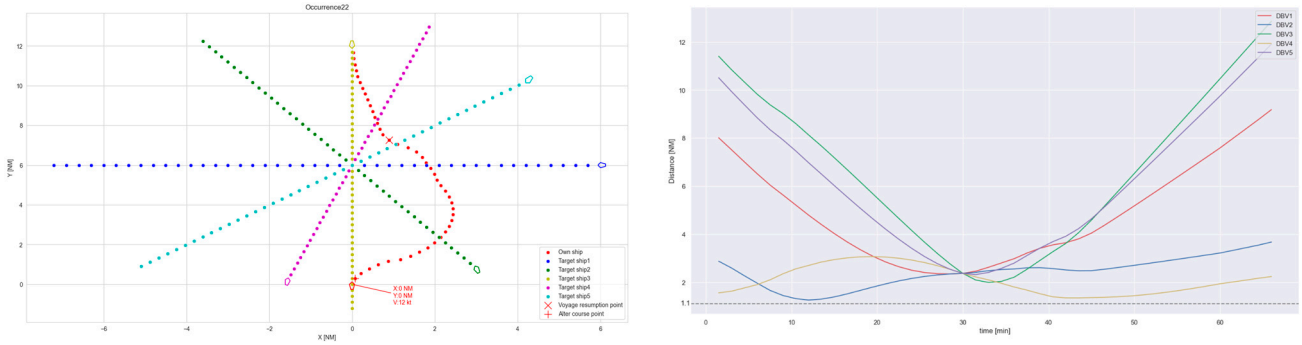


Figure 17. Comparison of PPO-max with other algorithms in complex navigational scenarios: occurrence 4.



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

Figure 18. Generalization performance. (**a**) Trajectory of ship in test scenario. (**b**) Variation in ship distance in test scenario.

Out of the 1000 random experiments conducted, the OS successfully reached its destination in 763 cases. In cases of failure, some are due to collisions, primarily occurring in the fourth type of encounter situation. Due to the dense arrangement of vessels in some random scenarios, the safe distance setting makes it easy for vessels to be deemed to have collided. Another cause is the failure to reach the destination in a timely manner after collision avoidance, primarily occurring in the overtaking situation. The conflicts between overtaking the TS and reaching the destination resulted in the OS being unable to arrive at its destination on time. Further research will be conducted to improve the algorithm's generalizability based on these issues.

4.3. Discussion

The validation experiments were carried out on same vessel types, but the core principles of this method are equally applicable to different types of ships, requiring only adjustments to the vessel's configuration parameters. Despite the potential of DRL in collision avoidance applications for ships, several challenges remain in terms of safety guarantees, predictability, and reproducibility. To ensure practical applicability, the algorithm must accurately define ships and their behaviors in various scenarios. Furthermore, the continuous action space training algorithm often has low heading stability, limiting the reliability of the collision avoidance obtained from the current algorithm. As a result, the proposed method can only be utilized as a reference route for ship collision avoidance in practical applications.

In the subsequent phase of our research, we aim to further optimize the algorithm through enhancements in state vectors, multi-agent architectures, reward design, and the development of a coordinated vessel collision avoidance model to address scenarios includ-

ing vessel speed and heading variations. Moreover, to ensure the practical applicability of our model, future endeavors will explore sim2real strategies to successfully transfer policies trained in simulation to real-world environments.

5. Conclusions

The present study proposes a method for ship collision avoidance using the DRL algorithm with a continuous action space. To ensure optimal performance, appropriate reward functions were designed for different encounter situations, in compliance with the COLREGs. Furthermore, the reward function was tailored to account for the ship collision avoidance risks, actual ship navigation norms, and navigation resumption. In the event of multiple ship encounters, the proposed approach incorporates a judgment mechanism to assess ship danger and determine whether the navigation should be resumed. To facilitate this, the latest navigation resumption element variable was included in the state set of the algorithm.

The simulation optimized the Imazu problem to assess the effectiveness and generalization capability of the proposed algorithm. Additionally, the study innovated improvements to the DRL algorithm. Specifically, the algorithm incorporated techniques like behavior cloning, residual networks, and CPU-GPU dual-core parallel processing modules, using appropriate tricks to optimize the algorithm's performance. As a result, the algorithm's performance and operational efficiency were significantly enhanced. The experimental results indicate that the proposed method effectively addresses various ship encounter situations in the Imazu problem. This confirms the effectiveness of incorporating different encounter situations and relevant factors into the reward function of the algorithm, which guides the OS to adhere to the COLREGs.

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

Funding: This research was supported by the National Natural Science Foundation of China (grants No. 52171345 and 52272413).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: We are especially grateful to the Marine Intelligent Transportation Research Team for their technical support.

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

Appendix A

Table A1. The setting of Imazu problem.

	Target Ship 1			Target Ship 2			Target Ship 3		
Occurrence	**X (nm)**	**Y (nm)**	φ_t (°)	**X (nm)**	**Y (nm)**	φ_t (°)	**X (nm)**	**Y (nm)**	φ_t (°)
1	0.000	12.000	180						
2	6.000	6.000	−90						
3	0.000	1.800	0						
4	−4.243	1.757	45						
5	0.000	12.000	180	6.000	6.000	−90			
6	1.042	0.091	−10	4.243	1.757	−45			

Table A1. *Cont.*

	Target Ship 1			Target Ship 2			Target Ship 3		
Occurrence	**X (nm)**	**Y (nm)**	φ_t (°)	**X (nm)**	**Y (nm)**	φ_t (°)	**X (nm)**	**Y (nm)**	φ_t (°)
7	0.000	1.800	0	4.243	1.757	−45			
8	3.000	0.804	−30	6.000	6.000	−90			
9	−1.553	0.204	15	6.000	6.000	−90			
10	3.000	0.804	−30	−6.000	6.000	90			
11	0.000	12.000	180	4.243	1.757	−45	−1.042	0.091	10
12	0.000	12.000	180	−4.243	1.757	45	−1.042	0.091	10
13	6.000	6.000	−90	4.243	1.757	−45	1.042	0.091	−10
14	6.000	6.000	−90	4.243	1.757	−45	0.000	1.800	0
15	6.000	6.000	−90	−2.970	3.030	45	−6.000	6.000	90
16	−1.042	0.091	10	0.000	1.800	0	4.243	1.757	−45
17	4.243	10.243	−135	1.553	0.204	−15	3.000	0.804	−30
18	4.243	10.243	−135	1.553	0.204	−15	−1.553	0.204	15
19	6.000	6.000	−90	1.553	0.204	−15	0.000	1.800	0
20	6.000	6.000	−90	1.553	0.204	−15	−1.553	0.204	15
21	6.000	6.000	−90	3.000	0.804	−30	0.000	1.800	0

References

1. International Maritime Organization. *Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)*; International Maritime Organization: London, UK, 1972.
2. Tang, P.; Zhang, R.; Liu, D.; Huang, L.; Liu, G.; Deng, T.-Q. Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle. *Ocean Eng.* **2015**, *106*, 128–140. [CrossRef]
3. Harris, C.J.; Hong, X.; Wilson, P.A. An intelligent guidance and control system for ship obstacle avoidance. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **1999**, *213*, 311–320. [CrossRef]
4. Smierzchalski, R.; Michalewicz, Z. Modeling of ship trajectory in collision situations by an evolutionary algorithm. *IEEE Trans. Evol. Comput.* **2000**, *4*, 227–241. [CrossRef]
5. Lee, S.-M.; Kwon, K.-Y.; Joh, J. A Fuzzy Logic for Autonomous Navigation of Marine Vehicles Satisfying COLREG Guidelines. *Int. J. Control Autom. Syst.* **2004**, *2*, 171–181.
6. Zhuo, Y.; Hearn, G.E. A ship based intelligent anti-collision decision-making support system utilizing trial manoeuvres. In Proceedings of the 2008 Chinese Control and Decision Conference, Yantai, China, 2–4 July 2008; pp. 3982–3987.
7. Ahn, J.-H.; Rhee, K.-P.; You, Y. A study on the collision avoidance of a ship using neural networks and fuzzy logic. *Appl. Ocean Res.* **2012**, *37*, 162–173. [CrossRef]
8. Su, C.M.; Chang, K.-Y.; Cheng, C.-Y. Fuzzy Decision on Optimal Collision Avoidance Measures for Ships in Vessel Traffic Service. *J. Mar. Sci. Technol.* **2012**, *20*, 38–48. [CrossRef]
9. Szlapczynski, R. Evolutionary Planning of Safe Ship Tracks in Restricted Visibility. *J. Navig.* **2014**, *68*, 39–51. [CrossRef]
10. Gao, M.; Shi, G.; Li, S. Online Prediction of Ship Behavior with Automatic Identification System Sensor Data Using Bidirectional Long Short-Term Memory Recurrent Neural Network. *Sensors* **2018**, *18*, 4211. [CrossRef] [PubMed]
11. 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]
12. Xie, S.; Garofano, V.; Chu, X.; Negenborn, R.R. Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks. *Ocean Eng.* **2019**, *193*, 106609. [CrossRef]
13. Gosavi, A. Reinforcement Learning: A Tutorial Survey and Recent Advances. *INFORMS J. Comput.* **2009**, *21*, 178–192. [CrossRef]
14. Geng, H.; Liu, H.; Wang, B.; Sun, F. Reinforcement Extreme Learning Machine for Mobile Robot Navigation. In *Proceedings of ELM-2016*; Springer: Cham, Switzerland, 2018.
15. Watkins, C.J.C.H.; Dayan, P. Q-learning. *Mach. Learn.* **2004**, *8*, 279–292. [CrossRef]
16. Peng, J.; Williams, R.J. Incremental multi-step Q-learning. *Mach. Learn.* **2004**, *22*, 283–290. [CrossRef]
17. Chen, Y.; Mabu, S.; Shimada, K.; Hirasawa, K. Enhancement of trading rules on stock markets using genetic network programming with Sarsa learning. In Proceedings of the SICE Annual Conference 2007, Takamatsu, Japan, 17–20 September 2007; pp. 2700–2707.
18. 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.
19. 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]
20. Van Dinh, N.; Viet, N.H.; Nguyen, L.A.; Dinh, H.T.; Hiep, N.T.; Dung, P.T.; Ngo, T.D.; Truong, X.T. An extended navigation framework for autonomous mobile robot in dynamic environments using reinforcement learning algorithm. In Proceedings of the 2017 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, Vietnam, 21–23 July 2017; pp. 336–339.

21. Xie, L.; Wang, S.; Markham, A.; Trigoni, A. Towards Monocular Vision based Obstacle Avoidance through Deep Reinforcement Learning. *arXiv* **2017**, arXiv:1706.09829.
22. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2016**, arXiv:1509.02971.
23. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. *arXiv* **2016**, arXiv:1602.01783.
24. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
25. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018.
26. Tai, L.; Zhang, J.; Liu, M.; Boedecker, J.; Burgard, W. A Survey of Deep Network Solutions for Learning Control in Robotics: From Reinforcement to Imitation. *arXiv* **2016**, arXiv:1612.07139.
27. Zhao, L.; Roh, M.-I. COLREGs-compliant multiship collision avoidance based on deep reinforcement learning. *Ocean Eng.* **2019**, *191*, 106436. [CrossRef]
28. Meyer, E.; Robinson, H.; Rasheed, A.; San, O. Taming an Autonomous Surface Vehicle for Path Following and Collision Avoidance Using Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 41466–41481. [CrossRef]
29. Engstrom, L.; Ilyas, A.; Santurkar, S.; Tsipras, D.; Janoos, F.; Rudolph, L.; Madry, A. Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO. *arXiv* **2020**, arXiv:2005.12729.
30. Sawada, R.; Sato, K.; Majima, T. Automatic ship collision avoidance using deep reinforcement learning with LSTM in continuous action spaces. *J. Mar. Sci. Technol.* **2020**, *26*, 509–524. [CrossRef]
31. Larsen, T.N.; Teigen, H.Ø.; Laache, T.; Varagnolo, D.; Rasheed, A. Comparing Deep Reinforcement Learning Algorithms' Ability to Safely Navigate Challenging Waters. *Front. Robot. AI* **2021**, *8*, 738113. [CrossRef] [PubMed]
32. Benjamin, M.R.; Curcio, J.A.; Leonard, J.J.; Newman, P. Navigation of unmanned marine vehicles in accordance with the rules of the road. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA 2006), Orlando, FL, USA, 15–19 May 2006; pp. 3581–3587.
33. Chauvin, C.; Lardjane, S. Decision making and strategies in an interaction situation: Collision avoidance at sea. *Transp. Res. Part F-Traffic Psychol. Behav.* **2008**, *11*, 259–269. [CrossRef]
34. Perera, L.P.; Carvalho, J.P.; Soares, C.G. Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions. *J. Mar. Sci. Technol.* **2011**, *16*, 84–99. [CrossRef]
35. Goodwin, E.M. A Statistical Study of Ship Domains. *J. Navig.* **1973**, *26*, 130. [CrossRef]
36. Szlapczynski, R.; Szlapczynska, J. Review of ship safety domains: Models and applications. *Ocean Eng.* **2017**, *145*, 277–289. [CrossRef]
37. Gang, L.; Wang, Y.; Sun, Y.; Zhou, L.; Zhang, M. Estimation of vessel collision risk index based on support vector machine. *Adv. Mech. Eng.* **2016**, *8*, 1687814016671250. [CrossRef]
38. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
39. Heiberg, A.; Larsen, T.N.; Meyer, E.; Rasheed, A.; San, O.; Varagnolo, D. Risk-based implementation of COLREGs for autonomous surface vehicles using deep reinforcement learning. *Neural Netw.* **2022**, *152*, 17–33. [CrossRef] [PubMed]
40. Rong, H.; Teixeira, A.P.; Soares, C.G. Ship collision avoidance behaviour recognition and analysis based on AIS data. *Ocean Eng.* **2022**, *245*, 110479. [CrossRef]
41. Cai, Y.; Hasegawa, K. Evaluating of marine traffic simulation system through imazu problem. *Proc. Jpn. Soc. Nav. Arch. Ocean Eng.* **2013**, *17*, 191–194.