Journal of
*Marine Science and Engineering*

MDPI

*Article*

# COLREGs-Based Path Planning for USVs Using the Deep Reinforcement Learning Strategy

**Naifeng Wen, Yundong Long, Rubo Zhang *, Guanqun Liu *, Wenjie Wan and Dian Jiao**

College of Mechanical and Electronic Engineering, Dalian Minzu University, Dalian 116600, China; wennaifeng@dlnu.edu.cn (N.W.); l2905975877@outlook.com (Y.L.); 19841141350@163.com (W.W.); jd0120666@163.com (D.J.)
* Correspondence: zhangrubo@dlnu.edu.cn (R.Z.); liuguanqun@dlnu.edu.cn (G.L.)

**Abstract:** This research introduces a two-stage deep reinforcement learning approach for the cooperative path planning of unmanned surface vehicles (USVs). The method is designed to address cooperative collision-avoidance path planning while adhering to the International Regulations for Preventing Collisions at Sea (COLREGs) and considering the collision-avoidance problem within the USV fleet and between USVs and target ships (TSs). To achieve this, the study presents a dual COLREGs-compliant action-selection strategy to effectively manage the vessel-avoidance problem. Firstly, we construct a COLREGs-compliant action-evaluation network that utilizes a deep learning network trained on pre-recorded TS avoidance trajectories by USVs in compliance with COLREGs. Then, the COLREGs-compliant reward-function-based action-selection network is proposed by considering various TS encountering scenarios. Consequently, the results of the two networks are fused to select actions for cooperative path-planning processes. The path-planning model is established using the multi-agent proximal policy optimization (MAPPO) method. The action space, observation space, and reward function are tailored for the policy network. Additionally, a TS detection method is introduced to detect the motion intentions of TSs. The study conducted Monte Carlo simulations to demonstrate the strong performance of the planning method. Furthermore, experiments focusing on COLREGs-based TS avoidance were carried out to validate the feasibility of the approach. The proposed TS detection model exhibited robust performance within the defined task.

**Keywords:** COLREGs; USV cooperative path planning; multi-agent proximal policy optimization; deep learning; target detection

check for updates

**Citation:** Wen, N.; Long, Y.; Zhang, R.; Liu, G.; Wan, W.; Jiao, D. COLREGs-Based Path Planning for USVs Using the Deep Reinforcement Learning Strategy. *J. Mar. Sci. Eng.* **2023**, *11*, 2334. https://doi.org/10.3390/jmse11122334

Academic Editor: Sergei Chernyi

Received: 30 October 2023
Revised: 7 December 2023
Accepted: 7 December 2023
Published: 11 December 2023

## 1. Introduction

The key to the unmanned surface vehicle (USV) cooperative path-planning problem is the adaptive selection of the best collision-avoidance action [1]. In the procedure, the subproblems below are predominantly considered, i.e., cooperatively avoiding static obstacles, avoiding collision within the USV fleet and between USVs and target ships (TSs) [2]. Several challenges are posed to the planning process. High-dimensional action and state spaces present a major difficulty to the efficiency of the online planner that is required to respond to various scenarios in time [3,4]. Meanwhile, the dynamic TS avoidance problem is complicated since it can greatly increase the dimension of the planning space [5,6].

The development of the International Regulations for Preventing Collisions at Sea (COLREGs)-compliant navigation has been in two areas: the complexity of ship encounter scenarios and the evolution in methodologies.

Studies [7–12] have contributed to the advancement in research on ship collision avoidance by presenting systematic approaches and providing insights into the interpretation of COLREGs rules. Research described in [7] offers valuable insights into ship collision avoidance based on COLREG 72, which can be useful for Officers of the Navigational Watch (OONW), both onboard and remotely, as well as for autonomous systems. However, the research is supported by examples drawn from various works that, despite their significant influence in current literature, may not be from the correct standpoint. In [8], Kim and Park provide insights into COLREGs sailing rules based on the perspectives of navigators and researchers. In [9], the recent progress in COLREGs-compliant navigation of USVs from traditional to learning-based approaches is reviewed in depth.

In [10], Yim and Park presented a systematic approach to model evasive action aimed at preventing collisions in a give-way situation at the minimum-distance moment. The researchers established a conceptual framework for such evasive action and identified COLREGs-compliant maneuvers through a simulation based on ship-handling scenarios. In [11], Kim and Park proposed a method for determining the appropriate timing and necessary actions to ensure ship collision avoidance in accordance with COLREGs rules, using Bayesian-regularized artificial neural networks (BRANNs). In [12], Hagen et al. expressed the COLREGs rules mathematically, providing insights into their interpretation through the selection of parameters and weights.

There are mainly four classes of conventional methods that make a great effort to accommodate COLREGs rules in their path-planning modules [9].

Rule-based methods involve hand-crafted designs and focus on simple ship-encounter scenarios and, hence, cannot be easily extended to more complex ship-encounter scenarios. Hybrid methods, such as A*-variants and rapidly exploring random-tree variants, suffer from the complexity of the multi-TS-avoiding problem and the high dimension of the multi-USV planning space. Reactive methods, such as the artificial potential field and velocity obstacle methods, have difficulties in uncertain TS course prediction. Optimization-based methods are also hard to conduct in complex TS encounter scenarios.

Additionally, traditional research usually focuses on simple 1–1 ship-encounter scenarios dictated by rules 13–16 in Part B of the COLREGs. However, the USV path-planning method becomes more challenging when considering multi-TS encounter scenarios. When planning those algorithms in simulated or real scenarios, simplified and basic assumptions of COLREGs are far from its real complexity [7,8].

Therefore, traditional methods are not able to fully use the seamanship of experienced mariners to solve complex situations, and they can hardly be considered powerful nonlinear approximators of the optimal value and policy functions.

Different from traditional methods, as a paradigm in the field of machine learning (ML) to achieve multi-agent collaboration, the deep reinforcement learning (DRL) model mainly studies the synchronous learning and evolution of agent strategies that are used to plan the coordinated movement of formation in real time [13]. By interacting with the environment, it continuously optimizes the agent's action strategy. The value function of different action strategies in the current state is estimated, and high-return actions are executed to avoid performing low-return or punitive actions. The deep network module of DRL is utilized to fit the motion model of USVs, enabling smooth control and avoiding falling into the local optimal solution [13]. DRL is well suited for situations where the optimal decision-making strategy is not known beforehand and for dealing with non-stationary environments where the underlying dynamics may change over time [13]. These characteristics make it an effective and powerful tool for our task.

DRL methods can be divided into three categories [14]. The value-based methods, such as deep Q-network variants, estimate the optimal values of all different states and then derive the optimal policy using the estimated values [15,16]. The policy-based methods, such as trust-region policy optimization (TRPO) [17] and proximal policy optimization (PPO) [18], optimize the policy directly without maintaining the value functions. Actor–critic methods, such as the deep deterministic policy gradient (DDPG) [19], twin-delayed DDPG (TD3) [20], and soft actor–critic (SAC) [21], can be viewed as a combination of the above two methods, and they maintain an explicit representation of both the policy (the actor) and the value estimates (the critic).

The vanilla policy gradient (VPG) method directly optimizes the policy by utilizing the gradient of the expected reward, which can lead to slow convergence [17]. In response to this challenge, TRPO ensures that each update to the policy parameters remains within a trusted region to prevent large parameter updates that could potentially degrade the performance of the policy [17]. Additionally, TRPO exhibits the capability of effectively handling high-dimensional state spaces. However, it is relatively complicated, and it is not compatible with architectures that include noise (such as dropout) or parameter sharing (between the policy and value function, or with auxiliary tasks), and it has poor data efficiency. To address this problem, the PPO method emerged. PPO converts the constraint term of TRPO into a penalty term to decrease the complexity of constrained optimization problems, and it uses only first-order optimization [18]. Multi-agent PPO (MAPPO) is a version for the multi-agent partially observable Markov's decision-making process [22]. The multi-agent DDPG (MADDPG) method is another state-of-the-art method besides MAPPO. In MADDPG, each agent takes the other agents as part of the environment, and agents in the same region cooperate with each other to determine the optimal coordinated action [19]. However, it is hard to achieve stability due to the complexity of the hyperparameters.

In [23], a multi-USV automatic collision-avoidance method was employed based on a double deep Q network (DDQN) with prioritized experience replay. In [24], the PPO algorithm and a hand-crafted reward function are used to encourage the USV to comply with the COLREGs rules.

Sawada et al. [25] proposed a collision-avoidance method based on PPO, and it uses a grid sensor to quantize obstacle zones by target and a convolutional neural network (CNN) and long–short-term memory (LSTM) network to control the rudder angle. Xu et al. [26] proposed a COLREGs intelligent collision-avoidance (CICA) algorithm that tracks the current network weight to update the target network weight, which improves stability when learning the optimal strategy.

The study in [27] employed a collision-avoidance framework that divides all encounter scenarios into seven types according to the avoidance constraints of the COLREGs for different encountered scenes.

In [28], the COLREGs and ship maneuverability were considered in the reward for achieving multi-ship automatic collision avoidance, and the optimal reciprocal collision-avoidance (ORCA) algorithm was used to detect and reduce the risk of collision.

In the work by [6], the action space and reward function were improved by incorporating the reciprocal velocity obstacle (RVO) scheme. Gate-recurrent unit-based networks were utilized to directly map the state of varying the number of surrounding obstacles to the corresponding actions.

In [5], the collaborative path-planning problem was modelled as a decentralized partially observable Markov decision-making process and used to devise an observation model, a reward function, and an action space suitable for the MAPPO algorithm for multi-target search tasks. In the research described in [29], a system was constructed to switch between path-following and collision-avoidance modes in real time, and the collision hazard was perceived through encounter identification and risk calculation.

In [30], the Q learning method was applied for optimizing the state action pairs to obtain the initial strategy, then PPO was used to fine-tune the strategy. In [31], agents were trained using a mixture of observations from different training environments and linearity constraints were imposed on both the observation interpolations and the supervision (e.g., associated reward) interpolations. In the study described in [32], multiple targets were simultaneously optimized to enhance the PPO.

The comparative methods are outlined in Table 1. The majority of current DRL methods focus on generating COLREGs-compliant paths using a COLREGs-based reward function. However, the high degree of randomness in early-stage action selection can lead to unpredictable strategy gradient updates, making it challenging to achieve model convergence.

**Table 1.** Introduction to comparative methods.

| | Method | Advantage | Limitation |
|---|---|---|---|
| Path replanning methods | Rule-based methods | COLREGs rules integrated into path replanning | relies on hand-crafted design; hard to extend to complex ship-encounter scenarios |
| | Hybrid methods: A star | fast; COLREGs-compliance incorporated into path replanning | hard to extend to more complex ship-encounter scenarios |
| | Reactive methods: velocity obstacle | fast; COLREGs compliance enforced by integrating forbidden zones | accurate TS course information required |
| | Optimization-based methods | optimal; COLREGs rules naturally formulated as constraints | relatively high computational burden |
| DRL methods | Value-based methods: Q-learning, DQN | optimal policy derived from estimates of the optimal values of all different states | overestimation bias; high dimensionality; hard to strike a balance between exploration and exploitation |
| | Policy-based methods: MAPPO, TRPO | directly optimizes the policy without maintaining the value functions | careful design of reward functions required |
| | Actor–critic methods: MADDPG, MATD3, | explicit representation of both the policy and the value estimates | computationally expensive; hyperparameter sensitivity |

Moreover, in the case of USVs, it is crucial for their paths to be both feasible and optimal, while ensuring that multiple USVs can maintain their formation and reach their respective goals [33,34]. Additionally, being COLREGs-compliant does not guarantee an ideal evasive behavior as it may result in overly conservative or inattentive responses to unexpected TSs.

By leveraging a large dataset of pre-recorded USV trajectory data and employing powerful DRL-based methods, it is possible to derive promising solutions that not only ensure that USV navigation adheres to COLREGs rules but that it also replicates the good seamanship exhibited by experienced mariners. Furthermore, path planning in dynamic environments poses a complex problem due to the need to plan multiple paths simultaneously while ensuring collision avoidance within the USV fleet and between USVs and TSs. Therefore, it is crucial that the planner is efficient. MAPPO offers high efficiency in learning, fast convergence, and improved stability, making it an ideal choice as the basic path planner for our task.

Subsequently, this research seeks to propose a two-stage multi-agent reinforcement learning (MARL) scheme based on the MAPPO algorithm, incorporating a centralized training and a decentralized execution strategy. The following innovations are presented:

1. We introduce a COLREGs-compliant action-evaluation module to compute action probabilities that align with COLREGs regulations when encountering multiple TSs. The module parameters are learned from a dataset of pre-recorded USV trajectories. By fusing the probability vector and the candidate action vector from the actor network, we select an action that is the most feasible for the encountered situation. Our reward function incorporates both COLREGs and seamanship considerations, providing a dual heuristic approach to guide the selection of COLREGs-compliant actions.

2. We propose a policy network that can handle multiple aggregation goals, obstacles, and dynamic TSs. To achieve this, we have defined the action space, observation space, and reward function for the policy network. Additionally, we have designed actor and critic networks.
3. A TS motion-detection network is constructed to provide guidance for the decision-making process of the MARL model.

## 2. Planning Problem Definition

### 2.1. USV Motion Model

Figure 1 shows the Earth-fixed inertial frame {i} and the body-fixed frame {b}. The positive direction of the X-axis of frame {b} coincides with the USV heading direction, and the origin is located at the barycenter of the USV.

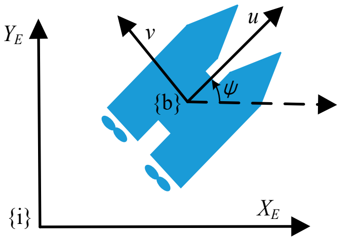


**Figure 1.** Schematic of the USV motion model.

The kinematic model is as follows:

$$\begin{cases} \dot{x} = u \cdot \cos\psi - v \cdot \sin\psi \\ \dot{y} = u \cdot \sin\psi + v \cdot \cos\psi \\ \dot{\psi} = r \end{cases} \tag{1}$$

To guarantee the feasibility of the resulting path, we give zero mask values to the actions that violate motion constraints; for example, the maximum velocity and acceleration, which are defined by the performance limitations of the USV.

### 2.2. Cooperation Problem Description

The cooperation is defined as a loosely coupled one whereby USVs choose the aggregation targets freely in terms of the minimum aggregation cost and collision-avoidance actions. The objective is illustrated as follows:

$$\begin{gathered} \text{argmin}\left\{\sum_{i=1}^{N}\sum_{j=1}^{M} s_{ij}\right\} \quad s.t.|p_m - p_t| > r_o \text{ and} \\ \min_r(|p_{iM} - q_i| < d_{agg}) \end{gathered} \tag{2}$$

where $N$ represents the number of USVs and the variable $M$ is the number of waypoints on the path of the $ith$ USV, $s_{ij}$ is the cost of the $ith$ USV reaching the $jth$ waypoint on its path, and $s_{ij}$ considers the turning angle, velocity, and the distance travelled. If the distance between the end-point of a path and the corresponding USV's aggregation point is smaller than $d_{agg}$, then the USV will have achieved the aggregation target. The planned paths must be collision-free, meaning that the minimum distance between USVs as well as that between USVs and TSs must exceed the safety radius $r_o$. Here, $p_m$ represents the position of a USV, and $p_t$ indicates the position of a sailing obstruction.

### 2.3. COLREGs Rules for Collision Avoidance

A USV may encounter head-on, crossing, and overtaking situations, as shown in Figure 2.

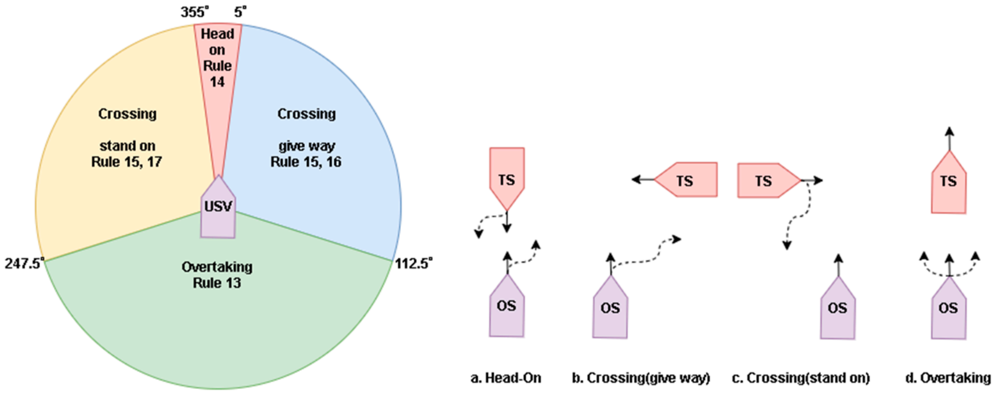


**Figure 2.** Illustrations of the COLREGs.

Referring to studies [7–12], the applicable COLREGs rules for this research are as follows:

- Rule 14: head-on. When two power-driven vessels are meeting on reciprocal or nearly reciprocal courses such that this involves the risk of a collision, each shall alter its course to starboard so that each shall pass on the port side of the other.
- Rule 15: crossing situation. The USV has the option to either stand on or give way to the TS. When two power-driven vessels are crossing such that this involves the risk of a collision, the vessel that has the other on its own starboard side shall keep out of the way and shall, if the circumstances of the case permit, avoid crossing ahead of the other vessel.
- Rule 16: action by the give-way vessel. Every vessel, which is directed by these rules to keep out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear of another vessel.
- Rule 17: action by the stand-on vessel. (i) Where one of two vessels is to keep out of the way, the other shall keep her course and speed. (ii) The latter vessel may, however, take action to avoid collision by its maneuver alone, as soon as it becomes apparent to it that the vessel required to keep out of the way is not taking appropriate actions in compliance with these rules. When, from any cause, the vessel required to keep her course and speed finds itself so close that collision cannot be avoided by the action of the give-way vessel alone, it shall take such action as will best aid to avoid collision.
- Rule 13: overtaking. 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 USV.

Most of the recent research has focused on addressing more complex scenarios, such as areas with restricted visibility that have obstructions and busy narrow channels governed by a traffic separation scheme. These scenarios involve interactions with vessels that may not comply with the COLREGs. For these scenarios, rules 2(b), 8, and 17 should be considered [9].

Rule 2(b): responsibility. Under special circumstances, a departure from the rules may be made to avoid immediate danger.

Rule 8: actions to avoid collision. Actions shall be made with ample time. If there is sufficient sea-room, alteration of course alone may be the most effective. Reduce speed, stop, or reverse if necessary. Action by a ship is required if there is a risk of collision, and when the ship has right-of-way.

Rule 17: actions by the stand-on vessel. Where one of two vessels is to keep out of the way, the other shall keep her course and speed. 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 actions in compliance with these rules.

In Figure 2, the yellow area represents the situation where the ship is crossing from the port side of other ships, the blue area represents the starboard crossing situation, the red area represents the head-on situation, and the green area represents the overtaking situation in which the USV is being overtaken. Figure 2a–d illustrate the actions that each vessel should take according to the COLREGs.

Figure 3 shows a typical collision-avoidance situation when a TS crosses the USV path [2,35], where $v_i$ denotes the USV velocity, $v_o$ represents the velocity of the TS, $v_{io}$ is the relative velocity of the USV with respect to the TS, $r_o$ is the safe radius from the USV to the TS, $p_i = (x_i, y_i)$ denotes the position of the USV, $p_o = (x_o, y_o)$ denotes the position of the TS, and the turning direction is represented by $n_{\mathrm{io}\perp}$, which is perpendicular to the line $p_i p_o$.

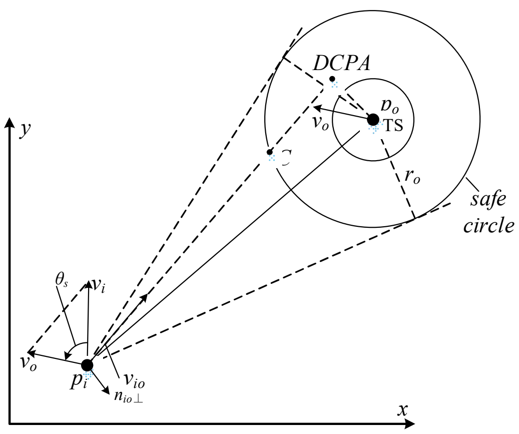


**Figure 3.** TS avoidance situation evaluation according to the COLREGs.

We observe that the direction of collision avoidance for the give-way vessel, as specified by the COLREGs, is primarily determined by the absolute angle ($\theta_s$, $0 < | \theta_s | < \pi$) from the motion direction of the USV to that of the TS in the crossing situation. Therefore, the collision-avoidance direction for the give-way vessel corresponds to the steering direction of the USV that increases the magnitude of $| \theta_s |$ [35].

Since encounters are dynamic situations, continuous monitoring is required. We assume that USVs remain vigilant for other vessels. If another vessel (including TSs or other USVs) does not comply with the COLREGs, the USV should take collision-avoidance actions in time.

### *2.4. Collision Detection*

The collision prediction method is the collision-cone method [2,35], which determines the collision area according to the velocity relationship between the USVs and between USVs and TSs. The collision-cone method operates as follows. Taking the TS avoidance problem as an example, when a TS enters the visible range of the USV ($d_o \leq 600$), the collision navigational angle range is computed according to the relative velocity of the USV with respect to the TS.

$$\begin{cases} v_\theta = v_i \sin(\alpha - \theta) - v_o \sin(\beta - \theta) \\ v_r = v_i \cos(\alpha - \theta) - v_o \cos(\beta - \theta) \end{cases} \tag{3}$$

As illustrated in Figure 4, we denote $v_\theta$ to be the component of $v_{io}$ in the direction perpendicular to the line $p_i p_o$, and we denote C to be the intersection of the expected USV path with the safe circle of the TS, if the USV and the TS keep their velocities. We also define $v_r$ to be the component of $v_{io}$ in the direction of $p_i p_o$. If $v_\theta = 0$, then the relative trajectory of the USV to the obstacle is right on the line of $p_i p_o$. If $v_r > 0$, then the projection of $v_{io}$ on $p_i p_o$ is positive, and the USV has the tendency to get close to the obstacle; otherwise, the USV is more likely to leave the TS alone. Figure 4 shows that collision probably occurs at the point C. If $v_r > 0$, and the direction of $v_{io}$ is within the domain from $p_iA$ to $p_iB$, which are the tangents of the circular bounding box of the TS from the USV location, then a collision possibly occurs. We denote $(v_\theta)_{pA}$ and $(v_\theta)_{pB}$ to be the components of the $v_{io}$ in the direction perpendicular to the line $p_iA$ and $p_iB$, respectively. If any one of the conditions $v_\theta = 0$ and $v_r > 0$, $(v_\theta)_{pA} \cdot (v_\theta)_{pB} \leq 0$, or $v_r > 0$, is met, then the USV will probably collide with the obstacle. The practical style of the second condition is ${d_o}^2{v_\theta}^2 \geq {r_o}^2\left({v_r}^2 + {v_\theta}^2\right)$, where $d_o$ is the distance between the USV and the obstacle. The collision detection algorithm is as follows:

$$\begin{cases} v_\theta = 0 \; and \; v_r > 0 \\ v_r > 0 \; and \; d_{io}^2 \cdot v_\theta^2 \geq r_o^2 \cdot (v_r^2 + v_\theta^2) \end{cases} \tag{4}$$

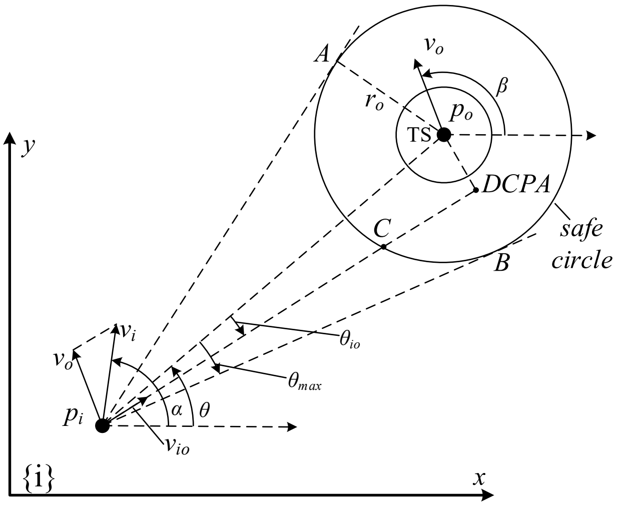


**Figure 4.** Illustration of the collision detection via the collision-cone method.

## 3. MAPPO Algorithm Design

### *3.1. Observation Space Design*

Each USV can observe the environment information, including the information of TSs within its sensing range, and they share their observations to construct the global observation through communication [5]. The observation consists of the positions and navigation states of both USVs and TSs, as well as the environmental information, such as the positions and shapes of obstacles. The observation vector has a fixed dimension, and any missing values are filled with zeros.

The inner formation observational features are employed to describe the state of the USVs within the communication range. For $USV_i$, the observational feature of the neighboring $USV_j$ is set to $o_{ij} = \{l, v_x, v_y, x_i, y_i, x_j, y_j, \Delta v_x, \Delta v_y, e_i\}$, where $l$ is the location of the estimated aggregation target that is available to all USVs, and $v_x$ and $v_y$ represent the velocity components of $USV_i$ in the $x$ and $y$ directions, respectively. To avoid complex coupling between velocities, orthogonal components are used to express the actions of the USV. In this context, $x_i$ and $y_i$ represent the position of the first vessel (denoted as $USV_i$), while $x_j$ and $y_j$ represent the position of another vessel (denoted as $USV_j$). The variable $e_i$ represents the TSs and obstacle features observed by $USV_i$. The variables $\Delta v_x$ and $\Delta v_y$ represent the differences between the velocities of $USV_i$ and $USV_j$, respectively. Specifically, this equation considers three USVs, four obstacles, and three TSs in the vicinity of a USV.

### 3.2. USV-State Feature Design

The USV-state feature is represented as $S_i$ = { $name$, $done$, $s_c$, $r_c$, $x$, $y$, $v_x$, $v_y$, $c_{ij}$, $n_c$}, where *name* represents the name of the USV state, *done* indicates whether the USV has reached an aggregation goal, $s_c$ represents the collision state of the USV, $r_c$ represents the optional actions determined by the current USV state, and $x$ and $y$ represent the USV position.

To address the restricted visibility problem in narrow channels, practical communication is simulated by incorporating observations from USVs with added noises into a global observation during the model training process. The USV can take communication actions to broadcast information to other USVs, and $c_{ij}$ represents the communication utterance transferred from $USV_j$ to $USV_i$, while $n_c$ represents the communication noise. Noise is added to the communication utterance of a USV to simulate the practical environment.

### 3.3. Reward Function Design

The reward calculation combines guidance rewards and sparse rewards. During task execution, each USV receives a guiding reward at each time step, which aims to guide the USVs to chase the aggregation targets and to form a formation. The average reward of all USVs is calculated as the collective reward of the USV formation. The reward function, $R$, consists of guiding rewards, COLREGs-compliant rewards, collision rewards, and time-consumption rewards weighted accordingly.

To guide the USVs to approach the aggregation targets during the early stage of training, the USVs receive negative rewards based on their distances to the targets at each time step, until the USVs reach the target.

$$r_t = \begin{cases} 0, & d_t \le d_{agg} \\ -w_{ta} \cdot (d_t - d_{agg}), & d_t > d_{agg} \end{cases} \tag{5}$$

where $w_{ta}$ is the coefficient that limits the range of reward changes; if the distance $d_t$ of a USV to its nearest target is less than $d_{agg}$, the USV is considered to have reached the aggregation point. We set $w_{ta}$ = 1, $d_{agg}$ = 20, and all the hyperparameters are set empirically.

The guiding reward is the estimation of USVs reaching the targets, while the time-consuming rewards reflect the actual time consumption from the start points. A USV will receive a negative reward of $\Delta t$ ($-0.1$) if it takes one additional step.

The control on turnings is more difficult than that on a linear path; thus, a USV will receive a negative reward if it needs to perform a steering maneuver. We assign a reward of $R_s$ = $-0.6$ to turning points when the trajectory turning angle exceeds 45 degrees.

The obstacle-avoidance reward consists of two parts, which are the static obstacle-avoidance reward and the dynamic obstacle-avoidance reward. The static obstacle-avoidance reward is calculated using the following formula.

$$r_{ca} = -R_c \cdot e^{-k_c d_o} \tag{6}$$

where $d_o$ is the minimum distance between a USV and its obstacles; $k_c$ is the weight, which we set to be 1; and $R_c$ represents the reward amplitude. If $d_o$ > 600, then $R_c$ = 0; when 300 $\le d_o \le$ 600, $R_c$ = 1.5; when 100 $\le d_o$ <300, $R_c$ = 2; and if $d_o$ < 100, then $R_c$ = 10. The reward is a negative value that decreases exponentially along with the decrease in the distance between the USV and the obstacles.

The dynamic avoidance reward considers the USVs avoiding sailing obstructions (including other USVs and TSs) in terms of the COLREGs. If a USV action enters the collision areas of other USVs or TSs and the minimum distance, $d_o$, from the $USV_i$ to other sailing obstructions is smaller than 600, then a negative reward, denoted as $r_i$, is added to the total reward. As shown in Formulas (7) and (8), the reward consists of two parts: the COLREGs-compliant reward, which is based on adherence to the rules, and the obstacle-avoidance reward, which focuses on avoiding vessels regardless of the rules.

Once the encountering scenario is determined by the COLREGs, the calculation of the COLREGs-compliant reward will remain unchanged. During the dynamic obstacle-avoidance procedure, the USV continuously monitors other vessels. If another vessel that is responsible for giving way according to the COLREGs does not comply with the rules, the USV takes collision-avoidance actions based on their distance, regardless of the specific rules.

$$r_i = -R_{rule} \cdot e^{-k_c d_o} \tag{7}$$

The dynamic obstacle avoidance constant, $R_{rule}$, is computed as:

$$R_{rule} = \alpha \cdot R_c + R_{col} \tag{8}$$

where the weight, $\alpha$, is set to be 0.5.

The head-on reward is shown in Formula (9). According to the COLREGs, the USV should turn to the starboard side in a head-on scenario; if the USV turns left, a penalty will be added to the reward. The rewards for the give-way and stand-on situations, shown in Formulas (10) and (11), are similar to the head-on reward. There is no guidance on the action the USV should take to avoid a collision in the overtaking situation. Therefore, we do not define a specific reward function for this scenario but, instead, directly use the distance to evaluate the avoidance actions of the USV. Finally, the head-on reward, give-way reward, and stand-on reward are combined to form the COLREGs-compliant reward, as shown in Formula (12).

$$R_{h\_on} = \begin{cases} 0, & if\ turns\ to\ the\ starboard\ side \\ -R_{port}, & if\ turns\ to\ the\ port\ side \end{cases} \tag{9}$$

Give-way reward:

$$R_{g\_way} = \begin{cases} 0, & if\ turns\ to\ the\ starboard\ side \\ -R_{port}, & if\ turns\ to\ the\ port\ sidet \end{cases} \tag{10}$$

Stand-on reward:

$$R_{s\_on} = \begin{cases} 0, & if\ turns\ to\ the\ starboard\ side \\ -R_{port}, & if\ turns\ to\ the\ port\ side \end{cases} \tag{11}$$

The aggregate:

$$R_{col} = R_{h\_on} + R_{g\_way} + R_{s\_on} \tag{12}$$

### 3.4. Action Space Design

To ensure the planning efficiency, we use the discrete action space that distributes uniformly by 45 degrees around a USV. As shown in Figure 5, the action probabilities are determined by the actor network; the appropriate actions in green are selected, while actions in red are suppressed.

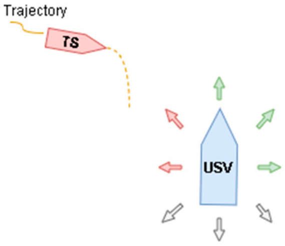


**Figure 5.** Illustration of the action space.

### 3.5. Policy Network Design

Figure 6 shows the policy network in this research. The environment considers that USVs sail confronting static obstacles and TSs. An observation block is constructed for each USV to map the USV state feature, $S$, to the observation feature, $O$, by interacting with the environment.

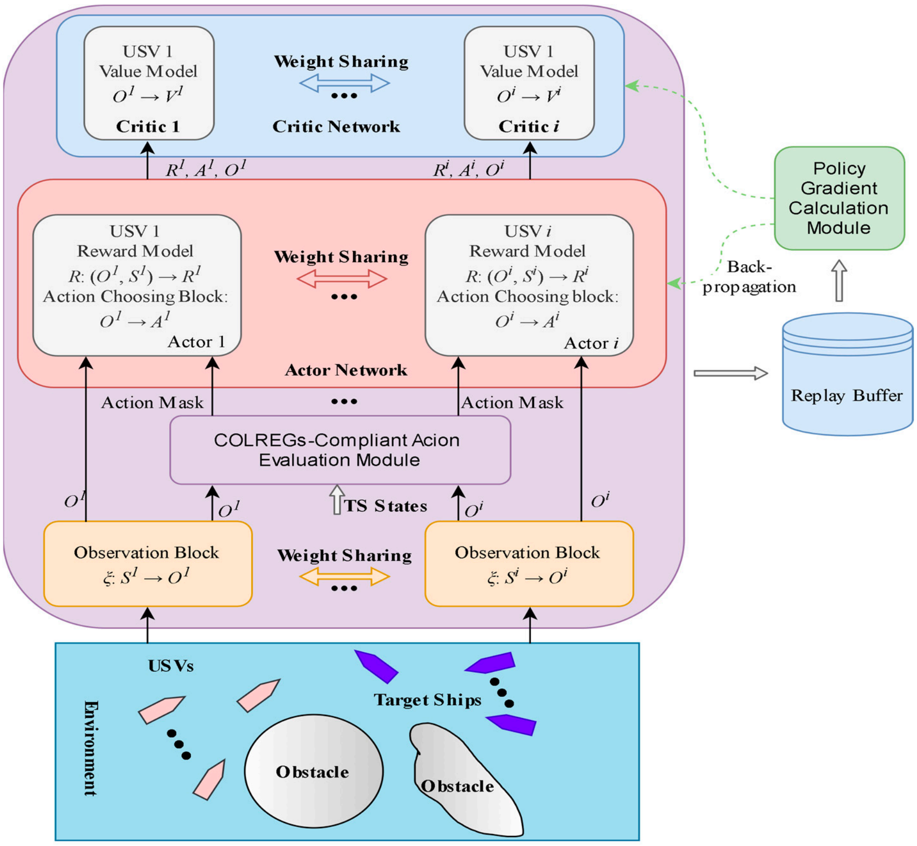


**Figure 6.** Illustration of the policy network.

The COLREGs-compliant action-evaluation module is used to distinguish the actions satisfying the COLREGs from those violating the rules. The MLP-based actor network is responsible for calculating the reward, $R$, and mapping the observation feature into the optimal action, $A$. The critic network, which assesses the chosen action, is also established using the MLP network, and it outputs the value $V$.

To optimize policies, the network alternates between sampling data from the policy and performing optimization on the sampled data. The policies represented by the $S$, $O$, $A$, $R$, $V$ vectors are saved in the replay buffer. We use the mini-batch method. After collecting data for multiple epochs, the data is transferred from the buffer to the policy gradient calculation module for the optimization of the actor and critic networks. By sharing weights among networks, the planner can learn from the cooperative action-selection experiences. When the policy network is used for path planning in a real environment, the actor networks plan actions based on the observation features without critic networks.

The loss function of the network adopts the clip strategy [18], with $\varepsilon = 0.1$, as follows.

$$L^{CLIP}(\theta) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{k=1}^{M}\min\left\{r_{\theta,i}^{(k)}A_i^{(k)}, clip(r_{\theta,i}^{(k)}, 1-\varepsilon, 1+\varepsilon)A_i^{(k)}\right\} \tag{13}$$

### 3.6. COLREGs-Compliant Action-Evaluation Network Design

The COLREGs-compliant action-evaluation module is built based on the MLP network. We used a pre-training strategy, training the COLREGs-compliant action-evaluation network separately from the other parts of the policy network. The COLREGs-compliant action-evaluation network is pre-trained using pre-recorded USV trajectory data, which includes the state vector and global observation vector, as well as the USV actions for avoiding the TSs. The action labels align with our action space. The final established dataset consists of 5000 data items, with an almost equal number of avoidance behaviors for each category (head-on, crossing and give-way, crossing and stand-on, and overtaking), ensuring a balanced dataset.

The COLREGs-compliant action-evaluation network consists of three fully connected (FC) layers, with parameter sizes of 18 × 64, 64 × 64, and 64 × 7 for the respective layers. The Tanh activation function is applied after the first two FC layers, and the softmax function is used following the last FC layer to obtain the COLREGs-compliant probabilities of actions. During the pre-training procedure, the CrossEntropyLoss function is utilized. The remaining parameters, including the optimizer and the learning rate, are set to be identical to those of the policy network. The pre-trained network is utilized directly without applying any parameter updates via the policy gradient algorithm.

The probability vector of actions compliant with the COLREGs will be multiplied by the dot product with the probability vector of chosen actions to select the final action with the highest result.

In Figure 7, the illustration depicts that the gym environment supplies the state vectors of the entities to the path-planning model. Additionally, the TS motion-detection network offers TS motion information, which includes courses and bearings, to the planner. Subsequently, the state vector and the observation vector are concatenated to form the input vector for both the actor network and the COLREGs-compliant action-evaluation network.

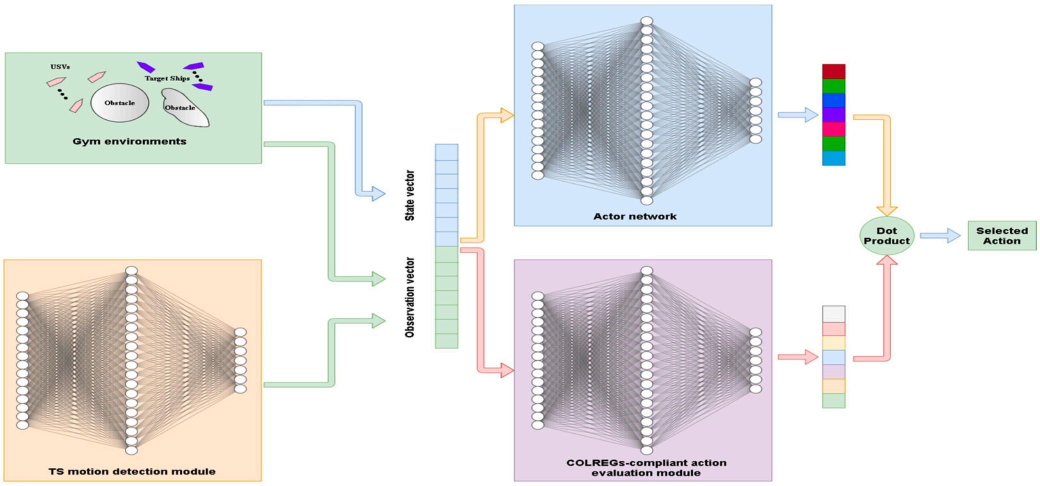


**Figure 7.** COLREGs-compliant action selection.

By integrating the prediction vectors from both the actor and action-evaluation networks, the policy network is capable of making well-informed decisions concerning the USV's path-finding and collision-avoidance actions while following the COLREGs.

## 4. TS Motion-Detection Network Design

This research recognizes the TS intention by the time series of RGB images. The bow, stern, and whole body of TSs on consecutive frames are detected. TSs are detected at different scales since they are sometimes relatively small in the overall picture. Subsequently, the motions of the TSs can be identified using the inter-frame difference scheme.

Practical TS images are relatively rare; thus, we adopt the pre-train and fine-tune paradigm to train the network. Firstly, the network is pre-trained using the SeaShip dataset [36]. The dataset is designed to train and evaluate ship object-detection algorithms. The dataset consists of 31,455 images covering six common ship types (ore carriers, bulk carriers, general cargo ships, container ships, fishing vessels, and passenger ships). All images are from approximately 10,080 real-world video footage captured by surveillance cameras in a deployed coastline video surveillance system. The data has been carefully selected to cover all possible imaging variations, such as different proportions, hull components, lighting, viewpoints, backgrounds, and occlusion.

After our TS detection model is pre-trained on the SeaShip dataset, the model is finetuned using the real-scene images to generalize it to our task.

The TS detection model is shown in Figure 8. The network is constructed based on the YOLO model [37]. The network utilizes the spatial pyramid pooling-based feature (SPPF) network to capture spatial features of images. The cross-stage partial (CSP) network is employed to enhance the CNN block and to reduce the computational burden [37]; CSPi_j refers to the *i*th CSP module utilizing *j* residual components. The CBS module comprises a convolutional network, a batch normalization network, and SiLU activation. It is utilized for feature extraction.

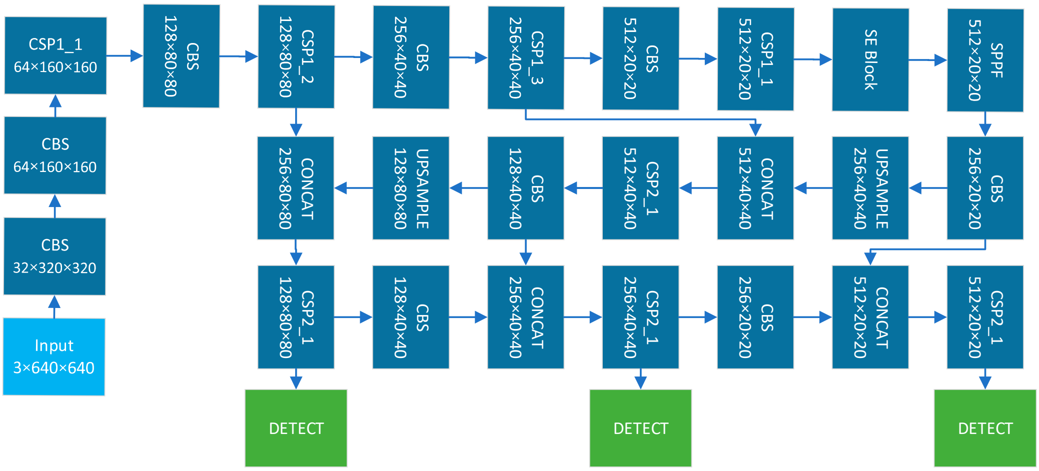


**Figure 8.** Illustration of the TS detection network. The dark blue boxes indicate the body of the model. The light blue box represents the model input block, while the green boxes represent the detection block.

The channel attention mechanism, namely squeezing and excitation (SE), is combined with the YOLO network to boost the feature-extraction ability, as shown in Formula (14) and Figure 9.

$$SE(x) = FC(x) \cdot Sigmoid(FC(Relu(FC(AvePool(x))))) \tag{14}$$

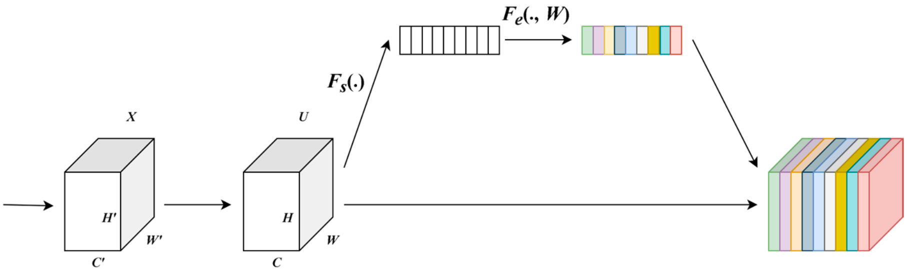


**Figure 9.** Illustration of the squeezing and excitation mechanism.

## 5. Experiments

### 5.1. Experimental Setting

Our MAPPO-based path-planning program runs on a computer equipped with an Intel(R) Core(TM) i7-7820HQ CPU @ 2.90 GHz and 16 GB of memory. Our planning model is compared with the MADDPG model in experiments. Due to the complexity of the MADDPG algorithm, we utilize a more powerful computer to run the program. The configuration is as follows: the operation system is Ubuntu 20.04.6 LTS, the CPU is Intel(R) Xeon(R) Silver 4214 @ 2.20 GHz with 48 cores and 2 threads per core, and the computer has 2 memories, each with a capacity of 8192 MB.

### *5.2. The Path-Planning Model Training Pipeline*

The MAPPO-based path-planning model is trained using a multi-threaded approach, where each thread establishes a virtual environment to fully select data points by importance.

Observations are captured first, then step actions for a USV are captured in terms of rewards from the actor network until an episode is reached. The step rewards of the episode or a path are summed up and then normalized. The normalization is used to eliminate the impact of different scales of metrics.

The maximum number of steps per episode is 25, which means that the planner generates a local path of no more than 25 steps in the local planning time domain. The critic network aims to assess the cooperative action-selection strategy of the actor network by solving the value function. The obtained values are also normalized.

We adopt the mini-batch scheme, and the batch size is set to be 32 episodes. After collecting a batch of data, the actor and critic networks are trained three times, and the average metrics are used to compute the strategy gradient.

The learning rate is set to be $5 \times 10^{-4}$. The discount factor when calculating the rewards is set to be 0.99. The policy entropy coefficient is set to be 0.01. The epsilon value of the Adam optimization is set to be $1 \times 10^{-5}$. The simulation time step is 0.1 s. We use the learning-rate-decay method and gradient clip tricks when updating the network parameters. The orthogonal parameter initialization scheme is applied to the MLP layers of both the actor and critic networks to keep the weight matrixes orthogonal, thereby reducing the problem of the gradient vanishing and exploding.

The MLP blocks of the actor network consists of three fully connected (FC) networks, and the parameters are $18 \times 64$, $64 \times 64$, and $64 \times 7$ for the respective layers. The Tanh activation function is applied after the first two FC layers. Subsequently, the softmax function is used following the last FC layer to obtain the probabilities of choosing actions.

The MLP blocks of the critic network consist of three FC networks, with parameters of $54 \times 64$, $64 \times 64$, and $64 \times 1$ for the respective layers. The Tanh activation function is applied after the first two FC layers. The value can be obtained from the last FC layer.

### *5.3. Path-Planning Results*

We configure Monte Carlo simulation environments for training the planner network whereas the aggregation targets for USVs and obstacles are randomly located, and the linear TS routes are also randomly generated. The planning objective of the USVs is to reach the aggregation targets while avoiding TSs and obstacles.

We trained our model for $1 \times 10^6$ iterations. Figure 10 shows the results of the Monte Carlo experiment for USV cooperation. The black-filled circles represent obstacles, the colored circles represent the aggregation positions of USVs, stars indicate the estimated aggregation targets, the triangles indicate the start points of the USVs, and the colored lines depict the obstacle-avoidance paths. Different colors are used to distinguish the relevant elements of different USVs. Since the planner needs to be trained based on the current model in a new environment that is significantly different from the known one, we train the planner for $1 \times 10^6$ iterations for efficiency.

We have observed that our planner can efficiently generate obstacle-avoidance paths in less than 1 s, and these paths can effectively avoid collisions while reaching the aggregation targets to maintain the formation. The success rate of achieving multiple objectives is higher than 95%. These observations testify to the efficiency and effectiveness of our method.

We train our planner for a total of $1 \times 10^6$ and $6 \times 10^6$ iterations. We evaluate the network and record metric values per 5000 training iterations, and the evaluation reward curves are shown in Figure 11. The action-space dimension is 7 and the observation-space dimension is 24. Figures 12 and 13 depict the training-reward curves for the MADDPG and multi-agent TD3 (MATD3) algorithms. MATD3 improves upon the original DDPG algorithm by incorporating additional techniques, such as using twin critic networks and delayed policy updates, to improve the coordination and learning in complex environments. Due to the complexity of the training process and the long duration required for each iteration, we trained both the MADDPG and MATD3 algorithms for 1 × $10^6$ iterations.

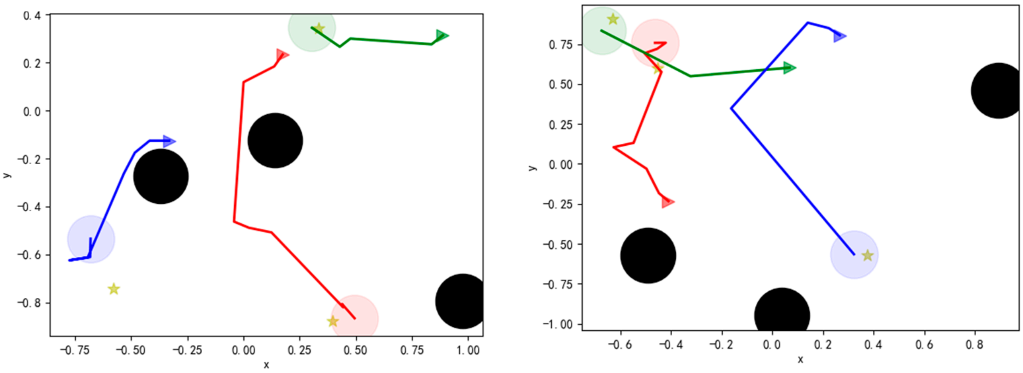
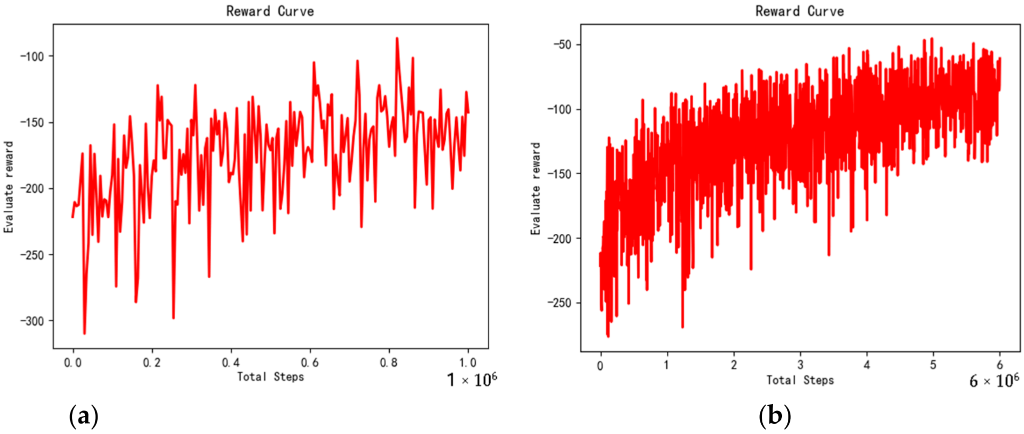


**Figure 10.** Monte Carlo experiment results.



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

**Figure 11.** Reward curves of the MAPPO-based planner. (**a**) Reward curve of training for 1 × $10^6$ iterations; (**b**) Reward curve of training for 6 × $10^6$ iterations.

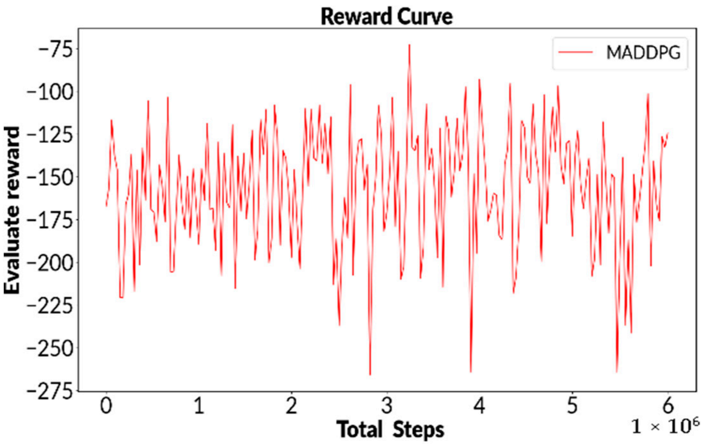


**Figure 12.** Reward curve of MADDPG.

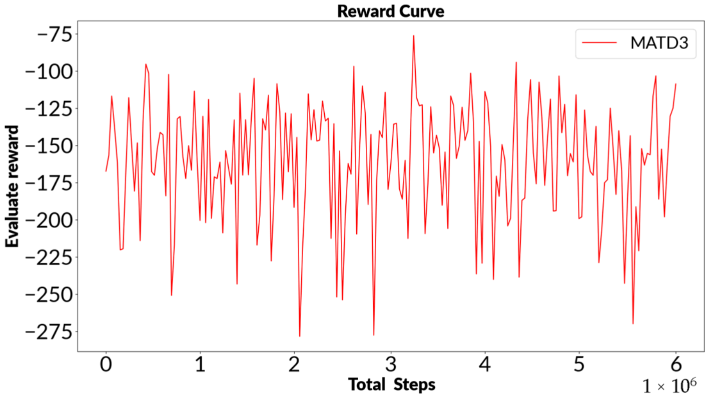


**Figure 13.** Reward curve of MATD3.

We observed that the reward curve of our method converges as the training iterations approach 1 × $10^6$. The oscillation is relatively low, indicating that the method has relatively high stability. Figure 11b shows the training-reward curve for 6 × $10^6$ training iterations. The rewards decrease as the training iterations increase, indicating that the method scales well with a large number of training iterations.

The curve became steady as the iteration time approached 5 × $10^6$, which probably indicated that the model was sufficiently trained. Since training efficiency is also an important metric for the online planner model, we benchmarked our method against the state-of-the-art MADDPG model and the MATD3 model after training for 1 × $10^6$ iterations. The oscillations in the reward curves of MADDPG and MATD3 are significant, and the decreases in rewards are not obvious. The observation implies that MADDPG and MATD3 may require more training iterations than our method.

Meanwhile, the duration of training for our MAPPO-based method with 6 × $10^6$ iterations was less than 5 h, whereas the training durations for MADDPG and MATD3 with only 1 × $10^6$ iterations exceeded 10 h. The above observations demonstrate that our method has a much higher training efficiency than MADDPG and MATD3 due to the low complexity of MAPPO.

The results of the Monte Carlo simulations show that our MAPPO algorithm outperforms both the MADDPG and MAPPO algorithms in terms of the average path length and the steering angle. Specifically, the average path length of our MAPPO algorithm is 0.489, which is lower than that of the MADDPG and MAPPO algorithms (0.683 and 0.582, respectively). This indicates that our algorithm is able to plan more optimal paths. Moreover, the average steering angle of our MAPPO algorithm is 2.14 rad, which is lower than that of the MADDPG and MATD3 algorithms (2.35 and 2.23, respectively). This suggests that our algorithm is also more efficient in steering the vehicle. Overall, these observations provide evidence that our method is capable of planning more optimal paths than the MADDPG and MATD3 models after being trained for 1 × $10^6$ iterations in the simulation environment of the Monte Carlo simulations. This result highlights the potential of our algorithm to improve the performance of autonomous driving systems.

### *5.4. COLREGs-Based Collision-Avoidance Experiments*

Figure 14 depicts the simulation results of USVs avoiding TSs, with randomly generated positions for both the USVs and the obstacles. The TS paths are also generated randomly. The obstacles are represented by black-filled triangles, rectangles, and circles. The USVs plan their paths to reach the aggregation targets, considering both low-cost paths and collision avoidance among the USVs and TSs. The red curves represent the paths of the three USVs, while the blue curves depict the paths of the TSs. The start points of the TS paths are indicated by blue triangles.

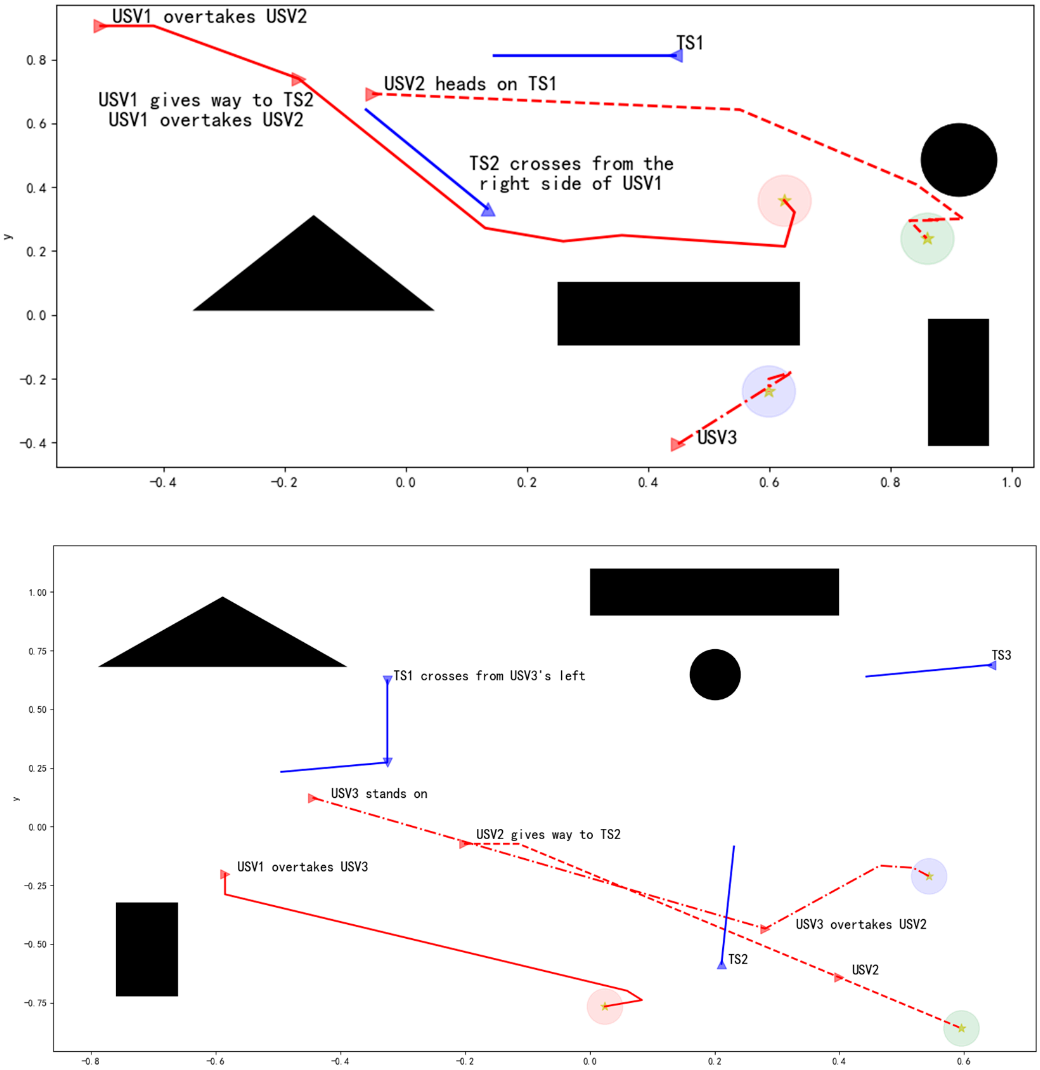


**Figure 14.** Simulations of USVs avoiding TSs in accordance with the COLREGs.

Figure 14 illustrates two representative scenarios of avoiding collisions in accordance with the COLREGs. USVs typically prioritize the most significant collision risks in the nearby area and subsequent collision-avoidance issues, taking actions according to the COLREGs to avoid a collision. The upper sub-figure illustrates that the first USV not only overtakes the second USV by turning to the starboard side in accordance with the COLREGs but also gives way to the TS crossing from the right by turning to the starboard side. Similarly, the second USV follows the COLREGs by turning to its starboard side when heading toward a TS.

The lower sub-figure illustrates that the first USV overtakes the third USV by maneuvering to its starboard side. When a TS crosses from the port side of the third USV, the USV stands on its course if it can pass the TS without risking a collision, in accordance with the COLREGs. In the subsequent segments of the voyage, the third USV turns to its port side to overtake the second USV. Simultaneously, the second USV adjusts its course to the starboard side to give way to the TS.

Figure 14 demonstrates that our path-planning method is capable of effectively planning multiple USV paths to achieve their targets while ensuring collision avoidance among USVs and TSs in accordance with the COLREGs regulations.

### 5.5. TS Detection Results

In the TS detection experiments, we detect the head, stern, and the entire body (i.e., all) of the TS. The corresponding PR curves are shown in Figure 15. When any part of the TS is detected, the motion intention of the TS can be captured by the detection results in consecutive frames using the inter-frame difference scheme. The blue curve indicates the mean average precision (mAP) curve when the intersection-over-union (IoU) threshold is 0.5. The metric values may imply that the model achieves a certain balance between recall and accuracy, and it demonstrates a relatively high overall performance. In the evaluation of the TS detection network on the SeaShips dataset, the 10-fold cross-validation method was employed. This method involves dividing the dataset into 10 subsets or folds of approximately equal size. For each fold, 70% of the images were used as the training set, while the remaining 30% of images were used as the test set. It is important to note that the training set and the test set did not have any overlapping images.

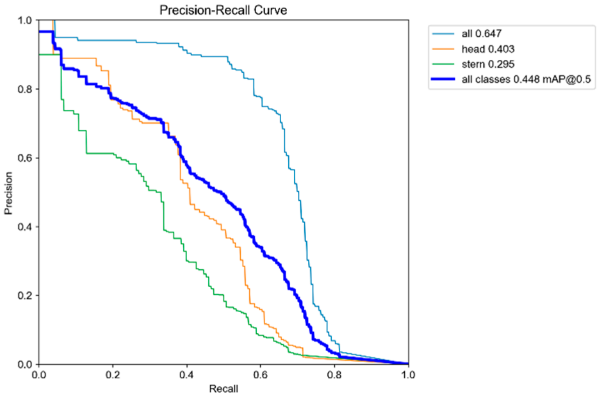


**Figure 15.** PR and mAP curves of the TS detection model.

Figure 16 presents the TS detection result curves from different perspectives. The training loss and validation (val) loss curves demonstrate that the model converges quickly. The precision, recall, and mAP curves imply high performance of the model.

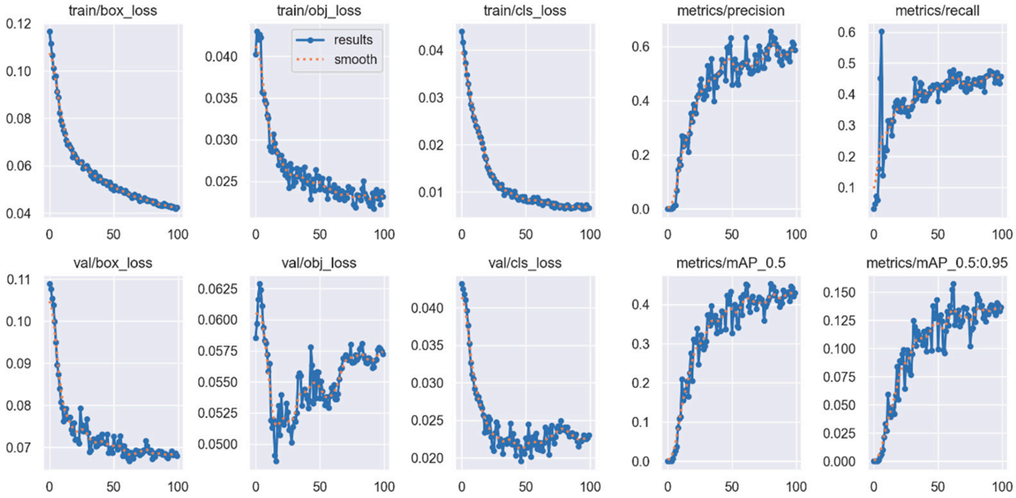


**Figure 16.** Result curves of the TS detection algorithm.

Figure 17 shows the result of the model detecting two TSs, and the model performed well in our TS detection task. The effectiveness of our model may benefit from the following factors. Since we use the time serial images of TSs, the detection of any part of a TS in consecutive frames is enough to recognize the motion of the TS. Meanwhile, the detections are conducted several times per second.

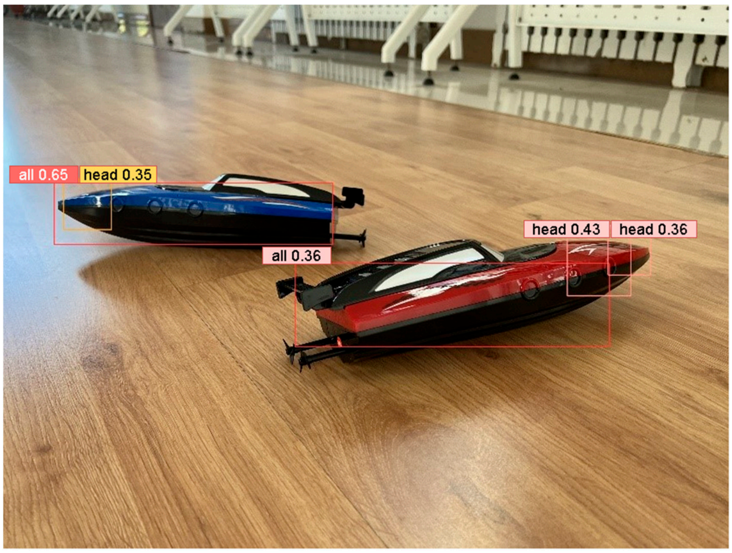


**Figure 17.** Illustration of TS detection.

## 6. Conclusions

This research proposes a two-stage path-planning method for multiple USVs based on the COLREGs. The method combines a cooperation module, a COLREGs-compliant action-evaluation module, and a TS detection module. The cooperative path-planning model for collision avoidance among USVs is constructed based on the MAPPO strategy, which utilizes a policy network capable of handling multiple aggregation goals, obstacles, and TSs. To achieve this, we define the action space, observation space, and reward function for the policy network, and design actor and critic networks.

- Monte Carlo experimental results confirm the effectiveness and efficiency of our path-planning method for formation aggregation and collision avoidance. We conducted these experiments by randomly specifying the positions of the USVs and obstacles. This approach allowed us to evaluate the performance of our method in diverse scenarios and validate its robustness.
- We benchmarked the simulation results against the MADDPG and MATD3 methods to validate the efficiency and the optimization performance of our approach.

After training the COLREGs-compliant action-evaluation calculation module using TS-avoiding trajectories, violations of USV actions that go against the COLREGs can be recognized and suppressed. This judgment is then used as heuristics for the actor network. Our reward function considers both COLREGs and seamanship principles.

- We conducted further experiments to test the feasibility of our collision-avoidance scheme based on the COLREGs within the USV fleet, as well as between USVs and TSs. We were able to confirm the practicality and effectiveness of our method in a realistic scenario.

The TS detection network is constructed based on the YOLO network and the squeeze-and-excitation scheme.

- Our proposed TS detection model performs well in our specific environment.

Primarily, we evaluate the algorithm through simulations conducted in gym environments. Additionally, we have conducted experiments on a semi-physical simulation platform where our algorithm acts as a local path planner, guiding the navigation of the system. The cooperative path-planning module is executed on individual ship-borne computers (PC-104) installed on each USV member. VxWorks 6.6 is utilized as the operating system, with Workbench 3.0 as the chosen development package.

In our future experiments, we aim to further advance our research by translating the algorithm into physical USVs. This will allow us to conduct practical implementation and comprehensive testing of the algorithm's capabilities.

**Author Contributions:** Conceptualization, N.W., R.Z. and G.L.; methodology, N.W., G.L., R.Z. and Y.L.; software, N.W., W.W. and Y.L.; data curation, Y.L., W.W. and D.J.; writing—original draft preparation, N.W.; writing—review and editing, N.W. and D.J.; visualization, N.W., Y.L. and W.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by grants from the Natural Science Foundation of Liaoning Province (No. 2020-MS-135), the National Natural Science Foundation of China (No. 62272067), and the Key Laboratory of Intelligent Perception and Advanced Control of State Ethnic Affairs Commission (No. MD-IPAC-2019103).

**Data Availability Statement:** The core code is available at https://github.com/WenNaifeng/Reinforcement-Learning-MAPPO.

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

## 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. Chakravarthy, A.; Ghose, D. Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **1998**, *28*, 562–574. [CrossRef]
3. Liang, X.; Qu, X.; Wang, N.; Li, Y.; Zhang, R. Swarm control with collision avoidance for multiple underactuated surface vehicles. *Ocean Eng.* **2019**, *191*, 106516. [CrossRef]
4. Liang, X.; Qu, X.; Wang, N.; Li, Y.; Zhang, R. A Novel Distributed and Self-Organized Swarm Control Framework for Underactuated Unmanned Marine Vehicles. *IEEE Access* **2019**, *7*, 112703–112712. [CrossRef]
5. Xia, J.; Luo, Y.; Liu, Z.; Zhang, Y.; Shi, H.; Liu, Z. Cooperative multi-target hunting by unmanned surface vehicles based on multi-agent reinforcement learning. *Def. Technol.* **2022**, *29*, 80–94. [CrossRef]
6. Xue, D.; Wu, D.; Yamashita, A.S.; Li, Z. Proximal policy optimization with reciprocal velocity obstacle based collision avoidance path planning for multi-unmanned surface vehicles. *Ocean Eng.* **2023**, *273*, 114005. [CrossRef]
7. Maza, J.A.G.; Argüelles, R.P. COLREGs and their application in collision avoidance algorithms: A critical analysis. *Ocean Eng.* **2022**, *261*, 112029. [CrossRef]
8. Kim, J.K.; Park, D.J. Understanding of sailing rule based on COLREGs: Comparison of navigator survey and automated collision-avoidance algorithm. *Mar. Policy* **2024**, *159*, 105894. [CrossRef]
9. Hu, L.; Hu, H.; Naeem, W.; Wang, Z. A review on COLREGs-compliant navigation of autonomous surface vehicles: From traditional to learning-based approaches. *J. Autom. Intell.* **2022**, *1*, 100003. [CrossRef]
10. Yim, J.B.; Park, D.J. Modeling evasive action to be implemented at the minimum distance for collision avoidance in a give-way situation. *Ocean Eng.* **2022**, *263*, 112210. [CrossRef]
11. Kim, J.K.; Park, D.J. Determining the Proper Times and Sufficient Actions for the Collision Avoidance of Navigator-Centered Ships in the Open Sea Using Artificial Neural Networks. *J. Mar. Sci. Eng.* **2023**, *11*, 1384. [CrossRef]
12. Hagen, I.B.; Vassbotn, O.; Skogvold, M.; Johansen, T.A.; Brekke, E.F. Safety and COLREG evaluation for marine collision avoidance algorithms. *Ocean Eng.* **2023**, *288*, 115991. [CrossRef]
13. Yang, Y.; Wang, J. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv* **2020**, arXiv:2011.00583.
14. Wang, X.; Wang, S.; Liang, X.; Zhao, D.; Huang, J.; Xu, X.; Dai, B.; Miao, Q. Deep reinforcement learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**. [CrossRef] [PubMed]
15. 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. Off. J. Int. Neural Netw. Soc.* **2022**, *152*, 17–33. [CrossRef] [PubMed]
16. Li, L.; Wu, D.; Huang, Y.; Yuan, Z. 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]
17. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; PMLR: Lille, France, 2015; pp. 1889–1897.
18. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
19. Lowe, R.; Wu, Y.I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; Mordatch, I. *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*; NeurIPS: Long Beach, CA, USA, 2017; pp. 1–12. [CrossRef]

20. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the International Conference on Continuous Control with Deep Reinforcement Machine Learning, Stockholm, Sweden, 10–15 July 2018; PMLR: Lille, France, 2018; pp. 1587–1596.
21. 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* **2015**, arXiv:1509.02971.
22. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of ppo in cooperative multi-agent games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.
23. 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]
24. 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]
25. Sawada, R.; Sato, K.; Majima, T. Automatic ship collision avoidance using deep reinforcement learning with LSTM in continuous action spaces. *J. Mar. Sci. Technol.* **2021**, *26*, 509–524. [CrossRef]
26. 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]
27. 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. [CrossRef]
28. Wei, G.; Kuo, W. COLREGs-compliant multi-ship collision avoidance based on multi-agent reinforcement learning technique. *J. Mar. Sci. Eng.* **2022**, *10*, 1431. [CrossRef]
29. Rongcai, Z.; Hongwei, X.; Kexin, Y. Autonomous collision avoidance system in a multi-ship environment based on proximal policy optimization method. *Ocean Eng.* **2023**, *272*, 113779. [CrossRef]
30. Skrynnik, A.; Yakovleva, A.; Davydov, V.; Yakovlev, K.; Panov, A.I. Hybrid policy learning for multi-agent pathfinding. *IEEE Access* **2021**, *9*, 126034–126047. [CrossRef]
31. Wang, K.; Kang, B.; Shao, J.; Feng, J. Improving generalization in reinforcement learning with mixture regularization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 7968–7978.
32. Khoi, N.D.H.; Van, C.P.; Tran, H.V.; Truong, C.D. Multi-Objective Exploration for Proximal Policy Optimization. In Proceedings of the 2020 Applying New Technology in Green Buildings (ATiGB), Da Nang, Vietnam, 12–13 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 105–109.
33. Tam, C.K.; Richard, B. Collision risk assessment for ships. *J. Mar. Sci. Technol.* **2010**, *15*, 257–270. [CrossRef]
34. Statheros, T.; Howells, G.; Maier, K.M. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *J. Navig.* **2008**, *61*, 129–142. [CrossRef]
35. Wen, N.; Zhao, L.; Zhang, R.B.; Wang, S.; Liu, G.; Wu, J.; Wang, L. Online paths planning method for unmanned surface vehicles based on rapidly exploring random tree and a cooperative potential field. *Int. J. Adv. Robot. Syst.* **2022**, *19*, 1–22. [CrossRef]
36. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. *IEEE Trans. Multimed.* **2018**, *20*, 2593–2604. [CrossRef]
37. Kim, J.H.; Kim, N.; Park, Y.W.; Won, C.S. Object detection and classification based on YOLO-V5 with improved maritime dataset. *J. Mar. Sci. Eng.* **2022**, *10*, 377. [CrossRef]

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