Meta-reinforcement Learning for Ship Autonomous Collision Avoidance

Xinyu Jia [a], Shu Gao [b,*]

[a] *School of Computer and Artificial Intelligence, Wuhan University of Technology, Wuhan, Hubei Province, P.R.China*
E-mail: jxy12000@gmail.com

[b] *School of Computer and Artificial Intelligence, Wuhan University of Technology, Wuhan, Hubei Province, P.R.China*
E-mail: gshu@whut.edu.cn

Abstract. Autonomous collision avoidance is critical for intelligent ship navigation. Ships encounter a variety of complex scenarios in real-world navigation environments, necessitating improvements in the adaptability and effectiveness of collision avoidance policies. Therefore, a meta-reinforcement learning method is proposed for ship autonomous collision avoidance. Inspired by meta-learning, we designed a two-layered recurrent model to enhance the adaptability and effectiveness of collision avoidance policies. Then, we created a task sampling method to train vessel agents in making collision avoidance decisions for high-risk encounter situations. The objective function and policy gradient method for risk assessment are designed to enable vessel agents to thoroughly evaluate the risk situation of the current encounter scenario and optimize the collision avoidance policy. Lastly, we conducted the simulation experiments to validate the feasibility of our work. The outcomes indicate that the collision avoidance policies outperform various comparative methods, exhibiting competitive advantages in adaptability, effectiveness, and safety across diverse encounter scenarios. Overall, our novel method provides a safer solution to enhance intelligent ship navigation.

Keywords: Meta-reinforcement Learning, Reinforcement Learning, Artificial intelligence, Autonomous ship, Collision Avoidance

1. Introduction

Intelligent ships with autonomous navigation capabilities are becoming the inevitable trend for transportation. Particularly, with the rapid advancement of artificial intelligence, big data, and maritime Internet of Things technologies, research on critical technologies for ship intelligence has gained tremendous importance and attention. One of the foundational problems that need to be solved first is the issue of ship autonomous collision avoidance.

Recent developments in ship collision avoidance have employed deep reinforcement learning, a more human-like approach to artificial intelligence. Nevertheless, most ship collision avoidance techniques using deep reinforcement learning face the following primary challenges: limited adaptability to new maritime environments. Deep reinforcement learning algorithms focus on acquiring policies for specific tasks. For this reason, each navigation environment necessitates trial-and-error learning from scratch and accumulating experience. Even for the same navigation environment, obtaining a reasonably effective collision avoidance policy requires thousands of interactions between the ship and the collision avoidance environment. These challenges have made the application of these methods considerably difficult.

Meta-reinforcement learning can be an effective approach for improving the performance of autonomous collision avoidance for ships. Meta-reinforcement learning is a learning framework capable of acquiring general policies from multiple tasks, and it en-

[*] Corresponding author. Shu Gao, Department of Computer and Artificial Intelligence, Wuhan University of Technology, Wuhan, Hubei Province, P.R.China E-mail: gshu@whut.edu.cn.

ables agents to learn how to learn more efficiently during training and to facilitate experience sharing across tasks. As a result, it has been applied extensively to studying autonomous driving, robotics routing planning, and gaming intelligence[1]. Since Markov games exhibit similarities with autonomous collision avoidance for ships, the vessels can be mapped into agents, and meta-reinforcement learning can address the autonomous collision avoidance problem. Specifically, as vessel agents learn general avoidance policies for different encounter situations, they can rapidly generate corresponding avoidance decisions in new situations to enhance the adaptability of avoidance methods. Furthermore, vessel agents can use experiences of avoidance learned from past scenarios to enhance the training effectiveness of the avoidance policy, enabling them to plan more appropriate avoidance policies and ultimately improve the effectiveness of the avoidance decision. Therefore, using meta-reinforcement learning provides a promising new solution for realizing autonomous collision avoidance for ships.

However, while ships are sailing, especially in restricted waters or narrow channels where multiple vessels are present, there are high-risk encounter situations where the probability of vessel collision increases. This poses a threat to the vessels and has severe consequences, such as environmental damage, injury, or death of personnel. Therefore, reducing collision risk in these situations is crucial. However, when using meta-reinforcement learning directly, the collision avoidance policy learned by vessel agents generally performs poorly in high-risk encounter situations because vessel agents have not received specific training for these scenarios. Consequently, to learn safer collision avoidance policies, it is necessary to improve the sampling methods during collision avoidance policy training and optimize the objective function and policy gradient method of the collision avoidance policy.

To summarize, we have made the following notable contributions:

1) Building a collision avoidance decision-making model for ships based on meta-reinforcement learning (SACMRL) is a two-layered recurrent model. In the inner loop, the vessel agent interacts with the collision avoidance environment and utilizes accumulated knowledge from multiple encounter scenarios to devise appropriate autonomous collision avoidance policies for a given situation, enhancing the effectiveness of the policy. In the outer loop, vessel agents learn and optimize meta-collision avoidance policies for different scenarios, enabling them to make decisions based on meta-collision avoidance policies in new encounter situations, improving the adaptability of the policy. Through iterative optimization of the inner and outer loops, the vessel agents can quickly design effective and highly adaptable avoidance policies for complex encounter scenarios. To our knowledge, this represents the first application of meta-reinforcement learning to solve the problem of autonomous collision avoidance for ships.

2) Improving sampling strategy for collision avoidance during training by sampling scenarios in which collision is most likely to occur. This sampling strategy enhances the safety of SACMRL under high-risk encounter scenarios, thereby fortifying the vessel agents' training. Additionally, we optimized the objective function and policy gradient method of the collision avoidance policy to ensure that the policy could accurately assess collision risks.

3) Building a simulation platform to verify the collision avoidance model achieved competitive outcomes in adaptability and effectiveness in new encounter scenarios. The collision avoidance policy exhibited better safety in high-risk encounter situations, thus laying the foundation for further advancing ship intelligent sailing capabilities.

2. Related Work

Deep reinforcement learning algorithms can train intelligent agents and equip them with autonomous collision avoidance capabilities, making them suitable for solving the problem of autonomous ship collision avoidance. Shen et al.[2]proposed an intelligent collision avoidance algorithm based on Deep Q Network (DQN), utilizing concepts such as ship domains and predicted danger zones to describe areas of obstacles and facilitate intelligent collision avoidance in complex water environments. Chen et al.[3] proposed a ship collision avoidance method based on Q-Learning path planning, enabling autonomous ship navigation along suitable paths or navigation policies without requiring human expertise, and compared the algorithm against traditional path planning methods, showing promising results. Xu et al.[4]proposed a deep reinforcement learning path planning and dynamic collision avoidance algorithm based on COLREGs, which enables safe navigation for Unmanned Surface Vehicles (USVs) in water environments. Zhai et al. [5] proposed an improved DQN-based multi-ship autonomous collision avoidance method, which quantifies regions of potential collision risks in the future as inputs to the intelligent agent and utilizes a prioritized experience replay method to accelerate model convergence. Sui et al.[6] proposed a ship collaborative collision avoidance decision-making method based on multi-agent deep reinforcement learning, enhancing vessel agent decisions' coordination, safety, and practicality in collision avoidance scenarios. Nonetheless, these deep reinforcement learning-based algorithms may need more adaptability. Collision avoidance policies learned by vessel agents may not achieve desired avoidance outcomes when encountering new collision scenarios.

To address the issue of limited task adaptability in deep reinforcement learning algorithms, some researchers have started exploring the integration of meta-learning into deep reinforcement learning. They proposed meta-reinforcement learning algorithms to improve the performance of reinforcement learning algorithms by learning how to learn. Meta-reinforcement learning algorithms focus on acquiring meta-policies that enable agents to adapt and learn in new tasks or environments quickly. By leveraging meta-learning, these algorithms aim to enhance deep reinforcement learning systems' adaptability and generalization capabilities. Rakelly et al.[7] introduced Probabilistic Embeddings for Actor-critic RL (PEARL) algorithm, which incorporates explicit context as a policy input and combines off-policy reinforcement learning with probabilisticly latent context. It separates task inference and agent learning, thereby improving meta-training efficiency to enable fast adaptation to new tasks. Luisa et al.[8] proposed the variational Bayes-Adaptive Deep RL (variBAD) algorithm, which addresses discrete problems by discretizing the continuous action space and employing a model-based reinforcement learning approach. The algorithm also utilizes an explorer-exploiter pair method, allowing it to effectively explore different regions of the state space using discrete actions. Seyed et al. [9]proposed a Brain Inspired Meta Reinforcement Learning (BIMRL), inspired by the neural principles of the brain. They designed a hierarchical architecture and a memory module, aiming to adapt to new tasks and retain experience quickly. By incorporating an intrinsic reward mechanism to promote exploration, the model's sample efficiency and generalization performance are significantly enhanced. Precisely, through training and optimization using meta-reinforcement learning algorithms, intelligent ships' learned collision avoidance policy can be adapted to different encounter situations.

Reinforcement learning involves agents interacting with the environment to gather data for policy learning and optimization. As a result, efficiently sampling and utilizing samples to optimize policies poses a significant challenge. Zha et al. [10]proposed the Experience Replay Optimization (ERO) algorithm, which is built upon the Deep Deterministic Policy Gradient (DDPG) framework. They introduced the concept of sample selection networks, which apply reinforcement learning ideas to choose more efficient samples for action policy optimization. ERO enhances the action policy indirectly through the sample selection networks and updates the sample selection network parameters based on feedback from sample manipulation. However, the effectiveness of this approach is constrained by the challenging process of network training. Liu et al. [11] introduced the Constrained Variational Policy Optimization (CVPO) algorithm to enhance the utilization efficiency of samples. The algorithm decomposes the problem of safety reinforcement learning into two stages: convex optimization and supervised learning. In the convex optimization stage, the algorithm calculates a feasible distribution and seeks the optimal nonparametric variational distribution. In the supervised learning stage, the algorithm enhances the policy parameters based on the optimal variational distribution within a trust region, achieving a more stable training performance. Greenberg et al. [12]employed the Cross Entropy Method (CEM) to select more challenging tasks from the task distribution. They used Conditional Value-at-Risk (CVaR) as the objective function and oversampled complex tasks to enhance the algorithm's robustness. Using CEM and focusing on challenging tasks, their approach improved robustness in solving complex problems. During the actual navigation process of ships, there are often high-risk areas where collisions are more likely to occur. It is vital to provide extensive training to the intelligent agents operating in these high-risk zones to ensure the reliability of the autonomous collision avoidance policies on the ship. Therefore, it is crucial to adequately sample high-risk scenarios during the meta-reinforcement learning algorithms' training process. Additionally, it is vital to enable vessel agents to perform comprehensive and accurate risk assessments during the learning process of collision avoidance policy. This approach ensures vessel agents can acquire reliable collision avoidance policies and prevent collision accidents in high-risk areas.

3. Methods

3.1. Model Architecture

Fig. 1 shows the architecture of the Ship Autonomous Collision Avoidance Decision Model based on meta-reinforcement learning(SACMRL). This model uses the idea of meta-learning to design the inner loop and outer loop. An additional loop layer is introduced to optimize the update of collision avoidance policies and the interaction between vessel agents and the environment within the basic framework of the ship autonomous collision avoidance meta-reinforcement learning algorithm. This two-layered loop setting enhances the learning of efficient collision avoidance policies and improves adaptability in different encounter scenarios.

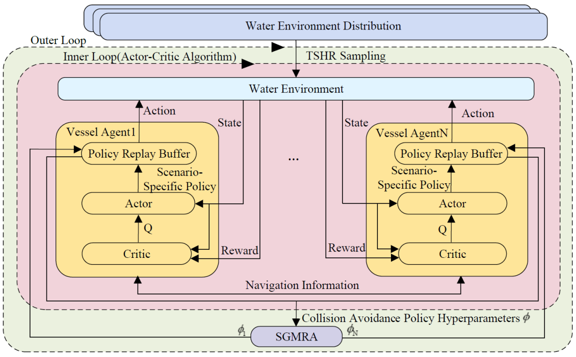


Fig. 1. The model architecture of SACMRL

The inner loop mainly consists of vessel agents and the environment. Since ship autonomous collision avoidance decision-making bears similarities to Markov decision processes, ships are modeled as vessel agents in autonomous collision avoidance decision-making, with their state space, action space, and reward function are designed (according Section 3.2). In the inner loop, the vessel agents receive the meta-collision avoidance policy transmitted from the outer loop (which is a function of collision avoidance policy hyperparameters). Then, using the TSHR method, high-risk encounter scenarios are sampled, allowing the collision avoidance policy to interact with them during the training process, thereby enhancing the safety of the collision avoidance policy (according Section 3.3). Finally, the vessel agents exchange navigation information (such as current position and heading) with each other. Using the Actor-Critic framework-based reinforcement learning algorithm, they learn collision avoidance policies applicable to specific encounter scenarios.

After the completion of the inner loop, the hyperparameters of collision avoidance policies applicable to specific encounter scenarios obtained from the inner loop are fed back to the outer loop. The outer loop is mainly responsible for updating the meta-collision avoidance policy. In order to ensure the safety of the collision avoidance policy, an objective function that considers collision risk and the corresponding policy gradient method SGMRA are designed (according Section 3.4). Based on performance metrics in specific encounter scenarios, the hyperparameters of collision avoidance policies are evaluated and optimized, and the meta-collision avoidance policy is updated to further enhance its performance. Subsequently, the updated meta-collision avoidance policy is transmitted back to the inner loop for use in the next round of training and interaction. This iterative interaction process between the inner and outer loops is continuously performed until the preset termination condition is met.

To summarize, vessel agents interact with the environment within the inner loop, acquiring collision avoidance policies tailored to specific encounter scenarios. They then update the sampled environment using the outer loop. This process allows vessel agents to learn adaptive collision avoidance policies, especially in high-risk encounter scenarios. The continuous iteration of the two-layered loop ensures the constant optimization of collision avoidance policies. As a result, vessel agents possess more effective, safer, and more adaptable autonomous collision avoidance policies.

3.2. Design of State Space, Action Space, and Reward Function

This section focuses on designing the state space, action space, and reward function for vessel agents, enabling them to make effective decisions in autonomous collision avoidance. The vessel agents' state space defines their current information and environmental conditions. The action space determines the available actions. The reward function evaluates the outcome of the vessel agents' actions across various states. By leveraging the reward function, vessel agents can acquire effective collision avoidance policies. Consequently, this establishes the essential groundwork for the meta-reinforcement learning-based decision model in ship autonomous collision avoidance.

3.2.1. Design of State Space

Design of the state space of vessel agents is based on the characteristics and navigation objectives of ships, as shown in Equation 1.

$$o_t^i = \{\psi_t^i, V_t^i, R_t^i, P_t^i, \left\|P^{g_i} - P_t^i\right\|, \left\|P_t^j - P_t^i\right\|\} \quad (1)$$

Here, o_t^i is the navigation information of the vessel agent i at time step t,ψ_t^i, V_t^i, R_t^i and P_t^i respectively represent the heading angle, speed, rudder angle, and position of the vessel agent i at time step t . $\left\|P^{g_i} - P_t^i\right\|$ is the relative positions of the target point g_i with respect to the vessel agent i. $\left\|P_t^j - P_t^i\right\|$ is relative positions of the vessel agent j with respect to the vessel agent i.

3.2.2. Design of Action Space

During the navigation process, ships typically prefer maintaining a consistent heading and speed. In practice, the ship's engine speed remains constant while the rudder maintains a slight angle during turning. As a result, we assume that the ship's engine speed remains constant, making the rudder angle the sole variable controlled by the vessel agent. Hence, the vessel agent's action space is solely defined by the range of possible rudder angles. In real-world navigation, the rudder angle generally varies between $[-35°, 35°]$. Therefore, the action space is designed with five actions, specifically $[-35°, -15°, 0°, 15°, 35°]$, where negative values represent turning the rudder to the left. Combining A and B, the state space of the vessel agent is adjusted as shown in Equation 2 due to the assumption that the ship's speed remains relatively constant.

$$o_t^i = \{\psi_t^i, R_t^i, P_t^i, \left\|P^{g_i} - P_t^i\right\|, \left\|P_t^j - P_t^i\right\|\} \quad (2)$$

The variables in Equation 2 have identical meanings as those in Equation 1.

3.2.3. Design of Reward Function

Integrating ship navigation objectives and collision avoidance safety in the design of a reward function enables vessel agents to develop effective collision avoidance policies. During the process of autonomous collision avoidance, vessel agents consider crucial factors such as heading deviation, potential collisions between vessel agents, and potential collisions between vessel agents and static obstacles (such as islands and reefs), influencing their decision-making.

i. Heading deviation The reward value for navigation is calculated based on the distance and angle between the vessel agent and the target point. A higher reward value is assigned when the heading direction aligns closely with the line connecting the vessel agent and the target point when it is closer to its destination. This reward guides the vessel agent in moving toward the target. The distance $R_t^i(dist)$ between the position of ship agent i at time step t and its target point is calculated as shown in Equation 3.

$$R_t^i(dist) = -\left\|P^{g_i} - P_t^i\right\| \quad (3)$$

Here, P^{g_i} is the target point location for vessel agent i. P_t^i is the position of vessel agent i at time step t. $\left\|P^{g_i} - P_t^i\right\|$ is the distance between the target point and the current position of vessel agent i.

The reward value $R_t^i(angle)$ for heading is shown in Equation 4.

$$R_t^i(angle) = -(\psi_{in} - \psi_{iv}) = - \triangle \psi \quad (4)$$

Here, ψ_{in} is the angle between the line connecting vessel agent i and the target point and the x-axis. ψ_{iv} represents the heading angle. $- \triangle \psi$ represents the heading angle difference.

Therefore, the reward $R_t^i(lane)$ for vessel agent's heading deviation is shown in Equation 5.

$$R_t^i(lane) = R_t^i(dist) + R_t^i(angle) \quad (5)$$

ii. The occurrence of collisions between vessel agents When a collision transpires among vessel agents, a substantial negative reward value should be assigned, as indicated in Equation 6.

$$R_t^i(boat) = \begin{cases} -\lambda_{collision}, & collision \\ 0, & non-collision \end{cases} \quad (6)$$

Here, $\lambda_{collision} \in R^+$.

iii. The occurrence of collisions between vessel agent and static obstacles When a collision transpires between the vessel agent and static obstacles, a substantial negative reward value should be assigned, denoted as indicated in Equation 7.

$$R_t^i(obst) = \begin{cases} -\lambda_{collision}, & collision \\ 0, & non-collision \end{cases} \quad (7)$$

Based on these three factors, the reward function for vessel agent i at time step t is shown in Equation 8.

$$R_t^i = \begin{cases} R_t^i(lane) + R_t^i(boat) + R_t^i(obst), & collision \\ R_t^i(dist), & non-collision \end{cases} \quad (8)$$

3.3. Task Sampling Method for High-Risk Encounter Scenarios

During ship navigation, particularly in constrained and narrow waterways, there are chances where multiple ships navigate simultaneously, leading to high-risk encounter situations. In such scenarios, the likelihood of collisions among ships rises. However, traditional random sampling methods predominantly focus on low-risk encounter scenarios during the training process, posing a challenge for vessel agents to acquire efficient collision avoidance policies for high-risk encounters. Hence, there is a necessity to enhance sampling methods to improve the safety of collision avoidance policies.

The Cross Entropy Method (CEM) [13] is a widely used approach for rare event sampling and optimization. It iteratively samples from a parameterized probability distribution, with a focus on the tail region of the distribution, which is characteristic of infrequent but significant low-probability events. This enables better exploration and identification of samples with higher values rather than being dominated by ordinary events. Therefore, employing the CEM as a sampling technique in the training process of optimizing ship autonomous collision avoidance policies allows the vessel agents to pay more attention to encounter scenarios with higher-risk probabilities. This will facilitate learning how to handle these complex and critical situations, ultimately enhancing the safety of collision avoidance policies.

During the training of vessel agents, a higher frequency of observed collision incidents in encountered scenarios indicates a higher collision risk. Thus, the average reward obtained by vessel agents during training in these scenarios serves as an evaluation index for collision risk. A higher average reward signifies a lower collision risk in encountered scenarios, whereas a lower average reward implies a higher collision risk.

In conclusion, our Task sampling method for high-risk encounter scenarios (TSHR, Algorithm 1) utilizes the CEM to sample encounter scenarios from the environment with lower average reward values, thereby constructing a training set with high collision risk. This training set is utilized to update the distribution parameters of the ship's autonomous collision avoidance task, which adjusts the probability distribution of encounter scenario samples. This adjustment significantly enhances the likelihood of selecting task samples involving high-risk collisions, ultimately bolstering the safety of the ship's autonomous collision avoidance policy in high-risk encounter scenarios.

Algorithm 1 Task sampling method for high risk collision scenarios (TSHR)

```
Input: Initial vessel agent encounter scene distribution $D_\phi$, Number of iterations $T$, Sampling quantity $N$, Number of high-risk samples $h$
Output: Final vessel agent encounter scene distribution $D_T$
Initialize the parameters of the vessel agent encounter scene distribution $\phi_0$
for $t \in \{1, 2, ..., T\}$ do
    Sample $N$ ship encounter scene samples $Z_N$ from distribution $D_{\varphi_t - 1}$
    for $Z_i \in \{Z_1, Z_2, ..., Z_N\}$ do
        Evaluate the risk value $R_i$ of $Z_i$
        Sort $Z_i$ based on $R_i$ from smallest to largest
    end for
    High-risk encounter scenario set $H = \{Z_1, ..., Z_h\}$
    for $Z_i \in H$ do
        Calculate the probability $P_i$ of each $Z_i$ occurring, where $P_i = \frac{F_{Z_i}}{T}$, $F_{Z_i}$ is the frequency of occurrence of $Z_i$
        Update the ship encounter scenario distribution parameter $\phi_i = P_i$
    end for
    $\phi = \{\phi_1, ..., \phi_h\}$
    Employ the maximum likelihood estimation method to refine the distribution parameter $\phi$, Here, the likelihood function is represented by $L(\phi) = f(Z_1; \phi) \times ... \times f(Z_h; \phi)$; $f(Z_h; \phi)$ is the probability density function of the encounter scenario samples under the given parameter $\phi$
end for
Obtain the final ship encounter scene distribution $D_T$ based on the final ship encounter scene distribution parameter $\phi_T$
```

3.4. The Objective Function and Policy Gradient Method for Risk Assessment

In the outer loop of the SACMRL (refer to the green area in Fig. 1), meta-collision avoidance policies are employed. These policies utilize the objective function as guidance and continually update and optimize to acquire collision avoidance policies with high safety. However, directly applying meta-reinforcement learning algorithms to ship autonomous collision avoidance problems may result in the vessel agent overlooking potential risks associated with its chosen actions. This occurs because the main objective of meta-collision avoidance policies is to maximize reward values. Consequently, the vessel agent may struggle to select appropriate safe actions in potentially hazardous situations. Therefore, while updating meta-collision avoidance policies, it is imperative to design an objective function that takes collision risks into account in order to ensure the safety of the collision avoidance policies.

Conditional Value at Risk(CVaR)[12] is a risk metric employed to evaluate the level of risk associated with a portfolio or decision. By employing CVaR as the objective function to update the meta-collision avoidance policy, the vessel agent can probabilistically assess the risk of collisions. This approach facilitates a comprehensive and precise evaluation of collision risks by the learned meta-collision avoidance policy, leading to a reduction in potential hazards and an enhancement in the safety of ship autonomous collision avoidance policies. Equation 9 is the objective function employed for updating meta-collision avoidance policies, aiming to maximize the expected return while accounting for collision risk.

$$\begin{gathered} \underset{\pi}{argmax} J_{\alpha}^{\theta}(R) \\ J_{\alpha}^{\theta}(R) = CVaR_{\tau}^{\alpha}[R] = \int_{-\infty}^{q_{\alpha}^{\theta}(R)} x P_{\tau}^{\theta}(x) dx \end{gathered} \tag{9}$$

Here, α is the learning rate. θ is the policy parameters. τ is the distribution of encounter scenarios in ship navigation task D. R is the return of the meta-collision avoidance policy in task τ of ship autonomous collision avoidance. P_{τ}^{θ} is the conditional probability density function of the return value R, and $q_{\alpha}^{\theta}(R)$ is the α-quantile of R.

The meta-collision avoidance policies can be acquired using the Policy Gradient (PG) method. This approach calculates the gradient of the objective function for the policy parameters, indicating the degree of policy performance variation in response to parameter changes. By updating the parameters based on the gradient direction, the meta-collision avoidance policies can explicitly incorporate the consideration of risk level throughout the training process. Consequently, the Strategy Gradient Method for Risk Assessment (SGMRA) is designed to:

$$\nabla_{\theta} J_{\alpha}^{\theta}(R) \approx \frac{1}{\alpha N} \sum_{i=1}^{N} 1_{R_i} \leq \hat{q}_{\alpha}^{\theta} \sum_{m=1}^{M} g_{i,m} \tag{10}$$

$$g_{i,m} = (R_{i,m} - b) \sum_{k=1}^{K} \sum_{t=1}^{T} \nabla_{\theta} \log \pi_{\theta} \left(a_{i,m,k,t}; \widetilde{s}_{i,m,k,t} \right) \tag{11}$$

where $\hat{q}_{\alpha}^{\theta}$ is an estimator of the current return quantile, M is the number of independent and identically distributed meta-rollouts, where each meta-rollout contains K episodes. $R_{i,m}$ is the return obtained by the vessel agent for task i in the m meta-rollout. b is an arbitrary baseline independent of θ. T is the number of steps in each episode. $a_{i,m,k,t}$ and $\widetilde{s}_{i,m,k,t}$ represent the action and state of the vessel agent under encounter scenario i, meta-rollout m, episode k, and step t.

By designing an objective function for risk assessment and the corresponding policy gradient method, our model enables vessel agents to make decisions while considering collision risk, thereby improving the safety of collision avoidance policies.

3.5. Algorithm for SACMRL

The SACMRL algorithm utilizes a double-loop structure, introduces the TSHR method, and optimizes the risk-oriented assessment function, which offers an innovative solution for autonomous ship collision avoidance, as explicitly outlined in Algorithm 2.

4. Experiments

4.1. Experimental Environment Setup

4.1.1. Experimental Environment

The simulation framework, based on ROS and Gazebo, provides a highly accurate and efficient simulation environment. ROS simulates a wide range of functional devices, including sensors and controllers. Gazebo uses multiple physics engines to calculate the motion and the influence of forces on objects while providing 3D visualization. The ROS-Gazebo framework is widely regarded as the most suitable 3D simulation software currently available. The USVsim simulator [14] provides modular components for navigation and control, facilitating simulation testing scenarios for ships. Therefore, we construct a simulation test environment for ship autonomous collision avoidance using the USVsim simulator. This environment serves to verify and evaluate the performance of the proposed ship autonomous collision avoidance decision model.

The ship autonomous collision avoidance decision-making model based on meta-reinforcement learning was evaluated for its adaptability under different encounter situations in two distinct Water Areas, A and

Algorithm 2 Ship Autonomous Collision Avoidance Decision Model based on meta-reinforcement learning(SACMRL)

```
Input: Starting heading angle, starting coordinates, target point coordinates, and static obstacle coordinates of $n$ vessel agents
Output: Autonomous collision avoidance policies of each vessel agent
//Meta-training phase
Initialize environment parameters, initial task distribution $D_\psi$, autonomous ship collision avoidance meta-task set $T$, autonomous ship collision avoidance policy hyperparameters $\phi_0$, number of sample $N$ per round, number of meta-rounds $M$, and number of steps $T$ executed in each episode within the inner loop
Sampling a small batch of samples $D_T$ from $D_\phi$ using Algorithm 1
for $D_{T_i} \in \{D_{T_1}, ..., D_{T_M}\}$ do
    Initialize the meta collision avoidance policy parameter $\theta_z$
    for $episode \in \{1, 2, ..., K\}$ do
        Vessel agent obtains current environmental information
        for $t \in \{1, 2, ..., T\}$ do
            Vessel agent selects the actions using the current meta collision avoidance policy $\pi_{\theta_t}$
            Vessel agent executes the selected action and observes the next state and reward
            Vessel agent records the current experience information and obtains the meta-reward $R_{i,m}$ using Equation 8
        end for
    end for
    Use Equation 10 and 11 to update the policy avoidance hyperparameters $\theta_z$
end for
//Meta-testing phase
Initialize the meta-test samples $D_{test}$ and the number of meta-test rounds $K_t$
for $episode \in \{1, 2, ..., K_t\}$ do
    Vessel agent obtains current environmental information
    for $t \in \{1, 2, ..., T\}$ do
        Vessel agent selects the actions using the current meta collision avoidance policy $\pi_{\theta_t}$
        Vessel agent executes the selected action and observes the next state and reward
        Vessel agent records the current experience information and obtains the meta-reward $R_i$ using Equation 8
        Calculate the total reward
        Use Equation 10 and 11 to update the policy avoidance hyperparameters $\theta_z$
    end for
end for
```

B (Fig. 2). Water Area A served as the training environment, where the ship interacted and learned adaptive collision avoidance policies. The ships and target points in the training water area were not depicted in the figure, as they were randomly generated. To comprehensively assess the model's adaptability, water Area B was chosen as the testing area, incorporating vessel agents(R, G, W), starting points (initial positions of vessel agents), static obstacles (yellow circles, primarily representing breakwaters), and target points (DR, DG, DW).

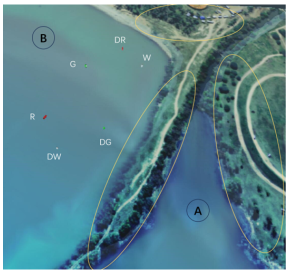


Fig. 2. General encounter experimental scenario

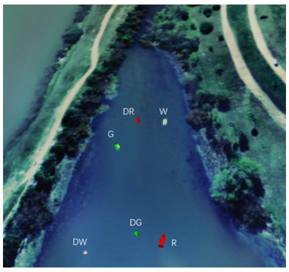


Fig. 3. High-risk encounter experimental scenario

4.1.2. Parameter Settings

Our experiment utilized an Ubuntu 16.04 virtual machine to establish the simulation software environment. The software was developed using the PyTorch framework and implemented in Python 3.7, incorporating the Gym, Numpy, and Pygame libraries. Extensive experimentation and parameter tuning were conducted to determine the optimal values for the main algorithm parameters, presented in Table 1.

4.1.3. Evaluation Metrics

We employ three evaluation criteria(adaptability, effectiveness, and safety) to assess collision avoidance policies. The adaptability of the policies is evaluated by training them in Water Area A and testing them in Water Area B using newly designed encounter scenarios. A higher reward value from vessel agents in these scenarios indicates greater adaptability. The effectiveness of the policies is measured by the average cumulative reward values obtained during the training period, with higher rewards indicating more effective learned policies. The safety is assessed by recording the frequency of collisions during ship collision avoidance experiments. Additionally, we use the visualization of resulting path diagrams to verify the effectiveness of collision avoidance.

Table 1

Parameter Settings

Parameter Name	**Parameter Meaning**	**Parameter Value**
policy_rl	Inner loop reinforcement learning algorithm	ATOC
lr_policy	Learning rate	0.002
policy_gamma	Reward discount rate	0.95
batch_size	Batch sample size	512
actor_lr	Learning rate of the actor neural network in the inner loop reinforcement learning algorithm	0.01
critic_lr	Learning rate of the critic neural network in the inner loop reinforcement learning algorithm	0.05
classifier_lr	Learning rate of the classifier in the inner loop reinforcement learning algorithm	0.005
rollout_length	Collision avoidance policy hyperparameter update frequency	4
γ	Discount factor	0.9

4.2. Experimental Results and Analysis

4.2.1. Evaluation of the Effectiveness of Collision Avoidance Policy

To assess the effectiveness of the SACMRL(ours), we conducted meta-training experiments on ship collision avoidance policies within the Water Area A. The performance of the SACMRL (ours) was compared against the approaches proposed in MSCCADRL [6], TRPO[15], and MA2CL [16]. To ensure fair and unbiased comparisons, we extracted the essential algorithms from the references mentioned earlier, established an identical state space, action space, and reward function in the experimental environment of our work, and appropriately adjusted the relevant parameters.

The average cumulative reward values of the same episodes during the meta-training phase of the collision avoidance policy were calculated using the evaluation method described in Section 4.1.3. Fig. 4 visually presents the training reward values. Table 2 provides the statistical results, including the average cumulative reward values and their standard deviation for the last 5000 episodes. Fig. 4 and Table 2 showcase the improved effectiveness of the SACMRL algorithm(ours) for enhancing ship autonomous collision avoidance policies. This improvement is due to the a two-layered design of the SACMRL algorithm(ours). In the inner loop, the vessel agent can interact with the collision avoidance environment and optimize the avoidance policies based on experiential knowledge from multiple encounter scenarios, resulting in a highly effective collision avoidance policy.

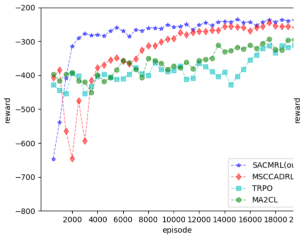


Fig. 4. Meta-training reward values

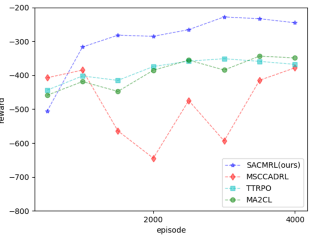


Fig. 5. Meta-testing reward values

Table 2

Average cumulative rewards and standard deviation during meta-training

Algorithm	**Average Cumulative Rewards**	**Standard Deviation**
MSCCADRL	-257.99	6.40
TRPO	-329.76	22.26
MA2CL	-310.07	11.88
SACMRL(ours)	-240.75	5.02

Table 3

Average cumulative rewards and standard deviation during meta-testing

Algorithm	**Average Cumulative Rewards**	**Standard Deviation**
MSCCADRL	-465.44	81.8
TRPO	-359.23	5.91
MA2CL	-358.17	15.89
SACMRL(ours)	-243.06	14.43

4.2.2. Evaluation of the Adaptiveness of Collision Avoidance Policy

To assess the adaptability of the SACMRL algorithm(ours), we conducted meta-testing experiments on ship collision avoidance policies. The policy learned in Water Area A was applied in Water Area B. Similarly, comparative experiments were conducted between the SACMRL algorithm(ours) and the methods presented in MSCCADRL[6], TRPO[15], and MA2CL [16]. The parameter adjustment was performed following the same procedure as described in Section 4.2.1.

We present statistical data on the average cumulative reward of ship collision avoidance policies in Test Water Area B during 4000 learning episodes(Fig. 5). Table 3 provides statistical information on the average cumulative reward and standard deviation for the last

2000 episodes. The analysis of Fig. 5 and Table 3 reveals that the SACMRL algorithm(ours) achieves the highest average reward for vessel agents in the testing area, indicating its strong adaptability to the collision avoidance policy in the test scenario. Moreover, the standard deviation of the average cumulative reward for the last 2000 episodes using the SACMRL algorithm(ours) is lower than those in MSCCADRL algorithm and MA2CL algorithm, implying higher stability in the obtained collision avoidance policy. Thus, the experiment demonstrates that the SACMRL algorithm can effectively generate collision avoidance policies with strong adaptability. This improvement is due to the two-layered design of the SACMRL algorithm, enhancing the adaptability of collision avoidance policies. By sampling and updating the environment in the outer loop, the vessel agent gains exposure to multiple diverse environments during training. This allows the vessel agent to leverage learned knowledge and updated meta-collision avoidance policies from previous experiences when encountering new scenarios. Consequently, the vessel agent is better equipped to make effective decisions in novel encounter situations, thereby improving the overall adaptability of collision avoidance policies.

4.2.3. *Evaluation of the Safety of Collision Avoidance Policy*

To assess the safety of the SACMRL algorithm(ours), we constructed an experimental map in Fig. 3. Within this map, three vessel agents simultaneously go through a narrow water area, presenting a significant risk of potential collisions. This experimental setting depicts a high-risk encounter scenario.

The SACMRL algorithm (ours) is compared with the methods referenced in MSCCADRL [6], TRPO[15], and MA2CL[16] through a comprehensive experimental analysis. We generated and evaluated the algorithm on 100 random experimental maps, carefully measuring and documenting the number of ship collisions in 100 avoidance experiments. Fig. 6 presents a statistical graph illustrating the frequency of ship collisions. The graph analysis indicates that the SACMRL algorithm (ours) outperforms the compared algorithms, exhibiting a notably lower incidence of collisions. These results indicate that the SACMRL algorithm (ours) offers enhanced safety. Some possible reasons are:

- The SACMRL algorithm(ours) utilizes the TSHR method to sample high-risk encounter scenarios, facilitating comprehensive training of collision avoidance policies in such scenarios to improve safety.
- The SACMRL algorithm(ours) optimizes the objective function of the collision avoidance policy and employs the SGMRA method to accurately evaluate collision risks, enhancing safety through a comprehensive and precise assessment approach.

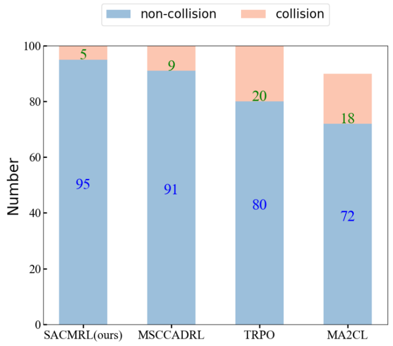


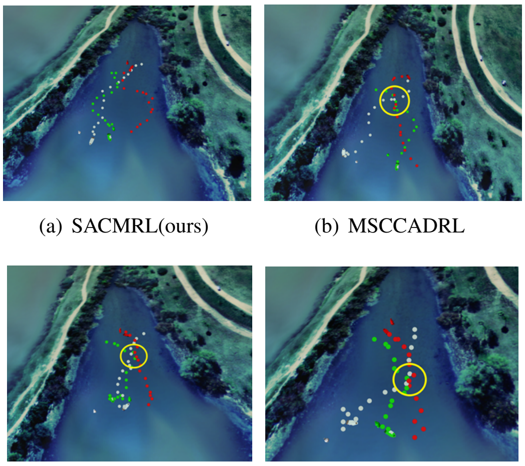
Fig. 6. Collision frequency results in high-risk encounter scenario

Figure 7 is a visualization of resulting path diagrams in one randomly selected instance out of the 100 experiments conducted. As depicted in Figure 7(a), the collision avoidance path planned by the SACMRL algorithm (ours) does not exhibit any notable collision risks. Conversely, MSCCADRL [6], TRPO[15], and MA2CL [16] result in collision situations, as indicated within the yellow circles in Figure 7(b), (c), and (d). Consequently, the SACMRL algorithm (ours) can generate collision avoidance policies with enhanced safety attributes, offering valuable insights into the realization of intelligent navigation for ships.

4.2.4. *Ablation Experiment*

To validate the effectiveness of TSHR and SGMRA, a comparative experiment was conducted between the SACMRL algorithm(ours) and the SACMRL/TS algorithm without the TSHR and SGMRA. The experiment utilized a randomly constructed environment (Fig. 2) where vessel agents underwent meta-training and meta-testing, and the occurrences of ship collisions were recorded for statistical analysis.

Following the initial training phase with 20,000 episodes in Water Area A, the vessel agents will undergo 100 random tests in the same area. The collision frequency results are presented in Fig. 8(a). The SACMRL algorithm(ours) proposed in this research exhibits a lower collision rate compared to SACMRL/TS, indicating the effectiveness of the TSHR and SGMRA in improving the safety of the collision avoidance policy during the meta-training phase.



(a) SACMRL(ours) (b) MSCCADRL

(c) TRPO (d) MA2CL

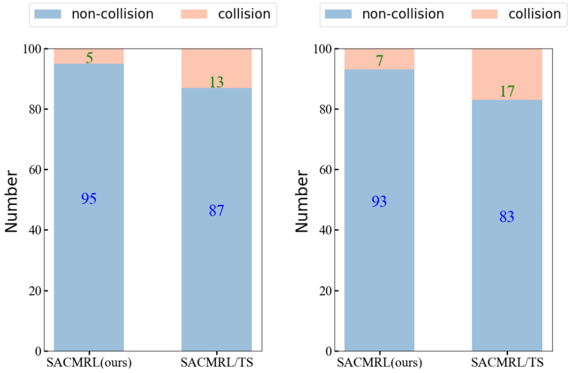
Fig. 7. Visualization of resulting path diagrams

During the meta-testing phase, the vessel agent applies the acquired policy training in Water Area A to complete 4000 training episodes in Water Area B. Subsequently, the vessel agent performs 100 random tests in Water Area B to assess the collision frequency(Fig. 8(b)). The results show that the SACMRL algorithm (ours) exhibits a lower collision frequency than SACMRL/TS, thus demonstrating the effectiveness of the TSHR and SGMRA in improving the safety of the collision avoidance policy in the testing environment. Some possible reasons are:

- The TSHR method trains vessel agents effectively by providing them with high-risk encounter avoidance task samples. This approach ensures that vessel agents are well-prepared to handle challenging scenarios with potentially high risks. Consequently, when vessel agents encounter such situations, they can devise and implement safer and more effective collision avoidance policies.
- With the risk-oriented evaluation objective function and SGMRA method, the collision avoidance policy of vessel agents can consider potential risks. This comprehensive evaluation reduces the likelihood of collisions, consequently enhancing overall safety.

5. Conclusion

Our work focuses on developing an autonomous collision avoidance decision-making model for ships using meta-reinforcement learning. We propose a two-layered recurrent model based on meta-learning. To effectively train the vessel agent, we utilize the cross-entropy method and introduce the TSHR method, explicitly addressing high-risk encounter scenarios sampled from the environment. For comprehensive risk assessment and policy refinement in collision avoidance, we employ the CVaR as the objective function for our meta-collision avoidance policy. Additionally, we introduce the SGMRA method to update and optimize our collision avoidance policies. We enhance the adaptability, effectiveness, and safety of collision avoidance policies in various encounter scenarios, providing valuable insights into intelligent ship navigation.



(a) Collision frequency results test during meta-training (b) Collision frequency results test during meta-testing

Fig. 8. Ablation experiment evaluation results diagram

Recognizing that ship collision avoidance is not only affected by other vessels, reefs, and islands but also by factors such as wind, waves, and ocean currents, future research should thoroughly consider the influencing factors in ship collision avoidance decision-making. Additionally, there is a need to investigate the integration of perception and prediction techniques into collision avoidance policy prediction methods. The utilization of advanced perception and prediction technologies can aid intelligent ships in creating precise dynamic models and state representations, thereby improving their understanding of the surrounding environment and the behavior of other vessels and enhancing the effectiveness and safety of collision avoidance policies.

References

[1] Gerrit Schoettler, Ashvin Nair, Juan Aparicio Ojea, Sergey Levine, and Eugen Solowjow. Meta-reinforcement learning for robotic industrial insertion tasks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9728–9735. IEEE, 2020.

[2] Haiqing Shen, Hirotada Hashimoto, Akihiko Matsuda, Yuuki Taniguchi, Daisuke Terada, and Chen Guo. Automatic collision avoidance of multiple ships based on deep q-learning. *Applied Ocean Research*, 86:268–288, 2019.

[3] Chen Chen, Xian-Qiao Chen, Feng Ma, Xiao-Jun Zeng, and Jin Wang. A knowledge-free path planning approach for smart ships based on reinforcement learning. *Ocean Engineering*, 189:106299, 2019.

[4] Xinli Xu, Peng Cai, Zahoor Ahmed, Vidya Sagar Yellapu, and Weidong Zhang. Path planning and dynamic collision avoidance algorithm under colregs via deep reinforcement learning. *Neurocomputing*, 468:181–197, 2022.

[5] Pengyu Zhai, Yingjun Zhang, and Wang Shaobo. Intelligent ship collision avoidance algorithm based on ddqn with prioritized experience replay under colregs. *Journal of Marine Science and Engineering*, 10(5):585, 2022.

[6] Lirong Sui, Shu Gao, and Wei He. Ship cooperative collision avoidance strategy based on multi-agent deep reinforcement learning. *Control and Decision*, 38(05):1395–1402, 2023.

[7] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

[8] Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: Variational bayes-adaptive deep rl via meta-learning. *The Journal of Machine Learning Research*, 22(1):13198–13236, 2021.

[9] Seyed Roozbeh Razavi Rohani, Saeed Hedayatian, and Mahdieh Soleymani Baghshah. Bimrl: Brain inspired meta reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9048–9053. IEEE, 2022.

[10] Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. Experience replay optimization. *arXiv preprint arXiv:1906.08387*, 2019.

[11] Zuxin Liu, Zhepeng Cen, Vladislav Isenbaev, Wei Liu, Steven Wu, Bo Li, and Ding Zhao. Constrained variational policy optimization for safe reinforcement learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 13644–13668. PMLR, 17–23 Jul 2022.

[12] Ido Greenberg, Shie Mannor, Gal Chechik, and Eli Meirom. Train hard, fight easy: Robust meta reinforcement learning. *arXiv preprint arXiv:2301.11147*, 2023.

[13] Fatemeh Rastgar, Houman Masnavi, Basant Sharma, Alvo Aabloo, Jan Swevers, and Arun Kumar Singh. Priest: Projection guided sampling-based optimization for autonomous navigation. *arXiv preprint arXiv:2309.08235*, 2023.

[14] Marcelo Paravisi, Davi H. Santos, Vitor Jorge, Guilherme Heck, Luiz Marcos Gonçalves, and Alexandre Amory. Unmanned surface vehicle simulator with realistic environmental disturbances. *Sensors*, 19(5):1068, 2019.

[15] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.

[16] Haolin Song, Mingxiao Feng, Wengang Zhou, and Houqiang Li. Ma2cl: Masked attentive contrastive learning for multi-agent reinforcement learning. *arXiv preprint arXiv:2306.02006*, 2023.

[17] Kefan Jin, Jian Wang, Hongdong Wang, Xiaofeng Liang, Yongjin Guo, Mianjin Wang, and Hong Yi. Soft formation control for unmanned surface vehicles under environmental disturbance using multi-task reinforcement learning. *Ocean Engineering*, 260:112035, 2022.

[18] Charles Packer, Pieter Abbeel, and Joseph E Gonzalez. Hindsight task relabelling: Experience replay for sparse reward meta-rl. *Advances in Neural Information Processing Systems*, 34:2466–2477, 2021.

[19] Alireza Fallah, Kristian Georgiev, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of debiased model-agnostic meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3096–3107, 2021.

[20] Yuemin Zheng, Zhongxin Liu, Jin Tao, Qinglin Sun, Hao Sun, Mingwei Sun, and Zengqiang Chen. Double deep q network optimized linear active disturbance rejection control for ship course keeping. In *Proceedings of 2021 Chinese Intelligent Systems Conference: Volume I*, pages 259–274. Springer, 2022.

[21] Liang Lihua, Zhao Peng, Zhang Songtao, and Yuan Jia. Simulation analysis of fin stabilizers on turning circle control during ship turns. *Ocean Engineering*, 173:174–182, 2019.

[22] Ye Hu, Mingzhe Chen, Walid Saad, H Vincent Poor, and Shuguang Cui. Distributed multi-agent meta learning for trajectory design in wireless drone networks. *IEEE Journal on Selected Areas in Communications*, 39(10):3177–3192, 2021.

[23] Suneel Belkhale, Rachel Li, Gregory Kahn, Rowan McAllister, Roberto Calandra, and Sergey Levine. Model-based meta-reinforcement learning for flight with suspended payloads. *IEEE Robotics and Automation Letters*, 6(2):1471–1478, 2021.

[24] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.

[25] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023.

[26] Yingjun Hu, Anmin Zhang, Wuliu Tian, Jinfen Zhang, and Zebei Hou. Multi-ship collision avoidance decision-making based on collision risk index. *Journal of Marine Science and Engineering*, 8(9):640, 2020.

[27] Wang Shaobo, Zhang Yingjun, and Li Lianbo. A collision avoidance decision-making system for autonomous ship based on modified velocity obstacle method. *Ocean Engineering*, 215:107910, 2020.

[28] Zhengyu Zhou, Yingjun Zhang, and Shaobo Wang. A coordination system between decision making and controlling for autonomous collision avoidance of large intelligent ships. *Journal of Marine Science and Engineering*, 9(11):1202, 2021.