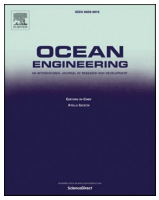
Ocean Engineering 309 (2024) 118416

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng



Research paper

Dynamic navigation: Integrating GL-STGCNN and MPC for collision avoidance with future Awareness

Weiqiang Liao [a,b], Yuegao Wu [a], Peilin Zhou [c], Haibin Wang [c,*], Wanneng Yu [a,b], Changkun Zhang [a], Chenghan Luo [a]

[a] School of Marine Engineering, Jimei University, Xiamen, 361021, China
[b] Marine Engineering College and Key Laboratory of Fujian province Marine and Ocean Engineering, Jimei University, Xiamen, 361021, China
[c] Department of Naval Architecture, Ocean and Marine Engineering, University of Strathclyde, Glasgow, G4 0LZ, UK

ARTICLE INFO

Keywords:
Trajectory prediction
Model predictive control
Dynamic collision avoidance
Multi-ship interaction

ABSTRACT

Existing ship dynamic collision avoidance methods mostly rely on the instantaneous motion information of surrounding ships to make decisions. This makes it difficult to adapt to changes in the motion states of surrounding ships, which may lead to collisions between ships. To improve the safety of dynamic collision avoidance methods, this paper combines the multi-ship trajectory prediction model GL-STGCNN with model predictive control for ship dynamic collision avoidance tasks. Firstly, the interaction between ships is extracted through GL-STGCNN to predict the future trajectories of surrounding ships. Then, the objective function based on the artificial potential field method and the velocity obstacle method is optimized to control the ship to complete the dynamic collision avoidance task. The performance of the dynamic collision avoidance method is verified and analyzed in the ship navigation scenario simulated by AIS data. The experiments show that the new ship dynamic collision avoidance method not only complies with the COLREGs, but also can flexibly select the collision avoidance method according to different scenarios. In addition, the theoretical collision avoidance threshold distance based on the MPC objective function shows a high degree of fit with the actual collision avoidance trigger distance observed in the simulation verification.

1. Introduction

Ship collisions during navigation pose a serious threat in maritime accidents (Ugurlu and Cicek, 2022). Analysis of numerous investigation reports on ship collision incidents indicates that approximately 80% of these accidents are attributable to human errors (Kim and Na, 2017). With the gradual improvement of electronic sensing devices on ships, extracting interactions among ships and predicting the future positions of surrounding ships have become possible. Providing rational dynamic collision avoidance guidance for ships during navigation is crucial for enhancing safety, efficiency, and overall maritime navigation.

To guide the behavior of ships and reduce the risk of collisions encounters, the International Maritime Organization established COLREGS in 1972 (Demirel and Bayer, 2015). Faced with complex and dynamic navigation scenarios, relying solely on the rigidly defined navigation rules in COLREGS may not enable ships to make optimal decisions. To address the limitations in COLREGS and enhance adaptability and flexibility in complex situations, more intelligent algorithms have been applied to ship collision avoidance tasks (Burmeister and Constapel, 2021). These algorithms operate by perceiving and processing instantaneous motion information from surrounding ships. By incorporating processed data and employing suitable optimization algorithms, they achieve a balance between global planning and local collision avoidance. Some of these algorithms include the A-star algorithm (Singh et al., 2018; Wang et al., 2021; He et al., 2022), the Rapidly Exploring Random Tree algorithm (Enevoldsen et al., 2021; Zhang et al., 2022a), the Particle Swarm Optimization algorithm; (Lu et al., 2022) and the Reinforcement Learning (Chun et al., 2021; Yang and Han, 2023; Guan et al., 2023). However, when considering long-distance collision avoidance to prevent falling into local optima, these algorithms may have slow computational speeds, posing challenges for real-time ship collision avoidance tasks.

Simultaneously, the ability of a ship to navigate according to the planned route is crucial for collision avoidance tasks. Starting from the principles of ship kinematics, (Hagen et al., 2018; Du et al., 2021; Lindqvist et al., 2020) establish a nonlinear dynamic model for ships using model predictive control, enhancing the accuracy and stability of

* Corresponding author.
E-mail address: haibin.wang@strath.ac.uk (H. Wang).

https://doi.org/10.1016/j.oceaneng.2024.118416
Received 3 April 2024; Received in revised form 27 May 2024; Accepted 5 June 2024

List of abbreviation

GL-STGCNN	Graph Learning-Spatiotemporal Graph Convolutional Neural Network
MPC	Model Predictive Control
APF	Artificial Potential Field
VO	Velocity Obstacle
AIS	Automatic Identification System
COLREGs	International Regulations For Preventing Collisions At Sea
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory Network
RNN	Recurrent Neural Network
GNN	Graph Neural Network
STGCNN	Spatio Temporal Graph Convolutional Neural Network
TXPCNN	Temporal Xceptional Parallel Convolutional Neural Network
MMG	Mathematical Model Group
ADE	Average Displacement Error
FDE	Final Displacement Error

collision avoidance planning. Real-time collision avoidance decisions are made during navigation based on instantaneous environmental and sensor information. However, under actual navigation conditions, inferring the future positions of surrounding ships solely from momentary motion information may lead to suboptimal planning or situations where collision avoidance is impossible.

To address the limitation of relying solely on the instantaneous motion information of surrounding ships in dynamic ship collision avoidance tasks without considering changes in the motion states of these ships, using ship trajectory prediction models to predict the trajectories of surrounding ships is a viable approach. Different trajectory prediction algorithms have various advantages and application scenarios. RNN performs well in handling individual time-series trajectory data and is often applied to traffic flow prediction, pedestrian path prediction, and vehicle speed prediction (Alahi et al., 2016; Cheng et al., 2018; Davis et al., 2020; Lu et al., 2020). However, RNN faces challenges in effectively capturing complex spatial dependencies in trajectory data. GNN can flexibly represent attributes of multiple targets and relationships between targets, having been applied in various tasks such as predicting traffic flow (Zhang et al., 2019; Wang et al., 2020; Wang et al., 2022; Li et al., 2023), predicting chemical molecule properties (Wieder et al., 2020), predicting pedestrian trajectories (Mohamed et al., 2020) and predicting ship trajectories (Feng et al., 2022). However, in dynamic environments, modeling the interaction adjacency matrix representing relationships between ships can be challenging, and there will always be some error between the predicted trajectories and the actual ship trajectories (Prabhakar and Rahimi, 2019).

When obtaining the future positions of ships through trajectory prediction, it is common not to consider sudden situations that may arise at sea. To allow for more time and space to react in emergency scenarios, ships need to maintain a certain reaction distance between them (Fan et al., 2020). The APF method can assign attractive forces to targets and repulsive forces to obstacles, ensuring a certain safety distance between vessels (Iswanto et al., 2019; Bounini et al., 2017). However, in complex environments with multiple dynamic vessels, the APF method may fall into a local optimum, and it cannot guarantee the vessels' heading (Zhang et al., 2022b). To address this, appropriate configurations need to be made based on collision avoidance tasks and environmental characteristics.

In addressing the issues associated with existing methods in the current stage, this paper focuses on ship collision avoidance planning tasks. Initially, the GL-STGCNN ship trajectory prediction model is employed to obtain the future positions of surrounding ships. Subsequently, MPC is introduced into the ship collision avoidance task. Drawing inspiration from the APF method and the VO method, the predicted future positions of nearby ships are incorporated into the objective function of the MPC. By considering the gradients of destination objective functions and collision avoidance objective functions at different distances, and employing an optimization method based on gradient descent, the ship collision avoidance process automatically achieves motion planning behaviors in three stages. Simultaneously, the planned path ensures the navigability of ships.

The contributions of the paper are as follows.

- Proposed the GL-STGCNN multi-ship trajectory prediction model, introducing the ship interaction adjacency matrix extraction module and trajectory correction module. This addresses the issue of using only the distance between ships to extract the interaction adjacency matrix and the probability distribution of predicted trajectories, which cannot be directly applied to dynamic collision avoidance.
- Established a mathematical model of ship motion considering environmental conditions, incorporating the future trajectories of surrounding ships into the MPC objective function. Through the principles of APF and VO collision avoidance, optimized the collision avoidance behavior without the need for cumbersome decision condition checks, achieving three-stage motion planning tasks.
- Validated the effectiveness of the ship collision avoidance algorithm through AIS data simulation of multiple ships navigating in a shared maritime area. This resolves the problem of previous algorithms being validated in relatively simple scenarios without complex navigation situations.

The remaining parts of the paper are as follows: In Section 2, the paper introduces the workflow and overall structure of dynamic ship collision avoidance. Section 3 introduces the design of the MPC objective function for dynamic collision avoidance. Section 4 provides a quantitative and qualitative analysis of the ship trajectory prediction model. Section 5 analyzes the performance of the dynamic ship collision avoidance model quantitatively and qualitatively. Section 6 concludes the paper.

2. Ship dynamic collision avoidance model

The dynamic ship collision avoidance model in Fig. 1 consists of two main modules: the GL-STGCNN multi-ship trajectory prediction module and the MPC motion planning module. In a scenario involving the interaction of multiple ships, the process begins with preprocessing and the ship interaction adjacency matrix extraction module in GL-STGCNN. This module converts historical trajectories into a node matrix representing ship attributes and an adjacency matrix representing the interaction strength between ships. Subsequently, the probability distribution of the predicted trajectory is obtained through spatiotemporal graph convolution and temporal extrapolation convolution.

As probability distribution cannot be directly applied to dynamic collision avoidance tasks, the corrected predicted trajectory is obtained through the MPC trajectory correction. The corrected predicted trajectory is then fed into the MPC motion planning module, where, considering the future trajectories of surrounding ships, environmental parameters, and the current motion state of the case ship, the MPC control variables are optimized and updated using gradient descent. The control variables are passed to the ship maneuvering MMG to control the case ship's next actions within the next t seconds, continuously updating the current state of the case ship to achieve dynamic ship collision avoidance functionality.

The combination of GL-STGCNN ship trajectory prediction with MPC allows the proposed dynamic ship collision avoidance method in this paper to consider both the rapid changes in ship motion within a short time and to perform long-term ship dynamic collision avoidance

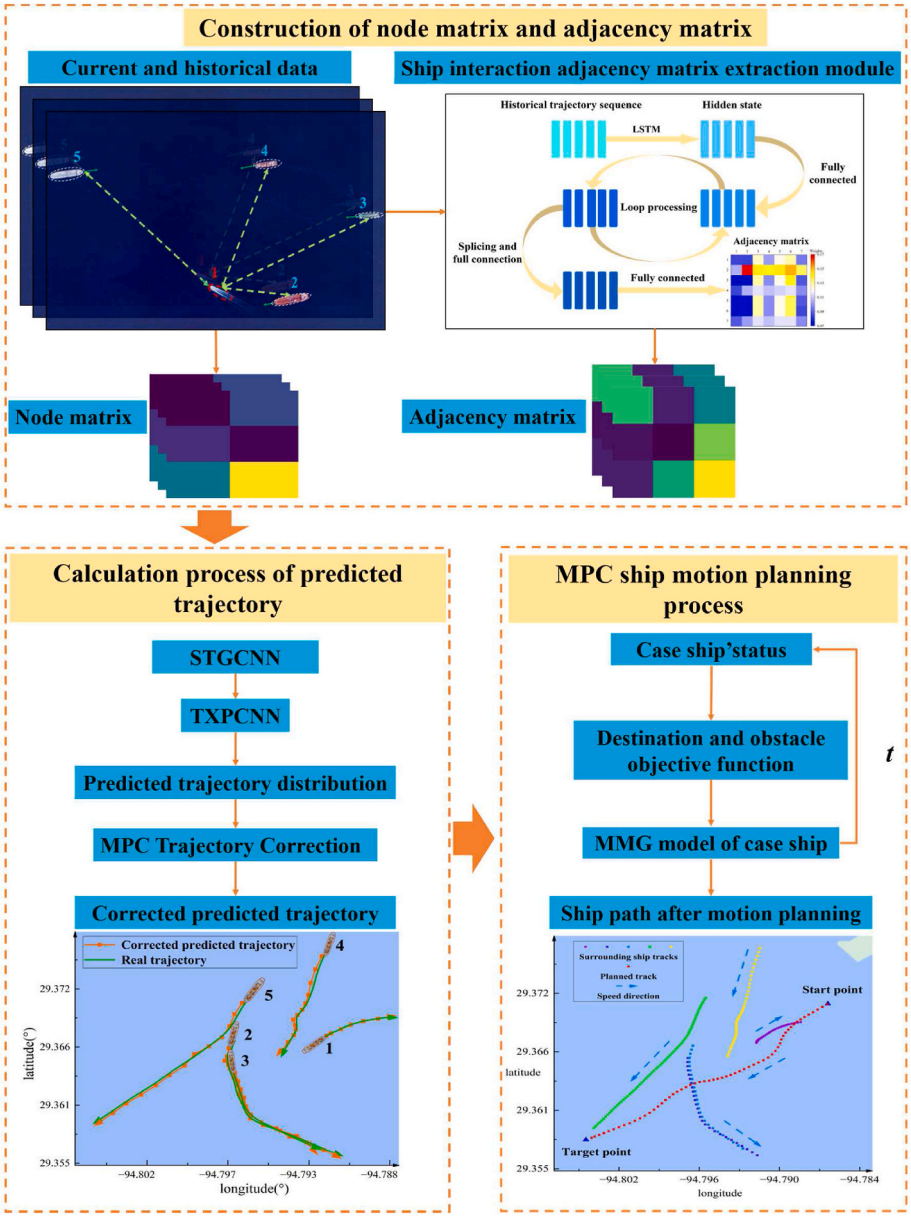


Fig. 1. Ship dynamic collision avoidance process.

systematically. This ensures the safety of ship navigation.

2.1. Graph convolution calculation

Graph convolution is one of the core operations in graph neural networks, widely used for processing data with a graph structure. The purpose of graph convolution computation is to leverage the information of nodes and edges in the graph structure. By performing calculations on the node matrix and adjacency matrix, it extracts feature representations for each node.

2.1.1. Node matrix

When creating the node matrix, first select a time window of length (T) to slice and obtain the time series data. For each ship within each time window, extract its features at each moment and combine these features into a single node feature vector. The node feature vector includes position, speed, and heading angle. Therefore, each ship node will have T time steps. The feature vector for each time step is in Eq (1).

$$\mathrm{v}_t = [x_t, y_t, v_t, \alpha_t] \tag{1}$$

In Eq (1), (x,y) represents the ship's position, (v) represents the velocity, and (α) represents the heading angle.

Stacking the feature vectors of T time steps together will form the node matrix representing a single ship, and The node matrix of a single ship is shown in Eq (2).

$$\mathrm{V} = [\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_t] \tag{2}$$

Since deep learning requires fixed-size inputs and outputs, the node matrix constructed in this paper needs to accommodate a variable number of ships. When the number of ships is small, missing values must be supplemented to meet the fixed size of the node matrix. Therefore, after integrating the temporal feature information of all ships, the resulting complete node matrix will be used as input to the graph neural network. The complete node matrix is shown in Eq (3).

$$V = \left[\mathrm{V}^1, \mathrm{V}^2, \ldots, \mathrm{V}^N\right] \tag{3}$$

2.1.2. Matrix calculation

When using graph neural networks for computation, the information in the scenario can be be represented as the structure of a graph $G = (V, A)$, Here, V is the node matrix representing ship positional motion information, and A is the adjacency matrix representing the interaction intensity between ships. Based on the shape of the adjacency matrix, a directed adjacency matrix is a non-symmetric matrix, and an undirected adjacency matrix is a symmetric matrix. A trajectory of T time steps length is processed using the Ship Interaction Intensity Extraction Module to obtain the adjacency matrix (A) between nodes. The calculation of the node matrix and the adjacency matrix is as follows Eq (4).

$$\widetilde{V} = \sigma(VAW) \tag{4}$$

2.1.3 In Eq (4), $\widetilde{V}$ is the node matrix output after calculation. σ represents the activation function. W is a training matrix, which is related to the number of nodes. By performing multiple graph convolutions, the node matrix is repeatedly processed with the adjacency matrix, capturing the complex spatial relationships between nodes and extracting feature information.STGCNN

The time graph convolution module, as a part of the STGCNN, performs convolution in the time dimension to extract patterns and features between different time steps. Sharing parameters across different time steps helps capture patterns, features, and structures in sequential data, enabling the model to better understand and process time series data. To perform the specific operation, node matrix and adjacency matrix information about the graph are required. The node matrix is represented as V, with a shape of (C, H, W), where C represents the time steps, H represents the number of nodes, and W represents the node attributes. During convolution in the time dimension, the channels of the node matrix become the number of nodes H, and operations are performed on a two-dimensional plane of shape (C, W).

2.1.3. TXPCNN

The time extrapolation convolution module (TXPCNN) operates in the spatial dimension, performing convolution on the input in each channel. By setting the number of channels, it operates on the predicted time steps. For the node matrix V, with a shape of (C, H, W), where C represents the time steps, H represents the number of nodes, and W represents the node attributes. During convolution in the spatial dimension, operations are performed on a two-dimensional plane of shape (H, W).

2.2. Ship interaction adjacency matrix extraction module

Fig. 2 illustrates the detailed workflow of the ship interaction adjacency matrix extraction module proposed in this paper. The MLP (Multi-Layer Perceptron) is a feedforward neural network consisting of multiple fully connected layers. In this paper, the ReLU activation function is used.

Node2Edge and Edge2Node correspond to two different matrices after one-hot encoding. Node2Edge represents the relationships between nodes and edges in the graph neural network, while Edge2Node represents the relationships between edges and nodes in the graph neural network. This facilitates message passing and feature updating within the graph neural network, helping to model relationships and update features between nodes.

In this paper, Concat refers to the operation of concatenating tensors along a specific dimension. This is used to integrate feature information obtained from different network layers in the model, enhancing the model's expressive power and performance.

When performing the operations shown in Fig. 2, for Node2Edge and Edge2Node, we process the output of the previous layer with the one-hot encoded matrices through matrix multiplication operations. Additionally, in terms of iterative selection, this paper conducts a total of two loop.

2.3. Ship prediction trajectory correction MPC model

2.3.1. Sampling process

The predicted trajectories based on the deep learning methods are typically discrete and localized points rather than continuous and smooth paths. In such cases, the predicted trajectories may violate the principles of ship motion dynamics and cannot be directly applied to dynamic ship collision avoidance tasks. Therefore, it is necessary to enhance the rationality of the predicted trajectories through trajectory correction.

As shown in Fig. 3, the basic idea of trajectory correction is as follows: first, sample a series of trajectory points from the probability distribution of predicted trajectories. Then, use model predictive control to track the sampled points, achieving the effect of trajectory correction. Finally, obtain the corrected trajectory.

For the probability distribution of predicting trajectories, the coordinate information (x, y) is composed of the corresponding mean μ and variance σ^2. First, two random numbers U_1 and U_2 are randomly obtained on the interval [0, 1]. Then, the Box-Muller transform is used to

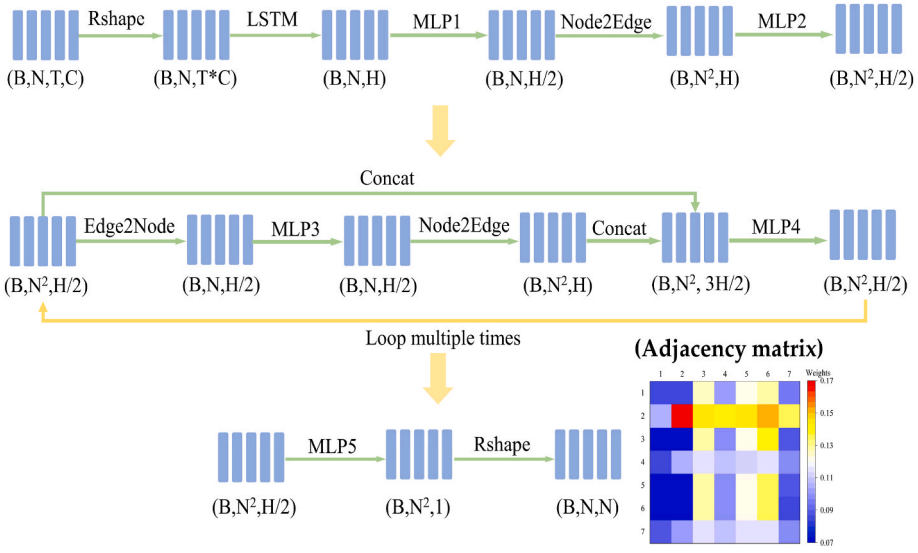


Fig. 2. Ship interaction adjacency matrix extraction module.

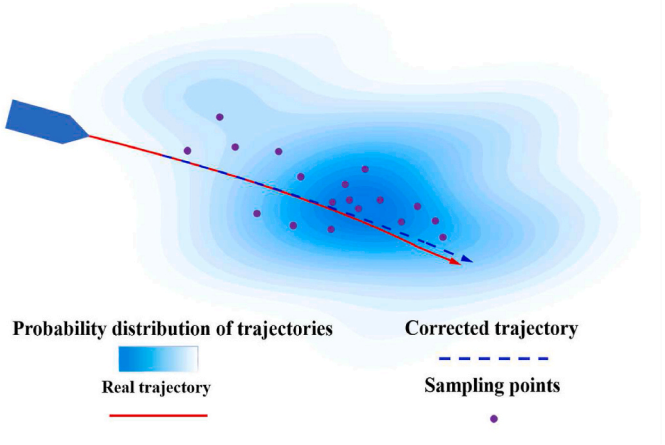


Fig. 3. Trajectory correction schematic diagram.

convert the uniformly distributed random numbers into normally distributed random numbers (Nelson and Thistleton, 2021).

$$Z_0 = \sqrt{-2\ln(U_1)}\cos(2\pi U_2) \tag{5}$$

In Eq (5), Z_0 is a sampled value from the standard normal distribution.

$$X = \mu + Z_0 \cdot \sigma \tag{6}$$

In Eq (6), X is a numerical value obtained by sampling from a distribution with mean μ and variance σ^2.

Through the above steps, it is possible to sample deterministic values from the mean and variance of a trajectory distribution. Moreover, the above computational process can be conveniently implemented using the deep learning framework PyTorch.

2.3.2. Motion model

Establishing a comprehensive ship kinematics model for each individual ship poses significant challenges. As shown in Fig. 4, when constructing a kinematics model used to correct ship trajectories, only the current position and velocity of the ship are considered.

$$\xi(t) = \begin{bmatrix} x \\ y \end{bmatrix}$$
$$u(t) = \begin{bmatrix} v \\ \alpha \end{bmatrix} \tag{7}$$

Eq (7) neglects ship parameters and performance indicators. The established ship kinematics model only needs to consider the ship's case

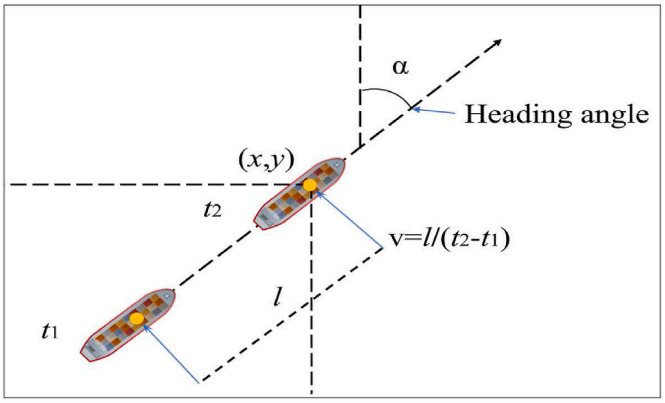


Fig. 4. Ship motion model diagram.

position (x,y), velocity (v), and heading angle (α). ξ and u represent functions of the state variables and control variables, respectively.

2.3.3. Linearization

$$\xi' = f(\xi, u) \tag{8}$$

$$\begin{cases} f(x) = \dfrac{f(x_0)}{0!} + \dfrac{f'(x_0)}{1!}(x - x_0) + \ldots + \dfrac{f^{(n)}(x_0)}{n!}(x - x_0)^n + R_n(x) \\ \xi' = f(\xi_r, u_r) + \dfrac{\partial f}{\partial \xi}(\xi - \xi_r) + \dfrac{\partial f}{\partial u}(u - u_r) \end{cases} \tag{9}$$

In Eq (8), the derivative ξ' of the state variable ξ is a nonlinear function that cannot be directly represented and requires linearization processing. The Taylor expansion is commonly employed to linearize nonlinear systems, thereby modeling system dynamics as linear systems. Specifically, we need to transform the error between the current system state and the desired state into control variables for optimization. Eq (9) represents the fundamental formula of Taylor expansion, where $f(x)$ is the function being expanded, and the derivative of ξ is approximated using the first-order Taylor expansion. $R_n(x)$ denotes the error term in the final approximation (Kouvaritakis and Cannon, 2016).

$$\begin{cases} \widetilde{\xi} = \xi - \xi_r \\ \widetilde{u} = u - u_r \end{cases} \tag{10}$$

$$\widetilde{\xi}' = \frac{\partial f}{\partial \widetilde{\xi}}\widetilde{\xi} + \frac{\partial f}{\partial \widetilde{u}}\widetilde{u} = \mathrm{A}\widetilde{\xi} + \mathrm{B}\widetilde{u} \tag{11}$$

The result in Equation (10) represents the error between the current state variables ξ and control variables u, and the reference state variables ξ_r and control variables u_r. The derivative of the error $\widetilde{\xi}'$ is shown in Eq (11).

$$\mathrm{A} = \begin{bmatrix} \dfrac{\partial f_1}{\partial \xi_1} & \dfrac{\partial f_1}{\partial \xi_2} \\ \dfrac{\partial f_2}{\partial \xi_1} & \dfrac{\partial f_2}{\partial \xi_2} \end{bmatrix}$$
$$\mathrm{B} = \begin{bmatrix} \dfrac{\partial f_1}{\partial u_1} & \dfrac{\partial f_1}{\partial u_2} \\ \dfrac{\partial f_2}{\partial u_1} & \dfrac{\partial f_2}{\partial u_2} \end{bmatrix} \tag{12}$$

The calculation for A and B results in Eq (12).

2.3.4. Discretization

The linearization process described above needs to be further discretized, transforming the continuous-time model into a discrete-time model, which facilitates control.

$$\begin{cases} \widetilde{\xi}' = \dfrac{\widetilde{\xi}(\mathrm{k}+1) - \widetilde{\xi}(\mathrm{k})}{\mathrm{T}} = \mathrm{A}\widetilde{\xi}(\mathrm{k}) + \mathrm{B}\widetilde{u}(\mathrm{k}) \\ \widetilde{\xi}(\mathrm{k}+1) = \widetilde{\mathrm{A}}\widetilde{\xi}(\mathrm{k}) + \widetilde{\mathrm{B}}\widetilde{u}(\mathrm{k}) \end{cases} \tag{13}$$

$$\widetilde{\mathrm{A}} = \begin{bmatrix} \mathrm{T}\dfrac{\partial f_1}{\partial \xi_1} + 1 & \mathrm{T}\dfrac{\partial f_1}{\partial \xi_2} + 1 \\ \mathrm{T}\dfrac{\partial f_2}{\partial \xi_1} + 1 & \mathrm{T}\dfrac{\partial f_2}{\partial \xi_2} + 1 \end{bmatrix}$$
$$\widetilde{\mathrm{B}} = \begin{bmatrix} \mathrm{T}\dfrac{\partial f_1}{\partial u_1} & \mathrm{T}\dfrac{\partial f_1}{\partial u_2} \\ \mathrm{T}\dfrac{\partial f_2}{\partial u_1} & \mathrm{T}\dfrac{\partial f_2}{\partial u_2} \end{bmatrix} \tag{14}$$

In Eq (13) and Eq (14), T denotes the time interval, and the time interval

needs to be reasonably adjusted based on conditions to achieve the best results. Since $\widetilde{A}$ consists of constants, its numerical values cannot be changed in this paper. The control process is completed solely by adjusting the parameters in $\widetilde{B}$ to minimize the error between the current point and the reference point.

2.3.5. Constraints

As shown in Eq (15), during the trajectory correction process, it is necessary to impose some constraints on the model, such as constraints on the ship's speed, heading angle, etc., to address the issue of unrealistic maneuvers in predicting ship trajectories.

$$\begin{cases} \alpha_{\min} \le \alpha_k \le \alpha_{\max} \\ -\Delta\alpha \le \alpha_{k+1} - \alpha_k \le \Delta\alpha \\ -\Delta v \le v_{k+1} - v_k \le \Delta v \end{cases} \tag{15}$$

Considering that ships have significant inertia while navigating, it is difficult to change speed in a short time. Therefore, the selection of constraint conditions needs to be done carefully. For the heading angle (α), its range is 2π. It is important to note that in Eq (15), we did not set a range limit for the speed. This is primarily because, with only AIS data, we cannot determine the speed limits of the ship, and adding speed limit constraints might introduce unnecessary complications. In this paper, the speed control variable is constrained through the rate-of-change constraints on speed. For the heading angle, we constrain it by setting the range and rate of change of the heading angle.

The rate of change for the heading angle ($\Delta\alpha$) is set to 1.5%π per second, and the rate of change for speed (Δv) is set to 1% of the current speed per second.

2.4. Own ship dynamic collision avoidance MMG model

2.4.1. Ship motion analysis

In Fig. 5, the analysis of forces acting on the ship is conducted with the ship's center of mass as the origin. Regarding angles, the north direction is considered as 0°, with angles increasing clockwise. F_1 represents the thrust from the ship's propeller. V_1, V_2, and V_3 denote the absolute velocities of the ship, wind, and water flow, respectively. θ_1, θ_2, θ_3, and θ_4 represent the heading angle, wind direction, water flow direction, and rudder angle, respectively. Wind and water flow, changing with the ship's heading angle, exert forces on the ship that can either impede or facilitate its forward movement. T represents the torque caused by various forces acting on the ship.

Table 1 presents the fundamental parameters of the ship, which are used to establish a comprehensive mathematical model of ship motion.

In Eq (16), F_1 represents the propeller thrust, and F_2, F_3 are the forces exerted by wind and water flow on the ship considering the ship's speed.

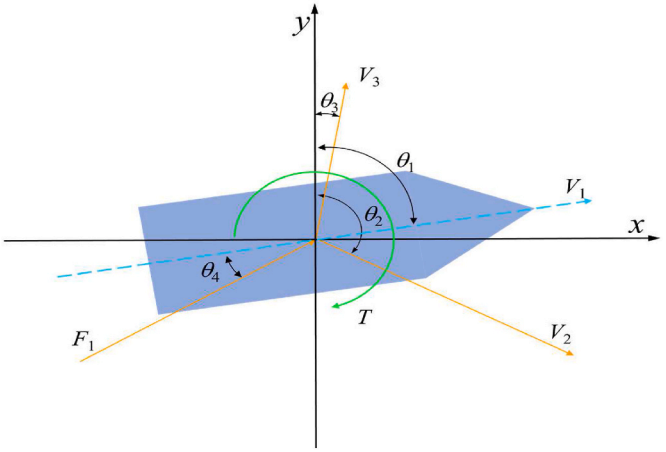


Fig. 5. Ship force analysis.

Table 1
Parameters of case ship.

Ship parameters	Parameter value
Total length	35m
Full load displacement	398t
Maximum draft	3.1m
Height from water surface	2.0m
Turbine power	441 kW
Turbine rated speed	400RPM
Rudder angle	−35°~ +35°
Full load speed	12.8kn

t_p is the thrust deduction factor of the propeller, ρ_2 is the density of seawater, ρ_3 is the density of air, D is the diameter of the propeller, a_0, a_1, a_2 are coefficients, w_p is the wake fraction, N is the propeller rotation speed. C_2 and C_3 represent the resistance coefficients of the ship in air and seawater, and S_2, S_3 represent the cross-sectional areas in air and seawater (Xiao et al., 2016).

$$\begin{cases} F_1 = (1 - t_p)F_T \\ F_T = \rho_2 N^2 D^4 K_T(J_P) \\ K_T(J_P) = a_0 + a_1 J_P + a_2 J_P^2 \\ J_P = \dfrac{(1 - w_p)V_1}{ND} \\ F_2 = \dfrac{1}{2}\rho_2 C_2 S_2 \left(V_1^2 + V_2^2 + 2|V_1||V_2|\cos(\theta_2 - \theta_1)\right)^2 \\ F_3 = \dfrac{1}{2}\rho_3 C_3 S_3 \left(V_1^2 + V_3^2 + 2|V_1||V_3|\cos(\theta_3 - \theta_1)\right)^2 \end{cases} \tag{16}$$

As shown in Eqs (17) and (18), adopting the idea of MMG separation modeling, the ship's motion is influenced by the thrust of the propeller, wind resistance, and water resistance. α_w is the angular acceleration, L is the ship length, I is the total moment of inertia of the ship, α_x is the acceleration of the ship in the x-axis direction, α_w is the acceleration of the ship in the y-axis direction, m is the mass of the ship.

$$\begin{cases} \alpha_w = \dfrac{F_1 \sin(\theta_4) L}{I} \\ \alpha_x = \dfrac{F_1 \cos(\theta_4) + F_2 \cos(\theta_4) + F_3 \cos(\theta_3)}{m} \\ \alpha_y = \dfrac{F_1 \sin(\theta_4) + F_2 \sin(\theta_4) + F_3 \sin(\theta_3)}{m} \end{cases} \tag{17}$$

$$\begin{cases} \dot{X} = V_1 \sin(\theta_4) + \alpha_x t \\ \dot{Y} = V_1 \cos(\theta_4) + \alpha_y t \\ \dot{\theta}_4 = \omega_4 + \alpha_w t \end{cases} \tag{18}$$

The ship's motion typically involves six degrees of freedom, including movement along the x, y, z axes, and rotation. Ship trajectory planning usually occurs in the plane of the x and y axes, without involving the z-axis direction. Therefore, the mathematical model of the ship's motion is represented as a three-degree-of-freedom model, considering the forward and backward movement along the x and y axes and the heading offset along the forward direction.

3. Design of dynamic collision avoidance MPC objective function

3.1. Expectation of MPC objective function

The maritime environment is dynamic, and the motion states of other ships may change at any time. Conducting collision avoidance behavior in stages allows better adaptation to the dynamic environment, flexibly choosing appropriate actions based on different distance scenarios.

As shown in Table 2, this paper divides the dynamic collision avoidance of ships into three stages and takes different actions according to the distance between the case ship and other ships. When the distance is far, the case ship takes going to the destination as the primary objective. When the distance is moderate, the case ship's navigation will consider both collision avoidance and heading for destination. When the distance between ships continues to approach, the case ship will take safe collision avoidance as the top priority.

Overall, this staged collision avoidance method aims to balance the ship's heading to the target point and collision avoidance requirements, ensuring appropriate actions are taken in different situations to improve navigation safety and efficiency. This paper achieves the above three tasks through MPC control of the ship, and balances the ship's execution of different operations in different situations by setting reasonable target point objective function and obstacle avoidance objective function.

3.2. Objective function analysis of MPC

$$f_{objective} = \sum_{n=1}^{N}\sum_{t=1}^{T} e^{-\frac{t}{T}} f_l\left(\sqrt{(x_t-\bar{x}_{nt})^2+(y_t-\bar{y}_{nt})^2}\right) + \sum_{t=1}^{T}(-a\cos(\theta_t)+b)l_t \tag{19}$$

Eq (19) is the complete objective function used for dynamic collision avoidance tasks. The first term is used to keep away from surrounding obstacle ships during the MPC optimization process, and the second term is used to steer the own ship towards its target point during the optimization process. In the following subsections, the components of this complete objective function will be explained in detail.

3.2.1. Target point objective function

In Fig. 6, the distance between the ship and the target point is denoted as l. θ represents the angle between the ship's velocity direction and the angle relative to the target point. Based on the concept of VO collision avoidance, both θ and l are incorporated into the objective function for controlling the ship's arrival at the target point. During the optimization process, the goal is to continuously minimize the value of the objective function. The overall function is represented by Eq (20).

$$f_{t\ \arg\ et} = (-a\cos(\theta)+b)l \tag{20}$$

Eq (20) consists of two components, each responsible for controlling the ship's approach to the target point and controlling the ship's orientation towards the target point. Parameters a and b regulate the strength of maintaining the heading. In this paper, to make the values generated by different objective functions have the same order of magnitude, the parameters a and b are set to 1/2 and 5/2, respectively. Under these parameter settings, the range of Eq (20) is [2, 3], ensuring that the strength of maintaining the heading is higher than not maintaining the heading. With such parameter values, the overall function's gradient will not exhibit significant changes, allowing for the control of the ship's heading towards the target point.

In Fig. 7, the visual representation allows for a more intuitive observation of the descending trend of a function, moving from the highest function values in the red region to the blue region. From the contour lines, it can be observed that, at the same level of l, smaller angles result in lower function values.

Table 2
The behavior of ships at different stages of navigation.

Navigation stage	Distance between ships	Primary objective
Stage one	Far	Destination
Stage two	Moderate	Collision avoidance and destination
Stage three	Close	Collision avoidance

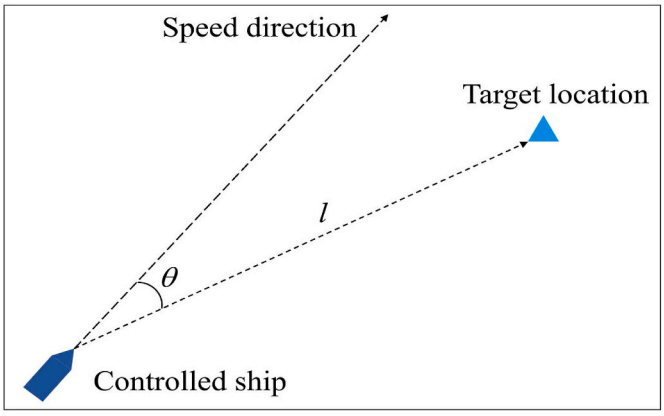


Fig. 6. Control ship to reach target point.

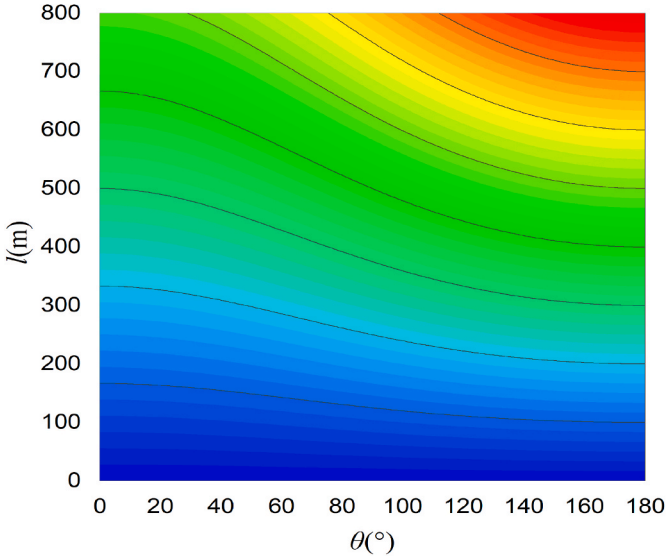


Fig. 7. Visualization of the destination objective function.

3.2.2. Obstacle avoidance objective function

When avoiding surrounding ships, we only use the predicted trajectories of the surrounding ships for a short period into the future to accomplish the avoidance task. Considering that the accuracy of trajectory prediction decreases over time. From a safety perspective in collision avoidance, it is necessary to incorporate a time-decaying function Eq (21) when performing collision avoidance calculations. This function aims to correct the continuously increasing prediction trajectory errors over time. Based on the concept of APF method, surrounding ships should have different degrees of influence on the own ship at different distances. Therefore, Eq (22) uses an exponential function to calculate the influence value of the surrounding ships. Additionally, to utilize the predicted trajectory, at each optimization step of the MPC output, the generated coordinates are extrapolated to obtain several coordinates matching the predicted trajectory data format. Using both extrapolated and predicted data, the spatiotemporal distance loss value is computed. This step helps assess whether the parameters optimized in each iteration provide the optimal overall collision avoidance loss. The expression for calculating the overall spatiotemporal distance loss value in scenarios with different numbers of ships is given by Eq (23).

$$f_t = e^{-\frac{t}{T}} \tag{21}$$

$$f_l = c^{l-\text{distance}} \tag{22}$$

$$f_{obstacle} = \sum_{n=1}^{N}\sum_{t=1}^{T} f_t f_l \left(\sqrt{(x_t - \bar{x}_{nt})^2 + (y_t - \bar{y}_{nt})^2}\right) \tag{23}$$

In Eqs (21)–(23), x and y represent the coordinates of the own ship's position, $\bar{x}$ and $\bar{y}$ represent the predicted coordinate position of the surrounding ships, t represents the time associated with each path point, T represents the total time for each optimization, l represents the distance between ships, N is used to denote different ships, distance represents the location where the gradient of the exponential function begins, c is the base of the exponential function.

The collision avoidance objective function based on the APF method is visualized in Fig. 8. When there are multiple surrounding ships in the scenario, the repulsive force on the case ship is illustrated in Fig. 9. As the distance between the case ship and surrounding ships decreases to a certain value, the collision avoidance objective function will exponentially increase, manifested as an increase in the objective function in model predictive control, thereby achieving collision avoidance. To prevent the function value from exponentially increasing and causing calculation errors, it is necessary to set reasonable values for the parameters c and distance in Eq (22), where these parameters control the distance and intensity at which collision avoidance begins. In this paper, c is set to 200/203, and distance is set to 500.

3.3. The theoretical calculation of obstacle avoidance distance

When calculating the distance at which the ship begins the third phase to avoid surrounding ships as the main task, considering the optimization process of MPC, which aims to find the direction in which the objective function descends most rapidly, from a gradient perspective, the objective function for controlling the ship to reach the target point can be simplified to a linear function, where the gradient value is a constant at this point. Combining this with the objective function used during collision avoidance, the derivative of the function can be expressed, as shown in Eq (24).

$$\begin{cases} c^{l-\text{distance}} \ln^c = -1 \\ l = \log_c^{-\frac{1}{\ln^c}} + \text{distance} \\ f_l' = c^{l-\text{distance}} \ln^c \end{cases} \tag{24}$$

By substituting the parameters c and distance into Eq (24), the theoretical distance to start the third phase is approximately 208 m. Meanwhile, since the distance is set to 500, the theoretical distance to start the second phase should be less than 500 m.

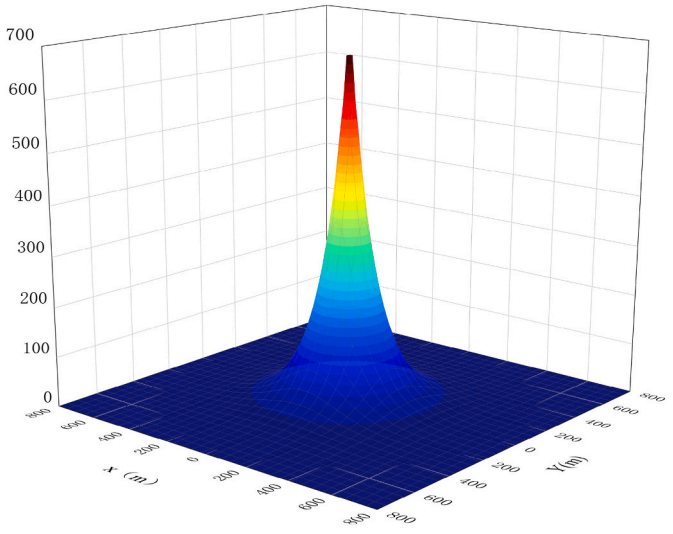


Fig. 8. Visualization of the Artificial potential field.

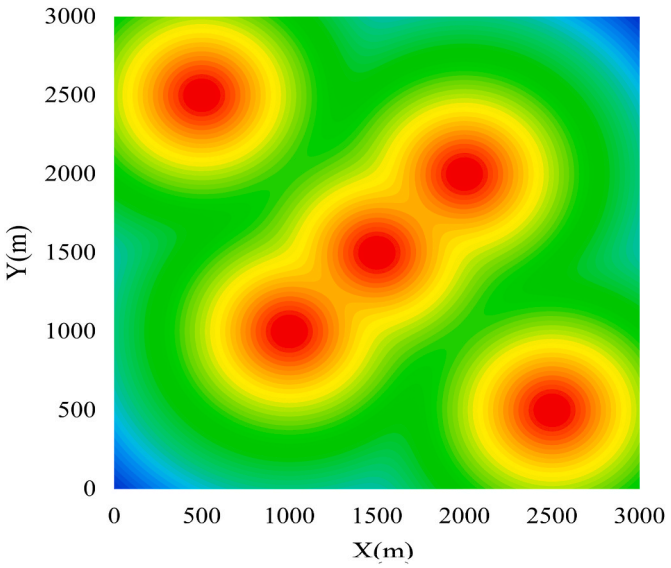


Fig. 9. Artificial potential field distribution of multiple ships.

It is worth noting that the threshold distances of 500 and 208 m obtained through calculation should not be regarded as the minimum distance between ships during the collision avoidance process. Furthermore, due to the presence of a time-decaying function in the obstacle avoidance objective function, we simplified the calculation method when determining the balance point between the target point objective function and the obstacle avoidance objective function gradient. Therefore, the values obtained are estimations.

3.4. The constraints of dynamic collision avoidance MPC

$$\begin{cases} -\Delta N \le N_{k+1} - N_k \le \Delta N \\ N_{\min} \le N_k \le N_{\max} \\ -\Delta\theta \le \theta_{k+1} - \theta_k \le \Delta\theta \\ \theta_{\min} \le \theta_k \le \theta_{\max} \end{cases} \tag{25}$$

As shown in Eq (25), considering that the control variables of the own ship are the turbine speed (N) and the rudder angle (θ), the constraints for the dynamic collision avoidance MPC are designed with practical and safety considerations in mind. The turbine speed change (ΔN) is limited to within 2 RPM per second, and the rudder angle change ($\Delta\theta$) is limited to within 5° per second. The ranges for the turbine speed and rudder angle are provided in Table 1.

4. Trajectory prediction experimental design and analysis

4.1. Experimental conditions and parameter settings

The algorithm in this paper was implemented by using the open-source neural network framework PyTorch (version 3.10). The computer configuration includes 1 GPU (GeForce RTX 3080), CPU (Intel(R) Core (TM) i9-9980HK CPU @ 2.40 GHz), and 64 GB RAM.

In Table 3, the model is trained with a batch size of 128, employs the ReLU activation function, and utilizes the Adam optimizer with a learning rate of 0.001. The model undergoes a uniform training process, encompassing a total of 400 epochs. Additionally, to investigate the impact of different numbers of layers in STGCNN and TXPCNN on -model performance, ablation experiments are conducted with the number of layers set to 1, 3, 5, and 7 for both STGCNN and TXPCNN, respectively. In the specific prediction experiments, the model takes

Table 3
Trajectory prediction model parameter settings.

Model hyperparameters	value
Batch size	128
Input step	15s
Output step	60s
Generation method	One-time generation
Activation function	Relu
STGCNN layer number	(1,3,5,7)
TXPCNN layer number	(1,3,5,7)
Optimizer	Adam
Number of training epochs	400
Learning rate	0.001

input trajectories with a 15-s interval and generates the predicted trajectories for the next 60 s in one go.

4.2. Datasets

Regarding AIS trajectory data for multi-ship navigation, there is currently no publicly available dataset. Therefore, AIS data was collected around ports with high ship density using the Marine Cadastre website (NOAA Office for Coastal Management). Due to the fact that the raw AIS data cannot be directly used for the task, the trajectory data that meets the task requirements is filtered based on the following conditions.

Step 1: Remove data with garbled characters in the AIS messages.
Step 2: Classify the data according to the ship's call sign.
Step 3: Due to the large time span of the AIS data, slice the classified data according to time to prevent AIS data from different time periods from mixing with each other.
Step 4: Since the interval of AIS data broadcasting is not fixed, fill the original AIS data with the same interval data through quadratic interpolation.
Step 5: Select AIS data of ships sailing in the same sea area at the same time based on time.

Through the above processing steps, we finally obtained 1393 multi-ship interaction scenarios, among which 1/10 were randomly selected as the test set scenario data.

4.3. Loss function

To convert the preliminary output trajectories of GL-STGCNN into a probability distribution, considering that trajectory positions can be represented as (x, y), we assume that the trajectory coordinates follow a bivariate Gaussian distribution. In this study, model training is conducted by minimizing the negative log-likelihood value. The negative log-likelihood loss function is defined as shown in Eq (26). For specific procedures, please refer to this study (Mohamed et al., 2020).

$$L = -\sum_{t=1}^{\mathrm{T}} \log(\mathbf{P}(\mathrm{P}_t|\overline{x}_t, \overline{y}_t, \overline{\sigma}_t, \overline{\rho}_t)) \tag{26}$$

In Eq (26), $\overline{x}_t$ and $\overline{y}_t$ represent the mean of the predicted position distribution, $\overline{\sigma}_t$ represents the variance of the predicted coordinate distribution, $\overline{\rho}_t$ represents the correlation between the predicted coordinates, and **P** represents the conditional probability.

4.4. Model evaluation metrics

In evaluating the model's performance, the ADE and FDE are employed. Since the model outputs a predicted trajectory as a bivariate Gaussian distribution, the initial prediction cannot be directly compared with the target value. Sampling is performed on the initial predicted distribution to obtain predicted trajectory points, and ADE and FDE are calculated using these predicted trajectory points and the target values.

$$\begin{aligned} \mathrm{ADE} &= \frac{\sum_{n=1}^{\mathrm{N}} \sum_{t=1}^{\mathrm{T}} \left\| \widehat{p}_t^n - p_t^n \right\|^2}{\mathrm{N} \times \mathrm{T}} \\ \mathrm{FDE} &= \frac{\sum_{n=1}^{\mathrm{N}} \left\| \widehat{p}_{\mathrm{T}}^n - p_{\mathrm{T}}^n \right\|^2}{\mathrm{N}} \end{aligned} \tag{27}$$

In Eq (27), p represents the positions of the trajectory points, T represents the time point, and N represents different ships.

4.5. Trajectory prediction quantitative analysis

In the paper "Social-STGCNN" on pedestrian trajectory prediction (Mohamed et al., 2020), a series of ablation experiments were conducted. The optimal trajectory prediction performance was achieved when the number of STGCNN layers and TXPCNN layers was set to 1 and 5, respectively. In subsequent experiments, the numbers of STGCNN and TXPCNN layers were fixed at 1 and 5.

The model performance, tested and compared on two different datasets, is shown in Table 4. While the performance of the same prediction model may vary on different datasets, the relative performance among different models usually does not change significantly due to dataset variations. Based on average performance metrics, the performance ratios of the Sosial-STGCNN model to other models (SocialLSTM, SocialGAN, GAT) were 0.76, 0.48, and 0.62, respectively. These values are comparable to the ones reported in the original paper "Sosial-STGCNN" (0.58, 0.73, 0.94). GL-STGCNN, compared to the model with the best performance (RAIN) in the Table, showed an increase of 31.8% in ADE and 16.8% in FDE. Regarding the model's inference time, there was no significant difference in computational performance among different models.

The predictive performance of the GL-STGCNN model on the training and test sets is shown in Table 5. For the training data, the model performs better due to the feedback from the loss function during training. However, the performance difference between the training and test sets is small, indicating that the model does not suffer from overfitting. This demonstrates that the model can generalize well to unseen data, exhibiting a certain level of generalization ability.

The performance testing discussed above is based solely on the original model structure, without considering the impact of trajectory correction on the performance of various models. The influence of trajectory correction on model performance is further explored in the following sections.

The ablation experiments in the paper "Social-STGCNN" indicate that the optimal model performance is achieved when the ratio of STGCNN layers to TXPCNN layers is set to 1:5(Mohamed et al., 2020). Table 6 presents the results of the ablation experiments for GL-STGCNN conducted in this paper. In Table 6, the numbers 1, 3, 5, and 7 in the first row represent the number of STGCNN layers, and the numbers 1, 3, 5, and 7 in the first column represent the number of TXPCNN layers. By combining the number of STGCNN layers from the first row with the number of TXPCNN layers from the first column, there are 16 different combinations of results in Table 5. Through detailed evaluation, it can be observed that when the number of STGCNN layers is 3 and the number of TXPCNN layers is 1, the model performance reaches its optimal level.

This analysis emphasizes the significance of hierarchical structure configuration in achieving optimal performance in trajectory prediction tasks.

From Table 7, it can be observed that as the number of ships gradually increases, the model's prediction accuracy shows a decreasing trend. This is due to the rising complexity of interaction problems as the number of targets increases, leading to a decrease in performance.

Table 4
Performance of different prediction models.

Model	Near Port of Houston		Near Port of Los Angeles		Average		Inference time (ms)
	ADE	FDE	ADE	FDE	ADE	FDE	
NRI	0.0656	0.0658	0.0767	0.0775	0.0711	0.0716	41.5
Social-Gan	0.0536	0.0578	0.0652	0.0691	0.0594	0.0634	89.4
GAT	0.0415	0.0482	0.0519	0.0591	0.0467	0.0536	57.2
LSTM	0.0346	0.0399	0.0423	0.0439	0.0384	0.0419	**13.7**
SocialI-LSTM	0.0326	0.0373	0.0434	0.0479	0.038	0.0426	27.4
Graph-LSTM	0.0321	0.0354	0.0421	0.0428	0.0371	0.0391	26.8
MTGCN	0.0267	0.0343	0.0346	0.0354	0.0306	0.0348	39.5
RAIN-STGCNN	0.026	0.0312	0.0360	0.0399	0.031	0.0355	48.9
Rain-LSTM	0.0247	0.0276	0.0344	0.0364	0.0295	0.032	21.3
Social-STGCNN	0.0245	0.0251	0.0337	0.0378	0.0291	0.0314	45.6
NRI-LSTM	0.0241	0.0261	0.0317	0.0362	0.0279	0.0315	29.1
RAIN	0.0188	0.0203	0.0221	0.0236	0.0204	0.0219	47.6
GL-STGCNN	**0.011**	**0.015**	**0.0167**	**0.0214**	**0.0139**	**0.0182**	49.6

Table 5
GL-STGCNN performance on training and test datasets.

Dataset	Near Port of Houston		Near Port of Los Angeles	
	ADE	FDE	ADE	FDE
Training set	**0.009**	**0.012**	**0.015**	**0.021**
Test set	0.011	0.015	0.0167	0.0214

Table 6
The influence of different layer combinations on the model.

	1	3	5	7
1	0.023/0.013	**0.004/0.006**	0.005/0.009	0.022/0.026
3	0.005/0.007	0.01/0.012	0.006/0.01	0.015//0.02
5	0.011/0.015	0.018/0.022	0.013/0.019	0.008/0.013
7	0.007/0.011	0.007/0.01	0.007/0.009	0.021/0.024

Table 7
The trajectory correction module improves the performance of prediction models.

Model	number	ADE	FDE
w/o trajectory correction	5	0.00288	0.00453
	6	0.00391	0.00675
	7	0.00480	0.00756
w/trajectory correction	5	0.00173	0.00223
	6	0.00204	0.00307
	7	0.00266	0.00367

However, through trajectory correction, the GL-STGCNN model demonstrates better performance in scenarios with different numbers of ships. The more ships there are, the more significant the improvement in the model's predictive performance. In scenarios with different numbers of ships, trajectory correction can enhance the performance of GL-STGCNN by 44.5%.

This indicates that capturing the interactions between ships becomes more challenging as the number of ships increases. However, trajectory correction proves to be an effective method for improving trajectory prediction performance.

4.6. Trajectory prediction qualitative analysis

In actual maritime traffic, the number and movement patterns of ships are not constant but are influenced by various factors. To assess the effectiveness of our prediction method from different perspectives, as shown in Fig. 10, simulating diverse maritime traffic situations in the real world. Through these tests and analyses, we gain a comprehensive understanding of the adaptability and accuracy of our model under various operating conditions, allowing for a more thorough evaluation of its performance and providing strong support for practical applications.

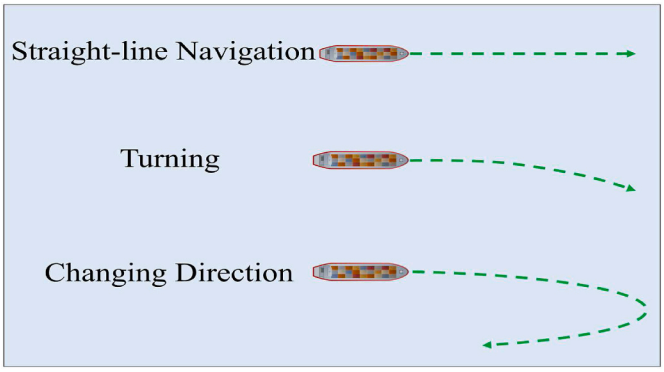


Fig. 10. Movement patterns of different ships.

4.6.1. Probability distribution of predicted ship trajectories

Fig. 11 shows the predicted trajectory distributions under different ship motion patterns and scenarios. Overall, the trajectory distributions obtained by GL-STGCNN can reflect the real ship trajectories, but there are still some problems. In the straight-line motion pattern, the predicted trajectory distributions of ship 4 in Fig. 11(a), ship 5 in Fig. 11(b), and ship 1 in Fig. 11(c) have certain deviations from the real trajectories, which usually exhibit a directional offset. In the turning pattern, the deviations between the predicted trajectory distributions of ship 1 in Fig. 11(a) and ship 2 in Fig. 11(c) and the real trajectories are larger than those in the straight-line motion pattern, this shows that the neural network has difficulty understanding the motion constraints of ships. This problem is more pronounced in the case of larger turning amplitudes, as shown in Fig. 11(c) for ship 6, where the predicted trajectory distribution has a large deviation from the real trajectory. As the number of ships in the scene increases, the interaction between ships becomes more complex, which leads to a gradual decrease in the trajectory prediction performance.

4.6.2. Ship interaction adjacency matrix in different navigation scenarios

The ship trajectory data in Fig. 11(a, b, c) is processed by GL-STGCNN to obtain visualized ship interaction adjacency matrices, which can reflect the interaction influence between ships, corresponding to Fig. 12(a, b, c) respectively.

For the analysis of column weights in Fig. 12(a), the weight values range from [1.8, 1.91], indicating similar levels of interaction among ships. However, the column corresponding to ship 2 has the relatively smallest weight, while the column for ship 4 has the relatively largest

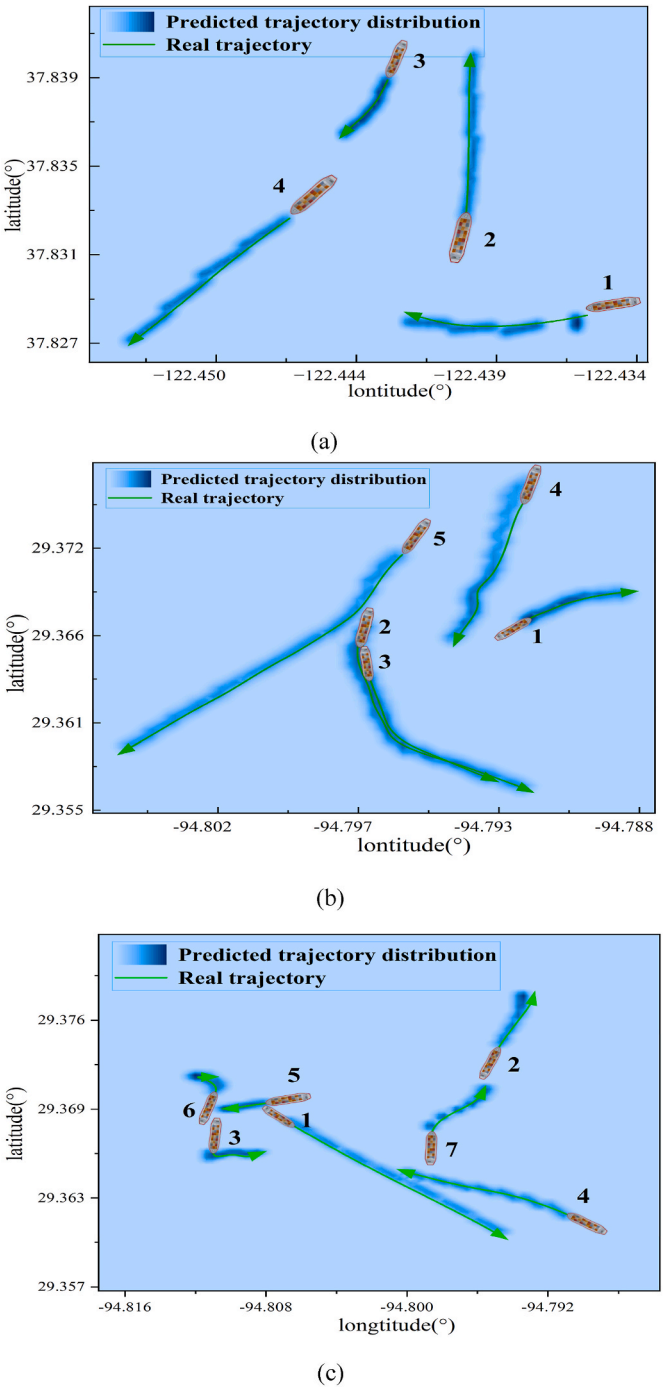


Fig. 11. Distribution of predicted trajectories under different ship movement mode and ship number scenarios.

weight. This is because ship 2 initially occupies a central position in the scene, while ship 4 consistently moves away from surrounding ships.

For the analysis of column weights in Fig. 12(b), the weight values range from [0.08, 0.18], suggesting significant differences in interaction levels among ships. The column for ship 1 has the relatively smallest weight, while the column for ship 4 has the relatively largest weight. This is due to ship 1 initially moving away from other ships, while ship 4 consistently approaches surrounding ships.

For the analysis of column weights in Fig. 12(c), the weight values range from [0.07, 0.17], indicating significant differences in interaction levels among ships. The columns for ships 3, 5, and 6 have the relatively largest weight values, as ships 3, 5, and 6 continuously maneuver and

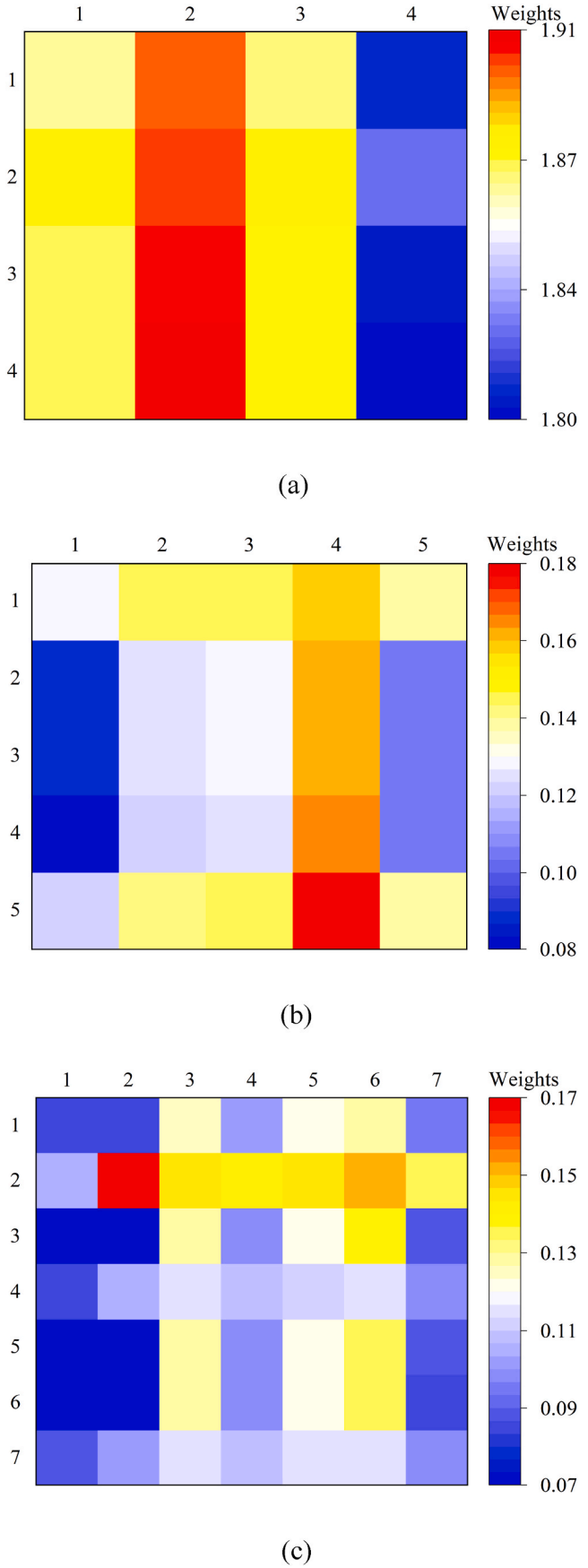


Fig. 12. Ship interaction adjacency matrix under different ship number and movement mode scenarios.

turn, with the smallest distances between them.

The analysis above utilizes weight analysis to quantify the interaction levels among different ships, providing in-depth insights into the relationships between ships. This type of analysis contributes to understanding the ship interaction adjacency matrix output by the model, offering crucial insights for optimizing model performance and adapting to diverse scenarios.

4.6.3. Corrected predicted ship trajectory

The corrected predicted trajectory distributions in Fig. 11(a, b, c) are shown in Fig. 13(a, b, c) respectively. Despite the increasing number of ships in the scenes of Fig. 13(a, b, c), the accuracy of the corrected predicted trajectories does not show a significant decline, and the overall correction effect is excellent.

In the straight-line motion mode, as shown in Fig. 13(a) ship 4, Fig. 13(b) ship 5, and Fig. 13(c) ship 1, the deviation between the corrected predicted trajectory and the real trajectory is further reduced. In the turning motion mode, as shown in Fig. 13(a) ship 1, Fig. 13(c) ship 2, and 7, the corrected predicted trajectory correctly reflects the motion characteristics of the ship in the turning mode. In the change direction motion mode, due to the large deviation between the original predicted trajectory distribution and the real trajectory, the corrected predicted trajectory of ship 6 in Fig. 13(c) does not correctly reflect the real motion of the ship.

The demonstrated effectiveness of ship trajectory prediction under various ship numbers and movement patterns attests to the reliability and accuracy of multi-ship trajectory prediction. This provides feasibility for integrating the results of trajectory prediction into dynamic collision avoidance tasks for ships.

5. Ship collision avoidance experimental design and analysis

5.1. Dynamic collision avoidance test environment

To realistically simulate the navigation states of ships, as shown in Fig. 14(a), a specific maritime area was selected, and AIS data for a certain period was obtained. Starting from the AIS data, scenarios were chosen where multiple ships sail together in the selected maritime area. In the designated scenarios, ships were controlled to navigate from the starting point along a route towards the destination while avoiding surrounding ships. In Fig. 14(a), there are five designated points to control the ships to navigate along existing routes in the maritime area. The simulation results of the ship crossing, overtaking and head-on encounter scenarios are obtained, and the ship's operation is verified to follow the COLREGs in Fig. 14(b) (Demirel and Bayer, 2015).

Surrounding factors have a significant and unavoidable impact on the navigation of ships. Ships exhibit distinct maneuvering performances in calm sea conditions compared to adverse sea conditions. In the process of controlling a ship to avoid collisions and reach a destination, it is essential to consider the influence of the surrounding sea conditions on the ship. As shown in Fig. 15(a), the water current rose diagram illustrates that the water flow in the target area is mainly influenced by tides, with limited flow in two directions and relatively low flow speed. According to Fig. 15(b), the wind rose diagram indicates that the wind speed in the target area is relatively high, and the wind direction is variable. These environmental factors pose certain challenges to ship navigation planning.

In this paper, the wind speed in the target area is set to 10 m/s, with a southward wind direction. The water current speed is 0.5 m/s in the northwest direction.

5.2. Dynamic collision avoidance parameter settings

As shown in Table 8, in the process of implementing dynamic collision avoidance through MPC in this paper, we set the weights of the obstacle avoidance objective function and the target point objective

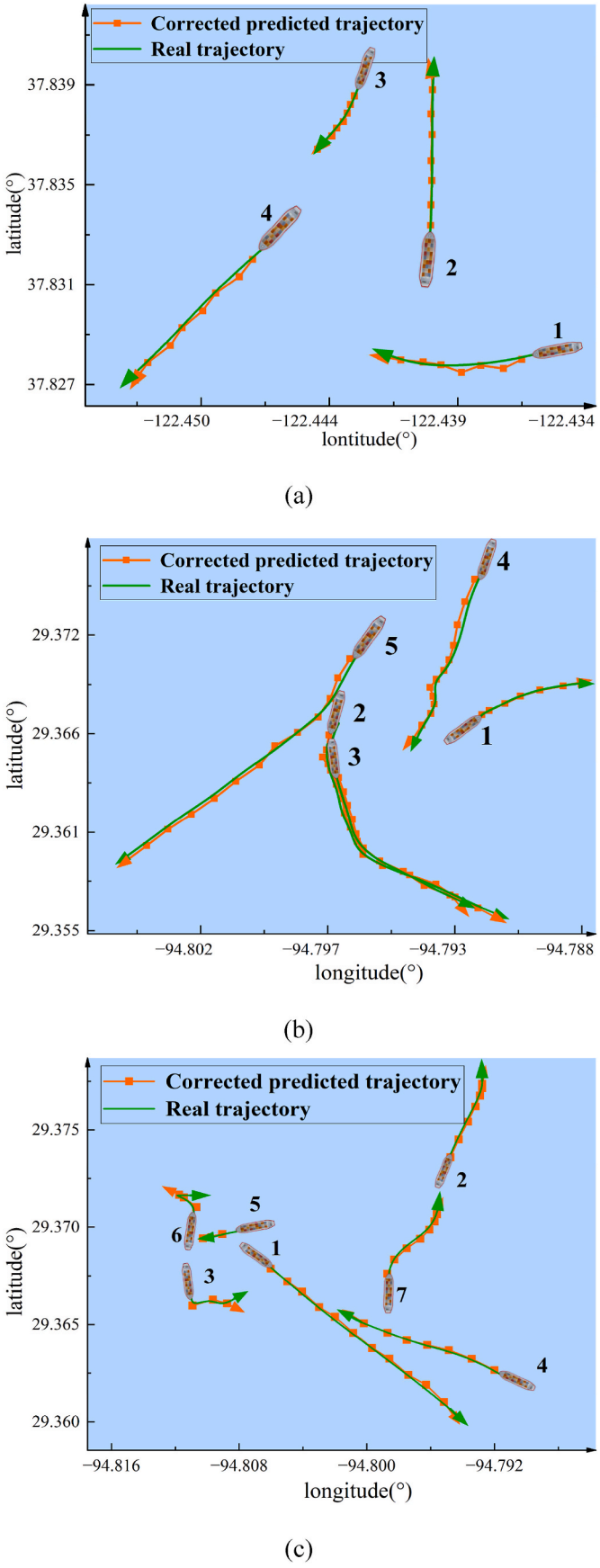
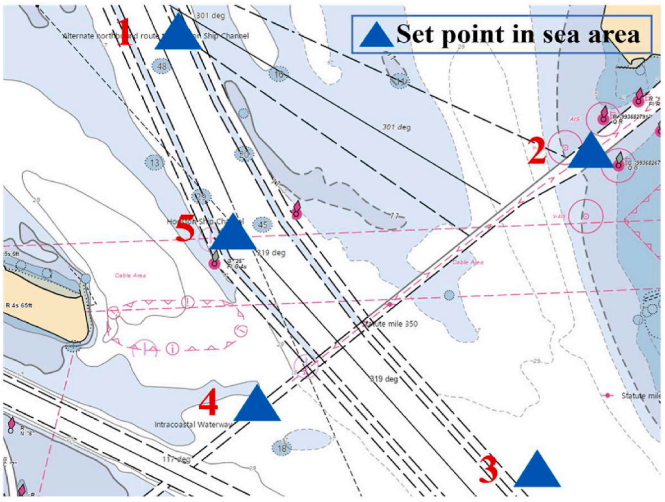
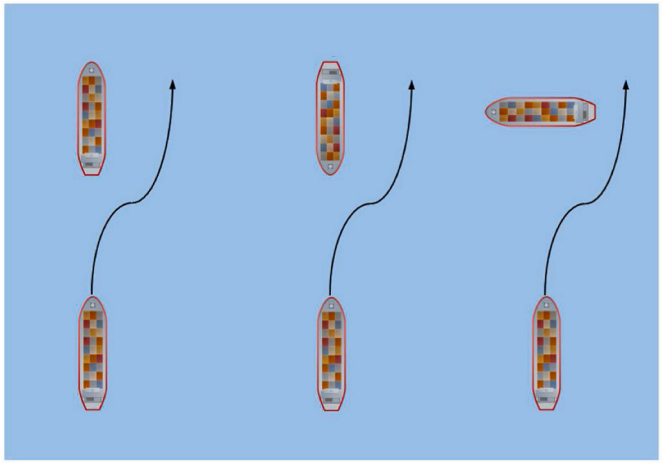


Fig. 13. Corrected ship prediction trajectories under different ship number and movement mode scenarios.



(a)



(b)

Fig. 14. Target sea area chart and COLREGs.

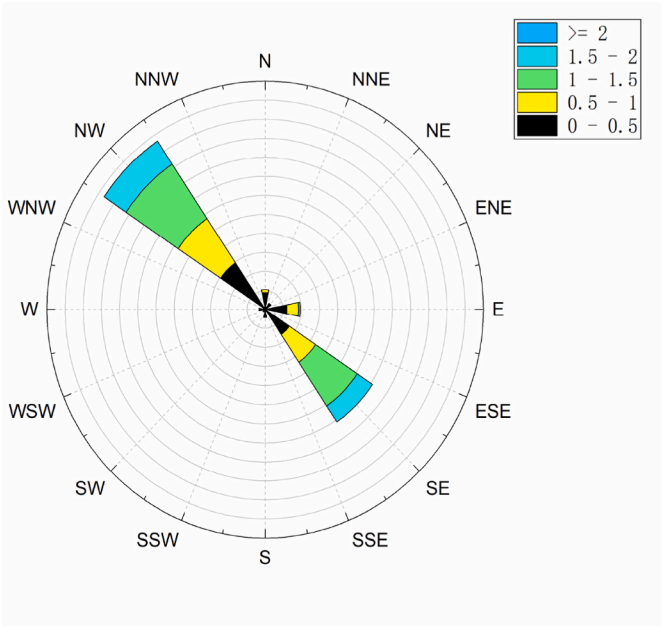
function to 0.01 and 1, respectively. This setting ensures that the sub-items of the objective function remain in the same order of magnitude. Since GL-STGCNN outputs the predicted trajectory with a step length of 60 s each time, we set the control horizon and prediction horizon parameters to 1 s and 60 s, respectively. For the initial speed and rudder angle of our ship, based on the given sailing performance of the ship, we set them to 5 m/s and 0°, respectively.

5.3. Single-ship encounter simulation verification

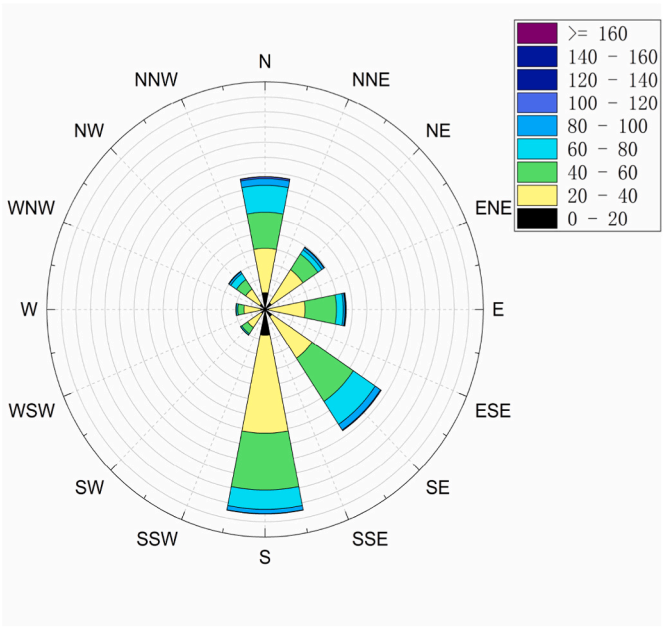
Fig. 16(a) shows a scenario of a crossing encounter. The case ship is sailing from point 2 to point 1 in Fig. 14(a). Based on the track, the case ship chooses to turn right to avoid collision with the surrounding ships. This maneuver not only avoids collision with the surrounding ships in the future but also makes the own ship closer to its destination.

According to Fig. 16(b), during the process of controlling the case ship to avoid collision and reach the destination, the case ship starts to avoid collision at 81 s and completes the avoidance process at 171 s. Then it sails straight to the target point.

Fig. 17 shows a scenario of a head-on encounter. The case ship is sailing from point 2 to point 4 in Fig. 14(a). During the voyage, the case ship chooses to turn right to avoid collision at 80 s. The avoidance process ends at 163 s, and then the case ship adjusts its heading and sails



(a)



(b)

Fig. 15. Surrounding environmental factors.

straight to the target point. This operation complies with the COLREGs in Fig. 14(b).

As shown in Fig. 18, we have added an overtaking collision avoidance test scenario in the revised manuscript. The planned trajectory of our ship is represented by red squares, while the trajectory of the obstacle ship to be overtaken is represented by blue squares. It can be observed that the blue obstacle ship has a lower speed compared to our ship. During the overtaking maneuver, our ship chooses to turn starboard and overtake from the right side of the obstacle ship. This behavior complies with the regulations of the International Regulations

Table 8
Dynamic collision avoidance parameter settings.

MPC hyperparameters	value
Obstacle avoidance weight	0.01
Target point weight	1
Control horizon	1s
Prediction horizon	60s
Initial velocity	5 m/s
Initial rudder angle	0°
Wind speed	10 m/s
Current speed	0.5 m/s

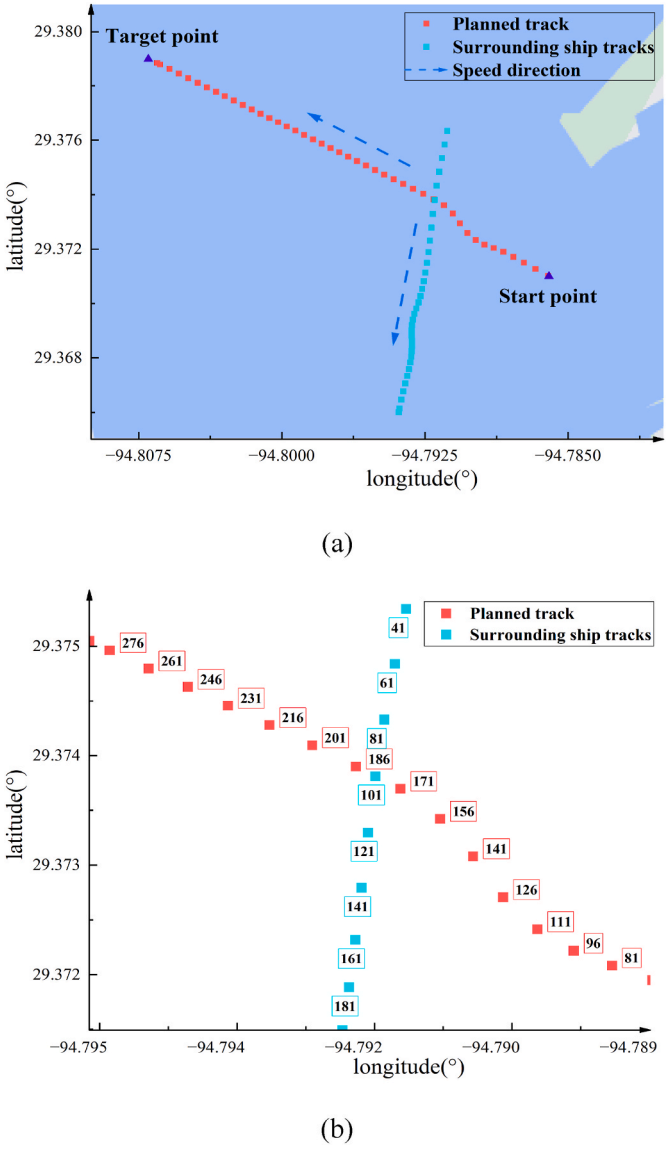


Fig. 16. The obstacle avoidance process and performance metrics for crossing encounters.

for Preventing Collisions at Sea (COLREGs), which stipulate that when overtaking a slower ship in the same direction, the overtaking ship should pass on the overtaken ship 's starboard side to reduce the risk of collision.

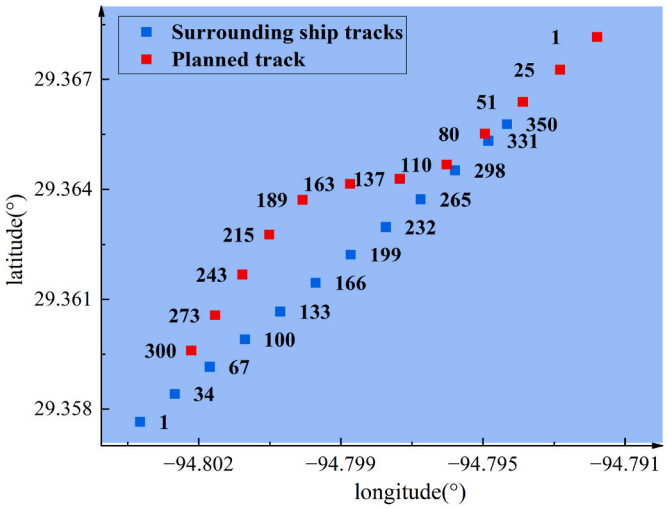


Fig. 17. The obstacle avoidance process and performance metrics for head-on encounters.

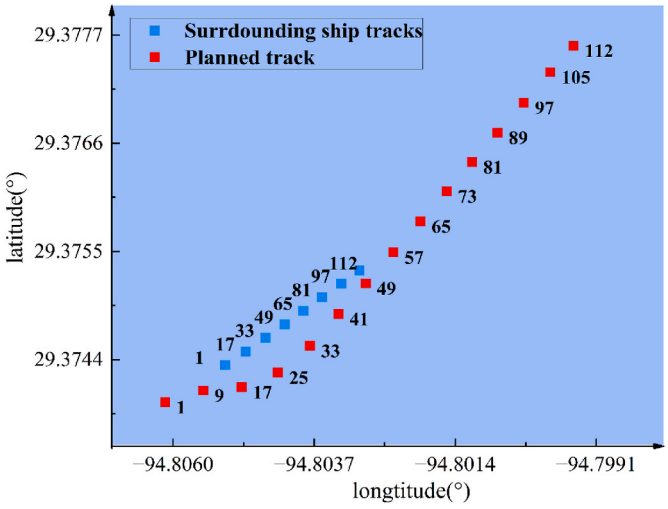


Fig. 18. The Obstacle avoidance process and performance metrics of overtaking.

5.4. Multi-ship interactive collision avoidance simulation verification

5.4.1. From point 1 towards point 2

According to Fig. 19(a,b), there are two collision avoidance processes in total when the case ship sails from point 1 to point 2 in Fig. 14(a). The first is to avoid the green ship, and the second is to avoid the orange ship.

According to Fig. 19(c), the collision avoidance process with the green ship starts at 50 s and ends at 108 s. The collision avoidance process with the orange ship starts at 150 s and ends at 261 s.

According to Fig. 19(d), the distances at which the collision avoidance processes with the green and orange ships are triggered are 512 m and 485 m, respectively. These distances match well with the theoretically calculated second-stage collision avoidance threshold distance of 500 m.

5.4.2. From point 2 towards point 4

According to Fig. 20(a,b,c), the case ship sails from point 2 to point 4 in Fig. 14(a). There are two collision avoidance processes in total in this dynamic collision avoidance process. The first is to avoid the purple ship, and the collision avoidance measure is to turn left. The second is a

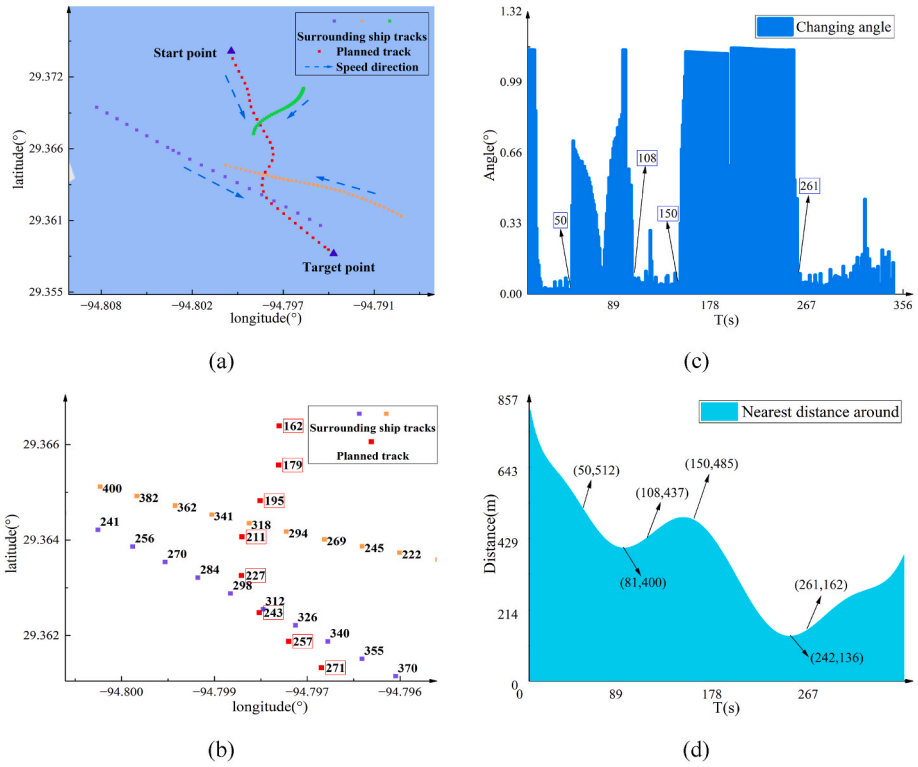


Fig. 19. The obstacle avoidance process and performance metrics for three ships.

crossing encounter, and the collision avoidance operation is to turn right, which complies with the COLREGs for crossing encounters.

According to Fig. 20(d), the first collision avoidance process starts at 52 s and ends at 194 s. The second collision avoidance process starts at 251 s and ends at 406 s.

According to Fig. 20(e), the distances at which the two collision avoidance processes are triggered are 464 m and 341 m, respectively. 464 m matches well with the theoretically calculated second-stage collision avoidance threshold distance of 500 m. For 341 m, before 251 s, the case ship was in the potential field of the previous ship and was driven away. However, after 251 s, the balance of the potential field strength of the two ships in front and behind was broken, so the case ship immediately entered the second collision avoidance process.

5.4.3. From point 3 towards point 5

According to Fig. 21(a, b, c), there are two collision avoidance behaviors in the scenario when the case ship sails from point 3 to point 5 in Fig. 14(a). The first is to avoid the green ship, and the collision avoidance operation is to turn right, which complies with the COLREGs for crossing encounters.

The second is to avoid the blue and purple ships. In this case, the case ship takes the operation of turning left for crossing encounters. Although turning left in this scenario does not comply with the COLREGs for crossing encounters, the distance between the ships is large in this scenario, and turning left makes the distance between the case ship and the target point closer. In terms of efficiency, it is more efficient for the ship to turn left in this scenario. According to Fig. 21(e), the minimum distance between the ships after turning left in the second collision avoidance behavior is 273 m, which is greater than the minimum distance of 212 m between the ships in the first collision avoidance behavior. In terms of ship safety, it is safer for the ship to turn left in this scenario.

According to Fig. 21(d), the first collision avoidance process starts at 142 s and ends at 235 s. The second collision avoidance process starts at 302 s and ends at 410 s.

According to Fig. 21(e), the distances at which the two collision avoidance processes are triggered are 449 m and 434 m, respectively. 449 m and 434 m match well with the theoretically calculated second-stage collision avoidance threshold distance of 500 m.

6. Conclusions

This paper integrates the GL-STGCNN multi-ship trajectory prediction model with the MPC model to propose a novel ship dynamic collision avoidance method. This method aims to achieve safe and efficient navigation of ships in dynamic and complex maritime environments. Initially, GL-STGCNN is used to model the interaction adjacency matrix between surrounding ships and obtain future trajectories, providing dynamic information for ship collision avoidance. Subsequently, based on the principles of APF and VO collision avoidance, the predicted trajectories of surrounding ships are incorporated into the objective function. This transforms the complex collision avoidance decision problem into an optimization problem, successfully achieving three stages of ship collision avoidance behavior. Through simulation and verification in multi-ship navigation scenarios using AIS data, the performance of the model is analyzed. GL-STGCNN, by modeling precise interaction adjacency matrices between ships, achieves more accurate trajectory predictions compared to previous algorithms. It shows a 31.8% improvement in the ADE metric and a 16.8% improvement in the FDE metric. Additionally, the ship dynamic collision avoidance model based on GL-STGCNN exhibits behaviors conforming to the COLREGs during ship-crossing encounters and head-on encounters. In scenarios with multiple ship interactions, the planned paths are not confined to international maritime collision avoidance rules. Different turning behaviors are adopted at appropriate distances to achieve dynamic collision avoidance. Moreover, the theoretical collision avoidance threshold distance calculated by the MPC objective function achieves a good match with the collision avoidance trigger distance obtained from the experimental simulation.

Additionally, we acknowledge that the dynamic collision avoidance

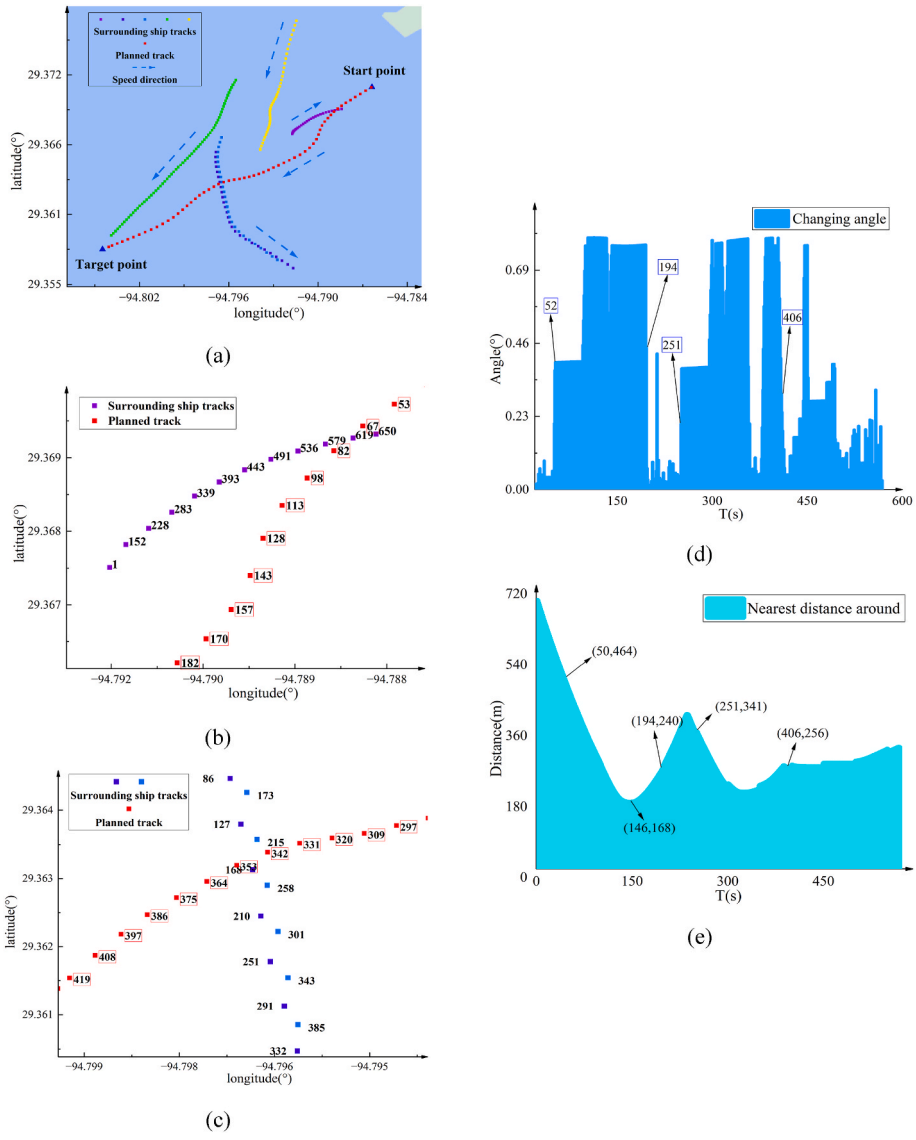


Fig. 20. The obstacle avoidance process and performance metrics for five ships.

method proposed in this paper relies to some extent on an adequate amount of AIS data within the maritime area to train the trajectory prediction model. In situations where AIS data is scarce or limited, the performance of this dynamic collision avoidance method may be compromised. Therefore, future research could explore the integration of more advanced trajectory prediction models with data augmentation techniques to enhance the adaptability and generalization performance of this method in various maritime environmental scenarios. We look forward to further research efforts that will refine and advance this field, providing more reliable and intelligent solutions for practical applications.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material; further inquiries can be directed to the corresponding author.

Funding

This paper was supported by the National Natural Science Foundation of China, China [52171308]; the Natural Science Foundation of Fujian Province, China [2022J01333]; and the Natural Science Foundation of Fujian Province, China [2022J01813].

CRediT authorship contribution statement

Weiqiang Liao: Writing – original draft, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Yuegao Wu:** Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **Peilin Zhou:** Writing – review & editing, Supervision, Methodology. **Haibin Wang:** Writing – review & editing, Visualization, Investigation, Formal analysis. **Wanneng Yu:** Writing – original draft, Visualization, Validation, Supervision, Project administration, Investigation, Funding acquisition. **Changkun Zhang:** Visualization, Supervision, Resources, Project administration, Methodology, Investigation. **Chenghan Luo:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence

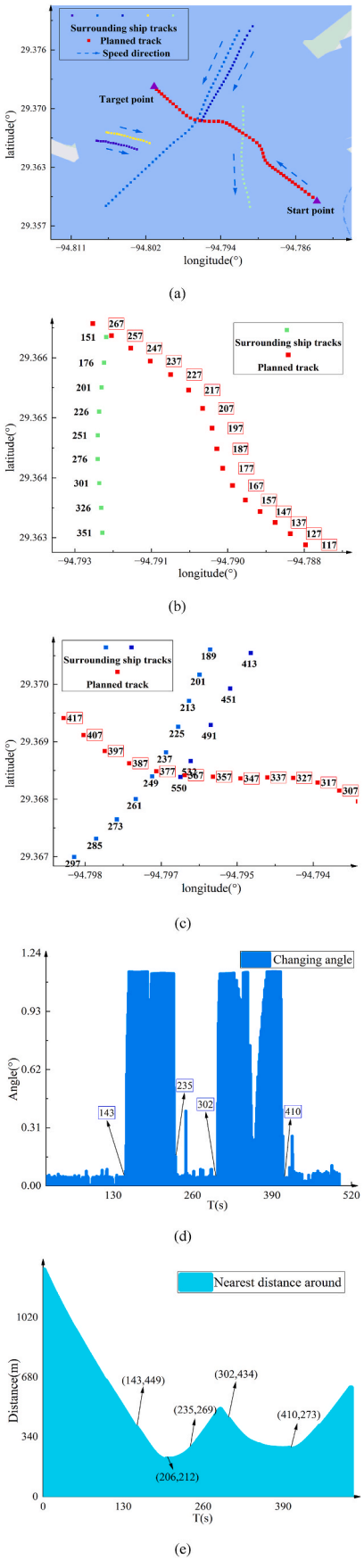


Fig. 21. The obstacle avoidance process and performance metrics for five ships.

the work reported in this paper.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, Fei-Fei, L., Savarese, S., 2016. Social lstm: human trajectory prediction in crowded spaces[C]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–971.

Bounini, F., Gingras, D., Pollart, H., Gruyer, D., 2017. Modified artificial potential field method for online path planning applications[C]. In: 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, pp. 180–185.

Burmeister, H.C., Constapel, M., 2021. Autonomous collision avoidance at sea: a survey. Frontiers in Robotics and AI 8, 739013.

Cheng, B., Xu, X., Zeng, Y., Ren, J., Jung, S., 2018. Pedestrian trajectory prediction via the Social-Grid LSTM model. J. Eng. 2018 (16), 1468–1474.

Chun, D.H., Roh, M.I., Lee, H.W., Ha, J., Yu, D., 2021. Deep reinforcement learning-based collision avoidance for an autonomous ship. Ocean Engineering 234, 109216.

Davis, N., Raina, G., Jagannathan, K., 2020. Grids versus graphs: Partitioning space for improved taxi demand-supply forecasts. IEEE Trans. Intell. Transport. Syst. 22 (10), 6526–6535.

Demirel, E., Bayer, D., 2015. Further studies on the COLREGs (collision regulations). TransNav: International Journal on Marine Navigation and Safety of Sea Transportation 9 (1), 17–23.

Du, Z., Reppa, V., Negenborn, R.R., 2021. MPC-based COLREGS compliant collision avoidance for a multi-ship ship-towing system[C]. In: 2021 European Control Conference (ECC). IEEE, pp. 1857–1862.

Enevoldsen, T.T., Reinartz, C., Galeazzi, R., 2021. COLREGs-Informed RRT* for collision avoidance of marine crafts[C]. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 8083–8089.

Fan, C., Wróbel, K., Montewka, J., Gil, M., Wan, C., Zhang, D., 2020. A framework to identify factors influencing navigational risk for Maritime Autonomous Surface Ships. Ocean Engineering 202, 107188.

Feng, H., Cao, G., Xu, H., Ge, S., 2022. IS-STGCNN: an improved social spatial-temporal graph convolutional neural network for ship trajectory prediction. Ocean Engineering 266, 112960.

Guan, W., Zhao, M.Y., Zhang, C.B., Xi, Z.Y., 2023. Generalized behavior decision-making model for ship collision avoidance via reinforcement learning method. J. Mar. Sci. Eng. 11 (2), 273.

Hagen, I.B., Kufoalor, D.K.M., Brekke, E.F., Johansen, T.A., 2018. MPC-based collision avoidance strategy for existing marine ship guidance systems[C]. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 7618–7623.

He, Z., Liu, C., Chu, X., Negenborn, R.R., Wu, Q., 2022. Dynamic anti-collision A-star algorithm for multi-ship encounter situations. Appl. Ocean Res. 118, 102995.

Iswanto, I., Ma'arif, A., Wahyunggoro, O., Cahyadi, A.I., 2019. Artificial potential field algorithm implementation for quadrotor path planning. Int. J. Adv. Comput. Sci. Appl. 10 (8).

Kim, H.T., Na, S., 2017. Development of a human factors investigation and analysis model for use in maritime accidents: a case study of collision accident investigation. Journal of Navigation and Port Research 41 (5), 303–318.

Kouvaritakis, B., Cannon, M., 2016. Model Predictive Control, vol. 38. Springer International Publishing, Switzerland, pp. 13–56.

Li, C., Zhao, L., Zhang, Z., 2023. MTGCN: multi-graph Fusion based temporal-spatial convolution for traffic flow forecasting[C]. In: 2023 IEEE 3rd International Conference on Computer Communication and Artificial Intelligence (CCAI). IEEE, pp. 75–80.

Lindqvist, B., Mansouri, S.S., Agha-mohammadi, A., Nikolakopoulos, G., 2020. Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles. IEEE Rob. Autom. Lett. 5 (4), 6001–6008.

Lu, N., Zhou, W., Yan, H., Fei, M., Wang, Y., 2022. A two-stage dynamic collision avoidance algorithm for unmanned surface vehicles based on field theory and COLREGs. Ocean Engineering 259, 111836.

Lu, Z., Lv, W., Cao, Y., Xie, Z., Peng, H., Du, B., 2020. LSTM variants meet graph neural networks for road speed prediction. Neurocomputing 400, 34–45.

Mohamed, A., Qian, K., Elhoseiny, M., Claudel, C., 2020. Social-stgcnn: a social spatio-temporal graph convolutional neural network for human trajectory prediction[C]. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14424–14432.

Nelson, K.P., Thistleton, W.J., 2021. Comments on "generalized Box-Müller method for generating q-Gaussian random Deviates". IEEE Trans. Inf. Theor. 67 (10), 6785–6789.

NOAA Office for Coastal Management. AccessAIS [EB/OL]. (2023-12-25) [2023-12-25]. https://marinecadastre.gov/accessais/.

Prabhakar, P., Rahimi Afzal, Z., 2019. Abstraction based output range analysis for neural networks. Adv. Neural Inf. Process. Syst. 32.

Singh, Y., Sharma, S., Sutton, R., Hatton, D., Khan, A., 2018. A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. Ocean Engineering 169, 187–201.

Ugurlu, H., Cicek, I., 2022. Analysis and assessment of ship collision accidents using fault tree and multiple correspondence analysis. Ocean Engineering 245, 110514.

Wang, F., Xu, J., Liu, C., Zhou, R., Zhao, P., 2020. MTGCN: a multitask deep learning model for traffic flow prediction[C]. In: Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24–27, 2020, Proceedings, Part I 25. Springer International Publishing, pp. 435–451.

Wang, Y., Jing, C., Xu, S., Guo, T., 2022. Attention based spatiotemporal graph attention networks for traffic flow forecasting. Inf. Sci. 607, 869–883.
Wang, Z., Li, G., Ren, J., 2021. Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm. Comput. Commun. 166, 49–56.
Wieder, O., Kohlbacher, S., Kuenemann, M., Garon, A., Ducrot, P., Seidel, T., Langer, T., 2020. A compact review of molecular property prediction with graph neural networks. Drug Discov. Today Technol. 37, 1–12.
Xiao, B.Q., Yong, Y., Xiu, F.Z., Li, Y., 2016. Influence of irregular disturbance of sea wave on ship motion. J. Traffic Transport. Eng. 16 (3), 116–124.
Yang, X., Han, Q., 2023. Improved reinforcement learning for collision-free local path planning of dynamic obstacle. Ocean Engineering 283, 115040.
Zhang, C., James, J.Q., Liu, Y., 2019. Spatial-temporal graph attention networks: a deep learning approach for traffic forecasting. IEEE Access 7, 166246–166256.
Zhang, G., Wang, Y., Liu, J., Cai, W., Wang, H., 2022a. Collision-avoidance decision system for inland ships based on velocity obstacle algorithms. J. Mar. Sci. Eng. 10 (6), 814.
Zhang, J., Zhang, H., Liu, J., Wu, D., Soares, C.G., 2022b. A two-stage path planning algorithm based on rapid-Exploring random tree for ships navigating in multi-obstacle water areas considering COLREGs. J. Mar. Sci. Eng. 10 (10), 1441.