journal homepage: www.brodogradnja.fsb.hr

Brodogradnja

An International Journal of Naval Architecture and Ocean Engineering for Research and Development

Deep reinforcement learning for integrated vessel path planning with safe anchorage allocation

Gil-Ho Shin[1], *Hyun Yang*[2*]

[1] *Graduate School of Korea Maritime and Ocean University, Busan, Republic of Korea*
[2] *Division of Maritime AI & Cyber Security, Korea Maritime and Ocean University, Busan, Republic of Korea*

ARTICLE INFO

Keywords:
Maritime safety
Reinforcement learning
Vessel Traffic Services (VTS)
Path planning
Deep reinforcement learning

ABSTRACT

This study addresses vessel path planning and anchorage allocation through a reinforcement learning approach. To improve maritime safety and efficiency, we developed an integrated system that combines Deep Q-Network and Artificial Potential Field concepts for path generation. The model implements a specialized grid extension method that accounts for actual vessel dimensions and wind direction, while incorporating differentiated safety distances for each anchorage area. Experimental validation using Automatic Identification System (AIS) data demonstrated that the system successfully generated efficient routes while maintaining all safety distance requirements during both navigation and anchoring phases. Additionally, the system ensured practicality through path simplification using the Douglas-Peucker algorithm while maintaining safety standards. The visualized optimal paths enhance navigational guidance, thereby improving both maritime traffic safety and port operational efficiency.

1. Introduction

Maritime logistics serves as a cornerstone of the global economy, driving international trade and economic development. Busan Port, as South Korea's premier trading port and a major logistics hub in Northeast Asia, consistently maintains a high ranking among world ports. As of 2023, according to the Busan Port Authority's Container Cargo Handling Statistics, Busan Port handles 23,154,000 TEUs of container cargo annually, ranking 7th globally in terms of container throughput. In transshipment volume, it ranks 2nd worldwide with 12,409,000 TEUs [1]. The port's significance is further evidenced by its high vessel traffic volume, with 90,453 vessels entering and departing in 2023 based on the Ministry of Oceans and Fisheries' Annual Vessel Entry and Departure Statistics [2]. At the same time, with the rapid development of science and technology, ships are moving towards intelligence, large-scale, and high-speed development, and the maritime navigation environment has become increasingly complex [3].

The efficient and safe management of this large-scale vessel traffic volume is a critical task that directly impacts national economic stability and growth beyond port operations. In particular, among various aspects

* Corresponding author.
E-mail address: yanghyun@kmou.ac.kr

http://dx.doi.org/10.21278/brod76305
Received 25 January 2025; Received in revised form 28 April 2025; Accepted 09 May 2025
Available online 18 May 2025
ISSN 0007-215X; eISSN 1845-5859

of port operations, anchorage allocation and optimal route planning are the most crucial elements. The efficiency of these processes directly affects overall port performance and determines both vessel safety and efficient utilization of port resources. According to ESKI and Tavacioglu [4], maritime accidents most frequently occur during navigation and anchorage operations, with a particularly high incidence during bunkering operations while at anchor.

These challenges highlight the need for digital transformation in maritime logistics and smart port development. The fourth industrial revolution, represented by artificial intelligence, big data, quantum information, and other technologies, is coming [5]. As one of the traditional transportation industries, maritime transportation is also developing towards automation and intellectualization [6]. In this context, artificial intelligence technologies are gaining attention for automating and optimizing complex decision-making processes like anchorage allocation. Among these technologies, Reinforcement Learning has emerged as a particularly effective approach for solving such complex decision-making problems. Reinforcement Learning is a branch of machine learning where agents learn optimal policies through trial-and-error interactions with their environment [7], and it can complement human limitations and support objective decision-making in optimal path planning and resource allocation problems that require complex decision-making [8]. The key advantages of Reinforcement Learning are as follows [9-11]:

1. Dynamic Environment Adaptation: The system can adapt to changing environments and find optimal solutions, making it suitable for complex and uncertain maritime environments.

2. Sequential Decision Making: It demonstrates effective handling of sequential decision-making problems, such as route planning.

3. Long-term Reward Optimization: The system is capable of making decisions considering both immediate and long-term outcomes.

4. Non-linear Problem Solving: It excels at learning complex non-linear relationships, making it suitable for port operations involving multiple interrelated variables.

5. Autonomous Learning: The system possesses the ability to self-improve through experience without explicit programming.

Based on these characteristics, this study proposes a system that plans optimal vessel routes using reinforcement learning, following objective and consistent criteria. Specifically, we implement optimal decision-making in complex maritime environments using Deep Q-Network (DQN), which has shown excellent performance in recent vessel path planning research [12-14]. Additionally, we incorporate the Potential Field concept, proven effective in vessel obstacle avoidance, to balance obstacle avoidance and goal tracking [15-17]. Notably, the Potential Field method has been successfully applied to vessel collision avoidance and path planning in studies conducted by Lyu and Yin [18] and Liu et al. [19], and in our research, we combine this with DQN for more effective path planning. Furthermore, we achieve practical route simplification through the Douglas-Peucker algorithm, which has proven effective in path simplification and practicality enhancement in studies by Du et al. [20], Guo et al. [21], and Lee and Kim [22].

The core functionality of this system is to automatically calculate and visually present optimal trajectories to anchoring positions for each vessel. This visualization capability can enhance safety and efficiency in maritime operations. The implementation of this system is expected to support VTS operator decision-making and enhance safety. As a result, overall port operational efficiency will improve while significantly reducing safety incident risks.

This paper is organized as follows.

Section 2 systematically reviews recent reinforcement learning studies related to vessel path planning to establish the academic positioning of this research.

Section 3 describes the simulation design modeling Busan Port's anchorage environment. In particular, it details methodologies for accurately reflecting real port conditions, including Busan Port's geographical characteristics, anchorage rules, grid-based framework, anchored vessel grid extension methods, and optimal anchoring position determination procedures. Because these environmental constraints and parameters form the basis for defining the agent's state, action, and reward structures, Section 4 then builds upon the simulation environment to present the reinforcement learning model.

Section 4 explains the reinforcement learning model implementation, providing detailed descriptions of state space, action space, dynamic modeling, reward function definitions, and the DQN algorithm implementation process.

Section 5 analyzes experimental results, conducting quantitative performance evaluations by comparing actual vessel paths with optimal paths generated by the proposed model for each anchorage area, focusing on path length reduction rates, safety distance compliance, and anchored vessel avoidance.

Finally, Section 6 summarizes the research achievements and suggests future research directions, including safety distance settings considering anchor chain length.

2. Literature Review

A comprehensive literature review was conducted using the Web of Science (WOS) database in November 2024 to identify relevant studies in vessel path planning using reinforcement learning. Initially, a search for 'Reinforcement Learning' AND 'anchorage' across all years yielded 10 documents, but none were directly relevant to reinforcement learning-based anchorage allocation. This indicated a research gap in the specific application of reinforcement learning to anchorage planning.

To understand recent research trends, we then focused on the past five years (2020-2024), broadening our search to include general vessel path planning studies using reinforcement learning. The search terms 'Reinforcement Learning' AND ('vessel path planning' OR 'ship path planning') initially returned 410 results. After excluding underwater vessel-related studies, 387 papers remained. Through careful review of titles, abstracts, and content, 16 papers were identified as directly relevant to our research focus. These papers were systematically analyzed and are summarized in Table 1.

Table 1 Summary of Related Studies in Maritime Path Planning

No.	Author(s)	Geographical Scope	Main Algorithm	Evaluation Method	Performance Metrics
1	Xu et al. [12]	Coastal water	MAPF-DQN	Comparative analysis with A-STAR and DQN algorithms	Path safety (minimum distance from obstacles), Path efficiency (path length), Planning success rate, Number of waypoints
2	Xiao et al. [23]	Simulated environment	Beta policy-based PPO	Comparative analysis with traditional algorithms (A-STAR, IDA-STAR, Dijkstra)	Planning time, Path length
3	Gao et al. [13]	Coastal water	Improved DQN	Comparative analysis with Original Q-learning, Original DQN, Double DQN, Dueling DQN algorithms	Planning success rate in different scenarios, Planning time, Path length, Model stability
4	Du et al. [20]	Simulated environment	DDPG with LSTM	Comparative analysis with DDPG, RRT, RRT*, APF, A-STAR, and BUG2 algorithms in multiple test environments	Path length, Number of turning points, Planning time, Model convergence speed
5	Yuan et al. [14]	Inland waterways	DQN	Comparative analysis between trained model and actual ferry trajectories with both undefined and defined crossing patterns	Convergence speed, Navigation safety (DCPA, TCPA, relative distances), Path efficiency (voyage length, navigation time), Crossing pattern accuracy
6	Guo et al. [21]	Simulated environment (Electronic chart-based)	DQN	Comparative analysis with traditional DQN, Q-learning, DDPG, A-STAR, BUG2, and APF algorithms in multiple test scenarios	Path length, Planning time, Number of path corners, Convergence speed, Model stability
7	Kim et al. [8]	Port area	Q-learning	Comparative analysis with A-STAR algorithm and PD controller	Route safety (UKC & navigation rules), Fuel efficiency, Path smoothness, Route deviation, Rudder angle changes
8	Chun et al. [24]	Simulated environment	PPO-based DRL	Comparative analysis with A-STAR algorithm in various scenarios	Minimum distance between ships, Maximum collision risk, Travel distance, Model convergence speed
9	Li et al. [25]	Coastal and inland waterways	DQN	Comparative analysis with DQN, DDPG, RRT, A-STAR, and APF algorithms	Path trajectory smoothness, Distance to destination, Displacement distance, COLREGS compliance
10	Lee and Kim [22]	Coastal waters between major ports	DQN	Comparative analysis with Q-learning and A-STAR algorithms	Navigation safety (adherence to rules), Path efficiency (distance

					reduction), Waypoint optimization
11	Wang et al. [26]	Simulated environment	Hybrid bi-level planning combining improved	Comparative analysis with standard PSO and traditional APF	Path length, Convergence speed, Computing time, Obstacle avoidance capability
12	Xie et al. [27]	Simulated environment	A3C with LSTM	Comparative analysis with model-free A3C and traditional optimization-based methods	Learning efficiency, Collision avoidance safety (CRI, minimum distances), Path optimization (rudder actions, heading errors), Model convergence
13	Chen et al. [28]	Simulated environment	Multi-agent DRL	Comparative analysis across three typical COLREGs scenarios	Collision avoidance safety, Cooperation effectiveness, COLREGs compliance, Path efficiency, Convergence in narrow waters
14	Guo et al. [29]	Simulated environment	DDPG	Comparative analysis with DQN, AC, DDPG and Q-learning algorithms across multiple encounter scenarios	Path length, Training convergence speed, Collision avoidance safety, Rule compliance, Decision time, Navigation stability
15	Cao et al. [30]	Inland waterways	Improved RRT	Comparative analysis with real ship AIS trajectories	Average error rate, Maximum error rate, Planning time, Path length, Safety distance maintenance
16	Zhen et al. [31]	Port area	Improved A-star	Comparative analysis with traditional A-star algorithm in simulation and real scenarios	Navigation safety (collision risk, grounding risk), Path length, Number of turns, Traffic rule compliance, Path smoothness, Computational efficiency

To systematically analyze the prior studies summarized in Table 2, we first examine their geographical scope followed by their algorithmic approaches. Through this analysis, we can identify the characteristics and limitations of maritime areas targeted by each study, while the analysis of algorithmic approaches provides insights into the current applications and development trends of reinforcement learning technologies.

2.1 Geographical Scope

Prior studies can be broadly categorized into two groups based on their geographical scope: research targeting actual maritime areas and research using simulated environments. Within these categories, studies of actual maritime areas can be further divided into those focusing on real ports and coastal waters versus inland waterways. Similarly, simulation-based studies can be subdivided into simplified virtual environments and electronic chart-based environments. Importantly, this classification reveals distinctive differences in research objectives and application scope.

Studies focused on actual ports and coastal waters made significant efforts to reflect realistic navigation environments. For instance, Gao et al. [13] conducted research in the Bohai Bay area (36.5°N-41.1°N, 117.0°E-125.5°E), specifically addressing complex terrain and maritime traffic characteristics. Kim et al. [8] validated path planning algorithms in real vessel operating conditions by studying the specific waters between Busan Port and Gamcheon Port (34°59'37"N-35°07'52"N,

128°57'08"E-129°09'37"E). Furthermore, Lee and Kim [22] investigated path optimization for long-distance navigation between Busan and Gwangyang ports.

Studies targeting inland waterways focused on specialized navigation conditions in restricted waters. Notable examples include research by Yuan et al. [14] and Cao et al. [30]. In particular, Cao et al. [30] studied path planning in narrow channels and complex traffic situations by examining the Shanghai waterway (20 km × 20 km) and Zhenjiang waterway (10 km × 10 km) of the Yangtze River. These studies emphasized the importance of precise navigation planning in confined waters.

Studies utilizing simulation environments demonstrated two main approaches. First, studies using simplified virtual environments focused on verifying basic algorithmic performance. Xiao et al. [23] conducted research using a 480×480 px environment with the Box2D physics engine, while Du et al. [20] performed studies in a virtual environment incorporating static obstacles and coastline information. In addition, studies using electronic chart-based environments structured their data to enable computer simulation while preserving real environmental characteristics. Guo et al. [21] utilized electronic charts quantified into 400×350 grids, while Zhen et al. [31] converted actual port data from Zhoushan Port and Hainan Port into electronic chart format.

Notably, the selection of geographical scope was closely related to each study's objectives. Research targeting actual maritime areas showed strengths in validating applicability within specific operational environments but faced difficulties in conducting repeated experiments across various scenarios. Specifically, studies by Gao et al. [13] in Bohai Bay and Kim et al. [8] in Busan Port accurately reflected actual maritime characteristics but struggled to control various variables such as weather conditions and maritime traffic volume changes.

Conversely, studies utilizing simulation environments offered the advantage of enabling repeated experiments under various conditions. Most notably, the research by Xiao et al. [23] and Du et al. [20] allowed systematic verification of algorithmic stability and scalability. However, these simulation-based studies had limitations in perfectly replicating the complex environmental elements of actual maritime areas. Although studies by Guo et al. [21] and Zhen et al. [31] using electronic chart-based environments attempted to reduce this gap, they still faced constraints in reflecting real-time maritime traffic situations and weather conditions.

2.2 Algorithmic Approaches

The algorithmic approaches in previous studies can be broadly categorized into single reinforcement learning-based approaches and hybrid approaches. Single reinforcement learning approaches directly applied algorithms such as DQN, Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO), while hybrid approaches combined reinforcement learning algorithms with existing path planning methods or control techniques.

In particular, DQN was the most widely used algorithm in single reinforcement learning approaches. Gao et al. [13] proposed an improved DQN combining k-means clustering and prioritized experience replay, which demonstrated superior performance compared to conventional Q-learning, standard DQN, Double DQN, and Dueling DQN. Yuan et al. [14] developed a DQN with modified state space, action space, and reward functions for ferry crossing behavior, which was validated through comparison with actual ferry trajectories. Additionally, Guo et al. [21] enhanced coastal vessel path planning performance by applying optimized reward functions to DQN, including potential energy rewards, target area rewards, and risk zone penalties.

Furthermore, more advanced forms of reinforcement learning algorithms were actively studied. Xiao et al. Xiao et al. [23] proposed a beta policy-based distributed sampling PPO, showing improved results in terms of planning time and path length compared to conventional A-STAR, IDA-STAR, and Dijkstra algorithms. Chen et al. [28] developed a multi-agent Deep Reinforcement Learning (DRL) to solve cooperative collision avoidance problems between vessels in narrow channels. Notably, this study demonstrated the possibility of efficient path planning while complying with International Regulations for Preventing Collisions at Sea (COLREGS) regulations.

Regarding hybrid approaches, researchers attempted to combine reinforcement learning with existing methodologies to leverage their respective advantages. Xu et al. [12] proposed Modified Artificial Potential Field-Deep Q-Network (MAPF-DQN), combining Artificial Potential Field (APF) with DQN. Specifically, this method enabled safer and more efficient path planning by combining APF's local obstacle avoidance capabilities with DQN's global path optimization abilities. Du et al. [20] integrated Long Short-Term Memory (LSTM) and optimized reward functions into DDPG, along with the Douglas-Peucker algorithm to improve path smoothing. Wang et al. [26] proposed a hybrid bi-level planning method combining improved Particle Swarm Optimization (PSO) for global path planning with enhanced APF for local obstacle avoidance.

Of particular significance were approaches related to collision avoidance. Li et al. [25] integrated APF-enhanced reward functions and COLREGS-based collision avoidance zones into DQN. Chun et al. [24] combined quantitative collision risk assessment using vessel domain and Closest Point of Approach (CPA) with PPO-based DRL. These studies effectively demonstrated methods for incorporating safety and regulatory compliance into the reinforcement learning framework.

Furthermore, some studies proposed post-processing methods to enhance path practicality. Guo et al. [21] and Lee and Kim [22] used the Douglas-Peucker algorithm to optimize path waypoints. This approach proved effective in simplifying calculated paths for easier vessel following while maintaining the essential characteristics of the original route.

Based on the analysis of these previous studies, we can present the distinctiveness and contributions of this research as follows:

First, while most previous studies focused solely on route planning, they did not address practical port operation issues such as anchorage allocation. In contrast, our study proposes realistic and practical solutions by explicitly considering Busan Port's actual anchorage operation rules and vessel size-based anchorage assignment criteria.

Second, previous studies typically validated algorithmic performance using data from limited time periods. However, our study utilized actual Automatic Identification System (AIS) data over a seven-day period from June 3 to June 9, 2023, verifying algorithmic performance under various temporal and operational conditions. In particular, by analyzing real vessel data, we more reliably demonstrated the algorithm's practical applicability.

Third, existing studies largely simplified vessels' physical dimensions to a single point. In contrast, our study presents realistic anchorage space modeling by expanding grids based on vessels' actual Length Overall (LOA) and considering bow direction changes according to wind direction. This approach significantly contributes to efficient utilization and safety assurance of anchorage areas.

Fourth, while previous studies mainly considered theoretical safety distances, our study applied differentiated safety distances for each anchorage area, reflecting actual operational rules of Busan Vessel Traffic Service (VTS). Specifically, we implemented graduated safety distances ranging from 120 m to 900 m for anchorages N-1 through N-5, proposing more realistic anchorage operation methods.

Fifth, while most previous studies focused solely on improving reinforcement learning algorithm performance, our study attempted a practical approach considering integration with actual VTS systems.

These distinctive features demonstrate that our research goes beyond theoretical approaches to provide practical solutions directly applicable to actual port operations. In particular, our comprehensive approach to anchorage operation issues, based on real data and operational rules, represents a unique contribution of this study.

Through this analysis of previous research, we have identified current trends in reinforcement learning-based vessel path planning and our study's distinctiveness. Building on this theoretical foundation, the next section explains the specific implementation methodology of our proposed reinforcement learning-based anchorage allocation system.

3. Simulation Environment

This section provides a detailed explanation of the realistic simulation environment design for reinforcement learning agent training. This environment, which was developed based on the actual topography and anchorage information around Busan Port, incorporates various elements for safe navigation and efficient anchorage allocation.

All simulations were performed in a general computing environment, with calculations conducted on a system with sufficient computational capabilities. The hardware specifications used are shown in Table 2.

Table 2 Experimental platform and environment

Category	Details
CPU	AMD Ryzen 5 2600 Six-Core Processor 3.40 GHz
GPU	NVIDIA Geforce RTX 3060 Dual OC D6 12GB
RAM	16GB
Language	Python 3.8.18
Operating system	Windows 10 x64
Deep learning Framework	Tensorflow 2.10.0

3.1 Modeling of Busan Port Environment

To maximize the reflection of actual navigation conditions, the simulation environment was constructed based on nautical chart images of Busan Port. These charts are based on the World Geodetic System 1984 (WGS84) coordinate system, which expresses locations in terms of latitude and longitude. The spatial scope and anchorage classification used in the simulation environment are shown in Figure 1.

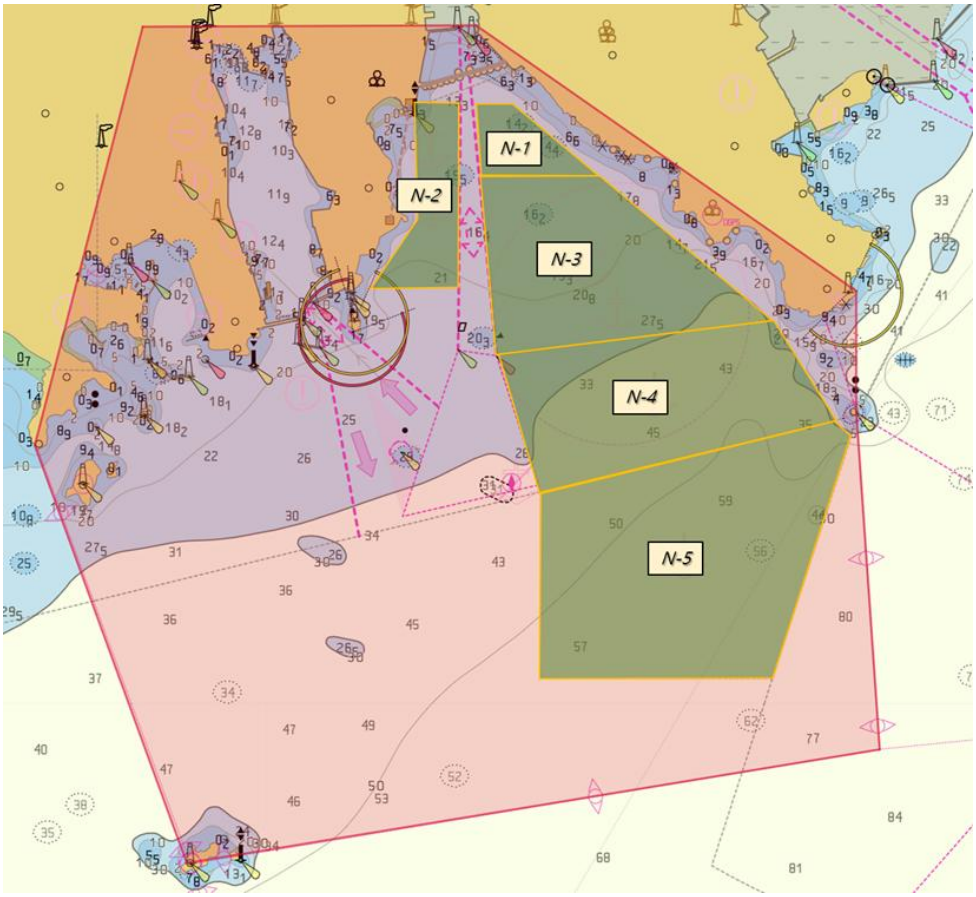


Fig. 1 Spatial scope and anchorage classification of Busan Port used in simulation environment

In Figure 1, the red area indicates the spatial scope used in this study. Additionally, the green areas represent the anchorage zones, covering a total area of approximately 29 km². The anchorage is classified into five zones from N-1 to N-5. The area and characteristics of each anchorage zone, as specified in the Rules on Navigation of Busan Port published by Busan Regional Office of Oceans and Fisheries, are presented in Table 3 [32].

Table 3 Classification and Characteristics of Anchorage Areas in Busan Port

Anchorage	Area (km^2)	Target Vessels (Gross Tonnage)	Minimum Safety Distance Between Vessels (m)	Maximum Number of Vessels
N-1	1.13	~ 1,000 t	120	8
N-2	1.77	1,000 ~ 2,999 t	360	7
N-3	6.80	3,000 ~ 9,999 t	540	18
N-4	7.27	10,000 ~ 29,999 t	720	8
N-5	12.02	30,000 t ~	900	7

Furthermore, to ensure accurate position estimation in the simulation environment, the coordinate system of the nautical charts was aligned with the simulation coordinate system. This alignment enabled the mapping of pixel coordinates from the chart images to actual geographical coordinates.

The simulation environment was implemented as a ShipEnv class using OpenAI's Gym library. This environment interacts with the reinforcement learning agent and incorporates the concepts of state, action, and reward. Specifically, the state represents various parameters such as the vessel's current position and direction to the target point, while actions define possible movement directions for the vessel. The reward quantifies how much the agent's actions contribute to achieving the target objective.

3.2 Anchorage Area Rules

Anchorage rules are essential considerations for ensuring safe navigation and anchoring of vessels. In this study, we implemented these rules in the simulation environment to enable agents to learn under realistic constraints. The following key elements were applied based on the rules used by Busan VTS:

1. Vessels are assigned to anchorage areas based on their gross tonnage. This implementation follows the 'Rules on Navigation of Busan Port' [32], allocating appropriate anchorage areas according to vessel tonnage. Specifically, larger vessels are assigned to larger anchorage areas to ensure vessel stability and efficient use of port facilities.

2. When arriving at their destination, vessels must maintain minimum safety distances from maritime obstacles and other vessels. Although there are no standardized regulations, this can be applied according to each port's VTS rules. In this study, we applied Busan Port's 'Minimum Safety Distance Rules for Anchoring' (Table 4). These safety distances vary depending on vessel size and navigation conditions, thereby minimizing collision risks.

3. During navigation toward the destination, vessels must maintain minimum safety distances from maritime obstacles and other vessels. This requirement applies more relaxed rules compared to arrival conditions, implementing Busan VTS's navigation safety distance of 100 m. This approach ensures that vessels maintain safety distances while efficiently approaching their destinations.

4. To ensure safety in boundary zones between anchorage areas, minimum safety distances from each anchorage boundary line were defined. These distances are based on Busan VTS safety regulations and are differentially applied considering anchorage scale and vessel size capacity. Specifically, minimum safety distances were set at 180 m for N-1, 240 m for N-2, 300 m for N-3, 360 m for N-4, and 400 m for N-5 anchorage areas. For conservative safety measures, the larger safety distance value was applied at boundaries

between adjacent anchorage areas. For example, 360 m was applied to the boundary between N-3 and N-4, and 400 m to the boundary between N-4 and N-5.

5. Since certain areas are restricted or prohibited from navigation, the agent was designed to avoid these zones. In this experiment, shallow water areas, maritime obstacles, land masses, and port entry/exit channels were designated as no-navigation zones.

These restrictions are strictly enforced within the simulation environment, with penalties applied if the agent violates these zones.

By meticulously incorporating these Busan Port anchorage rules into the simulation environment, we enabled the reinforcement learning agent to learn safe and efficient routes while considering various constraints that may be encountered in actual navigation situations.

3.3 Grid-Based Simulation Framework

To enhance the accuracy and efficiency of the simulation environment, we subdivided the waters around Busan Port into grids of 60 m × 60 m size. This grid division was implemented to ensure spatial precision considering both the nautical chart image resolution and actual geographical distances.

To divide the simulation area into an $N_{\text{lat}} \times N_{\text{lon}}$ sized grid, we first established the latitude and longitude ranges for the entire maritime area. Each grid cell was set to a fixed size of 60 m × 60 m, enabling uniform division of the area. To generate the grid, we calculated Δlat and Δlon, which represent the latitude and longitude ranges divided by the number of rows and columns in the grid:

$$\Delta\text{lat} = \frac{\text{lat}_{\text{max}} - \text{lat}_{\text{min}}}{N_{\text{lat}} - 1} \tag{1}$$

$$\Delta\text{lon} = \frac{\text{lon}_{\text{max}} - \text{lon}_{\text{min}}}{N_{\text{lon}} - 1} \tag{2}$$

Here, lat_{min} and lat_{max} represent the minimum and maximum latitudes of the simulation area, respectively, while lon_{min} and lon_{max} denote the minimum and maximum longitudes. Additionally, N_{lat} and N_{lon} represent the number of rows and columns in the grid, respectively.

The center latitude and longitude of each grid cell (i, j) are calculated using the following equations:

$$\text{lat}_{i,j} = \text{lat}_{\text{min}} + i \times \Delta\text{lat} \tag{3}$$

$$\text{lon}_{i,j} = \text{lon}_{\text{min}} + j \times \Delta\text{lon} \tag{4}$$

In these equations, i represents the row index and j represents the column index, where $i = 0,1,2,\ldots,N_{\text{lat}} - 1$ and $j = 0,1,2,\ldots,N_{\text{lon}} - 1$.

For each grid cell (i, j), we generate a safety distance map $D_{\text{safe}}(i, j)$ by calculating the minimum safety distance from surrounding obstacles and vessels. To accomplish this, we utilize the Haversine formula to calculate distances between two points.

The Haversine formula, which calculates the shortest distance between two points on a sphere, is expressed as follows:

$$d = 2R\arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \tag{5}$$

Here, R represents the Earth's mean radius, which we calculate as 6,371 km. ϕ_1 and ϕ_2 represent the latitudes of the two points in radians, while λ_1 and λ_2 denote their longitudes, also in radians.

Using this formula, we define $D_{\text{safe}}(i, j)$ by calculating the minimum safety distance between each grid cell (i, j) and surrounding obstacles or vessels as follows:

$$D_{\text{safe}}(i,j) = min\{d_{\text{haversine}}((\text{lat}_{i,j}, \text{lon}_{i,j}), (\text{lat}_{\text{obs}}, \text{lon}_{\text{obs}}))\} \quad (6)$$

where $(\text{lat}_{i,j}, \text{lon}_{i,j})$ represents the central coordinates of grid cell (i,j), and $(\text{lat}_{\text{obs}}, \text{lon}_{\text{obs}})$ denotes the coordinates of obstacles or anchored vessels. Specifically, $D_{\text{safe}}(i,j)$ is set as the distance to the nearest obstacle or anchored vessel from each grid cell, thereby ensuring that vessels maintain safety distances when utilizing these grid cells.

3.4 Grid Extension for Anchored Vessels

In this simulation, we extended the grid area representing anchored vessels beyond the single grid cell representation to account for the actual vessel dimensions, while considering wind direction. This extension was deemed necessary for two critical reasons: to reflect the characteristic positioning of AIS antennae at the stern of vessels and to secure adequate safety zones that account for the physical dimensions of anchored vessels. The grid extension was implemented exclusively along the vessel's length dimension. Notably, analysis of vessel data from June 3 to June 9, 2023, revealed that beam-wise extension was unnecessary, as the maximum vessel beam width of 58 m remained below the standard grid cell dimension of 60 m. The extent of longitudinal extension was determined based on the maximum vessel length recorded for each anchorage area during the same period.

The grid extension size for anchored vessels in each anchorage area was calculated using the following equation, which accommodates both linear (east, west, north, south) and diagonal (northeast, northwest, southeast, southwest) wind directions:

$$Grid_{\text{ext}} = \left\lceil \frac{LOA_{\max}}{Grid_{\text{side}}} \right\rceil \quad (7)$$

where $Grid_{\text{ext}}$ represents the total number of extended grid cells, $LOA_{\max}$ denotes the maximum vessel length (in meters) in the respective anchorage area, and $Grid_{\text{side}}$ represents either the length of a grid cell side (60 m) for linear wind directions or the diagonal length (approximately 84.85 m) for diagonal wind directions. The ceiling function $\lceil x \rceil$ returns the smallest integer greater than or equal to x. Applying Eq. (7) yielded the following grid extension results for each anchorage area:

1. For linear wind directions (east, west, north, south):

1.1 N-1 anchorage (maximum vessel length 100 m): $Grid_{\text{ext}} = \left\lceil \frac{100}{60} \right\rceil = 2$ (1 *grid extension*)

1.2 N-2 anchorage (maximum vessel length 110 m): $Grid_{\text{ext}} = \left\lceil \frac{110}{60} \right\rceil = 2$ (1 *grid extension*)

1.3 N-3 anchorage (maximum vessel length 142 m): $Grid_{\text{ext}} = \left\lceil \frac{142}{60} \right\rceil = 3$ (2 *grid extension*)

1.4 N-4 anchorage (maximum vessel length 179 m): $Grid_{\text{ext}} = \left\lceil \frac{179}{60} \right\rceil = 3$ (2 *grid extension*)

1.5 N-5 anchorage (maximum vessel length 399 m): $Grid_{\text{ext}} = \left\lceil \frac{399}{60} \right\rceil = 7$ (6 *grid extension*)

2. For diagonal wind directions (northeast, northwest, southeast, southwest):

2.1 N-1 anchorage (maximum vessel length 100 m): $Grid_{\text{ext}} = \left\lceil \frac{100}{84.85} \right\rceil = 2$ (1 *grid extension*)

2.2 N-2 anchorage (maximum vessel length 110 m): $Grid_{\text{ext}} = \left\lceil \frac{110}{84.85} \right\rceil = 2$ (1 *grid extension)*

2.3 N-3 anchorage (maximum vessel length 142 m): $Grid_{\text{ext}} = \left\lceil \frac{142}{84.85} \right\rceil = 2$ (1 *grid extension)*

2.4 N-4 anchorage (maximum vessel length 179 m): $Grid_{\text{ext}} = \left\lceil \frac{179}{84.85} \right\rceil = 3$ (2 *grid extension*)

2.5 N-5 anchorage (maximum vessel length 399 m): $Grid_{\text{ext}} = \left\lceil \frac{399}{84.85} \right\rceil = 5$ (4 *grid extension*)

Figure 2 provides a visual representation of the extended grids for each anchorage area under easterly wind conditions, demonstrating the practical implementation of these calculations.

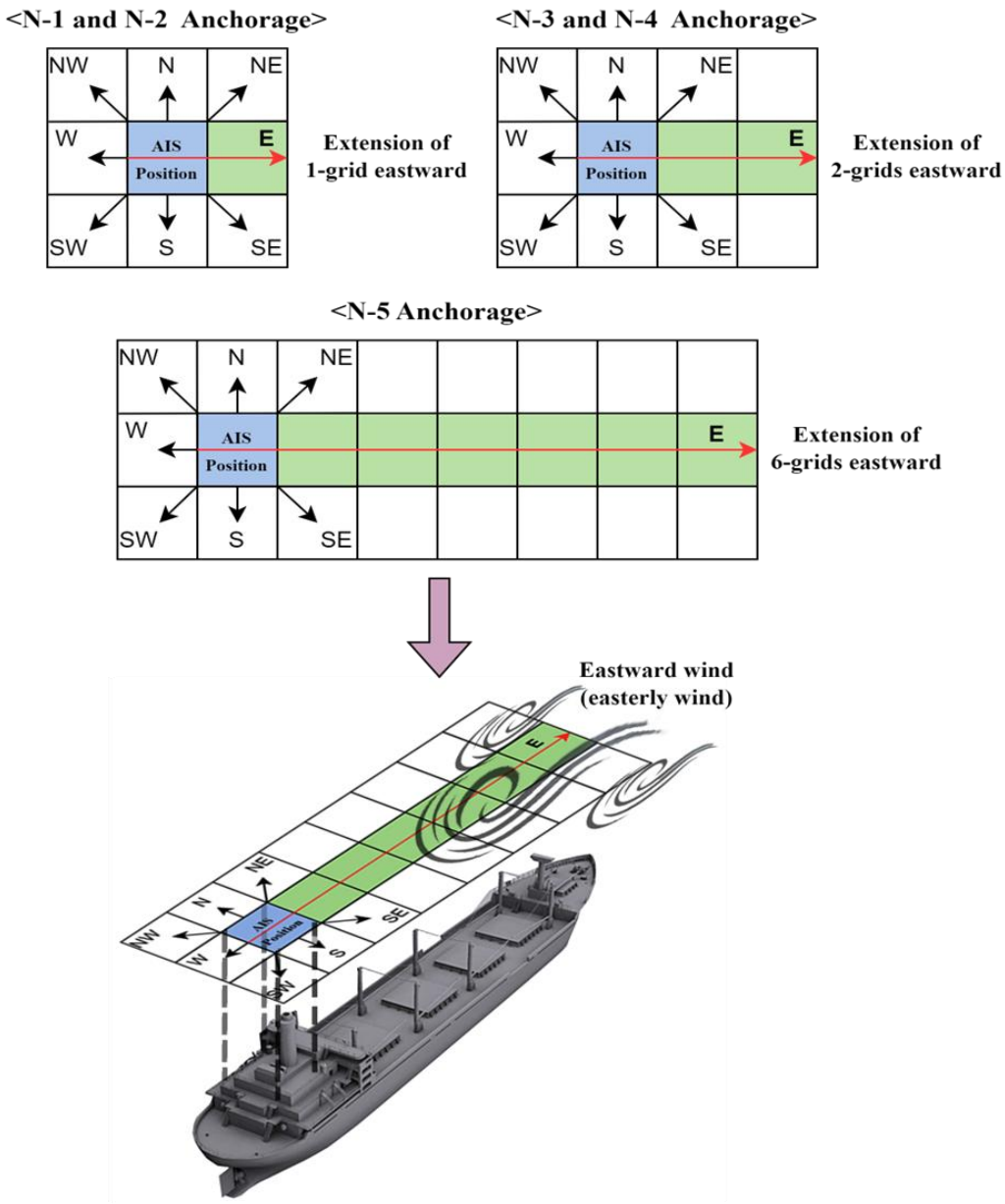


Fig. 2 Grid extension patterns for anchored vessels in each anchorage area under easterly wind conditions

In Figure 2, light blue cells indicate AIS antenna positions at the stern, while green cells represent the extended areas, with the final cell marking the bow position. The grid extension is oriented eastward to reflect the prevailing wind conditions, demonstrating how vessels typically align their bows with the wind direction. This extension methodology enhances simulation realism by incorporating both vessel physical dimensions and environmental factors, enabling safer anchorage allocation. In particular, the application of different extension sizes for each anchorage area ensures adequate space for the safe anchoring of vessels of varying dimensions.

3.5 Target Position Optimization

To enable objective comparison between the reinforcement learning agent's performance and actual vessel paths, we established a fundamental principle of setting the agent's target position (anchoring position) identical to the vessel's final anchoring position. However, since maintaining safety distances during anchoring is crucial for preventing serious maritime accidents such as vessel collisions, adjustments become necessary when actual vessel anchoring positions violate existing safety distance requirements. To address this challenge,

we propose an optimization method that identifies the nearest target position that satisfies all safety requirements while maintaining proximity to the actual anchoring position.

To evaluate safety with respect to anchored vessels at any position $q = (\text{lat}, \text{lon})$, we define a safety distance function $D_{\text{safe}}(q)$ and a violation indicator function $V(q)$ as follows:

$$D_{\text{safe}}(q) = \min_{i \in A}\{d_{\text{haversine}}(q, q_i)\}, \qquad V(q) = \begin{cases} 1 & \text{if } D_{\text{safe}}(q) < D_{\text{min}} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where A represents the set of positions of all currently anchored vessels, q_i denotes the position of the ith anchored vessel, $d_{\text{haversine}}$ is the Haversine distance defined in Eq. (5), and D_{min} represents the minimum safety distance requirement for the respective anchorage area, as specified in Table 4.

Based on these safety evaluation functions defined in Eq. (8), we consider three constraints for assessing the suitability of a target position q_{actual}:

$$D_{\text{safe}}(q) \geq D_{\text{min}}, \qquad q \in R_{\text{anchorage}}, \qquad N(q) = 1 \tag{9}$$

where $R_{\text{anchorage}}$ represents the valid region of the respective anchorage area, and $N(q)$ is a binary function indicating navigability at position q, returning 1 for navigable positions and 0 otherwise.

When any of the three conditions in Eq. (9) is not satisfied, we search for a new target position through the following optimization problem:

$$\begin{aligned} &\underset{q \in R_{\text{anchorage}},\ N(q)=1}{\text{minimize}} \; d_{\text{haversine}}(q, q_{\text{actual}}) \\ &\text{subject to: } D_{\text{safe}}(q) \geq D_{\text{min}} \end{aligned} \tag{10}$$

This optimization problem aims to find a position q that is closest to the actual anchoring position q_{actual} while satisfying all safety conditions. Given that this problem must simultaneously satisfy various safety-related constraints, an efficient search method is required. In this study, we employed an appropriate search algorithm to determine the optimal target position.

For example, if the distance between a vessel's actual anchoring position and the nearest anchored vessel in anchorage N-3 is 450 m, this violates the minimum safety distance requirement of 540 m (Table 4) for that anchorage area. In this case, the optimization algorithm is applied to determine a new target position that is closest to the original while satisfying all constraints.

The proposed target position optimization method offers several key features. Through its objective function that minimizes distance from the actual anchoring position, it maximally preserves the VTS operator's original intent. Furthermore, it fundamentally prevents collision risks during anchoring by strictly maintaining minimum safe distances from other anchored vessels. Additionally, by comprehensively considering various operational constraints such as anchorage boundaries, it provides a foundation for the reinforcement learning agent to plan realistic and safe routes.

As a result, this method is expected to enhance overall port operational efficiency while significantly reducing safety incident risks.

Throughout Section 3, we have provided a detailed examination of the simulation environment design where the reinforcement learning agent learns and operates. Building upon this simulation framework, Section 4 presents the design and implementation of the reinforcement learning model for optimal route planning.

4. Reinforcement Learning Model

This section provides a detailed explanation of the design and implementation of the reinforcement learning model for optimal ship route planning. Reinforcement learning is a method where agents learn optimal policies through trial-and-error interactions with their environment, making it particularly effective for complex decision-making problems. In this study, we implemented a reinforcement learning model based

on the Deep Q-Network (DQN) algorithm. One of the key advantages of DQN is its ability to enable effective learning even in high-dimensional state spaces by approximating state-action value functions using deep neural networks. This section first describes the definitions of the core elements of the reinforcement learning model: state space, action space, and reward function. Subsequently, we detail the implementation specifications and learning process of the DQN algorithm.

4.1 *State Space Definition*

The state information provided to the reinforcement learning agent must include the vessel's current position, distance and direction to the target point, and safety distances from the surrounding environment. This comprehensive state information plays a crucial role in enabling the agent to accurately understand its environment and learn optimal routes. To achieve this, we defined the following state variables. First, we normalized the position information. The vessel's current grid indices (i, j) were normalized to the range [0, 1] by dividing by the total number of rows and columns in the grid, as follows:

$$s_1 = \frac{i}{N_{\text{lat}} - 1}, \qquad s_2 = \frac{j}{N_{\text{lon}} - 1} \tag{11}$$

where s_1 and s_2 represent the normalized position information in the latitude and longitude directions, respectively.

Second, we normalized the distance and direction to the target point. For the target point, the Haversine distance from the current position to the target point was calculated and normalized by dividing by the maximum possible distance $d_{\max}$ within the simulation area as follows:

$$d_{\text{goal}} = d_{\text{haversine}}\big((\text{lat}_{i,j}, \text{lon}_{i,j}), (\text{lat}_{\text{goal}}, \text{lon}_{\text{goal}})\big) \tag{12a}$$

$$s_3 = \frac{d_{\text{goal}}}{d_{\max}} \tag{12b}$$

where d_{goal} represents the distance from the current position to the target point, and $d_{\max}$ denotes the maximum possible distance within the simulation area.

Regarding the direction to the target point, we calculated the azimuth angle from the current position to the target point and normalized it to the range [0, 1] using the following equations:

$$\begin{aligned}\theta_{\text{goal}} = \arctan\ 2\big(&\sin(\Delta\lambda)\cos(\phi_{\text{goal}}), \cos(\phi_{i,j})\sin(\phi_{\text{goal}}) \\ &- \sin(\phi_{i,j})\cos(\phi_{\text{goal}})\cos(\Delta\lambda)\big)\end{aligned} \tag{13a}$$

$$s_4 = \frac{\theta_{\text{goal}} + \pi}{2\pi} \tag{13b}$$

where $\phi_{i,j}$ represents the latitude of the current position, ϕ_{goal} is the latitude of the target point, $\lambda_{i,j}$ is the longitude of the current position, λ_{goal} is the longitude of the target point, and $\Delta\lambda$ denotes $\lambda_{\text{goal}} - \lambda_{i,j}$. All angular measurements are expressed in radians.

Third, the safety distance from the surrounding environment was normalized by dividing $D_{\text{safe}}(i, j)$ derived from Eq. (6) by the maximum safety distance $D_{\max}$:

$$s_5 = \frac{D_{\text{safe}}(i, j)}{D_{\max}} \tag{14}$$

where $D_{\text{safe}}(i, j)$ represents the safety distance at grid cell (i, j), and $D_{\max}$ denotes the maximum safety distance defined in the simulation environment.

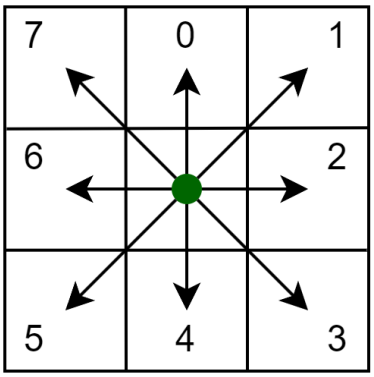
Therefore, the final state vector S is defined as:

$$S = [s_1, s_2, s_3, s_4, s_5] \tag{15}$$

Each component of the state vector represents a specific aspect of the vessel's navigational state: s_1 and s_2 represent the normalized position in latitude and longitude respectively, s_3 indicates the normalized distance to the target point, s_4 represents the normalized direction angle to the target point, and s_5 denotes the normalized safety distance from surrounding obstacles and vessels.

4.2 Action Space Definition

The action space is defined as the set of all possible actions that the agent can select. In this study, we configured the vessel to move in eight directions. These actions consist of cardinal movements (up, down, left, right) and diagonal movements (up-left, up-right, down-left, down-right), with each action represented by an index $a \in \{0,1,\ldots,7\}$. The grid movements corresponding to each action are shown in Figure 3.



$a = 0: (\Delta i, \Delta j) = (-1,0)$ (N)
$a = 1: (\Delta i, \Delta j) = (-1,1)$ (NE)
$a = 2: (\Delta i, \Delta j) = (0,1)$ (E)
$a = 3: (\Delta i, \Delta j) = (1,1)$ (SE)
$a = 4: (\Delta i, \Delta j) = (1,0)$ (S)
$a = 5: (\Delta i, \Delta j) = (1,-1)$ (SW)
$a = 6: (\Delta i, \Delta j) = (0,-1)$ (W)
$a = 7: (\Delta i, \Delta j) = (-1,-1)$ (NW)

Fig. 3 Definition of Eight-Directional Action Space for Grid-Based Exploration

Here, a represents the action index, while Δi and Δj denote the changes in grid indices according to action a. This definition of action space provides a foundation for the reinforcement learning algorithm to effectively learn vessel movements while enabling the simulation of various movements that may occur in actual navigation situations. Through this action space, the agent selects the vessel's movement direction and uses it as a basis for exploring optimal routes.

4.3 *Dynamic Modeling of States and Actions*

The reinforcement learning agent selects an action a_t in the current state S_t and transitions to a new state S_{t+1}. In this section, we mathematically define the dynamic model of states and actions related to vessel movement in the grid-based simulation environment.

First, the state vector at time t can be defined as follows, referring to the definition in Eq. (11):

$$S_t = [s_{1,t}, s_{2,t}, s_{3,t}, s_{4,t}, s_{5,t}] \tag{16}$$

where $s_{1,t}$ and $s_{2,t}$ represent the normalized position information as defined in Eq. (7):

$$s_{1,t} = \frac{i_t}{N_{\text{lat}} - 1}, \qquad s_{2,t} = \frac{j_t}{N_{\text{lon}} - 1} \tag{17a}$$

Here, i_t and j_t are the grid cell indices at time t, and N_{lat} and N_{lon} represent the number of rows and columns in the grid, respectively. For example, when $N_{\text{lat}} = 100$ and $N_{\text{lon}} = 100$, if $(i_t, j_t) = (50,50)$, then $s_{1,t} = 0.505$ and $s_{2,t} = 0.505$ after normalization.

$s_{3,t}$ represents the normalized distance to the target point as defined in Eq. (8b):

$$s_{3,t} = \frac{d_{\text{goal},t}}{d_{\max}} \tag{17b}$$

where $d_{\text{goal},t}$ is the distance from the current position to the target point, and $d_{\max}$ represents the maximum possible distance within the simulation area.

$s_{4,t}$ is the normalized azimuth angle as defined in Eq. (9b):

$$s_{4,t} = \frac{\theta_{\text{goal},t} + \pi}{2\pi} \tag{17c}$$

where $\theta_{\text{goal},t}$ is the azimuth angle from the current position to the target point, measured in radians. For example, when $\theta_{\text{goal},t} = 2.356$, $s_{4,t}$ is normalized to approximately 0.833.

$s_{5,t}$ is the normalized safety distance as defined in Eq. (10):

$$s_{5,t} = \frac{D_{\text{safe}}(i_t, j_t)}{D_{\max}} \tag{17d}$$

where $D_{\text{safe}}(i_t, j_t)$ represents the safety distance at the current position, and $D_{\max}$ is the maximum safety distance. In this study, we set the maximum safety distance $D_{\max}$ to 900 m (As shown in Table 4, this distance is the safety distance for anchorage N-5 and represents the largest maximum safety distance used in the study). For example, if $D_{\text{safe}}(i_t, j_t) = 450$ m, then $s_{5,t}$ is normalized to 0.5.

Next, when the agent selects an action $a_t \in \{0,1,\dots,7\}$ at time t, the new position (i_{t+1}, j_{t+1}) is determined by the grid index changes $\Delta i(a_t)$ and $\Delta j(a_t)$ as defined in Fig. 4:

$$i_{t+1} = i_t + \Delta i(a_t) \tag{18a}$$

$$j_{t+1} = j_t + \Delta j(a_t) \tag{18b}$$

where $\Delta i(a_t)$ and $\Delta j(a_t)$ represent the changes in grid indices according to action a_t, as defined in Fig. 4. For example, when action index $a_t = 0$ indicates northward movement, $\Delta i(a_t) = -1$ and $\Delta j(a_t) = 0$.

The new position (i_{t+1}, j_{t+1}) must satisfy several conditions. First, the vessel must remain within the simulation grid boundaries:

$$0 \le i_{t+1} < N_{\text{lat}}, \qquad 0 \le j_{t+1} < N_{\text{lon}} \tag{19a}$$

Second, the vessel must navigate within navigable areas:

$$\text{navigable}(i_{t+1}, j_{t+1}) = \text{True} \tag{19b}$$

where $\text{navigable}(i_{t+1}, j_{t+1})$ is a function indicating whether grid cell (i_{t+1}, j_{t+1}) is navigable, returning $False$ for cells containing obstacles or no-navigation zones. This condition must be satisfied for the vessel to move to that position. For example, port entry/exit channels or shallow water areas are designated as no-navigation zones, and movement is restricted when entering these areas.

Third, the vessel must maintain minimum safety distances at the new position (i_{t+1}, j_{t+1}). The safety distance $D_{\text{safe}}(i_{t+1}, j_{t+1})$ must satisfy:

$$D_{\text{safe}}(i_{t+1}, j_{t+1}) \ge D_{\min} \tag{19c}$$

where $D_{\text{safe}}(i_{t+1}, j_{t+1})$ represents the safety distance at the next position (i_{t+1}, j_{t+1}), and $D_{\min}$ is the minimum safety distance set according to each anchorage area's rules (Table 4). The safety distance requirements in our study are categorized into two distinct phases: First, during navigation, vessels must maintain a minimum safety distance of 100 m from obstacles and other vessels to ensure safe passage. Second, at the final destination (anchoring position), more stringent safety distances are applied according to each anchorage area's specific requirements. For example, in anchorage N-3, the minimum safety distance from other vessels is set to 540 m when anchoring. These dual-phase safety requirements act as constraints to minimize collision risks throughout the entire journey while ensuring proper spacing at anchor. Therefore,

when a vessel moves to a new position, it must verify that the area is navigable and that the appropriate safety distances for the current phase are maintained.

Only when these conditions are satisfied can the vessel move to the new position, and the new state vector S_{t+1} is updated as follows. First, the position information is updated by normalizing the new grid indices:

$$s_{1,t+1} = \frac{i_{t+1}}{N_{\text{lat}} - 1}, \qquad s_{2,t+1} = \frac{j_{t+1}}{N_{\text{lon}} - 1} \tag{20}$$

Next, we calculate and normalize the distance $d_{\text{goal},t+1}$ from the new position to the target point:

$$d_{\text{goal},t+1} = d_{\text{haversine}}\big((\text{lat}_{i_{t+1},j_{t+1}}, \text{lon}_{i_{t+1},j_{t+1}}), (\text{lat}_{\text{goal}}, \text{lon}_{\text{goal}})\big) \tag{21a}$$

$$s_{3,t+1} = \frac{d_{\text{goal},t+1}}{d_{\max}} \tag{21b}$$

where $d_{\text{haversine}}$ is the distance calculation function using the Haversine formula defined in Eq. (5), and $\text{lat}_{i_{t+1},j_{t+1}}$ and $\text{lon}_{i_{t+1},j_{t+1}}$ represent the latitude and longitude of the new position as defined in Eqs. (3) and (4). Additionally, we calculate and normalize the azimuth angle $\theta_{\text{goal},t+1}$ from the new position to the target point:

$$\begin{aligned}\theta_{\text{goal},t+1} = \arctan2\big(&\sin(\Delta\lambda_{t+1})\cos(\phi_{\text{goal}}), \cos(\phi_{i_{t+1},j_{t+1}})\sin(\phi_{\text{goal}}) \\ &- \sin(\phi_{i_{t+1},j_{t+1}})\cos(\phi_{\text{goal}})\cos(\Delta\lambda_{t+1})\big)\end{aligned} \tag{22a}$$

$$s_{4,t+1} = \frac{\theta_{\text{goal},t+1} + \pi}{2\pi} \tag{22b}$$

where $\phi_{i_{t+1},j_{t+1}}$ represents the latitude of the new position, ϕ_{goal} is the latitude of the target point, and $\Delta\lambda_{t+1}$ denotes the longitude difference. Finally, we calculate $s_{5,t+1}$ by normalizing the safety distance $D_{\text{safe}}(i_{t+1}, j_{t+1})$ at the new position by the maximum safety distance:

$$s_{5,t+1} = \frac{D_{\text{safe}}(i_{t+1}, j_{t+1})}{D_{\max}} \tag{23}$$

Using these updated values, the new state vector S_{t+1} is defined as:

$$S_{t+1} = [s_{1,t+1}, s_{2,t+1}, s_{3,t+1}, s_{4,t+1}, s_{5,t+1}] \tag{24}$$

Through this dynamic modeling, the agent learns to select appropriate actions in the current state and understands how these actions affect the environment. The state S_t consists of information such as the vessel's position, distance and direction to the target point, and safety distances from surrounding obstacles, and the agent selects one of the possible actions based on this state. The selected action changes the vessel's position on the grid, and a new state S_{t+1} is calculated. The new state provides information about whether the agent's movement has brought it closer to the target point and maintained safe routes, compared to the previous state. The agent experiences these state changes repeatedly and proceeds with learning to find optimal routes.

4.4 *Reward Function Design*

In reinforcement learning, the reward function serves as a fundamental element that determines the learning direction of the agent. To optimize vessel navigation while ensuring safety, this study employs the concept of Potential Field to design the reward function. The APF method, originally proposed by Khatib [33] for robot motion planning and later applied in maritime research such as [20, 21, 25, 26].

In the classical APF theory, the total potential energy is expressed as the sum of attractive and repulsive components:

$$U_{\text{total}}(q) = U_{\text{att}}(q) + U_{\text{rep}}(q) \tag{25a}$$

where $U_{\text{att}}(q)$ represents the attractive potential pulling the agent toward the goal, and $U_{\text{rep}}(q)$ denotes the repulsive potential pushing the agent away from obstacles.

Following standard reinforcement learning notation, the agent in state S_t at time t selects action a_t and transitions to a new state S_{t+1}. The reward r_t for this transition is defined as:

$$\begin{aligned} r_t = (U(S_t) - U(S_{t+1})) \times w_{\text{potential}} + \left(d_{\text{prev}} - d_{\text{current}}\right) \times w_{\text{distance}} + R_{\text{goal}} \times \delta_{\text{goal}} \\ - R_{\text{penalty}} \times \delta_{\text{penalty}} \end{aligned} \tag{25b}$$

where $U(S)$ represents the potential energy at state S, $w_{\text{potential}}$ is the weighting factor for potential energy changes, d_{prev} and d_{current} denote the previous and current distances to the target point, respectively, and w_{distance} is the weighting factor for distance changes. Additionally, R_{goal} represents the reward for reaching the target point with δ_{goal} being a binary indicator (1 if the target is reached, 0 otherwise), while R_{penalty} denotes the penalty magnitude with δ_{penalty} being a binary indicator (1 if safety distance violations or restricted area entry occur, 0 otherwise).

For computational efficiency in our reinforcement learning framework, we implement a simplified version of the attractive potential function:

$$U(S) = \frac{1}{d_{\text{goal}} + 1} \tag{26}$$

where d_{goal} represents the distance to the target point. This formulation ensures that the potential energy decreases as the vessel approaches its target point, thereby providing increasingly positive rewards for successful navigation toward the destination. The repulsive potential from obstacles is implicitly handled through the penalty term $R_{\text{penalty}} \times \delta_{\text{penalty}}$, which creates steep potential barriers around safety-critical regions.

This combined approach integrates the theoretical foundations of APF with the practical requirements of reinforcement learning in maritime environments, resulting in a theoretically sound reward mechanism that effectively balances goal-seeking behavior with safety constraints.

The weighting factors in Eq. (25) were determined through systematic experimentation. We conducted a grid search approach, testing values of [1, 5, 10, 50, 100, 200] for each weight parameter. The final values ($w_{\text{potential}} = 100$, $w_{\text{distance}} = 10$, $R_{\text{goal}} = 100$, $R_{\text{penalty}} = 200$) were selected based on their ability to balance safety requirements with path efficiency. Specifically, $w_{\text{potential}} = 100$ effectively encouraged target-oriented navigation while maintaining obstacle avoidance capability. The $w_{\text{distance}} = 10$ provided sufficient influence on path optimization without overly sacrificing safety constraints. The penalty magnitude R_{penalty} was set to twice the goal achievement reward (R_{goal}) to ensure the agent prioritized safety rule compliance over merely reaching the destination. Through extensive testing, we found that R_{penalty} values below 200 occasionally resulted in the agent choosing unsafe shortcuts, while values above 300 led to overly conservative paths with unnecessary length increases.

Furthermore, the reward mechanism implements a comparative approach: if the agent moves closer to the target point compared to its previous state, it receives a positive reward; conversely, if it moves farther away, a penalty is applied. This dynamic reward structure is further enhanced by additional rewards when the agent reaches its target point, while penalties are imposed for safety distance violations or entry into restricted navigation zones. Through this comprehensive reward system, the agent receives rewards at each step and continuously improves its policy through the reinforcement learning algorithm. Of particular significance is the implementation of the potential field concept, which enables the agent to naturally navigate toward the target point while simultaneously learning to avoid obstacles and hazardous zones in its vicinity. This dual-objective learning approach ensures optimal path learning while maintaining safe navigation parameters.

Having established the reward function design, we now turn our attention to the implementation of the DQN algorithm, which effectively utilizes this reward structure to learn optimal navigation policies.

4.5 *Deep Q-Network Algorithm*

In this study, we employed the DQN algorithm to enable the agent to learn optimal policies. DQN approximates the state-action value function $Q(S, a)$ through deep neural networks, enabling effective learning even in high-dimensional state spaces.

The agent, at time t, is in state S_t and transitions to a new state S_{t+1} after selecting action a_t. The Q-value is updated according to the following equation:

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(S_{t+1}, a'; \theta^-) - Q(S_t, a_t; \theta) \right] \tag{27}$$

where α represents the learning rate, determining how quickly new information is incorporated into existing Q-values. r_t is the reward at time t, calculated by the reward function. γ is the discount factor, indicating the importance of future rewards relative to immediate ones. Additionally, a' is a new variable representing all possible actions available in the next state S_{t+1}. θ and θ^- denote the weights of the current Q-network and target Q-network, respectively.

The agent updates Q-values through Eq. (27) and learns optimal policies through this process. In this learning framework, the introduction of experience replay memory and target networks plays a crucial role in enhancing learning stability and efficiency.

Experience replay memory serves as a buffer that stores the agent's experiences $(S_t, a_t, r_t, S_{t+1}, \text{done})$ acquired through interactions with the environment, from which mini batches are randomly sampled. This approach reduces data correlation and improves learning efficiency by utilizing diverse and rich experiences. The target network θ^- is updated periodically with the weights θ of the current network, mitigating potential instabilities during Q-value updates.

The neural network architecture consists of input layers, multiple hidden layers, and output layers. Specifically, the input layer receives the state vector S_t, which includes information such as the agent's current position and bearing to the target point. The hidden layers comprise multiple neurons and employ ReLU (Rectified Linear Unit) as the non-linear activation function to learn complex patterns and features. The output layer produces $Q(S_t, a; \theta)$ for all possible actions a, which is utilized for the agent's policy decisions. This architectural design enables the agent to effectively evaluate the value of each action in any given state.

The key hyperparameters used in the learning process are summarized in Table 4.

Table 4 Hyperparameter Settings for the Reinforcement Learning Algorithm

Hyperparameter	Value	Description
Input Dimension	3	Number of input features (state space dimension)
Hidden Layers	[128, 128, 128]	Three hidden layers with 128 neurons each
Output Layer Size	8	Number of neurons in output layer (action space dimension)
Activation Function	ReLU	Activation function used between layers
Learning Rate	0.01	Learning rate applied during neural network weight updates
Discount Factor	0.9	Represents the importance of future rewards to current value
Initial Exploration Rate	1.0	Initial degree of random exploration by the agent
Minimum Exploration Rate	0.05	Minimum value of exploration rate
Exploration Rate Decay Rate	0.995	Rate at which the exploration rate decreases each episode
Mini-batch Size	256	Size of the mini-batch used during neural network training
Experience Replay Memory Size	10000	Maximum number of experiences stored in the replay buffer for training
Target Network Update Frequency	Every 100 episodes	Frequency at which the target network weights are synchronized
Potential Energy Weight ($w_{\text{potential}}$)	100	Weight for potential energy change in the reward function
Distance Change Weight (w_{distance})	10	Weight for distance change to the goal in the reward function
Goal Achievement Reward (R_{goal})	100	Reward given when the agent reaches the goal
Safety Violation Penalty (R_{penalty})	200	Penalty applied for violating safety distances

5. Experimental Data and Results

This section presents a detailed analysis of experiments and results using real vessel's AIS data from Busan Port to evaluate the performance of the proposed reinforcement learning-based anchorage allocation and optimal path planning system. Specifically, quantitative evaluations were conducted in the following key aspects:

1. Optimization performance evaluation through analysis of path length and reduction rate compared to actual paths
2. Safety assessment including violations of navigation safety distances, anchored vessel safety distances, and anchorage boundary violations
3. Evaluation of practical applicability through path simplification using the Douglas-Peucker algorithm.

The analysis of experimental results consists of three main parts. First, we describe the characteristics and selection criteria of the experimental dataset. Then, we present quantitative comparative analysis results between actual paths and optimized paths for each anchorage area. Finally, we discuss comprehensive evaluations of path optimization results, including safety rule violations, along with detailed analysis results considering the characteristics of each anchorage area.

5.1 Experimental Data

The experiments utilized actual AIS path data from 115 vessels that anchored in anchorage areas N-1 through N-5 of Busan Port during a seven-day period from June 3 to June 9, 2023. These vessels were distributed across anchorage areas as follows: 4 vessels in N-1, 22 vessels in N-2, 44 vessels in N-3, 26 vessels in N-4, and 19 vessels in N-5. For the experiments, one vessel was randomly selected from each anchorage area.

The initial position of the reinforcement learning agent was set identical to the starting position of the actual vessel path. This setting enables direct comparison with real operational scenarios, allowing for realistic evaluation of the model's performance. As for the target positions, we applied the target position optimization method described in Section 3.5. Specifically, while using the actual vessel's anchoring position as a reference point, in cases where this position violated safety distance requirements, our optimization algorithm determined a new position that was closest to the original while satisfying all safety conditions. This approach enables the selection of anchoring positions that both preserve the VTS operator's original intent and ensure safety requirements are met. The detailed information of the selected vessels is summarized in Table 5.

Table 5 Ship data used in experiments

Anchorage	MMSI	Gross tonnage	Ship type	Start time	Start point (Lat, Long)	End time	End point (Lat, Long)
N-1	2733*****	651	Fishing Vessel	2023-06-05 17:09:08	35.0019, 129.0961	2023-06-05 18:00:15	35.0683, 129.0359
N-2	4415*****	2,994	Tanker	2023-06-04 10:15:26	34.9882, 129.0525	2023-06-04 10:50:20	35.0641, 129.0288
N-3	4571*****	6,142	General Cargo	2023-06-09 08:54:40	35.0178, 129.0950	2023-06-09 09:55:29	35.0540, 129.0580
N-4	2736*****	16,949	General Cargo	2023-06-03 05:58:22	34.9863, 129.0390	2023-06-03 06:30:57	35.02879, 129.0474
N-5	3746*****	35,832	General Cargo	2023-06-08 07:13:47	34.9916, 129.0771	2023-06-08 07:42:02	35.0209, 129.0654

This data played a crucial role in model training and validation. Information for each vessel includes MMSI (Maritime Mobile Service Identity), gross tonnage, ship type, navigation start/end times, and departure/arrival position coordinates. As shown in the table, vessels selected for each anchorage area have appropriate tonnage ranges for their respective areas (from 651 t in N-1 to 35,832 t in N-5) and represent various vessel types (fishing vessels, tankers, general cargo ships). This diversity allowed the reinforcement learning model to adapt to different vessel characteristics and operational patterns. Additionally, the actual departure and arrival time information was utilized to construct simulation environments that accurately reflected the maritime traffic and weather conditions at those times.

For these simulations using vessel data, the actual maritime conditions were accurately reflected by incorporating the positions of other vessels that were anchored at the time each test vessel began its journey. For instance, in the case of the test vessel in anchorage N-3, the simulation environment incorporated the positions of all vessels anchored in areas N-1 through N-5 as of 08:54:40 on June 9, 2023. Furthermore, the prevailing weather conditions, particularly wind direction, were considered in the simulation, affecting the bow orientation of anchored vessels.

This approach enables the reinforcement learning agent to learn and plan routes under realistic maritime traffic conditions, thereby allowing for more accurate evaluation of the system's applicability in actual operational environments.

5.2 Experimental Results

We applied the reinforcement learning-based optimal path planning algorithm to each anchorage area (N-1 through N-5) and conducted a comparative analysis between actual vessel paths and simulated paths. For path simplification, we implemented the Douglas-Peucker algorithm with dynamically adjusted epsilon values. The epsilon values were optimized to ensure that the simplified paths maintained all safety requirements while preserving the essential characteristics of the original paths. Figures 4 through 8 present visualizations comparing actual paths with optimized simulation paths for each anchorage area.

In each figure, yellow grids represent independent obstacles such as buoys, while green and blue grids indicate the positions of anchored vessels with extended grid cells as explained in Figure 2. The orange regions surrounding anchored vessels visualize the minimum safety distances defined for each anchorage area. Regarding the paths, actual vessel paths are shown in magenta, while the optimal paths generated by the reinforcement learning agent are displayed as red solid lines (Optimal path (Original)), and the simplified optimal paths processed through the Douglas-Peucker algorithm are shown as royal purple solid lines (Optimal path (Simplified)). When the original optimal path (Optimal path (Original)) and the simplified path (Optimal path (Simplified)) coincide, they are uniformly represented in royal purple and labeled as "Optimal path (Original, Simplified)".

The starting point of each path is marked with a green square, while the destinations are indicated by either a red asterisk for actual path goals or a yellow asterisk for optimized path goals. In cases where the actual path goal and the optimized path goal are identical, they are uniformly represented by a yellow asterisk.

Simulation 1 was conducted using a fishing vessel with a gross tonnage of 651 t, which began its journey at 17:09:08 on June 5, 2023. The prevailing wind direction at the time was southeasterly. The maritime traffic conditions showed varying occupancy across anchorage areas, with no vessels in N-1, three vessels in N-2, five vessels in N-3, one vessel in N-4, and one vessel in N-5. In the figure, the blue dotted line represents the anchorage boundary between N-1 and N-2. Analysis of the actual path's destination reveals a safety violation where the vessel's anchoring position infringed upon the anchorage boundary safety distance (shown as a black dotted area in the figure). The optimized path generated by the reinforcement learning agent successfully avoids this violation while maintaining all required safety distances.

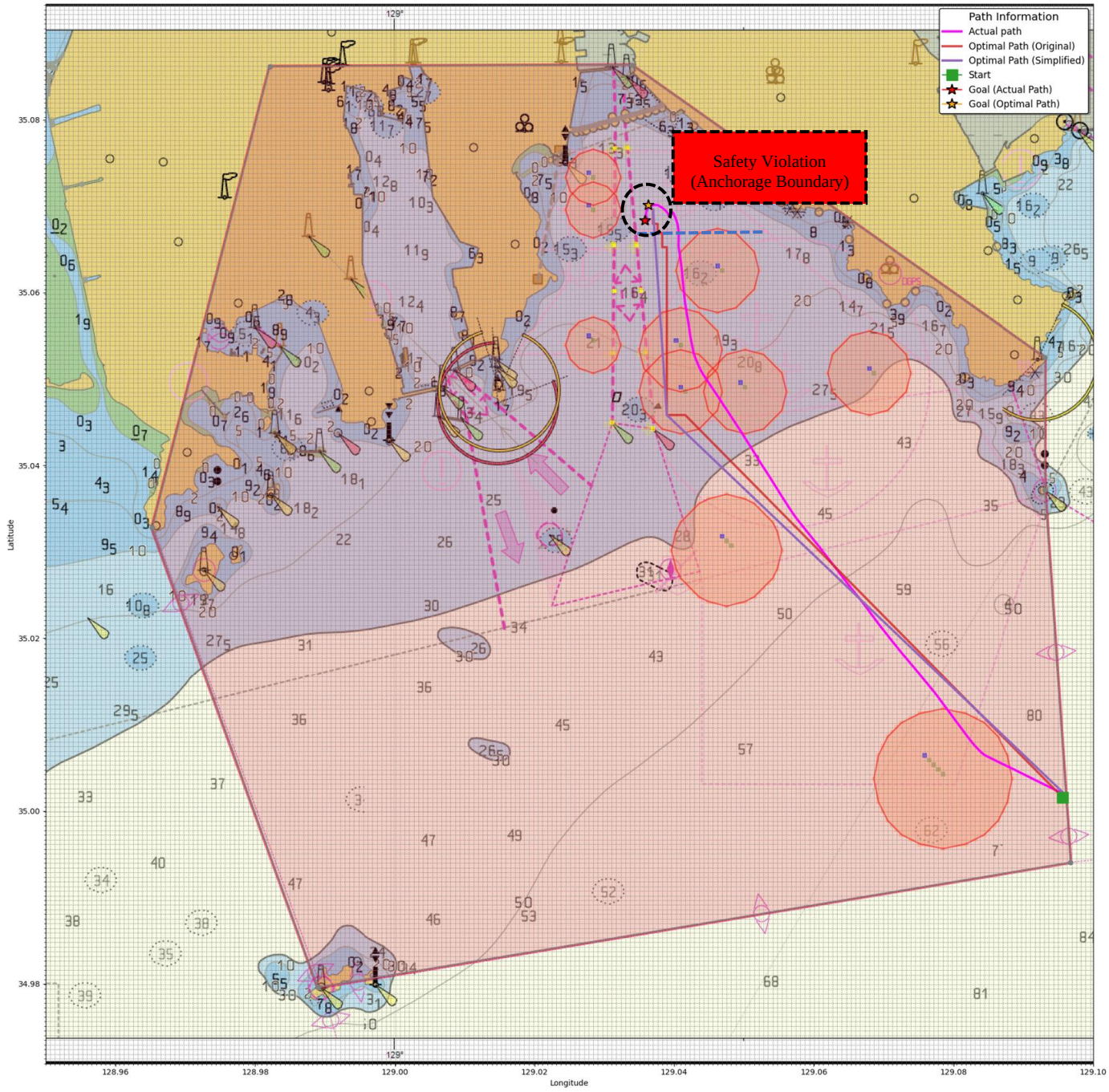


Fig. 4 Analysis of Actual and Simulated Paths in Anchorage N-1

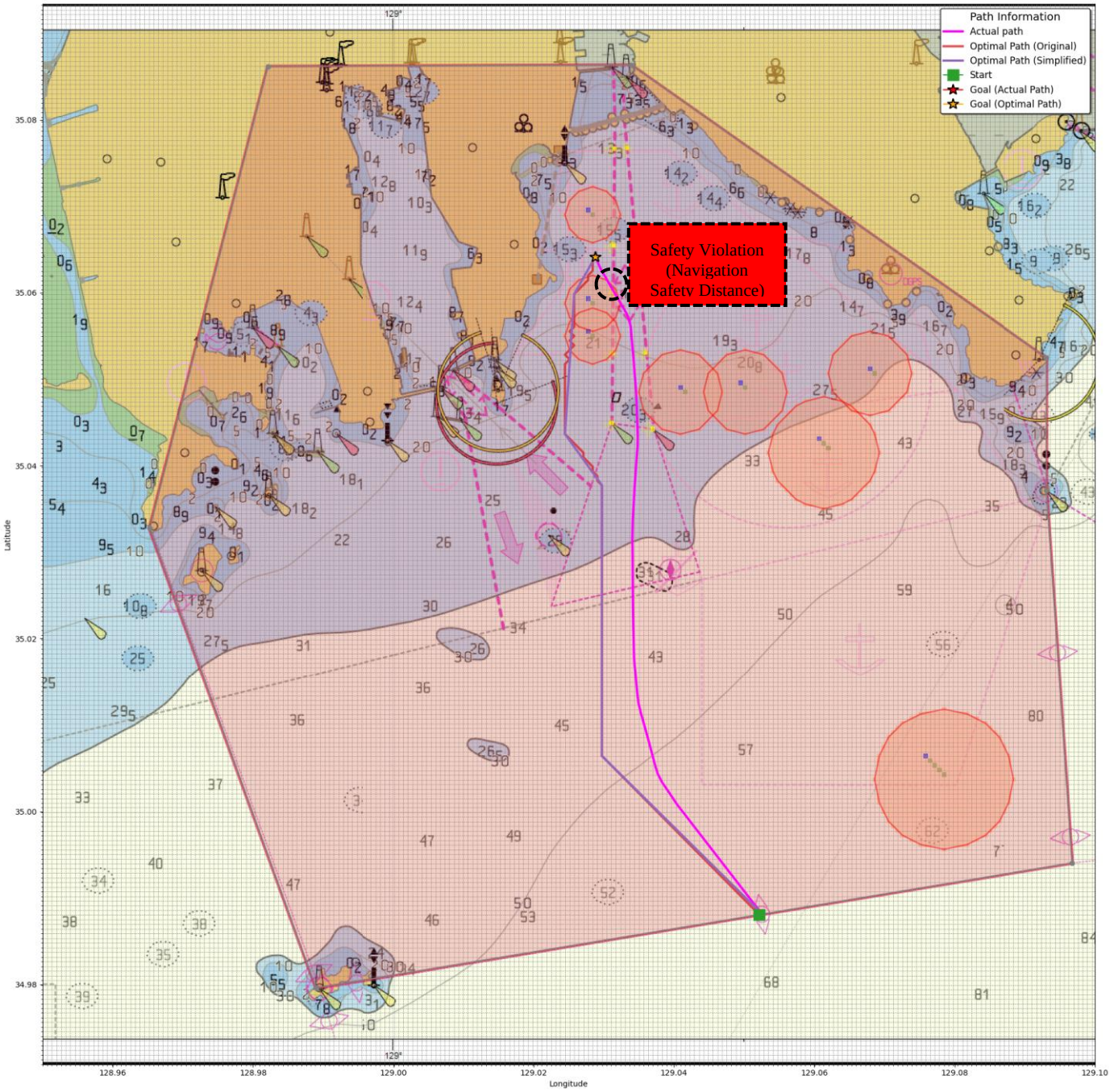


Fig. 5 Analysis of Actual and Simulated Paths in Anchorage N-2

Simulation 2 utilized data from a tanker of 2,994 gross tonnage, operating under southeasterly wind conditions on June 4, 2023, at 10:15:26. At the time of simulation, the anchorage occupancy was distributed as follows: no vessels in N-1, three in N-2, three in N-3, one in N-4, and one in N-5. The actual path shows a safety violation with respect to a nearby independent obstacle (indicated by the black dotted area), while the optimized path successfully maintains all required safety margins.

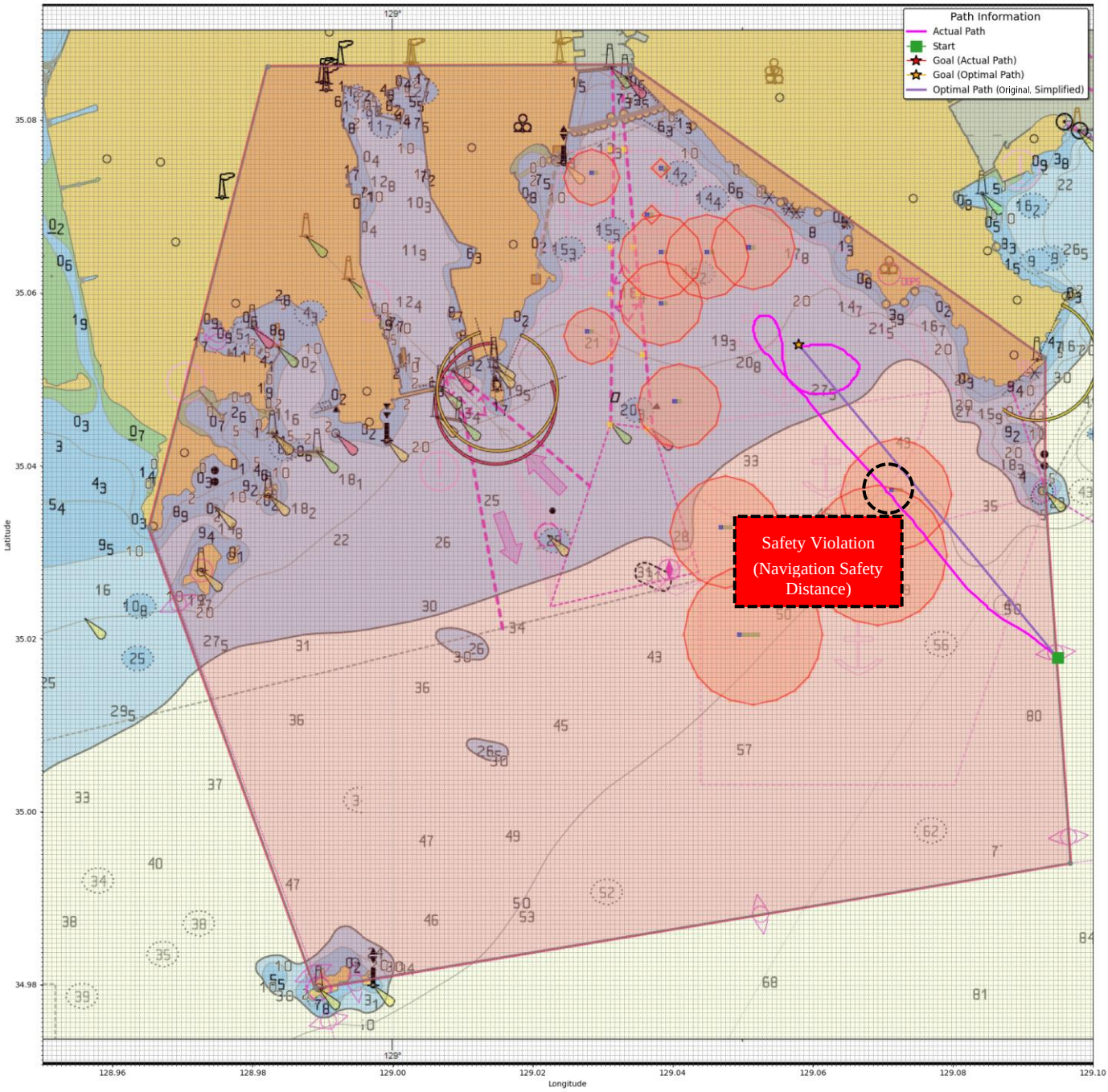


Fig. 6 Analysis of Actual and Simulated Paths in Anchorage N-3

Simulation 3 features a general cargo vessel of 6,142 gross tonnage, navigating under easterly wind conditions on June 9, 2023, at 08:54:40. The anchorage areas contained two vessels in N-1, two in N-2, five in N-3, two in N-4, and two in N-5. The actual path demonstrates a safety violation during navigation, specifically in maintaining the required distance from an anchored vessel in N-4 (highlighted by the black dotted area). The reinforcement learning algorithm generated an alternative path that maintains all safety requirements.

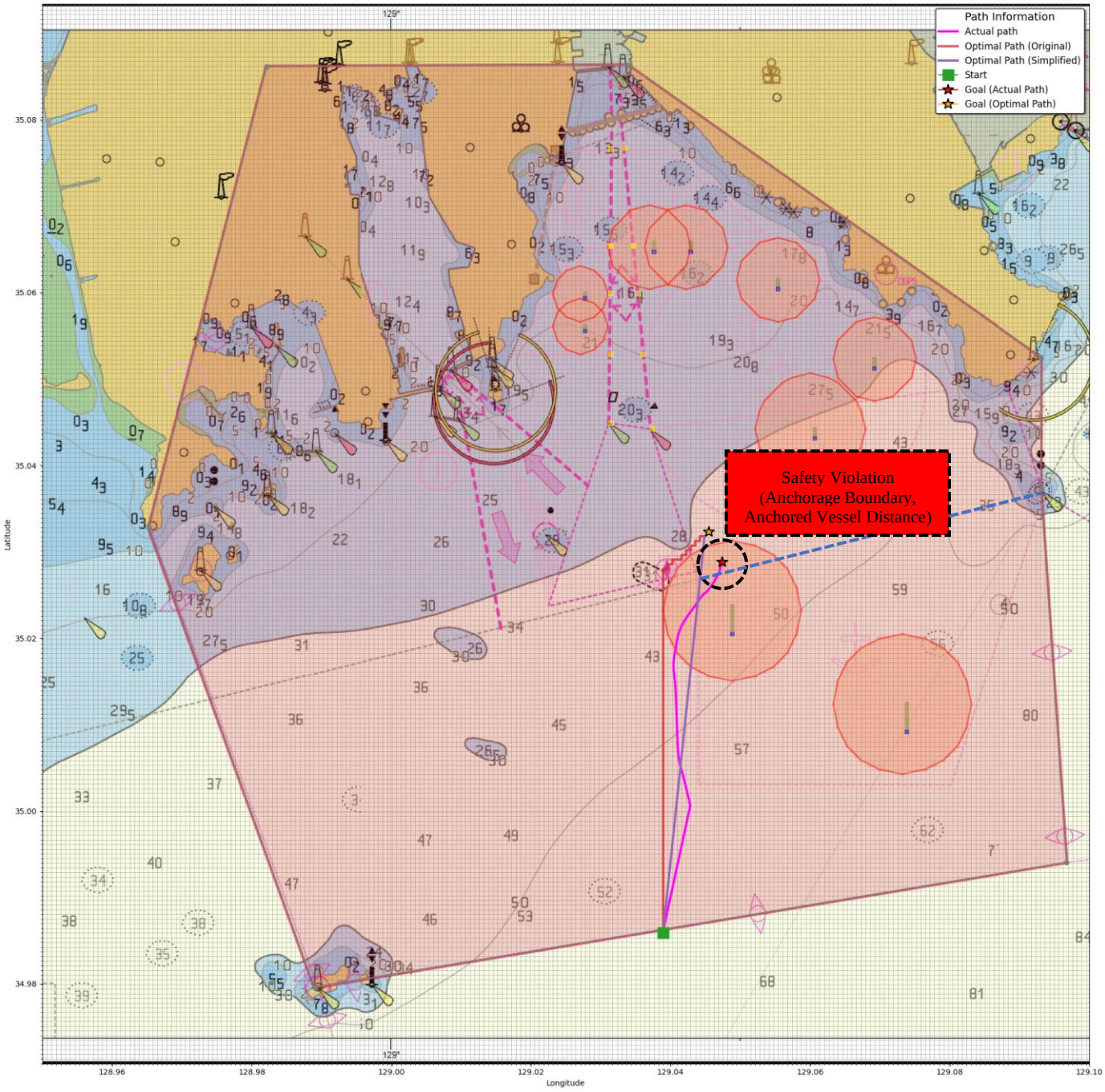


Fig. 7 Analysis of Actual and Simulated Paths in Anchorage N-4

Simulation 4 analyzed a general cargo vessel of 16,949 gross tonnage, operating under northerly wind conditions on June 3, 2023, at 05:58:22. The anchorage occupation status showed no vessels in N-1, two in N-2, four in N-3, one in N-4, and two in N-5. In the figure, the blue dotted line represents the anchorage boundary between N-4 and N-5. Analysis of the actual path's destination identified two safety violations: insufficient anchoring distance from a vessel in N-5 and violation of the anchorage boundary safety distance (marked with black dotted area). The optimized path generated by the reinforcement learning agent successfully addresses these safety concerns.

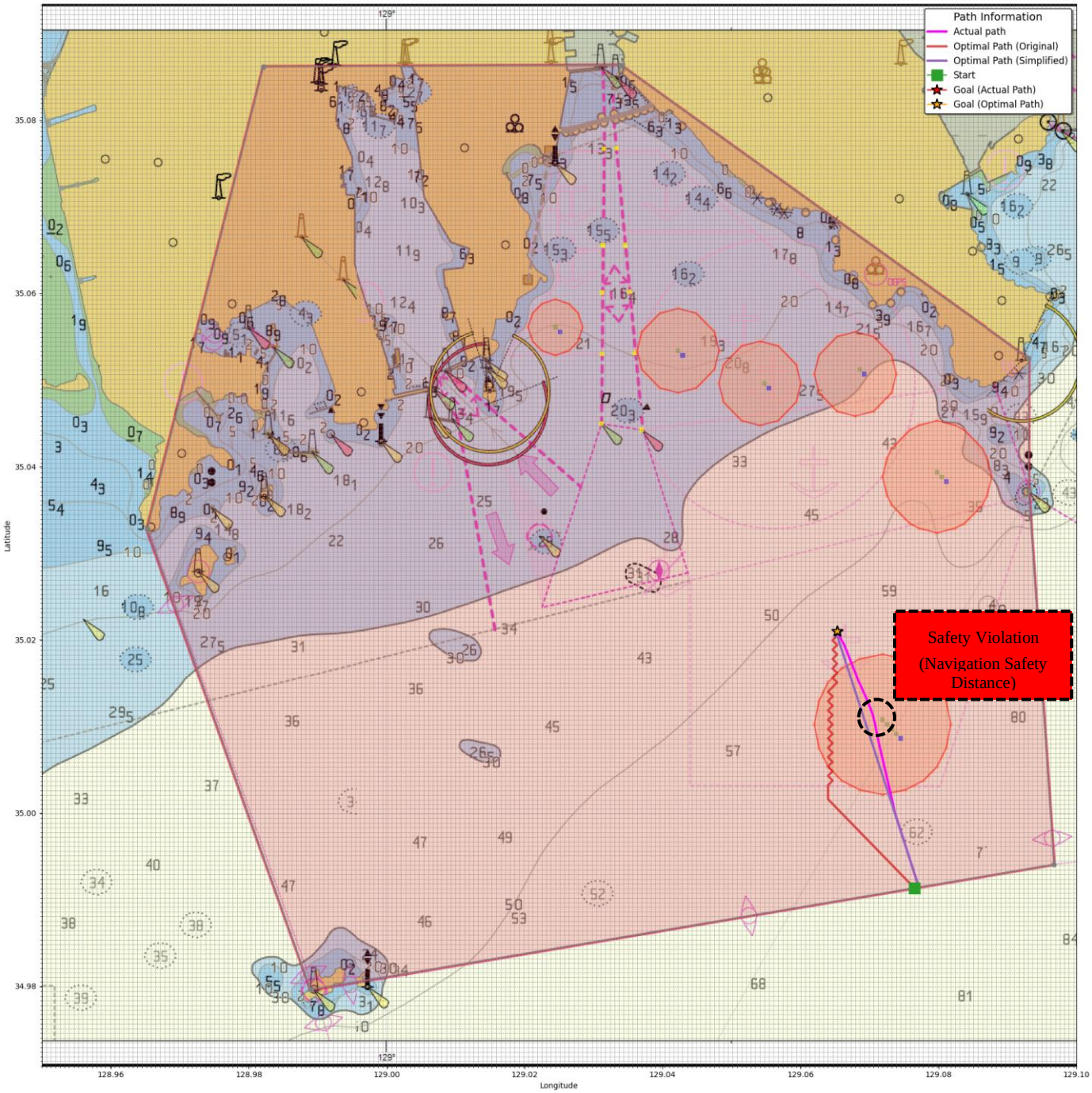


Fig. 8 Analysis of Actual and Simulated Paths in Anchorage N-5

Simulation 5 examined a general cargo vessel of 35,832 gross tonnage, navigating under northwesterly wind conditions on June 8, 2023, at 07:13:47. The anchorage areas contained one vessel in N-1, one in N-2, three in N-3, one in N-4, and one in N-5. The actual path shows a navigation safety distance violation with an N-5 vessel (indicated by the black dotted area). The optimized path maintains all required safety distances while efficiently reaching the designated anchoring position. A comprehensive performance analysis of these optimization results, including detailed path comparisons and safety assessment, is presented in the following section.

5.3 Comprehensive Performance Analysis and Discussion

This section presents a comprehensive evaluation of the path optimization results across different anchorage areas, encompassing both quantitative analysis and broader implications of the experimental findings. The quantitative assessment primarily focuses on comparing actual versus optimized path lengths,

compliance with safety distance requirements, and path simplification outcomes. Furthermore, this section discusses how these results address the limitations of current VTS systems and their practical implications for maritime traffic control operations. Table 6 provides a quantitative summary of path optimization results for each anchorage area, where Safety Violations have been comprehensively evaluated to include both safety distance violations between the agent and anchored vessels as well as violations with respect to independent obstacles.

Table 6 Quantitative Analysis of Path Optimization Results

Path Analysis			**Safety Violations**		
Anchorage	Path Length (Actual/Optimal, m)	Path length Reduction (%)	Navigation Safety Distance (Actual/Optimal, n)	Anchored Vessel Distance (Actual/Optimal, n)	Anchorage Boundary (Actual/Optimal, n)
N-1	10096.45/10035.95	0.60	No/No	No/No	Yes (1)/No
N-2	9052.45/9484.32	− 4.55	Yes (1)/No	No/No	No/No
N-3	9301.31/5245.89	43.60	Yes (1)/No	No/No	No/No
N-4	4965.86/5377.21	− 7.65	No/No	Yes (1)/No	Yes (1)/No
N-5	3733.11/3440.30	7.84	Yes (1)/No	No/No	No/No

Analysis of the results revealed distinct path optimization patterns across different anchorage areas. While path length reductions were achieved in anchorages N-1, N-3, and N-5, anchorages N-2 and N-4 showed increased path lengths due to safety considerations. The detailed analysis for each anchorage area is as follows:

In anchorage N-1, the actual path length of 10,096.45 m was reduced to 10,035.95 m through optimization, showing a path reduction rate of 0.60 %. This relatively modest reduction suggests that the vessel initially chose a fairly efficient route in terms of distance. However, from a safety perspective, the actual path's destination violated the anchorage boundary safety distance, indicating that both the VTS operator and vessel operator experienced significant difficulties in identifying a safe anchoring position.

For anchorage N-2, a path reduction rate of -4.55 % was recorded, indicating an increase from the actual path length of 9,052.45 m to 9,484.32 m after optimization. This increase in path length resulted from the selection of a detour route to address the safety distance violation with an independent obstacle (buoy) observed in the actual path, exemplifying the system's design philosophy that prioritizes safety over simple distance minimization.

Anchorage N-3 showed notable performance improvement through optimization. When comparing the complete trajectories from identical starting points to designated anchoring positions, the actual path followed by the vessel was considerably longer than the optimized path generated by our system. This difference highlights a common operational challenge in maritime traffic management, where vessels may follow inefficient routes due to practical limitations such as communication constraints between VTS operators and vessel captains, language barriers during VHF voice communications, or unfamiliarity with local anchorage zones for vessels entering Busan Port for the first time.

In the case of anchorage N-4, the actual path length increased from 4,965.86 m to 5,377.21 m after optimization, resulting in a path reduction rate of -7.65 %. This was necessary to address both anchorage boundary violations and safety distance violations with anchored vessels, incorporating destination adjustments for a safer anchoring position. This case serves as a crucial example of how safety-based decision-making should be implemented in actual operational environments.

For anchorage N-5, the actual path length of 3,733.11 m was reduced to 3,440.30 m through optimization, achieving a path reduction rate of 7.84 %. The actual path showed navigation safety distance violations with another N-5 anchored vessel. This case simultaneously demonstrates both the challenges vessel operators face in real-time monitoring and maintaining safety distances from surrounding vessels, and the cognitive

limitations of VTS operators in immediately calculating and suggesting optimal routes and anchoring positions while monitoring multiple vessels simultaneously.

Notably, by summing all path lengths from Table 7, the total distance for actual paths is approximately 37,149.18 m, whereas the optimal paths total about 33,583.67 m. Hence, there is an overall distance reduction of roughly 3,565.51 m (about 9.6 %), indicating the potential for meaningful savings in fuel and transit time.

In addition, a simple inspection of safety violations in Table 7 shows a total of six violations (across navigation safety distance, anchored vessel distance, and anchorage boundary) in the actual paths, whereas the optimized paths recorded zero. This result underscores the system's ability to enhance navigational safety by systematically avoiding risk-prone areas.

It is also important to note the characteristics of path patterns observed in Figures 5, 6, 8, and 9. The original optimal paths (shown as "Optimal path (Original)" in these figures) exhibit zigzag patterns due to the grid-based nature of the reinforcement learning environment. This pattern emerges naturally as the agent learns to navigate by moving between discrete grid cells. However, such paths present practical limitations for actual vessel operations, as they would require excessive directional changes.

To address this limitation, we applied the Douglas-Peucker algorithm as a post-processing step for path simplification. This algorithm preserves the essential features of the path while removing unnecessary intermediate points, resulting in smoother trajectories with diagonal movements (shown as "Optimal path (Simplified)" in the figures). These simplified paths maintain the quantitative improvements shown in Table 7 while significantly reducing the number of directional changes required during navigation, thereby enhancing maneuverability and practical applicability for vessel operations.

These comprehensive analysis results demonstrate that the proposed system can effectively address the limitations of the current VTS system. The experiments validated several advantages: minimizing complex voice communications through visualized optimal path provision, resolving communication issues arising from language barriers, and enabling safe guidance to anchorage areas even for vessels without prior knowledge of the zones. Although some anchorage areas generated longer paths to ensure safety, this rather serves as evidence that the system can support efficient operations while strictly adhering to safety standards required in actual maritime environments.

In particular, the significant drop in total safety violations (6 to 0) and the overall distance reduction of about 9.6 % clearly illustrate the potential operational gains and risk mitigation benefits. These characteristics not only promise overall maritime traffic safety improvements but also suggest potential fuel consumption reduction benefits through path length reductions achieved in most cases.

6. Conclusion

This study proposed a reinforcement learning-based optimal anchorage allocation and path planning system and validated its effectiveness using actual vessel operation data from Busan Port. The main achievements of this study are as follows:

First, the proposed system achieved path length reduction in most anchorage areas compared to actual vessel paths. Notably, anchorage N-3 recorded a significant path reduction rate of 43.60 %, while anchorages N-1 and N-5 achieved reductions of 0.60 % and 7.84 %, respectively. Although anchorages N-2 and N-4 showed path increases of 4.55 % and 7.65 % due to necessary detours for safety assurance, this reflects the system's design philosophy that prioritizes maritime safety.

Second, the grid extension methodology proposed in this study enabled realistic anchorage space modeling that considers vessels' actual physical dimensions and wind direction. This is expected to significantly contribute to efficient utilization and safety assurance of anchorage areas.

However, this study has the following limitations:

First, due to the constraint of not knowing the exact anchor chain calculation length for each vessel, safety distances were set by positioning the vessel's bow as the center of the safety distance zone. In reality, additional extension is needed from the bow section, which is the last part of the extended cell, by the length

of the anchor chain calculation. Future research could achieve more precise and safer anchorage allocation by investigating the exact anchor chain calculation length for each vessel.

Second, there is a limitation in that the grid extension distance was set based on the maximum length among vessels anchored during a certain period, rather than considering individual vessel lengths. This can lead to excessive occupation of anchorage space, and future research needs to develop a dynamic grid extension methodology that reflects the actual length of each vessel.

Third, while this study validated the model using Busan Port data, its applicability to other ports with different geographical characteristics and operational regulations requires further investigation. However, the methodological framework developed in this study has potential for adaptation through reconfiguration of simulation parameters and retraining of the learning model to accommodate different port environments.

Fourth. Although this study demonstrates promising results, its current computational overhead poses challenges for real-time VTS implementation. In particular, the detailed grid-based modeling and repeated learning iterations require considerable computational resources. Future research should thus explore more efficient or hybrid algorithms, high-performance computing resources, or localized path planning approaches to enable near-real-time decision-making under dynamic maritime conditions.

Fifth, the study did not explicitly incorporate seasonal variations in weather conditions, which significantly impact anchorage operations. Seasonal weather patterns such as typhoons, fog, high waves, and varying wind conditions considerably influence anchorage management and vessel safety. Therefore, integrating seasonal and real-time weather conditions into the reinforcement learning framework is essential to enhance the practical applicability and robustness of the proposed model.

Despite these limitations, the reinforcement learning-based anchorage allocation and path planning system proposed in this study demonstrated significant potential for improving the efficiency and safety of maritime traffic control operations. In particular, this study holds great significance in that it verified the system's practicality based on actual AIS data from real vessel operations in Busan Port.

Future research is expected to enhance the system's practicality and safety by addressing the aforementioned limitations and adding functionalities to respond to various maritime situations, such as path planning in severe weather conditions and alternative route generation in emergency situations. Additionally, meaningful follow-up research could involve applying the methodology proposed in this study to other major ports to verify its effectiveness.

ACKNOWLEDGEMENT

This work has supported by Korea Institute of Marine Science & Technology Promotion (KIMST) funded by the Korea Coast Guard (RS2023-00238652, Integrated Satellite-based Applications Development for Korea Coast Guard) and the Ministry of Oceans and Fisheries (20210046 / RS-2021-KS211406, Development of marine satellite image analysis application technology).

REFERENCES

[1] Busan Port Authority, 2024. 2023 Busan Port Container Cargo Handling and Transportation Statistics. https://www.busanpa.com/kor/Board.do?mode=view&mCode=MN0995&idx=32412 (accessed 14 October 2024)

[2] Ministry of Oceans and Fisheries, 2023. Vessel entry and departure status (General - by year). Statistics of Entry Vessels by Tonnage. https://kosis.kr/statHtml/statHtml.do?orgId=146&tblId=DT_MLTM_1292&conn_path=I2 (accessed 14 October 2024)

[3] Xu, L., Huang, L., Zhao, X., Liu, J., Chen, J., Zhang, K., He, Y., 2025. Navigational decision-making method for wide inland waterways with traffic separation scheme navigation system. *Brodogradnja* 76(2), 76201. https://doi.org/10.21278/brod76201

[4] ESKI, Ö., Tavacioglu, L., 2024. A combined method for the evaluation of contributing factors to maritime dangerous goods transport accidents. *Brodogradnja* 75(4), 1-20. https://doi.org/10.21278/brod75408

[5] Gao, J., Zhang, Y., 2024. Ship collision avoidance decision-making research in coastal waters considering uncertainty of target ships. *Brodogradnja* 75(2), 75203. https://doi.org/10.21278/brod75203

[6] Zhang, W., Zhang, Y., 2023. Research on classification and navigational risk factors of intelligent ship. *Brodogradnja* 74(4), 105-128. https://doi.org/10.21278/brod74406

[7] Kaelbling, L.P., Littman, M.L., Moore, A.W., 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 237-285. https://doi.org/10.1613/jair.301

[8] Kim, M.-K., Kim, J.-H., Yang, H., 2023. Optimal Route Generation and Route-Following Control for Autonomous Vessel. *Journal of Marine Science and Engineering*, 11, 970. https://doi.org/10.3390/jmse11050970

[9] Sutton, R.S. and Barto, A.G., 2018. Reinforcement Learning: An Introduction. *2nd ed. Cambridge, MA: A Bradford Book, MIT Press*.

[10] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., ... Hassabis, D., 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533. https://doi.org/10.1038/nature14236

[11] Li, Y., 2017. Deep reinforcement learning: An overview. *arXiv preprint arXiv:* 1701.07274.

[12] Xu, W., Zhu, X., Gao, X., Li, X., Cao, J., Ren, X., Shao, C., 2024. Manipulation-Compliant Artificial Potential Field and Deep Q-Network: Large Ships Path Planning Based on Deep Reinforcement Learning and Artificial Potential Field. *Journal of Marine Science and Engineering*, 12, 1334. https://doi.org/10.3390/jmse12081334

[13] Gao, X., Dong, Y., Han, Y., 2023. An Optimized Path Planning Method for Container Ships in Bohai Bay Based on Improved Deep Q-Learning. *IEEE Access*, 11, 91275-91292. https://doi.org/10.1109/ACCESS.2023.3307480

[14] Yuan, X., Yuan, C., Tian, W., Liu, G., Zhang, J., 2023. Path Planning for Ferry Crossing Inland Waterways Based on Deep Reinforcement Learning. *Journal of Marine Science and Engineering*, 11(2), 337. https://doi.org/10.3390/jmse11020337

[15] Lyu, H., Yin, Y., 2019. COLREGS-constrained real-time path planning for autonomous ships using modified artificial potential fields. *Journal of Navigation*, 72(3), 588-608. https://doi.org/10.1017/S0373463318000796

[16] Liu, M., Feng, H., Xu, H., 2020. Dynamic path planning for unmanned surface vessel based on improved artificial potential field. *Journal of Ship Mechanics*, 24(12), 1625-1635. https://doi.org/10.3969/j.issn.1007-7294.2020.12.010

[17] Wang, J., Xiao, Y., Li, T., Chen, C.L.P., 2023. A jamming aware artificial potential field method to counter GPS jamming for unmanned surface ship path planning. *IEEE Systems Journal*, 17(3), 4555-4566. https://doi.org/10.1109/JSYST.2023.3263581

[18] Lyu, H., Yin, Y., 2017. Ship's trajectory planning for collision avoidance at sea based on modified artificial potential field. *In Proceedings of the 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, Shanghai, China, 29–31 December 2017, 351-357. https://doi.org/10.1109/ICRAE.2017.8291397

[19] Liu, K., Zhang, Y., Ren, J., 2016. Path planning algorithm for unmanned surface vehicle based on an improved artificial potential field method. *Natural Science Journal of Hainan University*, 34(2), 99-104.

[20] Du, Y., Zhang, X., Cao, Z., Wang, S., Liang, J., Zhang, F., Tang, J., 2021. An Optimized Path Planning Method for Coastal Ships Based on Improved DDPG and DP. *Journal of Advanced Transportation*, 2021, 7765130. https://doi.org/10.1155/2021/7765130

[21] Guo, S., Zhang, X., Du, Y., Zheng, Y., Cao, Z., 2021. Path Planning of Coastal Ships Based on Optimized DQN Reward Function. *Journal of Marine Science and Engineering*, 9(2), 210. https://doi.org/10.3390/jmse9020210

[22] Lee, H.T., Kim, M.K., 2024. Optimal path planning for a ship in coastal waters with deep Q network. *Ocean Engineering*, 307, 118193. https://doi.org/10.1016/j.oceaneng.2024.118193

[23] Xiao, Q., Jiang, L., Wang, M., Zhang, X., 2023. An Improved Distributed Sampling PPO Algorithm Based on Beta Policy for Continuous Global Path Planning Scheme. *Sensors*, 23, 6101. https://doi.org/10.3390/s23136101

[24] 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. https://doi.org/10.1016/j.oceaneng.2021.109216

[25] Li, L., Wu, D., Huang, Y., Yuan, Z.M., 2021. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Applied Ocean Research*, 113, 102759. https://doi.org/10.1016/j.apor.2021.102759

[26] Wang, Z., Li, G., Ren, J., 2021. Dynamic path planning for unmanned surface vehicle in complex offshore areas based on hybrid algorithm. *Computer Communications*, 166, 49-56. https://doi.org/10.1016/j.comcom.2020.11.012

[27] Xie, S., Chu, X., Zheng, M., Liu, C., 2020. A composite learning method for multi-ship collision avoidance based on reinforcement learning and inverse control. *Neurocomputing*, 411, 375-392. https://doi.org/10.1016/j.neucom.2020.05.089

[28] Chen, C., Ma, F., Xu, X., Chen, Y., Wang, J., 2021. A Novel Ship Collision Avoidance Awareness Approach for Cooperating Ships Using Multi-Agent Deep Reinforcement Learning. *Journal of Marine Science and Engineering*, 9(10), 1056. https://doi.org/10.3390/jmse9101056

[29] Guo, S., Zhang, X., Zheng, Y., Du, Y., 2020. An Autonomous Path Planning Model for Unmanned Ships Based on Deep Reinforcement Learning. *Sensors*, 20(2), 426. https://doi.org/10.3390/s20020426

[30] Cao, S., Fan, P., Yan, T., Xie, C., Deng, J., Xu, F., Shu, Y., 2022. Inland Waterway Ship Path Planning Based on Improved RRT Algorithm. *Journal of Marine Science and Engineering*, 10, 1460. https://doi.org/10.3390/jmse10101460

[31] Zhen, R., Gu, Q., Shi, Z., Suo, Y., 2023. An Improved A-Star Ship Path-Planning Algorithm Considering Current, Water Depth, and Traffic Separation Rules. *Journal of Marine Science and* Engineering, 11, 1439. https://doi.org/10.3390/jmse11071439

[32] Busan Regional Office of Oceans and Fisheries, 2024. Rules on Navigation of Busan Port, etc. [Notice No. 2024-24 of Busan Regional Office of Oceans and Fisheries]. National Law Information Center, https://law.go.kr/admRulLsInfoP.do?admRulId=2088785&efYd=0.

[33] Khatib, O., 1986. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1), 90-98. https://doi.org/10.1177/027836498600500106