e-Navi

Available online at http://www.e-navi.kr/
e-Navigation Journal

Original article

Advancing Maritime Route Optimization: Using Reinforcement Learning for Ensuring Safety and Fuel Efficiency

Jisoo Kim, Byeonggong Hwang, Gi-Hyun Kim, Ung-Gyu Kim

[a] R&D Center, mapsea Corp., Seoul, Korea

Abstract

Efficient and safe maritime navigation in complex and congested coastal regions requires advanced route optimization methods that surpass the limitations of traditional shortest-path algorithms. This study applies Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) reinforcement learning (RL) algorithms to generate and refine optimal ship routes in East Asian waters, focusing on passages from Shanghai to Busan and Ulsan to Daesan. Operating within a grid-based representation of the marine environment and considering constraints such as restricted areas and Traffic Separation Schemes (TSS), both DQN and PPO learn policies prioritizing safety and operational efficiency. Comparative analyses with actual vessel routes demonstrate that RL-based methods yield shorter and safer paths. Among these methods, PPO outperforms DQN, providing more stable and coherent routes. Post-processing with the Douglas-Peucker (DP) algorithm further simplifies the paths for practical navigational use. The findings underscore the potential of RL in enhancing navigational safety, reducing travel distance, and advancing autonomous ship navigation technologies.

Keywords: Autonomous Ship Navigation, Reinforcement Learning, Deep Q-Network, Proximal Policy Optimization, Maritime Route Optimization, Traffic Separation Schemes, Maritime Safety

https://doi.org/10.52820/j.enavi.2024.23.041

1. Introduction

Maritime transportation underpins the global logistics network, handling over 90% of international trade (UNCTAD, 2020). In particular, East Asia—with major ports like Shanghai, Busan, and Singapore—serves as a pivotal hub for international shipping but also experiences intense traffic congestion and a heightened risk of marine accidents. Ensuring navigational safety in such complex waterways requires robust route optimization strategies.

Evidence suggests that heavily trafficked regions, such as the Singapore Strait and the East China Sea, face elevated collision and grounding rates due to navigational complexity and congestion (Kim et al., 2019; IMO, 2021). Meeting these challenges demands approaches that not only identify efficient routes but also continually assess dynamic environmental factors. As Johansen et al. (2016) emphasized, route optimization is integral to improving safety and efficiency in autonomous vessel operations. It includes real-time data—such as water depth, obstacles, and TSS—to support decision-making.

Traditional shortest-path algorithms often fail to incorporate essential safety constraints, prompting the adoption of reinforcement learning (RL) methods. This study applies DQN and PPO to maritime route optimization, comparing RL-generated routes against actual vessel paths. PPO, leveraging stable policy updates, demonstrates improved performance over DQN in complex coastal environments. Through route simplification via the DP algorithm, practical and navigationally efficient routes are obtained, laying groundwork for autonomous navigation advancement.

This study specifically uses DQN and PPO to optimize routes between Shanghai and Busan, and between Ulsan and Daesan, comparing the RL-generated routes with actual ship paths. Unlike previous research focusing on local route planning with static obstacles, this work accounts for complex ship dynamics, navigational constraints, and environmental factors. The PPO algorithm, in particular, offers advanced route searching in continuous state and action spaces, better reflecting real-world challenges.

By considering factors like turning radius, propulsion limits, and under-keel clearance, as well as dynamic conditions such as wind, waves, and currents (Mnih et al., 2015), the proposed DQN- and PPO-based approaches demonstrate the potential for enhancing both safety and operational efficiency in East Asian waters.

2. Related Work

2.1. Reinforcement learning

Reinforcement learning (RL) generates optimal ship routes by iteratively interacting with an environment, as illustrated in **Figure 1** (Sutton and Barto, 2018). At each time step t, an agent observes its current state s_t (e.g., geographic coordinates and heading), chooses an action a_t, and transitions to a new state s_{t+1}, receiving a reward r_{t+1}. Through repeated trial-and-error, the agent learns a policy that balances two key behaviors: exploration (testing new actions) and exploitation (leveraging learned strategies).

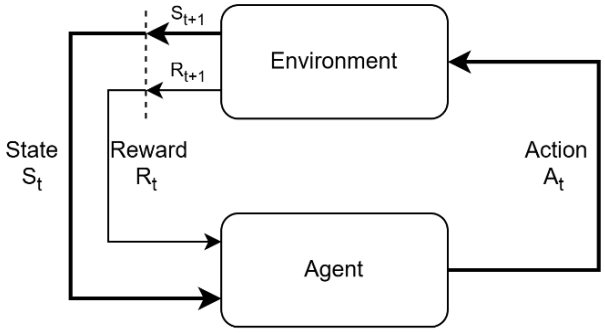

Figure 1. Interaction between the agent and the environment in a Markov Decision Process.

A fundamental requirement for RL is that the current state encapsulates all essential information, known as the Markov property. Overloading the agent with excessive details can cause confusion, so carefully selecting the state representation is crucial. In this study, the environment is discretized into a grid system (**Figures 2 and 3**), where the agent selects actions corresponding to movements between grid cells in both discrete and continuous spaces. This setup allows straightforward navigation along and around great-circle routes, controlling position (latitude/longitude adjustments) and speed. The agent's goal is to maximize cumulative rewards by balancing safety, efficiency, and compliance with navigational constraints.

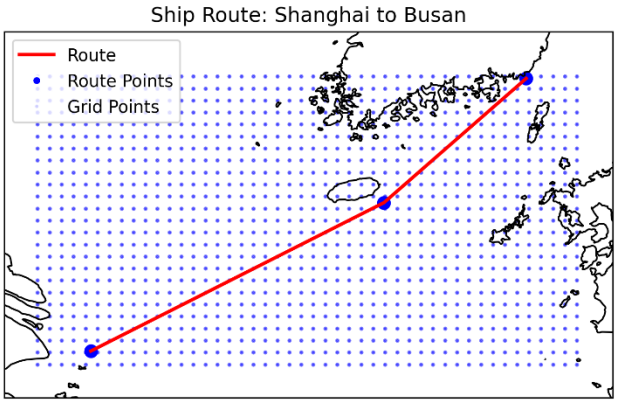

Figure 2. Grid system for the study of the Shanghai-to-Busan route.

Related Work on RL Approaches for Route Optimization

***Deep Q-Network (DQN)*:**

DQN integrates Q-learning with deep neural networks to handle high-dimensional state spaces. It estimates action-value functions and updates policies through experiences sampled from replay buffers, stabilizing learning. While effective in simpler scenarios, DQN can struggle in highly complex environments.

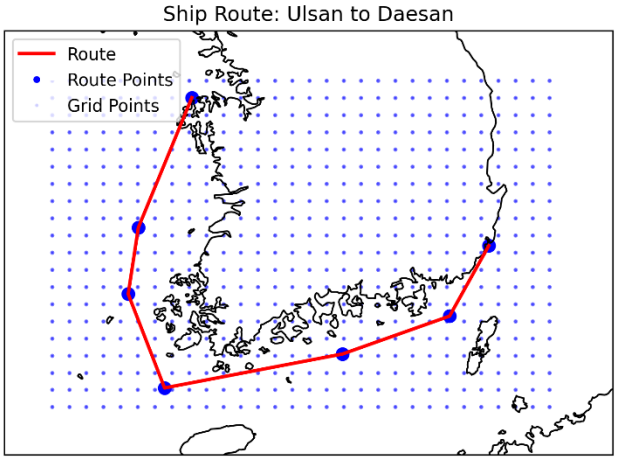

Figure 3. Grid system for the study of the Ulsan-to-Daesan route.

Proximal Policy Optimization(PPO):

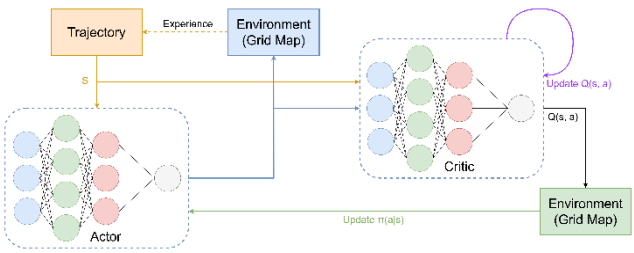

Figure 4. The PPO algorithm

PPO is a policy-gradient method that updates policies more stably than earlier algorithms like TRPO(Schulman et al., 2015). By clipping policy updates and balancing exploration and exploitation, PPO can handle complex, dynamic maritime conditions more robustly than DQN.

2.2. Electronic Navigational Chart

Electronic Navigational Charts (ENCs) are digital chart systems that provide essential maritime information for safe and efficient route planning during ship operations. Produced according to the S-57 standard format established by the International Hydrographic Organization (IHO), ENCs include electronic formats of geographic information such as depths, port locations, hazardous areas, and navigational data (IHO, 2018). These data are utilized through Electronic Chart Display and Information Systems (ECDIS), which are periodically updated to provide navigators with the latest information, enabling prompt responses to dynamic marine environments (Felski and Zwolak, 2020).

In this study, ENCs were employed to construct the grid-map environment and to incorporate critical maritime elements into the RL framework. Integrating ENC data into the grid map allows the agent to operate in an environment closely resembling real-world conditions. The ENC information includes TSS, navigational aids, and spatial data representing the characteristics of waterways around the Korean Peninsula, along with lighthouses, buoys, beacons, obstacles, anchorages, routes, current data, seabed topography, marine weather information, maritime traffic data, and risk areas. These elements are indispensable for safe navigation and efficient route planning at sea.

The quality and accuracy of ENC data directly affect navigational performance. Experiments were conducted using the most recent ENC data, adhering to international standards for route planning and monitoring. By integrating ENC data into the grid-map environment, the RL agent can test and refine decision-making strategies under realistic maritime conditions, thereby contributing to increased maritime safety, improved route planning efficiency, and establishing a foundation for future autonomous navigation.

2.3. Route Optimization

Maritime route optimization is critical for maximizing

safety and efficiency as vessels travel from departure points to destinations. This study seeks to overcome the limitations of traditional path-finding algorithms by applying DRL techniques within a grid-based map environment that captures complex and dynamic maritime conditions (including water depth, obstacles, TSS, wind, waves, and currents). Traditional algorithms typically emphasize the shortest distance, often failing to account for safety in areas with dense obstacles or intricate traffic-separation schemes.

By contrast, the RL algorithms utilized here—DQN and PPO—allow agents to learn composite reward functions through environment interaction. These reward functions incorporate hazardous-area avoidance (shallow waters, restricted zones, and congested traffic areas), fuel efficiency, and minimized navigation time. Consequently, agents adapt to environmental changes and update their policies, deriving safer and more economical routes.

Notably, the PPO algorithm ensures learning stability via a clipping technique that limits abrupt policy changes within its Actor-Critic framework, effectively approximating optimal policies in complex, continuous state-action spaces. As a result, PPO demonstrates more stable route-finding performance than DQN in intricate marine environments. Additionally, by integrating real-time maritime conditions—such as TSS, shallow waters near ports, reefs, fishing grounds, and restricted navigation zones—the route optimization process closely reflects actual environmental constraints.

However, the optimal routes generated by PPO or DQN may contain highly granular waypoints, which may require re-evaluation under real conditions involving currents or weather. As discussed in Section 2.4, a path simplification method such as the DP algorithm can be applied during post-processing to address this issue.

2.4. Douglas-Peucker Algorithm

Although RL algorithms can derive optimal routes, these routes may exhibit complex curvilinear forms and contain excessively detailed waypoints, impeding practical implementation. To address this issue, this study employs the Douglas-Peucker (DP) algorithm (Douglas and Peucker, 1973) to simplify the generated paths.

3. Methodology

3.1. Environment Setup and Grid System

To simplify the complex marine environment into a format suitable for RL algorithms, we employ a grid-map representation. Bathymetric data from the General Bathymetric Chart of the Oceans (GEBCO) was used to create a detailed grid covering the Shanghai–Busan and Ulsan–Daesan routes. Each grid cell represents an area of 4 km × 4 km, providing sufficiently granular depth information to ensure navigational safety without overwhelming computational resources.

Depth information is critical for safe navigation, especially for vessels with large drafts. Shallow areas are categorized as high-risk zones, incurring penalties for RL agents that plan routes through them (Meneghetti and Fraccaroli, 2020). The grid map also integrates navigational constraints such as restricted areas and TSS, all of which must be respected during route optimization.

Initially, the grid map was composed of 400 m × 400 m cells. To improve computational efficiency while maintaining broader navigational perspective, bicubic interpolation was used to downscale the resolution to 4 km × 4 km. This approach reduces environmental complexity while preserving essential details for decision-making, making it well-suited to coastal routes with fewer obstacles.

Figure 5 illustrates the conceptual layout of the grid map used in our experiments. Each cell is defined by latitude (lat) and longitude (lon) coordinates, with the central cell ($cell_c$) surrounded by eight neighboring cells. This configuration provides the agent with navigational information regarding adjacent cells, such as restricted zones or shallow areas, during its route exploration.

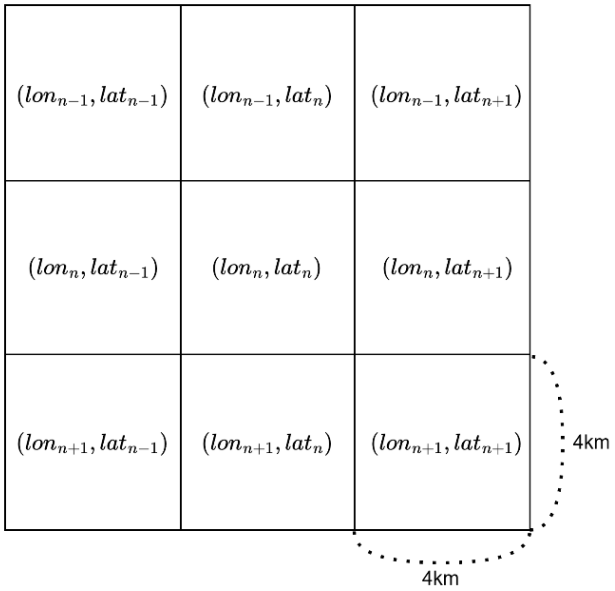

Figure 5. Conceptual composition of the grid map for reinforcement learning.

3.2. Data and Routes

Table 1 provides details about an actual ship sailing from Shanghai to Busan, while **Figure 6** depicts the map of the environment where the RL agent interacts in our experiments.

Table 1 Details of the ship operating on the Shanghai to Busan route

Ship to experience 1	
MMSI	440403000
Name	KEOYOUNG SEVEN
Type	Chemical/Oil Products Tanker
Draught	4.5 m
Length	70 m
Beam	12 m
Date	2024-09-24 07:24 - 2024-09-25 21:08 (GMT)
Distance	429.7 NM

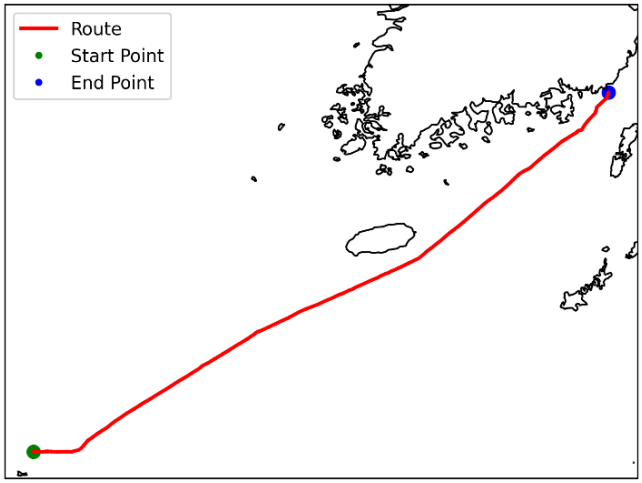

Figure 6. Environment and trajectory of the ship sailing from Shanghai to Busan.

Similarly, **Table 2** and **Figure 7** show the environment for a ship sailing from Ulsan to Daesan.

Table 2 Details of the ship operating on the Ulsan to Daesan route

Ship to experience 2	
MMSI	440027000
Name	KEOYOUNG BLUE 1
Type	Chemical/Oil Products Tanker
Draught	5.1 m
Length	72 m
Beam	12 m
Date	2024-10-01 16:46 ~ 2024-10-03 03:00 (GMT)
Distance	401.88 NM

3.3. Reward Functions and Hyperparameters

3.3.1. Parameter of DQN

The DQN algorithm is a value-based RL method that uses state–action value functions (Q-functions) to find optimal actions. It was applied here to explore safe and efficient routes in the maritime environment.

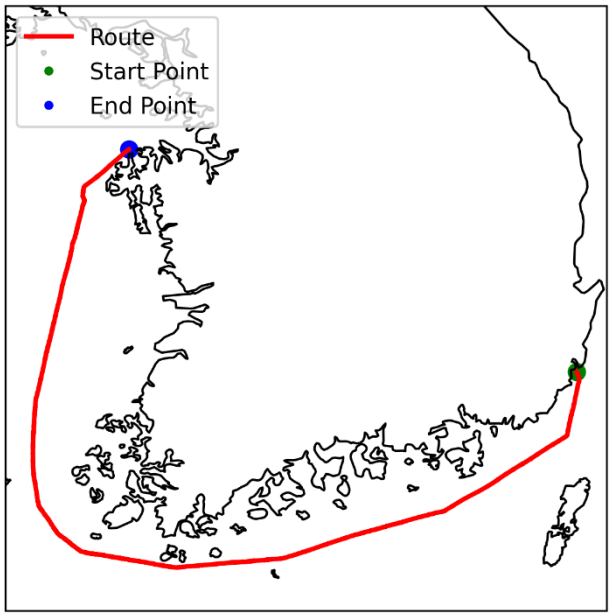

Figure 7. Reinforcement learning environment from Ulsan to Daesan.

(1) Reward Function Design

The agent receives rewards or penalties for navigating different zones or repeating actions, guiding it toward efficient paths. **Table 3** details the reward function for DQN.

Table 3 Reward function of DQN Algorithm

Zone/Action	Reward/ Penalty
Navigable Areas	0
Land and TSS Restricted Areas	-1
Visited Areas	-1
TSS Passable Areas	+0.5
Destination Reached	+1
Current action is the same as the previous action	0
Current action is adjacent to the previous action	-0.001
Current action is not adjacent to the previous action	-0.005
Total travel distance in the current episode is shorter than in the previous episode	+1

(2) Hyperparameter Settings

Table 4 lists the hyperparameters used for the DQN algorithm.

Table 4. Hyperparameters for the DQN Algorithm

Hyperparameters	Value
Total Episodes	25,000,000
Number of environments(n-envs)	128
Batch size	2,048
Buffer size	100,000
Learning rate	0.001
Discount factor(gamma)	0.8
Exploration rate(epsilon)	1.0

3.3.2. Parameter of PPO

We also conducted experiments using PPO on the same routes. Empirical results showed that PPO produced more optimal and stable routes than DQN.

(1) Reward Function Design

Table 5 outlines the reward function for PPO, similar in structure to DQN but with different values to promote stability.

Table 5 Reward function of PPO Algorithm

Zone/Action	Reward/ Penalty
Navigable Areas	0
Land and TSS Restricted Areas	-10
Visited Areas	-5
TSS Passable Areas	+5
Destination Reached	+10
Current action is the same as the previous action	0
Current action is adjacent to the previous action	-0.01
Current action is not adjacent to the previous action	-0.05
Total travel distance in the current episode is shorter than in the previous episode	+10

(2) Hyperparameter Settings

Table 6 summarizes the hyperparameters for PPO.

(3) Training Time

On average, training took around 1 hour and 5 minutes, ranging from 50 minutes to 1 hour 20 minutes per experiment.

Table 6. Hyperparameters for the PPO Algorithm

Hyperparameters	Value
Total Episodes	15,000,000
Number of environments(n-envs)	128
Batch size	1,024
Learning rate	0.0003
Discount factor(gamma)	0.91
Clipping parameter(epsilon)	0.1
Number of steps(n-steps)	2,048
Entropy coefficient	0.07

3.4. Post-Processing (Douglas-Peucker Algorithm)

Once the RL algorithms (DQN or PPO) generate a series of waypoints, the raw routes often contain unnecessary complexity. These paths can include numerous intermediate nodes that, while mathematically optimal, may be impractical for real-world navigation due to minor directional shifts that provide negligible benefit.

To address this issue, we apply the Douglas-Peucker algorithm, a well-established line simplification technique, to the RL-generated routes. The aim is to reduce the route's complexity while retaining the essential geometric characteristics and navigational safety. The process involves the following steps:

1. **Initial Segmentation**: The algorithm first considers the route as a polyline defined by a sequence of latitude-longitude coordinate pairs. A straight line is drawn between the first and last points of this polyline.

2. **Farthest Point Identification**: Among the remaining intermediate points, the algorithm locates the point that is farthest from the straight line. This point represents the location where the original route deviates most significantly from a simple linear path.

3. **Distance Threshold Check**: If the distance from this farthest point to the line segment connecting the start and end points is greater than a predefined tolerance threshold, that point is deemed necessary to maintain navigational accuracy. The original route is then split at this point, creating two sub-polylines. The algorithm recursively applies the same process to each sub-polyline.

4. **Iterative Simplification**: If the farthest point lies within the tolerance threshold, it can be safely removed without substantially altering the overall shape or safety criteria of the route. The algorithm continues in this manner, iteratively simplifying the route by removing redundant waypoints until no segment violates the tolerance threshold.

5. **Resulting Simplified Route**: The end result is a route composed of fewer waypoints, maintaining the route's critical navigational features. The tolerance level, carefully chosen based on factors such as vessel maneuverability, environmental complexity, and safety margins, ensures that essential turning points or hazards remain accurately represented.

By applying the Douglas-Peucker algorithm to the raw RL-generated routes, the final paths become more manageable and operationally friendly. Captains, pilots, or autonomous navigation systems can more easily interpret and follow these simplified yet reliable routes. Although the simplification process reduces complexity, careful consideration must be given to the chosen tolerance value to avoid discarding important route details—especially in areas with intricate coastal features, dense traffic, or numerous obstacles. In such scenarios, adjustments to the algorithm's parameters or the use of supplementary post-processing strategies may be necessary to preserve the integrity of the navigation plan.

4. Results and Discussion

4.1. Deep Q-Network (DQN) Results

4.1.1. Optimal Route Generation from Shanghai to Busan

Figure 8 shows the route generated by DQN for Shanghai to Busan. Although it was generally stable, the route included numerous detailed waypoints, creating unnecessary complexity. To address this issue, the Douglas-Peucker algorithm was applied to simplify the route, removing superfluous points while retaining essential navigational characteristics. The simplified result is presented in **Figure 9**.

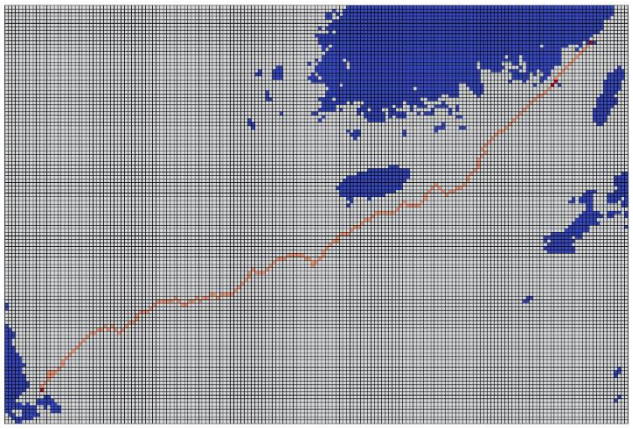

Figure 8. Original DQN-generated route for Shanghai → Busan.

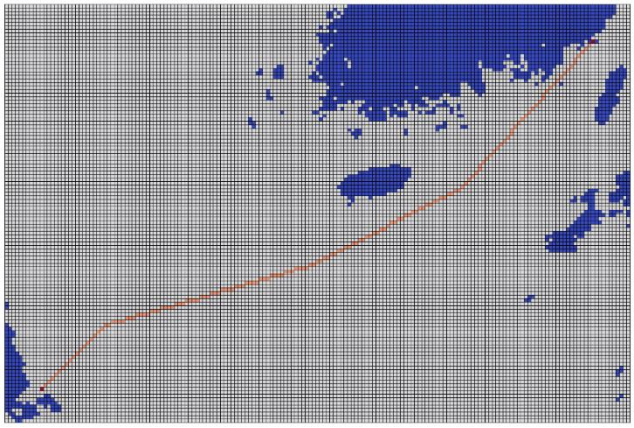

Figure 9. Shanghai → Busan route after applying the Douglas-Peucker algorithm to the DQN output.

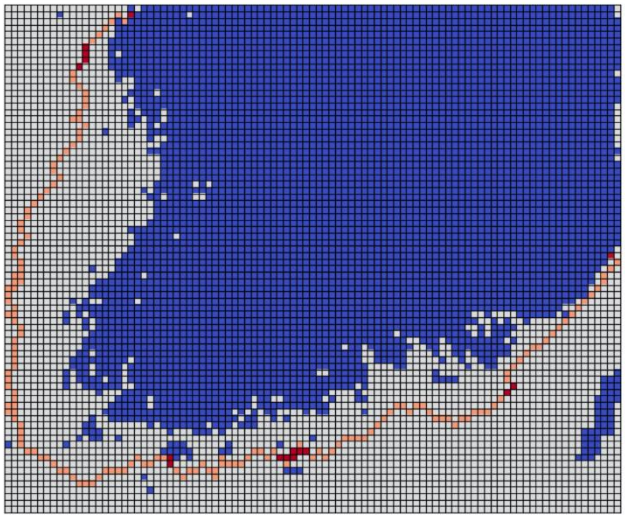

Figure 10. Original DQN-generated route for Ulsan → Daesan.

4.1.2. Optimal Route Generation from Ulsan to Daesan

Similarly, DQN was used to generate an optimal route from Ulsan to Daesan (**Figure 10**). Like the previous route, it included excessive waypoints, increasing overall complexity. After applying the Douglas-Peucker algorithm, the path was simplified, resulting in a more streamlined and navigationally efficient route (**Figure 11**).

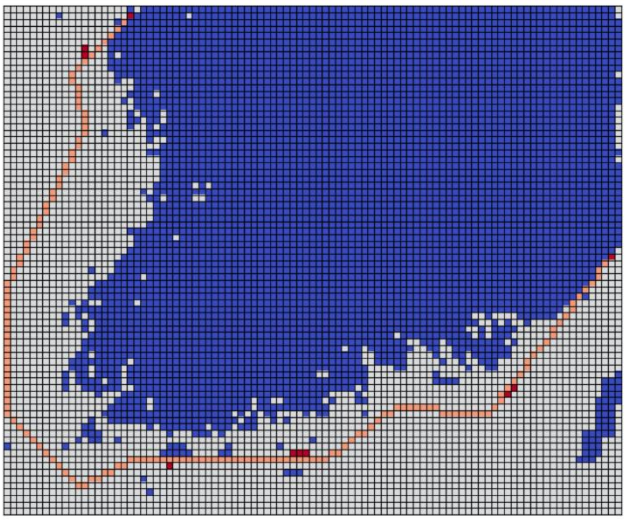

Figure 11. Simplified optimal route from Ulsan to Daesan using the Douglas-Peucker algorithm on the DQN model's output.

4.2. Proximal Policy Optimization (PPO) Results

4.2.1. Optimal Route Generation from Shanghai to Busan

The PPO algorithm, known for its stable policy updates, produced efficient routes even in complex maritime environments. **Figure 12** illustrates the PPO-generated route from Shanghai to Busan. Despite obstacles such as islands, shallow areas, and restricted zones, the PPO approach reached the destination safely and efficiently. Initially, the route included frequent directional changes. After applying the Douglas-Peucker algorithm (**Figure 13**), the route became more straightforward and suitable for practical navigation.

4.2.2. Optimal Route Generation from Ulsan to Daesan

The PPO algorithm also successfully generated an optimal route from Ulsan to Daesan (**Figure 14**). As before, applying the Douglas-Peucker algorithm simplified the route, removing unnecessary complexity and resulting in a more efficient and navigable path (**Figure 15**).

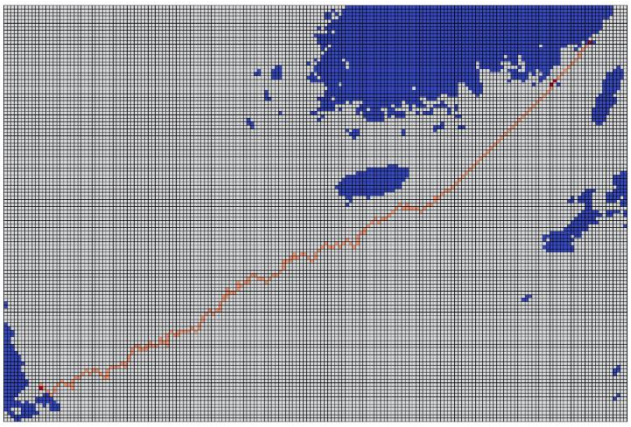

Figure 12. Original PPO-generated route for Shanghai → Busan.

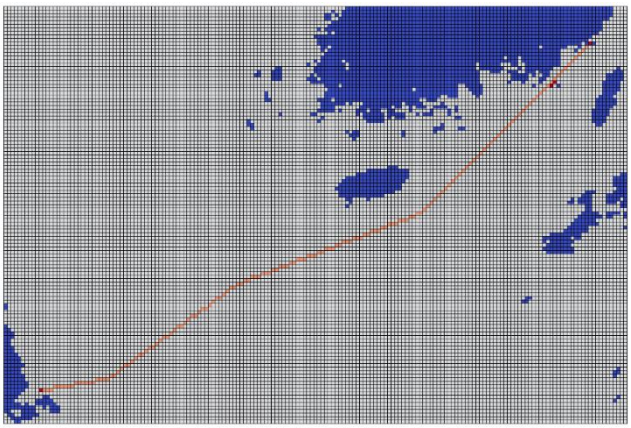

Figure 13. Shanghai → Busan route after applying the Douglas-Peucker algorithm to the PPO output.

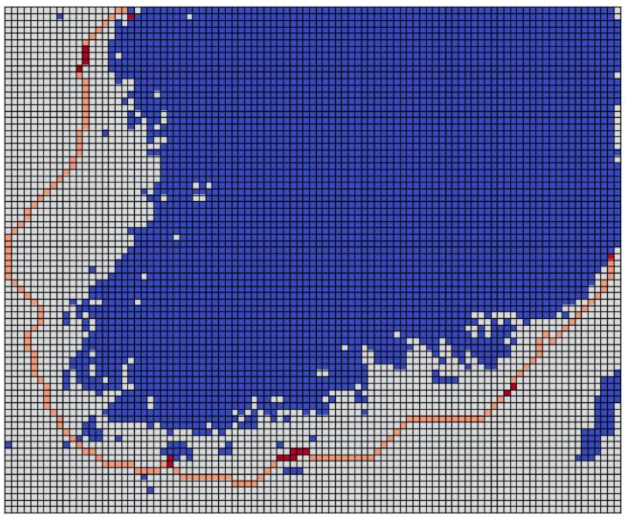

Figure 14. Original PPO-generated route for Ulsan → Daesan.

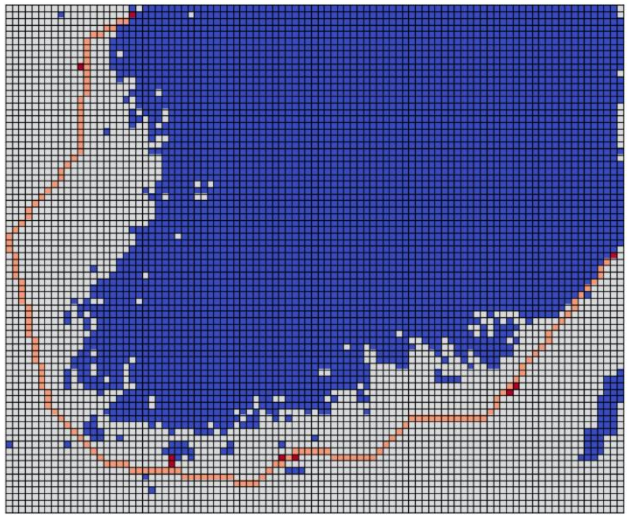

Figure 15. Ulsan → Daesan route after applying the Douglas-Peucker algorithm to the PPO output.

4.3. Comparative Analysis of Results

Table 7. Comparative Distances (NM)

Route	Shanghai → Busan	Ulsan → Daesan
Actual	574.76	605.02
DQN	442.10	413.56
PPO	422.74	406.21

PPO produced safe and efficient routes even in challenging environments, yielding more optimal paths than DQN. Particularly, route simplification via the DP algorithm provided routes suitable for practical navigation.

However, in areas with multiple islands, simplification using only the DP algorithm posed limitations. In such complex environments, it may be necessary to retain the original PPO-generated route or employ further algorithmic refinements to preserve both efficiency and safety.

4.4. Discussion

Experimental results demonstrate that PPO outperforms DQN in generating efficient routes for complex maritime environments. Moreover, applying the DP algorithm for route simplification enhances real-world navigability.

This study suggests that RL-based autonomous route generation can significantly contribute to maritime route planning, although further research is needed for more robust performance in highly complex marine

environments.

5. Conclusion

This study confirmed that applying reinforcement learning—specifically DQN and PPO—can produce safer and more efficient routes in complex East Asian waters. Both DQN and PPO outperformed conventional approaches in reducing distance and improving navigational safety compared to actual vessel routes. Among these methods, PPO yielded more stable and coherent solutions, demonstrating its adaptability to intricate maritime environments and dynamic constraints.

Nevertheless, the work has some limitations. Integrating real-time data such as weather patterns and live traffic information is essential to strengthen model robustness. Further enhancements include refining the grid map with higher-resolution data, using advanced sensors, and collaborating with maritime authorities for large-scale validation in real environments. These steps will be crucial milestones in the practical adoption of RL for ship navigation.

Overall, this research offers a foundation for integrating route optimization technologies into complex marine environments like East Asia. The results suggest potential contributions to route optimization and ship-operation management systems by incorporating weather information, navigational notices, ENCs, and real-time traffic volumes. This can serve as a critical basis for balancing environmental objectives and operational efficiency in the maritime industry. Future researchers may also explore additional real-time environmental variables to further expand this field.

Acknowledgments

This research was supported by Korea Institute of Marine Science & Technology Promotion(KIMST) funded by the Ministry of Oceans and Fisheries RS-2023- 00254860.

This work was supported by Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT(MSIT, Korea)&Gwangju Metropolitan City.

References

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization, 10(2), 112–122.

Felski, A., & Zwolak, K. (2020). The safety of autonomous ships: Operational and legal aspects. Scientific Journal of Gdynia Maritime University, 113, 7–17.

Felski, A., & Zwolak, K. (2020). The challenges of Electronic Chart Display and Information Systems (ECDIS) usage. Journal of Marine Science and Engineering, 8(1), 65.

International Hydrographic Organization. (2018). IHO standards for Electronic Navigational Charts (ENC) S-57 Edition 3.1. IHO Publication.

International Maritime Organization. (2021). Reports on marine casualties and incidents: Annual report 2020.

Johansen, T. A., Perez, T., & Cristofaro, A. (2016). Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment. IEEE Transactions on Intelligent Transportation Systems, 17(12), 3407–3422.

Kim, H. J., Kim, H. S., & Kim, J. H. (2019). Analysis of marine accidents in Korean waters using GIS. Journal of the Korean Society of Marine Environment and Safety, 25(2), 123–130.

Meneghetti, G., & Fraccaroli, D. (2020). A Dijkstra-based algorithm for ship route planning in ice-covered waters. Journal of Marine Science and Engineering, 8(9), 688.

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.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Schulman, J. (2015). Trust Region Policy Optimization. *arXiv preprint arXiv:1502.05477*.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of Go without human knowledge. Nature, 550(7676), 354–359.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

United Nations Conference on Trade and Development. (2020). Review of maritime transport 2020.

Received **09 December 2024**

1st Revised **30 December 2024**

Accepted **30 December 2024**