Journal of
Marine Science and Engineering

MDPI

Article

Manipulation-Compliant Artificial Potential Field and Deep Q-Network: Large Ships Path Planning Based on Deep Reinforcement Learning and Artificial Potential Field

Weifeng Xu [1], Xiang Zhu [1,*], Xiaori Gao [2], Xiaoyong Li [1], Jianping Cao [1], Xiaoli Ren [1] and Chengcheng Shao [1]

[1] College of Meteorology and Oceanography, National University of Defense Technology, Changsha 410073, China; xuweifeng.23@nudt.edu.cn (W.X.); sayingxmu@nudt.edu.cn (X.L.); caojp@nudt.edu.cn (J.C.); renxiaoli18@nudt.edu.cn (X.R.); shaoc@nudt.edu.cn (C.S.)

[2] Navigation College, Dalian Maritime University, Dalian 116026, China; gaoxiaori@dlmu.edu.cn

[*] Correspondence: zhuxiang@nudt.edu.cn

Abstract: Enhancing the path planning capabilities of ships is crucial for ensuring navigation safety, saving time, and reducing energy consumption in complex maritime environments. Traditional methods, reliant on static algorithms and singular models, are frequently limited by the physical constraints of ships, such as turning radius, and struggle to adapt to the maritime environment's variability and emergencies. The development of reinforcement learning has introduced new methods and perspectives to path planning by addressing complex environments, achieving multi-objective optimization, and enhancing autonomous learning and adaptability, significantly improving the performance and application scope. In this study, we introduce a two-stage path planning approach for large ships named MAPF–DQN, combining Manipulation-Compliant Artificial Potential Field (MAPF) with Deep Q-Network (DQN). In the first stage, we improve the reward function in DQN by integrating the artificial potential field method and use a time-varying greedy algorithm to search for paths. In the second stage, we use the nonlinear Nomoto model for path smoothing to enhance maneuverability. To validate the performance and effectiveness of the algorithm, we conducted extensive experiments using the model of "Yupeng" ship. Case studies and experimental results demonstrate that the MAPF–DQN algorithm can find paths that closely match the actual trajectory under normal environmental conditions and U-shaped obstacles. In summary, the MAPF–DQN algorithm not only enhances the efficiency of path planning for large ships, but also finds relatively safe and maneuverable routes, which are of great significance for maritime activities.

Keywords: large ships; path planning; safety; DQN; artificial potential field

check for updates

Citation: Xu, W.; Zhu, X.; Gao, X.; Li, X.; Cao, J.; Ren, X.; Shao, C. Manipulation-Compliant Artificial Potential Field and Deep Q-Network: Large Ships Path Planning Based on Deep Reinforcement Learning and Artificial Potential Field. *J. Mar. Sci. Eng.* **2024**, *12*, 1334. https://doi.org/10.3390/jmse12081334

Academic Editors: Marko Perkovic, Sebastian Feuerstack, Maciej Gucma, Andrej Androjna and Ivica Pavic

Received: 25 June 2024
Revised: 30 July 2024
Accepted: 2 August 2024
Published: 6 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Maritime transport is the lifeblood of the global economy, and large vessels play a pivotal role due to their exceptional cargo capacity. However, naval accidents are frequent and more than 80% are attributed to human factors [1], underscoring the importance of intelligent automated navigation systems. These systems can significantly reduce the rate of maritime accidents by minimizing human error, thereby ensuring the safety of personnel and assets at sea. Within intelligent navigation systems, route planning is crucial. It requires algorithms to ensure the safety and feasibility of the route while also demonstrating high adaptability and flexibility to cope with the ever-changing maritime conditions and potential emergencies. Researchers have developed various route planning methods, including bio-inspired algorithms [2–7], graph-based A* algorithms [8–12], artificial potential field methods [13–15], and data-driven intelligent algorithms [16–21], all aimed at improving the safety and efficiency of maritime navigation.

Bionic algorithms [2] perform probabilistic optimization searches by simulating biological behaviors. A prominent example is the ant colony algorithm [3,4,11], which is frequently applied in path planning research. Researchers have integrated it with other optimization techniques to improve performance, such as the bacterial foraging optimization algorithm [7] and the simulated annealing algorithm [6]. However, these approaches do not always ensure effective path planning under dynamic or time-varying conditions. To address this problem, Wang et al. [5] proposed a method that utilizes particle swarm acceleration for local path planning in dynamic navigation environments. Even with this progress, bionic algorithms still frequently need fine-tuning parameters and are prone to becoming stuck in local optima.

Unlike bionic algorithms, the A* algorithm [12] employs raster maps to discover paths with reduced costs and shorter distances. Yu et al. [8] enhanced the traditional A* algorithm by incorporating a surrogate value into its cost function, allowing ships to rapidly return to their predetermined course after avoiding obstacles. Li et al. [9] reduced both the path length and the number of inflection points by combining the A* algorithm with the dynamic window approach. To accurately represent the ship's current navigation situation, factors such as the genuine marine environment and the time consumption of expected routes are incorporated into the algorithm design [10,22]. Nevertheless, these algorithms often struggle with real-time performance and their search efficiency diminishes in environments with an abundance of nodes.

The artificial potential field method (APF) has been employed for ship path planning due to its real-time performance and ease of implementation. Liu et al. [14] enhanced this method by incorporating velocity and acceleration factors into the attractive and repulsive forces. However, this strategy relies on the ship's precise location, which can be uncertain in practice. To address this issue, Wang et al. [15] proposed an APF variant capable of detecting interference sources to determine their positions. The International Regulations for Preventing Collisions at Sea (COLREGS) is an internationally recognized convention aimed at preventing maritime collisions. Ohn and Namgung [23] found that the APF exhibits the highest adaptability to the COLREGS. Consequently, researchers have integrated several enhanced APF methods with COLREGS to develop algorithms for dynamically avoiding obstacles [13,24,25]. Despite the advantages of this approach, it has three significant flaws, i.e., local minimum traps, inability to reach the destination, and complex path execution.

Despite the individual strengths of bio-inspired algorithms, A* algorithms, and Artificial Potential Field (APF) methods in path planning, they may exhibit limitations as single models when confronted with complex and variable environments. These algorithms often lack the flexibility to rapidly adapt to unforeseen circumstances, such as sudden environmental changes. In such cases, further adjustments may be needed for prompt and effective response. To address these limitations, the academic community has begun exploring deep reinforcement learning, particularly Deep Q-Networks (DQNs) [16], as a strategic solution. The DQN aims to achieve more agile and adaptable path planning in complex navigational environments by learning the mapping between environmental states and actions. For instance, studies by Shen et al. [26], Chun et al. [20], Liu et al. [21], and Wen et al. [27] have applied DQNs to maritime path planning that adheres to the International Regulations for Preventing Collisions at Sea (COLREGS). These studies demonstrate the potential application of DQNs in various maritime tasks, including route planning within ferry terminals and search and rescue missions.

The application of DQNs in maritime path planning is constrained by the design of the reward function. Sparse reward signals can slow the learning process and affect the algorithm's convergence speed. To address this challenge, Du et al. [28] and Chen et al. [29] redesigned the reward function to introduce a denser reward distribution, thus accelerating the learning process and enhancing the exploratory capacity of the strategy. Furthermore, Yang et al. [30], Guo et al. [19], and Li et al. [31] attempted to simplify the design of the reward function using the APF method, building attractive and repulsive potential fields to guide the vessels to avoid obstacles and move toward their goals. However, the existing methods' oversight of kinematic and dynamic constraints in reward function formulation can lead to infeasible or hazardous navigation, especially in confined maritime settings [32]. To address this, we introduce a novel path-planning algorithm that integrates vessel dynamics, environmental variability, and risk assessment into its core design. The primary contributions of this paper are summarized as follows:

(1) Following the grid-based representation of the navigation environment, the reward function within the DQN algorithm has been enhanced using the APF method to improve learning efficiency and overcome the difficulties associated with local minimum traps and the inability to reach the destination.

(2) In response to the high inertia of large ships and the characteristics of the rudder servo systems, the experiment involves conducting sustained rotational trials using the nonlinear Nomoto mathematical model of the "Yupeng" vessel. The experimental paths are pruned, extended, and translated into the paths generated by the MAPF–DQN algorithm to obtain smooth trajectories with rudder positions.

The structure of this paper is as follows. Section 2 introduces the basic knowledge underlying the algorithms discussed in this paper. Section 3 details the framework of the proposed algorithm, integrating the Artificial Potential Field and Deep Q-Network processes. Section 4 demonstrates the performance of the proposed algorithm through experimentation, covering both path planning and path feasibility enhancement. Finally, Section 5 provides a comprehensive conclusion based on the experimental results.

2. Theoretical Background

2.1. Artificial Potential Field Method

The Artificial Potential Field (APF) method mimics the attraction and repulsion of charged particles to guide an entity around obstacles and toward its goal. Consequently, establishing attractive and repulsive fields is crucial for the effectiveness of this strategy. As the controlled entity increases its distance from obstacles, the magnitude of the repulsive force should decrease. On the contrary, as the distance from the target point increases, the magnitude of the attractive force should increase. This scheme ensures that the resultant forces collectively steer the entity toward the predetermined target.

In a two-dimensional plane, suppose a ship departs from a starting point under the dual effects of both a gravitational U_{Gi} and a repulsive field U_{Ri}. The spatial positions of the starting and target points are represented as $[x, y]$ and $\left[x_g, y_g\right]$, respectively, within the inertial coordinate system. The overall potential field acting on the ship can be expressed as U_{All} in Equation (1).

$$U_{All}(X) = U_{Gi}(X) + U_{Ri}(X) \tag{1}$$

where X denotes the coordinate position $\left[x_{ship}, y_{ship}\right]$ of the ship while sailing. At this point, the ship experiences the gravitational field function outlined in Equation (2).

$$U_{Gi}(X) = \frac{1}{2}K_G R_a^2 \tag{2}$$

where K_G represents an artificially determined gravitational coefficient. The term R_a refers to the Euclidean distance between the ship and the target point.

The repulsive field function is often modeled as a quadratic function in which the independent variable corresponds to the inverse of the Euclidean distance between the ship and the obstacle. It indicates that minor variations in the ship's trajectory can substantially affect the repulsive force, potentially intensifying the vibration phenomenon. Consequently, this study utilizes an exponential function [33] for the repulsive field function, which is mathematically formulated as shown in Equation (3).

$$U_{Ri}(X) = \begin{cases} K_{Ri} R_a^2 e^{\left(-\rho^2\right)}, \rho \leq \rho_d \\ 0 \quad\quad\quad\quad , \rho > \rho_d \end{cases} \tag{3}$$

where K_{Ri} represents an artificially determined repulsion coefficient, ρ denotes the Euclidean distance between the ship and the obstacle, and ρ_d is the obstacle's range of influence.

2.2. Deep Reinforcement Learning

The research framework of reinforcement learning is based on Markov Decision Processes (MDPs), where the core concept is that the state of the intelligent agent in the next time step depends solely on the current state and the action taken. This process involves the agent receiving a state signal S_t from the environment at a specific moment $t \in T$, selecting an action from the allowable set of actions A_t, updating its state based on the transition probabilities, and obtaining rewards. The particular manifestation of MDP is denoted in Equation (4).

$$S_t \xrightarrow{\pi(\mathrm{a}|\mathrm{s})} A_t \xrightarrow{p(s_{t+1}|s_t,a_t)} S_{t+1} \rightarrow R_{t+1} \tag{4}$$

where S represents the set of states of the agent; A denotes the set of actions that the agent can execute; P represents the transition probabilities between states; R signifies the rewards the agent receives upon reaching a certain state; and π, referred to as the policy, is the probability of transitioning from a state to an action.

The requite G_t of the agent from time step $t < T$ onwards encapsulates all future rewards, as depicted in Equation (5).

$$G_\mathrm{t} = R_{\mathrm{t}+1} + \gamma R_{\mathrm{t}+2} + \gamma^2 R_{\mathrm{t}+3} \ldots = \sum_{t=0}^{+\infty} \gamma^\tau R_{\mathrm{t}+ø+1} \tag{5}$$

The discount factor $\gamma \in [0,1]$ determines the balance between immediate rewards and future rewards. A value of 0 implies that the agent prioritizes immediate gains, while a value of 1 indicates that the agent values all future rewards equally. The action–value function $Q_\pi(s,a)$ denotes the expected return after the agent takes action (a) in the state (s), as defined in Equation (6).

$$Q_\pi(s,a) = E_\pi(G_t \mid S_t = s, A_t = a) \tag{6}$$

In conventional reinforcement learning, Q-values are often represented in tabular forms, indexed by various states and corresponding actions. Nevertheless, such methodology frequently proves impractical for real-world scenarios characterized by continuous state spaces. Consequently, an integration of deep learning techniques with reinforcement learning has emerged, wherein neural networks are employed to estimate Q-values. The schematic of this approach is depicted in Figure 1.

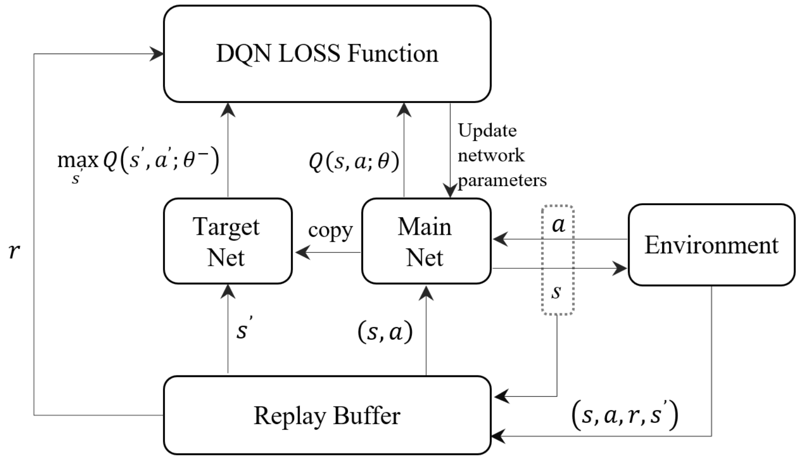


Figure 1. Structure of the DQN algorithm.

The following five steps are taken to analyze the structure of framework components.

(1) The main network is a convolutional neural network designed to approximate the state–action value function with hyperparameters θ. It accepts both the current state and the selected action by the agent as inputs and produces corresponding Q-values as outputs. This network aims to learn representations of state–action pairs to predict the expected return of taking a specific action in a given state.

(2) The target network is employed to mitigate discrepancies induced by temporal differences, of which the hyperparameters are θ^{-}. It periodically replicates the parameters (θ) from the primary network, maintaining consistency with it. This approach bolsters the algorithm's stability and generates training labels for the main network.

(3) Convolutional neural networks utilize the maximum likelihood estimation method to approximate the true Q-value, predicated on the assumption of independently and identically distributed training samples. However, correlations may be present in the data collected during the learning process. To address this issue and augment training stability, researchers have incorporated an experience replay pool. This pool persistently archives state (s), action (a), and reward information (r), facilitating random data extraction during the learning episodes.

(4) The loss function is shown in Equation (7).

$$L(\theta) = E\left[\left(r + \gamma \max_{a'} Q\left(s', a'; \theta^{-}\right) - Q(s, a; \theta)\right)^{2}\right] \tag{7}$$

The parameters of the main network are updated according to Equation (8).

$$\theta_{t+1} = \theta_t + \alpha\left[r + \gamma \max_{a'} Q\left(s', a'; \theta^{-}\right) - Q(s, a; \theta)\right] \bigtriangledown Q(s, a; \theta) \tag{8}$$

In Equation (8), the gradient of the main network is indicated by $\bigtriangledown Q(s, a; \theta)$ and α denotes the learning rate. In the algorithm flow,$r + \gamma \max_{a'} Q(s', a'; \theta^{-})$ represents the target Q value. Subsequently, the Q is updated using the temporal difference method, as per Equation (9).

$$Q(s, a) \leftarrow Q(s, a; \theta) + \alpha(Target\ Q - Q(s, a; \theta)) \tag{9}$$

(5) The traditional greedy strategy in the search process evenly allocates a certain probability to each action, while assigning the remaining probability to the optimal action. However, random searches conducted during the later stages of training can potentially disrupt the identification of the optimal strategy. To alleviate this disruption, adjusting the value of the parameter ϵ is recommended[1]. The specific equations are illustrated in Equations (10) and (11).

$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A(s)|}, if\ a = arg\ \max_a Q_\pi(s, a) \\ \frac{\varepsilon}{|A(s)|}\quad, if\ a \neq arg\ \max_a Q_\pi(s, a) \end{cases} \tag{10}$$

$$\varepsilon = \begin{cases} 1,\ if\ T_s < 100 \parallel P_e < \frac{1}{T_s^{0.2}} \\ 0, \qquad else \end{cases} \tag{11}$$

The variable T_s represents the training rounds of the algorithm, while P_e is a random number of [0,1]. In the initial phase, the algorithm searches for paths with 100% probability. After 100 training rounds, the search continues with some level of randomness. This method involves dynamically tuning the value of ϵ throughout the training process to strike a balance between minimizing computational overhead and optimizing the robustness of the resultant strategy.

3. General Framework of the Algorithm

In this paper, we propose an algorithmic framework based on deep reinforcement learning for the planning and smoothing of ship navigation paths. This framework has two main stages: the path planning stage and the path smoothing stage.The overall framework is depicted in Figure 2. During the path planning stage, the algorithm utilizes deep reinforcement learning techniques to initiate from the initial state, assess the environment, and select the optimal sequence of actions to generate a preliminary navigation path. The objective of this stage is to establish an efficient route to the destination, without considering the specific operational constraints of the ship.

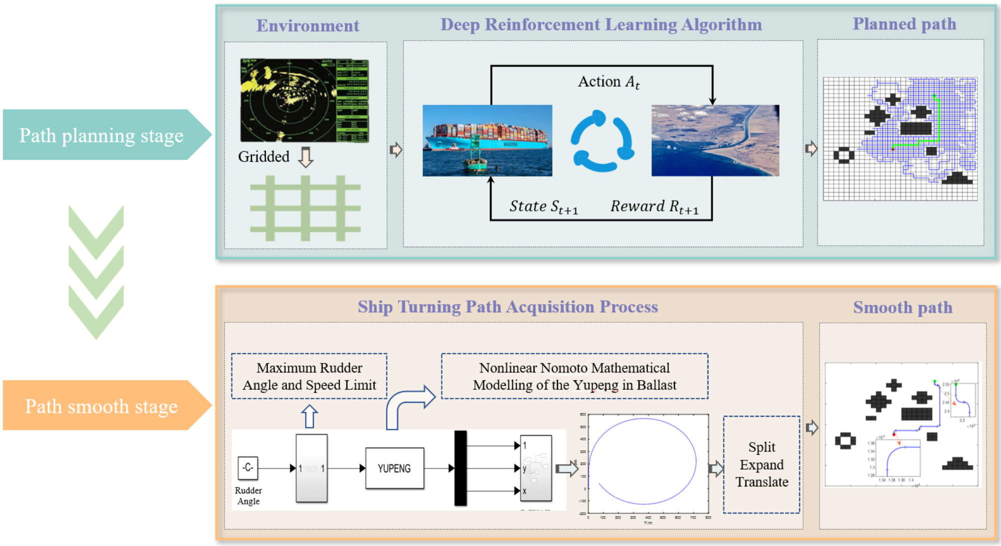

Path planning stage
Environment
Gridded
Deep Reinforcement Learning Algorithm
Action A_t
State S_{t+1}
Reward R_{t+1}
Planned path
Path smooth stage
Ship Turning Path Acquisition Process
Maximum Rudder Angle and Speed Limit
Nonlinear Nomoto Mathematical Modelling of the Yupeng in Ballast
-C-
Rudder Angle
YUPENG
Split Expand Translate
Smooth path

Figure 2. The overall framework of MAPF–DQN algorithm.

Subsequently, the algorithm advances to the path smoothing stage, actively integrating the nonlinear Nomoto mathematical model to simulate the dynamic behavior of the ship when it is in a ballast state. By segmenting, expanding, and transforming the preliminary path, the algorithm further optimizes the route, reducing its tortuosity and enhancing the smoothness of navigation. In this stage, the algorithm also considers the ship's maximum rudder angle and speed limitations, ensuring that the generated path complies with the ship's physical characteristics and meets the requirements of actual navigation.

The entire algorithm framework diagram depicts the comprehensive process, starting from environment initialization, moving through path planning, and finally reaching path smoothing. This progression showcases the logical and systematic nature of our algorithm design.

3.1. Path Planning Algorithm

In the path planning stage, we adopted deep reinforcement learning techniques to overcome the limitations of single models in terms of adaptability. In this field, the Deep Q-Network (DQN) algorithm, recognized as a classic and mature technology, has been widely acknowledged and applied to solve complex decision-making problems. The DQN algorithm works in various intricate navigation environments, but it encounters the challenge of reward sparsity. Although the APF is efficient, fast, and simple, it is hindered by local minimum traps and the inability to reach the destination. This research merges the APF technique with the DQN algorithm to surmount these constraints.

In the MAPF–DQN algorithm, the environment module of the framework is implemented as the APF environment module, as shown in Figure 3. The pseudo-code is shown in Algorithm 1.

The APF environment module is essentially a design of the original segmented reward function into the improved artificial potential field method mentioned in Section 2.1, as shown in Equation (12).

$$r = -U_{Gi}(X) - U_{Ri}(X) \tag{12}$$

Algorithm 1 Path Planning Algorithm.

```
Input: Initial environmental observation matrix $G$. The action set is defined as a discrete ensemble of movements, i.e., up, down, left, and right.
Parameter descriptions:
Observation period $T_{\text{obs}}$, the algorithm solely accumulates data into the replay buffer without performing random sampling.
Training period $T_{\text{train}}$, the algorithm stores data in the replay buffer and conducts random sampling.
Training round $T_{\text{episode}}$, the algorithm ceases searching and advances to the subsequent round, upon reaching the iteration limit.
Update interval $T_{\text{renew}}$, the main network synchronizes hyperparameters to the target network every 20 rounds.
Learning rate $\alpha$, the learning rate for updating the $Q$.
Gravitational coefficient $K_G$, adjusting gravitational potential field intensity.
Repulsive force coefficient $K_R$, adjusting repulsive potential field intensity.
Discount factor $\gamma$, balancing the trade-off between immediate and future rewards.

Output: Destination arrival path $S$, optimal path $S_{\text{best}}$
for $i = 1$ to $T_{\text{obs}}$ do
    for $T_m = 1$ to $T_{\text{episode}}$ do
        Determine $Q_{\text{now}}$ based on the current $Q_{\text{main}}$ network
        Select action based on time-changing-greedy policy and determine the next position
        Store the obtained environmental observation sequence $(s, a)$ in the experience pool
        if destination is reached then
            Record the path
        end if
    end for
end for
for $j = 1$ to $T_{\text{train}}$ do
    for $T_m = 1$ to $T_{\text{episode}}$ do
        Determine $Q_{\text{now}}$ based on the current $Q_{\text{main}}$ network
        Select action based on time-changing S-greedy policy and determine the next position
        Update the experience pool
        if $T_m \bmod T_{\text{renew}} = 0$ then
            Update the target neural network $Q_{\text{target}}$
        end if
        Update the $Q$ value using temporal difference method and reward function:
        $r = -U_{Gi}(X) - U_{Ri}(X)$
        $U_{Ri}(X) = \begin{cases} K_{Ri} R_a^2 e^{-\rho^2}, & \text{if } \rho \le \rho_d \\ 0, & \text{if } \rho > \rho_d \end{cases}$
        $U_{Gi}(X) = \frac{1}{2} K_G R_a^2$
        Train the $Q_{\text{main}}$ network with the updated $\alpha$ and experience pool data
        if destination is reached then
            Record the path
        end if
    end for
end for
```

The gravitational potential function U_{Gi} and the repulsive potential function U_{Ri} are components of the enhanced APF method detailed in Section 2.1. Equation (12) disregards the direction of the force, instead expressing the force magnitude as a reward value. When the ship approaches the target point, the gravitational force decreases, leading to increased reward. In contrast, approaching an obstacle intensifies the repulsive force, which decreases the reward value r. Integrating the artificial potential field into the reward function ensures continuity, making the deep reinforcement learning search process more directed and boosting the learning efficacy.

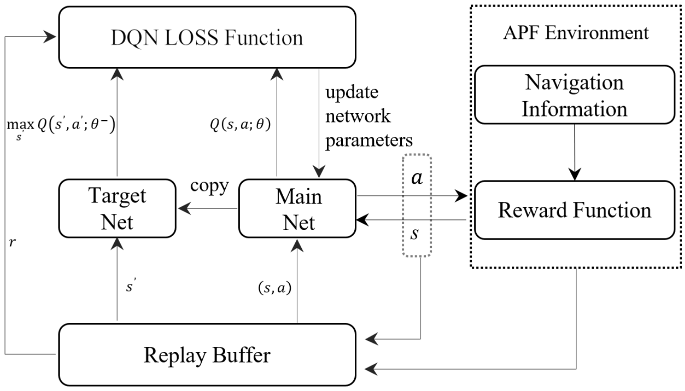


Figure 3. Structure of the MAPF–DQN algorithm.

3.2. Feasibility Enhancement Algorithm

The generated path effectively tackles the issues of local minimum traps and inability to reach the destination present in the artificial potential field method. The path planning algorithm should take into account various factors that affect the maneuvering performance of vessels [34], including vessel type, size, propulsion system, hydrodynamic characteristics, etc. As a result, this section centers on examining the turning behavior of massive ships and optimizing the planned route to coincide with the ship's handling traits, thereby achieving a smoother trajectory. Next, we will explore the acquisition of data regarding the ship's helm position for a 90° heading change across various bearings, while considering the unique attributes of the rudder servo mechanism.

To achieve a realistic simulation, a platform built within Simulink generates the experimental trajectory of the ship's continuous rotation. The structure depicted in Figure 4 integrates the rudder servo system, which includes the rudder angle saturation limit, the rudder angle rate of change limit, and a first-order inertia system. Zhang and Zhang [35] designed the first-order inertia system as the transfer function $\frac{1}{6s+1}$ to simulate the transition process of the ship's rudder angle response.

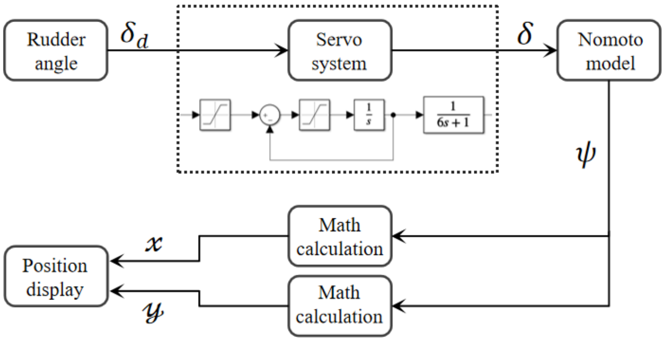


Figure 4. Simulation diagram of ship cyclotron experiment.

The framework specifically incorporates the Nomoto module, which features a nonlinear Nomoto model [36] designed to describe the motion characteristics of ships accurately.

The nonlinear Nomoto model was developed through a combination of theoretical development, empirical observations, and experimental data from model tests and full-scale ship trials. It typically takes the form of a set of nonlinear differential equations that describe the ship's motion in surge, sway. The model parameters can be directly obtained from the actual ships.

$$\ddot{\psi} + \frac{K}{T}\left(\alpha\dot{\psi} + \beta\dot{\psi}^3\right) = \frac{K}{T}\delta + \triangle \tag{13}$$

The parameters K and T are pivotal within this model, representing the ship's maneuverability indices. These indices are not constants but are influenced by a myriad of factors including, yet not limited to, the ship's hydrodynamic design, its operational conditions, the state of the hull, and the environmental conditions such as wind and current. Furthermore, ψ represents the actual heading, δ denotes the rudder angle, and $\triangle$ symbolizes external disturbances. This study focuses solely on examining the turning performance of the ship to ensure a smoother planned path, disregarding the influence of external interference.

To enhance the smoothness of the paths generated by the MAPF–DQN algorithm, it is essential to incorporate precise positional information during ship turning maneuvers, moving beyond sole reliance on the cumulative heading angle calculated from the heading rate of change. This critical conversion is depicted in Equation (14), signifying the computational procedure executed within the simulation environment.

$$\begin{cases} \dot{x} = U\cos\psi \\ \dot{y} = U\sin\psi \end{cases} \tag{14}$$

where (x, y) denotes the ship's position and U represents the magnitude of the ship's velocity. The heading, ψ, is determined by integrating the bow angular velocity.

The temporal lag inherent in the rudder response manifests exclusively during a vessel's 90° course alteration from its point of departure. We selectively preserve the trajectory segment to accurately reflect the practical scenario characterized by a time delay in steering. Subsequently, the preserved segment is expanded upon using the principle of symmetry. As illustrated in Figure 5, the trajectory located within the first quadrant corresponds to the positional data (x, y) acquired from the simulation experiment. The positional information throughout the remaining quadrants is extrapolated by employing the symmetry principle.

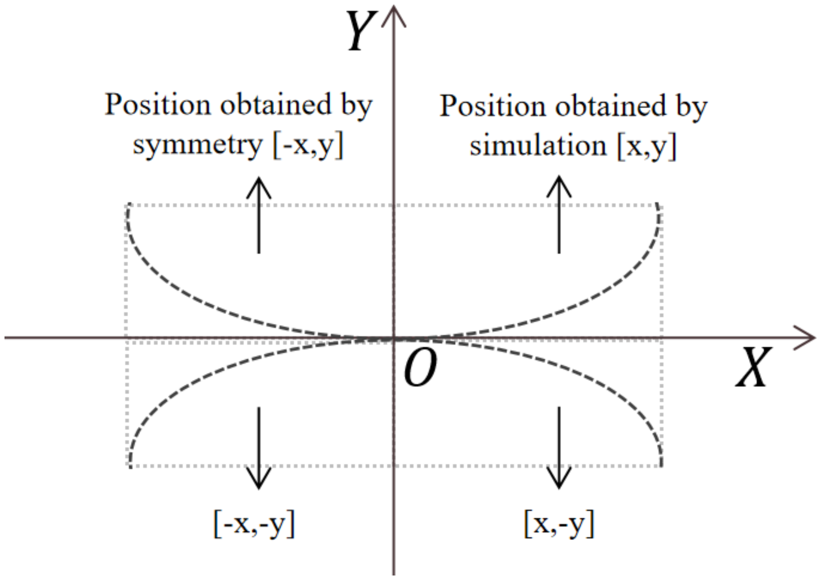

Y
Position obtained by symmetry [-x,y]
Position obtained by simulation [x,y]
O
X
[-x,-y]
[x,-y]

Figure 5. Principle of ship turn information expansion.

4. Experiments and Analyses

4.1. Path Planning Experiments

Within the scope of this experimental study, the ship navigates in a calm sea area with a size of 3000 m × 3000 m, a partitioned grid with dimensions of 30 × 30. The neural network is constructed with two hidden layers, each hosting 60 neurons. The precise parametric configurations are elucidated in Table 1. And the "Observation period" and "Training period" refer to the iterative rounds of the algorithm's training regimen. Each round is a critical phase for the model, integrating observation, decision-making, and learning updates. In all the following experiments, the safety, economy, and practicality of paths are evaluated through various parameters, including the minimal distance from the planned trajectory to obstacles (R_{min}), the length of the planned path (L), and the number of waypoints (N) and the number of turns back in situ (Z).

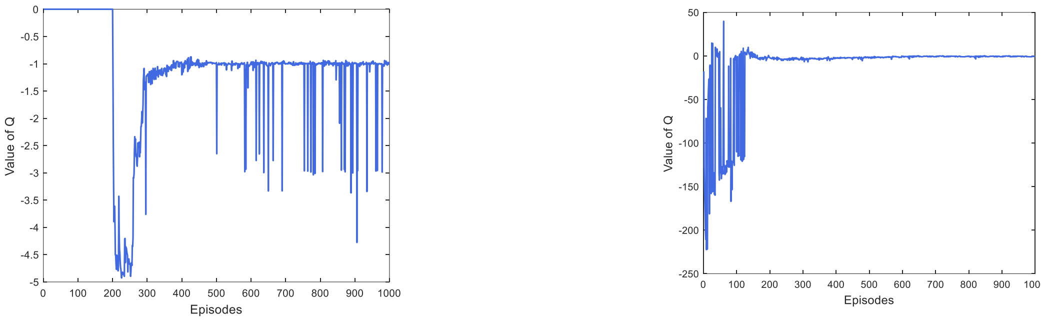
Table 1. Parameters of the MAPF–DQN algorithm.

Parameters	Symbols	Numeric	Parameters	Symbols	Numeric
Observation Period	$T_{obs}\ [times]$	200	Discount factor	γ	0.9
Training period	$T_{train}\ [times]$	800	Gravitational coefficient	K_{Gi}	0.9
Training round	$T_{episode}\ [iterations]$	300	Repulsive force coefficient	K_{Ri}	0.9
Update interval	$T_{renew}\ [times]$	20	Learning rate	α	1

4.1.1. Collision Avoidance Experiment in Conventional Narrow Waterways

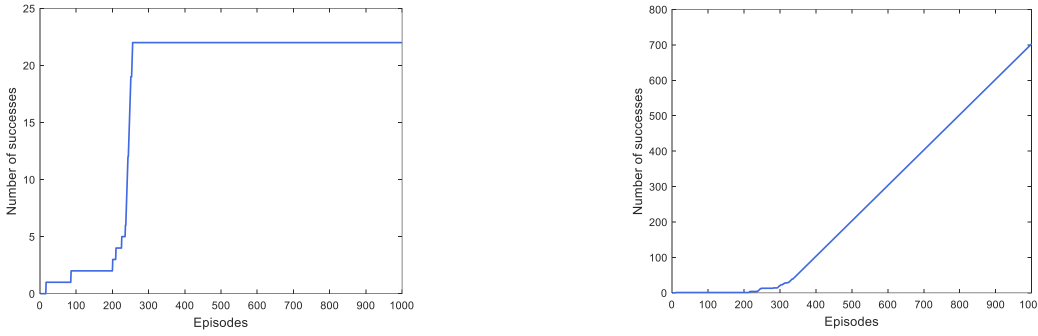
This investigation employs simulations to compare the MAPF–DQN, A*, and DQN algorithms for marine obstacle avoidance within an identical maritime environment, thereby assessing the efficacy of the MAPF–DQN approach. The results of these experiments are illustrated in Figures 6–8. Specifically, Figure 8 depicts the obstacle field in black, the successfully navigated path of the ship during learning training in blue, and the optimal planned path achieved during training, characterized by the least number of waypoints, in green.

Figure 6 illustrates the evolution of Q during the training of both DQN and MAPF–DQN algorithms. Q represents the expected cumulative reward for state–action pairs, assisting the agent in evaluating the contribution of each action to long-term rewards, thereby enhancing learning and decision-making. In the training of MAPF–DQN, the Q stabilizes after 200 iterations, indicating that the learning process may have converged to a steady state. The agent has learned to take optimal actions in given states to maximize its expected return. Figure 7 depicts the temporal evolution of successful path-finding attempts during the learning process, serving to evaluate the learning progress and performance enhancement of the MAPF–DQN algorithm in path planning tasks. Over time, there is a steady increase in the number of successful path-finding instances, indicating iterative optimization and effective adoption of environment-appropriate path planning strategies by the MAPF–DQN algorithm.



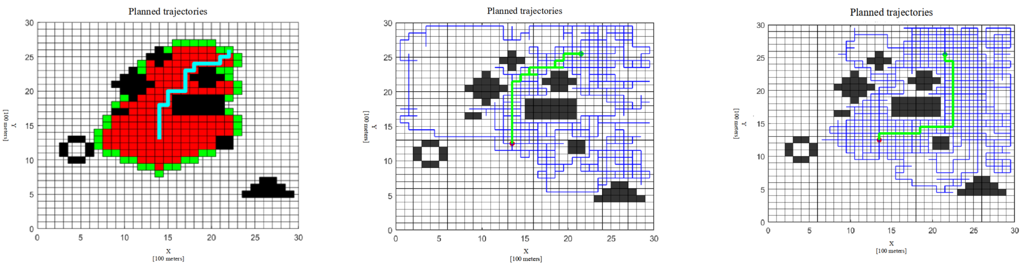
(**a**) (**b**)

Figure 6. Changes in Q value of obstacle avoidance experiment. (**a**) shows the Q-value variation process of DQN algorithm; (**b**) shows the process of Q value variation of MAPF-DQN algorithm.



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

Figure 7. Times of success of obstacle avoidance experiment. (**a**) shows the successful times of DQN algorithm; (**b**) shows the successful times of MAPF-DQN algorithm.



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

Figure 8. The planning path of obstacle avoidance experiment. (**a**) shows the planning path of A* algorithm; (**b**) shows the planning path of DQN algorithm; (**c**) shows the planning path of MAPF-DQN algorithm. (The green path of the DQN algorithm is obtained by the 250th cycle of the search. The green path of the MAPF–DQN algorithm is obtained by searching for the 968th cycle).

Figures 6 and 7 illustrate that the DQN algorithm randomly explored two distinct paths during the observational period. Although there was evidence of learning in the initial stages, the algorithm failed to identify a successful trajectory to the target point even after 300 rounds of training. This failure could be attributed to an error during this phase, which subsequent learning sessions did not rectify. Moreover, the learned strategy proved inadequate in guiding the vessel precisely to the target point by the conclusion of the training. In contrast, the number of successfully identified paths increased consistently when the MAPF–DQN algorithm commenced its training phase, demonstrating its capability to learn and avoid collisions with stationary obstacles. In Figure 8, the optimal path was achieved in the 968th cycle rather than the final 1000th cycle. This occurrence is due to the probability that the vessel may seek alternative trajectories based on the applied greedy strategy. However, these subsequent paths did not result in improvements beyond those attained in the 968th iteration. Notably, the frequency of reaching the target point increased from 22 to 703 times, signifying an improvement of over 30-fold. The significant increase suggests that the MAPF–DQN algorithm enhances the likelihood of converging upon an optimal learned strategy. By integrating the APF method's physical model with the data-driven DQN algorithm, the MAPF–DQN approach significantly boosts the latter's learning efficiency.

Figure 8 depicts the optimal paths planned by the A* algorithm, the DQN algorithm, and the MAPF–DQN algorithm. The A* algorithm selects a path along the edges of obstacles, which is the shortest path. However, the path is too close to obstacles which raises the risk of navigating narrow waterways. It could stem from the use of the Euclidean distance as the heuristic function, which, while enabling the identification of the shortest path, may not be suitable for real-world scenarios. Factors such as the representation of obstacles, the appropriateness of the heuristic function, specifics of the algorithm's implementation, the presence of local optima, and the choice of algorithm parameters could all contribute to the impracticality of the path generated. On the other hand, the optimal path obtained by the DQN algorithm traverses an area filled with obstacles while maintaining a certain distance from them. However, this path contains the most waypoints. Therefore, frequent adjustments to the rudder and potential reversals in heading pose significant challenges to the execution work and hurt navigation efficiency. In contrast, the trajectory computed by the MAPF–DQN algorithm, as shown in Figure 8, clearly lacks the local minima and unreachable target issues commonly associated with artificial potential field methods. Compared to the DQN algorithm, the best trajectory generated by the MAPF–DQN algorithm effectively passes through sparsely populated obstacle areas. It reaches the destination with fewer waypoints and wider spacing. Furthermore, the path generated by the MAPF–DQN algorithm focuses on the endpoint, thanks to the direction guidance imposed by the gravitational vectors. As shown in Table 2, the path planned by the MAPF–DQN algorithm exhibits improvements in safety, operational economy, and practical feasibility.

Table 2. Results of qualitative analysis. (Bold numbers indicate the best performance).

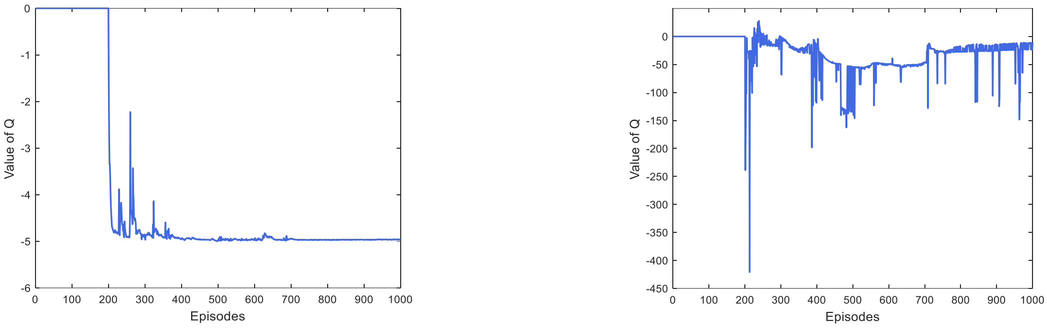
	A* Algorithm	**DQN Algorithm**	**MAPF–DQN Algorithm**
R_{min} $[m]$	50	50	**150**
L $[m]$	**2100**	2700	2300
N $[number]$	10	9	**6**
M $[number]$	-	22	**703**
Z $[number]$	**0**	2	**0**

In conclusion, the MAPF–DQN algorithm for planning routes enhanced learning efficiency, safety, economics, and feasibility. However, the presence of six right-angle bends in the track results in abrupt 90° changes in direction, which deviate from the actual sailing trajectory of large ships.

4.1.2. Collision Avoidance Experiment on U-Shaped Obstacle

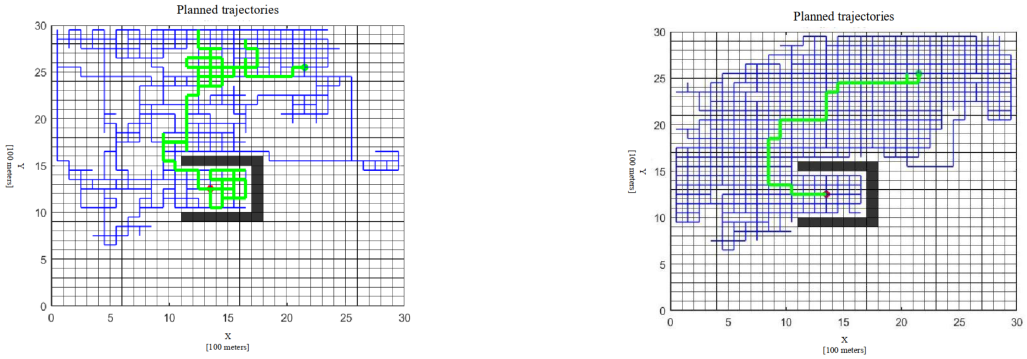
During actual maritime navigation, vessels routinely engage in berthing and unberthing maneuvers, with ports often characterized by U-shaped configurations. The conventional artificial potential field method for obstacle avoidance may encounter entrapment at specific points within such U-shaped geometries. We conduct a comparative simulation with the DQN algorithm for U-shaped obstacle avoidance to validate the efficacy of the MAPF–DQN algorithm in resolving local minimum traps issues. The results of the experimental evaluation are presented in Figures 9 and 10.

Figure 9 documents the evolution of Q-values for DQN and MAPF–DQN algorithms when encountering a U-shaped obstacle. The DQN algorithm exhibits relatively stable Q-value changes but stabilizes at -5, significantly deviating from the ideal target value of 0, indicating ineffective learning of strategies to reach the goal. In contrast, despite MAPF–DQN showing more pronounced Q-value fluctuations during training, it converges successfully to the ideal value of 0. This demonstrates MAPF–DQN's ability to learn strategies guiding the agent to navigate obstacles and reach the target effectively. The traditional artificial potential field method often exhibits suboptimal performance in U-shaped obstacles with a propensity to deadlock. Conversely, the DQN algorithm circumvents this issue; however, it suffers from diminished learning efficiency due to the reward function's sparsity, culminating in an impractical final trajectory. As demonstrated in Figure 9, the optimal path is achieved during the seventh iterative search, indicating that the terminally trained network does not converge to an optimal policy. In stark contrast, the MAPF–DQN algorithm outperforms its DQN counterpart in learning efficiency and path practicality. Furthermore, the algorithm effectively negotiates escape from U-shaped impediments, alleviating predicaments such as local minimum entrapment.



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

Figure 9. Changes in Q value of U-shaped obstacle experiment. (**a**) shows the Q-value variation process of DQN algorithm; (**b**) shows the process of Q value variation of MAPF-DQN algorithm.

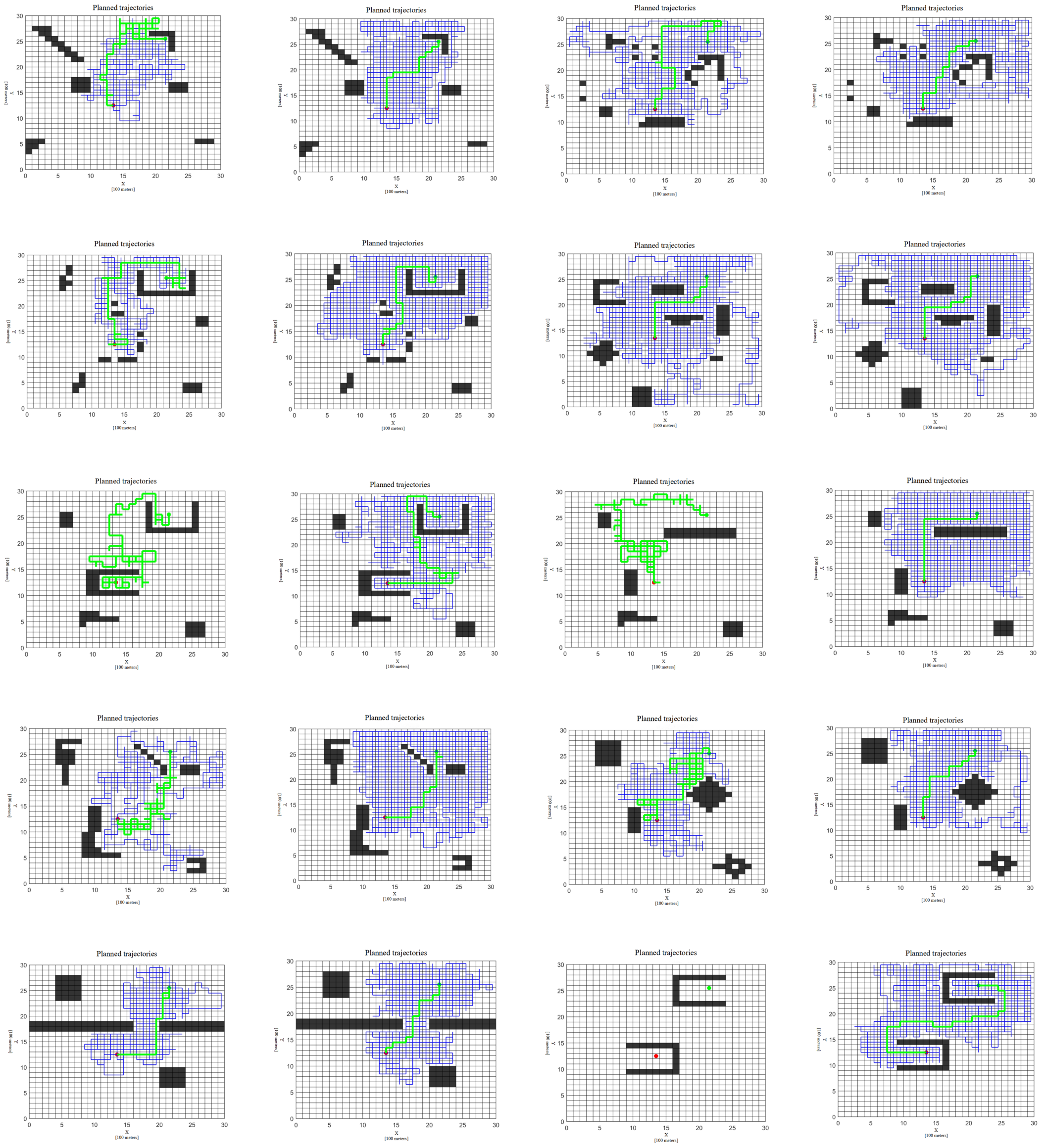


(**a**) (**b**)

Figure 10. The planning path of U-shaped obstacle experiment. ((**a**) The green path of the DQN algorithm is obtained by the 7th cycle of the search. (**b**) The green path of the MAPF–DQN algorithm is obtained by searching for the 804th cycle).

4.1.3. Collision Avoidance Experiments across Diverse Scenarios

This experiment aims to investigate the generalisability of the MAPF–DQN algorithm in various navigational environments and the performance difference of the DQN algorithm. As the previous comparative experiments were limited to specific environments, this study designed a set of experiments covering ten different navigation environments and conducted a detailed evaluation of the performance of both algorithms in these environments. The specific experimental results are detailed in Figure 11. In Figure 11, the green line distinctly marks the optimal path planned by the algorithm. In contrast, the blue lines represent the collection of all paths successfully reaching the target during the training process. We have set up a hypothetical navigation environment primarily consisting of narrow channels, and have specifically included challenging special channels, such as double-U-shaped obstacles. These complex navigation conditions pose a severe challenge for path-planning algorithms. Through comparative validation of experimental results, the advantages of the MAPF–DQN algorithm in planning paths in complex navigation environments are proven. Tables 3 and 4 illustrate the specific performance indicators of the DQN and MAPF–DQN algorithms in different navigation environments. To clearly demonstrate the changes in the MAPF–DQN algorithm on the five metrics, this study used the indicator values of the DQN algorithm as a baseline, calculated the relative indicator values of the MAPF–DQN algorithm, and provided the difference in values. However, since the optimal paths of the DQN algorithm in the respective harsh environments could not be counted for some of the metrics, they were replaced with the maximum values from other experiments, as shown in Figure 11. By analyzing these data, we aim to reveal the adaptability advantages and disadvantages of the two algorithms in diverse sailing environments.



(**a**) DQN (**b**) MAPF-DQN (**c**) DQN (**d**) MAPF-DQN

Figure 11. Results of the generalizability experiment.

The experimental results are shown in Figures 11 and 12. After comparing the number of successfully searched paths between the MAPF–DQN algorithm and the DQN algorithm in ten independent experiments, we found that the average number of successful searches for the MAPF–DQN algorithm was significantly higher than that of the DQN algorithm. The enhancement can be attributed to the revised reward function, which explicitly defines the rewards and aids in generating paths that are closer to the target point during the learning process. As a result of the improved search efficiency, the selected optimal paths also showed partial improvement in other performance metrics. Although the paths generated by the DQN algorithm had one less turn than the MAPF–DQN algorithm in the fourth experiment, the MAPF–DQN algorithm still demonstrated an advantage in the number of successful searches. This may indicate that the DQN algorithm probabilistically found a superior path in this experiment. However, the result cannot prove that the DQN algorithm outperforms the MAPF–DQN algorithm overall. In particular, in the experimental scenario containing two reversed U-shaped obstacles, the DQN algorithm failed to successfully discover any paths to the target point during the learning process. The MAPF–DQN algorithm successfully found 506 valid paths, and the final selected paths were relatively high in quality. Therefore, the advantage of the MAPF–DQN algorithm is more significant in more challenging environments.

Table 3. DQN algorithm metrics in generalization experiments. (- indicates that the value could not be counted).

	R_{min} $[m]$	L $[m]$	N $[number]$	Z $[number]$	M $[number]$
Scenario 1	50	9300	-	6	6
Scenario 2	50	4700	18	2	15
Scenario 3	50	10,800	22	6	4
Scenario 4	150	2000	6	0	621
Scenario 5	50	20,400	-	8	1
Scenario 6	50	18,500	-	-	1
Scenario 7	50	10,300	-	11	4
Scenario 8	50	13,900	-	5	8
Scenario 9	50	2700	9	0	26
Scenario 10	-	-	-	-	0
Average value	60	11,300	25.3	6	68.6

Table 4. MAPF–DQN algorithm metrics in generalization experiments.

	R_{min} $[m]$	L $[m]$	N $[number]$	Z $[number]$	M $[number]$
Scenario 1	50	2100	10	0	777
Scenario 2	150	2100	11	0	790
Scenario 3	50	3100	13	1	233
Scenario 4	150	2000	7	0	750
Scenario 5	50	6900	28	4	19
Scenario 6	150	2100	2	0	762
Scenario 7	50	2700	10	1	719
Scenario 8	250	2100	10	0	781
Scenario 9	150	2100	12	0	779
Scenario 10	150	4300	16	0	506
Average value	120	2950	11.9	0.6	611.6

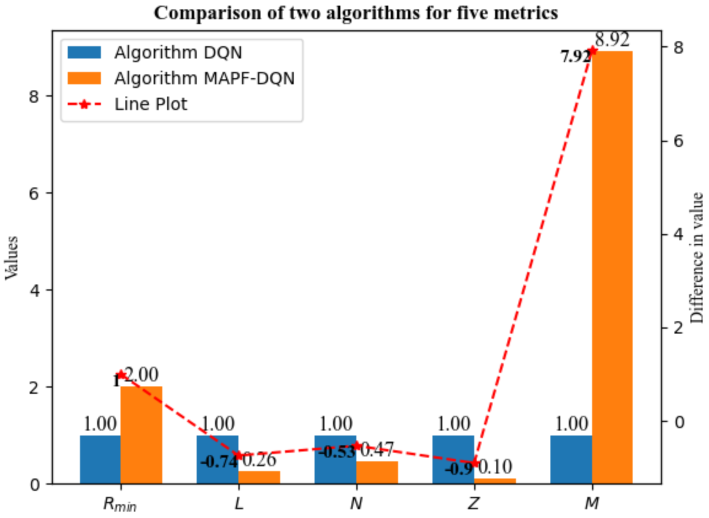


Figure 12. Comparison of two algorithms for five metrics.

In summary, the MAPF–DQN algorithm demonstrates strong generalization ability, can adapt to a variety of navigational environments, and shows better comprehensiveness in path selection.

4.2. Feasibility Enhancement Experiment

In this experiment, the path planned by the MAPF–DQN algorithm in Section 4.1.1 is processed to match the turning performance of "Yupeng", as shown in Figure 8. The principal parameters of the ship are delineated in Table 5. The derived parameters for the nonlinear Nomoto model are $K = 0.21, T = 107.78, \alpha = 13.17, \beta = 16,323.89$, as cited from Zhang and Zhang [35]. An examination of the rudder dynamics of the subject vessel reveals a maximum rudder angle of ±35° and a maximum steering rate of $\pm 5^{\circ}/\mathrm{s}$.

Table 5. Ship type parameters during the ballast of the practice ship "Yupeng".

Variant	Numeric	Variant	Numeric
Length between perpendiculars	189 m	Width	27.8 m
Draft	6.313 m	Discharge volume	22,036.7 m^3
Distance from center of gravity	−4.043 m	Block coefficient	0.661
Rudder area	31.67 m^2	Speed	17.26 kn

Figure 13 depicts the outcomes of the simulation experiment, revealing a longitudinal tactical diameter of approximately 694 m—equivalent to 3.7 times the vessel's length; and a transverse tactical diameter of about 717 m, which is 3.8 times the ship's length. These findings are consistent with the actual "Yupeng" vessel's constant rotation experimental results. For instance, Figure 14 illustrates the trajectory planned by the MAPF–DQN algorithm, as detailed in Section 4.1.1, wherein the "Yupeng" vessel initiates its journey from the starting point and navigates through an array of obstacles before finally reaching the destination. The red dot indicates the start point, the green dot the end point, the blue line the planned trajectory, and the black + the rudder position. The red dot signifies the starting point, the green dot denotes the endpoint, the blue line illustrates the planned trajectory, and the black "+" indicates the steering position. En route, the ship executes six turns that adhere to its maneuvering characteristics and maintains a safe distance of no less than 150 m from the obstacles.

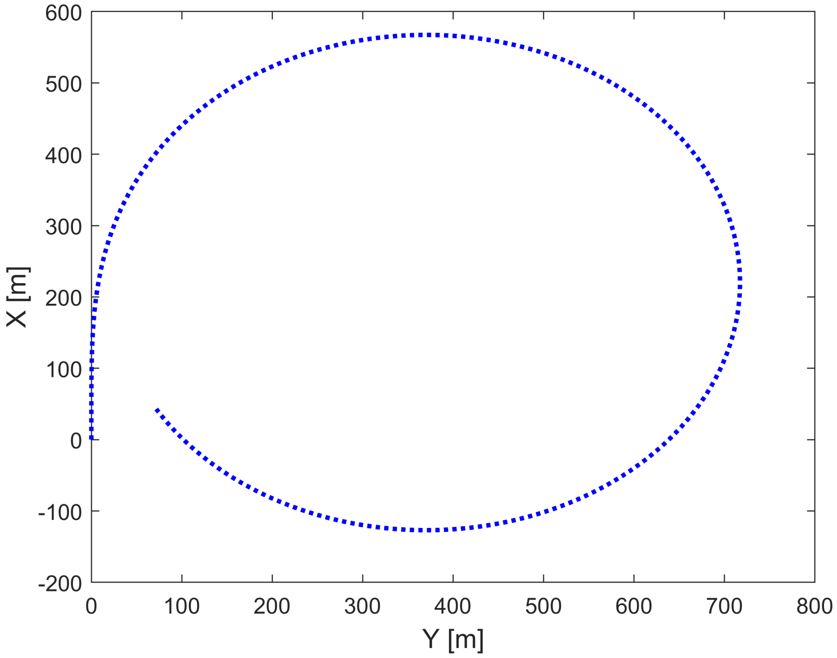


Figure 13. Results of the ship slewing experiment.

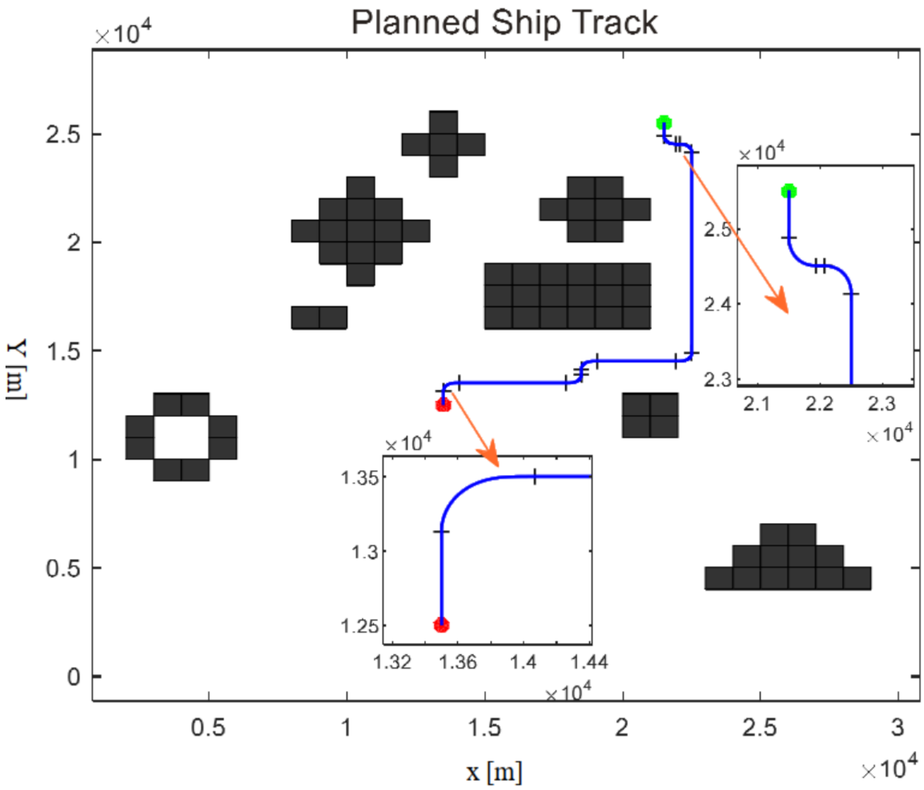


Figure 14. Final trajectory for path planning (+ indicates specific rudder position).

4.3. Dicussion for Seafarers-Related Training Issues

Role and Skill Requirement Transformation. The transition to automated vessels, exemplified by sophisticated algorithms such as MAPF–DQN, has redefined the maritime professional's role, evolving from manual operation to strategic oversight and technical stewardship. Seafarers must now be adept in the intricacies of advanced navigation systems, prepared to manage and troubleshoot in real time, especially in critical scenarios like navigating through hazardous areas with impaired or disabled AIS and radar systems. In such instances, relying on manual observations through telescopes to delineate no-go zones within the MAPF–DQN algorithm, seafarers must demonstrate agility and resourcefulness to ensure safe passage.

Evolution of Training Needs. With the advent of automation, seafarer training has become more complex, requiring not only a grasp of high-tech systems but also proficiency in traditional navigation skills. Training must encompass emergency response to situations where automated systems like the AIS and radar may be compromised, necessitating manual intervention to input no-go zones into the MAPF–DQN for safe navigation. This highlights the need for a dual-skills approach: maintaining traditional seamanship while advancing technical expertise.

Managerial Implications. Our research significantly impacts maritime management by enhancing safety, efficiency, and adaptability. The MAPF–DQN algorithm can assist shipping companies in optimizing routes, reducing fuel consumption, and minimizing travel time, which directly translates to cost savings and improved operational efficiency. Moreover, the ability to rapidly adapt to dynamic maritime conditions and emergencies can significantly improve safety standards.

Challenges in Management and Regulation. The shift toward automation introduces challenges in management and regulatory oversight. It requires the development of contingency protocols for scenarios wherein standard navigation aids are non-operational, and seafarers must manually guide the ship's path planning. Regulatory bodies must update standards to accommodate such manual interventions, ensuring they are safely and effectively integrated with automated systems like MAPF–DQN.

In conclusion, the maritime industry's progression toward automation demands a seafarer workforce that is versatile, capable of addressing both routine operations and unexpected emergencies. The integration of MAPF–DQN with traditional navigation skills underscores the need for a comprehensive training regimen that prepares seafarers to navigate safely, even when faced with the unexpected challenges of modern seafaring.

5. Conclusions

This paper addresses the static obstacle avoidance problem for large ships by designing a local path planning algorithm that combines deep reinforcement learning with the artificial potential field method. The algorithm consists of two main components: path planning and feasibility enhancement. In the path planning phase, the planned path in the simulation experiment overcomes local minimum traps and the inability to reach the destination. Specifically, the path search is more targeted, generalizable to various environments, and search efficiency is improved. After enhancing the feasibility of the paths found by the MAPF–DQN algorithm, the path turns are smoother and meet the maneuvering characteristics of the "Yupeng" ship. In conclusion, the path planned by the MAPF–DQN algorithm exhibits safety, economy, and feasibility, while also demonstrating improved learning efficiency. However, the MAPF–DQN algorithm faces challenges in rapidly adapting to dynamic environmental changes and does not yet incorporate collision avoidance regulations, potentially impacting its practicality in maritime navigation. The phased approach of the algorithm increases computational demands, which could hinder its real-time capabilities in systems with limited resources. To enhance the performance and practicality of the algorithm, we need to focus on several areas in the future. First, to improve the responsiveness of the algorithm so that it can rapidly adapt to dynamic changes in sea conditions; second, to integrate collision avoidance protocols to ensure the safety of vessels during navigation; in addition, we need to enhance the generalizability of the algorithm to cope with the constant changes in the navigation environment; finally, through testing in sea conditions, we continue to optimize and refine the performance of the algorithm.

Author Contributions: Conceptualization, X.Z., X.G. and J.C.; Methodology, W.X., X.Z., X.G. and C.S.; Software, W.X., X.Z. and C.S.; Validation, W.X. and X.L.; Formal analysis, J.C.; Investigation, J.C.; Resources, X.L. and J.C.; Writing— original draft, X.Z. and X.G.; Writing—review and editing, X.G., X.L. and X.R.; Visualization, X.R.; Project administration, X.R. and C.S.; Funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant Nos. 42275170, 42305170, 62202487), the Science and Technology Innovation Program of Hunan Province (2022RC3070), and the Natural Science Foundation of Hunan Province (Grant No. 2023JJ40678).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data presented in this study were obtained from simulation experiments. The algorithms used for these simulations are transparently described in the paper, and readers can independently design algorithms to generate similar data based on the provided pseudocode.

Conflicts of Interest: The authors declare no conflicts of interest.

Note

[1] CSDN, 2022. Dqn shortest path algorithm, matlab implementation with interface, ready to run!

References

1. Trucco, P.; Cagno, E.; Ruggeri, F.; Grande, O. A Bayesian Belief Network modelling of organisational factors in risk analysis: A case study in maritime transportation. *Reliab. Eng. Syst. Saf.* **2008**, *93*, 845–856. [CrossRef]

2. Sahu, B.; Das, P.K.; Kumar, R. A modified cuckoo search algorithm implemented with SCA and PSO for multi-robot cooperation and path planning. *Cogn. Syst. Res.* **2023**, *79*, 24–42. [CrossRef]
3. Liu, J.; Yang, J.; Liu, H.; Tian, X.; Gao, M. An improved ant colony algorithm for robot path planning. *Soft Comput.* **2017**, *21*, 5829–5839. [CrossRef]
4. Luo, Q.; Wang, H.; Zheng, Y.; He, J. Research on path planning of mobile robot based on improved ant colony algorithm. *Neural Comput. Appl.* **2020**, *32*, 1555–1566. [CrossRef]
5. Wang, X.; Feng, K.; Wang, G.; Wang, Q. Local path optimization method for unmanned ship based on particle swarm acceleration calculation and dynamic optimal control. *Appl. Ocean Res.* **2021**, *110*, 102588. [CrossRef]
6. Long, Y.; Su, Y.; Lian, C.; Zhang, D. Hybrid bacterial foraging algorithm for unmanned surface vehicle path planning. *J. Huazhong Univ. Sci. Technol. (Natural Sci. Ed.)* **2022**, *50*, 68–73. [CrossRef]
7. Mao, S.; Yang, P.; Gao, D.; Liu, Z. Path planning for unmanned surface vehicle based on bacterial foraging-improved ant colony optimization algorithm. *Control Eng. China* **2024**, *31*, 608–616. [CrossRef]
8. Yu, B.; Chu, X.; Liu, C.; Zhang, H.; Mao, Q. A path planning method for unmanned waterway survey ships based on improved A* algorithm. *Geomat. Inf. Sci. Wuhan Univ.* **2019**, *44*, 1258–1264. [CrossRef]
9. Li, W.; Wang, L.; Fang, D.; Li, Y.; Huang, J. Path planning algorithm combining A* with DWA. *Syst. Eng. Electron.* **2021**, *43*, 3694–3702. [CrossRef]
10. Liu, Y.; Wang, T.; Xu, H. PE-A* algorithm for ship route planning based on field theory. *IEEE Access* **2022**, *10*, 36490–36504. [CrossRef]
11. Zhang, Y.; Wen, Y.; Tu, H. A method for ship route planning fusing the ant colony algorithm and the a* search algorithm. *IEEE Access* **2023**, *11*, 15109–15118. [CrossRef]
12. Fransen, K.; van Eekelen, J. Efficient path planning for automated guided vehicles using A*(Astar) algorithm incorporating turning costs in search heuristic. *Int. J. Prod. Res.* **2023**, *61*, 707–725. [CrossRef]
13. Lyu, H.; Yin, Y. COLREGS-constrained real-time path planning for autonomous ships using modified artificial potential fields. *J. Navig.* **2019**, *72*, 588–608. [CrossRef]
14. Liu, M.; Feng, H.; Xu, H. Dynamic path planning for unmanned surface vessel based on improved artificial potential field. *J. Ship Mech.* **2020**, *24*, 1625–1635. [CrossRef]
15. Wang, J.; Xiao, Y.; Li, T.; Chen, C.L.P. A jamming aware artificial potential field method to counter GPS jamming for unmanned surface ship path planning. *IEEE Syst. J.* **2023**, *17*, 4555–4566. [CrossRef]
16. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
17. Khan, W.A.; Chung, S.H.; Awan, M.U.; Wen, X. Machine learning facilitated business intelligence (Part I) Neural networks learning algorithms and applications. *Ind. Manag. Data Syst.* **2020**, *120*, 164–195. [CrossRef]
18. Khan, W.A.; Chung, S.H.; Awan, M.U.; Wen, X. Machine learning facilitated business intelligence (Part II) Neural networks optimization techniques and applications. *Ind. Manag. Data Syst.* **2020**, *120*, 128–163. [CrossRef]
19. Guo, S.; Zhang, X.; Du, Y.; Zheng, Y.; Cao, Z. Path planning of coastal ships based on optimized DQN reward function. *J. Mar. Sci. Eng.* **2021**, *9*, 210. [CrossRef]
20. Chun, D.H.; Roh, M.I.; Lee, H.W.; Ha, J.; Yu, D. Deep reinforcement learning-based collision avoidance for an autonomous ship. *Ocean Eng.* **2021**, *234*, 109216. [CrossRef]
21. Liu, J.; Shi, G.; Zhu, K.; Shi, J. Research on mass collision avoidance in complex waters based on deep reinforcement learning. *J. Mar. Sci. Eng.* **2023**, *11*, 779. [CrossRef]
22. Jiang, R.; Yu, W.; Liao, W.; Wang, J. Optimal energy consumption based path planning for intelligent all-electric ships. *Shipbuild. China* **2021**, *62*, 245–254. [CrossRef]
23. Ohn, S.W.; Namgung, H. Requirements for optimal local route planning of autonomous ships. *J. Mar. Sci. Eng.* **2022**, *11*, 17. [CrossRef]
24. Zhang, L.; Mou, J.; Chen, P.; Li, M. Path planning for autonomous ships: A hybrid approach based on improved APF and modified VO methods. *J. Mar. Sci. Eng.* **2021**, *9*, 761. [CrossRef]
25. Lyu, H.; Yin, Y. Ship's trajectory planning for collision avoidance at sea based on modified artificial potential field. In Proceedings of the 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), Shanghai, China, 29–31 December 2017; pp. 351–357. [CrossRef]
26. Shen, H.; Guo, C.; Li, T.; Yu, Y. An intelligent collision avoidance and navigation approach of unmanned surface vessel considering navigation experience and rules. *J. Harbin Eng. Univ.* **2018**, *39*, 998–1005. [CrossRef]
27. Wen, N.; Long, Y.; Zhang, R.; Liu, G.; Wan, W.; Jiao, D. COLREGs-based path planning for USVs using the deep reinforcement learning strategy. *J. Mar. Sci. Eng.* **2023**, *11*, 2334. [CrossRef]
28. Du, Y.; Zhang, X.; Cao, Z.; Wang, S.; Liang, J.; Zhang, F.; Tang, J. An optimized path planning method for coastal ships based on improved DDPG and DP. *J. Adv. Transp.* **2021**, *2021*, 7765130. [CrossRef]
29. Chen, C.; Chen, X.Q.; Ma, F.; Zeng, X.J.; Wang, J. A knowledge-free path planning approach for smart ships based on reinforcement learning. *Ocean Eng.* **2019**, *189*, 106299. [CrossRef]
30. Yang, Q.; Wang, S.; Sang, J.; Wang, C.; Huang, G.; Wu, C.; Song, S. Path planning and real-time obstacle avoidance methods of intelligent ships in complex open water environment. *Comput. Integr. Manuf. Syst.* **2022**, *28*, 2030–2040. [CrossRef]

31. Li, L.; Wu, D.; Huang, Y.; Yuan, Z.M. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean Res.* **2021**, *113*, 102759. [CrossRef]
32. Öztürk, Ü.; Akdağ, M.; Ayabakan, T. A review of path planning algorithms in maritime autonomous surface ships: Navigation safety perspective. *Ocean Eng.* **2022**, *251*, 111010. [CrossRef]
33. Liu, K.; Zhang, Y.; Ren, J. Path planning algorithm for unmanned surface vehicle based on an improved artificial potential field method. *Nat. Sci. J. Hainan Univ.* **2016**, *34*, 99–104. [CrossRef]
34. Vo, A.K.; Mai, T.L.; Yoon, H.K. Path planning for automatic berthing using ship-maneuvering simulation-based deep reinforcement learning. *Appl. Sci.* **2023**, *13*, 12731. [CrossRef]
35. Zhang, X.; Zhang, G. *Nonlinear Feedback Theory and Its Application to Ship Motion Control*; Dalian Maritime University Press: Dalian, China, 2020.
36. Zhao, B.; Zhang, X. An improved nonlinear innovation-based parameter identification algorithm for ship models. *J. Navig.* **2021**, *74*, 549–557. [CrossRef]

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