Journal of
Marine Science and Engineering

MDPI

Article

Adaptive Sampling Path Planning for a 3D Marine Observation Platform Based on Evolutionary Deep Reinforcement Learning

Jingjing Zhang *, Yanlong Liu and Weidong Zhou

College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China
* Correspondence: zhangjingjing@hrbeu.edu.cn

Abstract: Adaptive sampling of the marine environment may improve the accuracy of marine numerical prediction models. This study considered adaptive sampling path optimization for a three-dimensional (3D) marine observation platform, leading to a path-planning strategy based on evolutionary deep reinforcement learning. The low sampling efficiency of the reinforcement learning algorithm is improved by evolutionary learning. The combination of these two components as a new algorithm has become a current research trend. We first combined the evolutionary algorithm with different reinforcement learning algorithms to verify the effectiveness of the combination of algorithms with different strategies. Experimental results indicate that the fusion of the two algorithms based on a maximum-entropy strategy is more effective for adaptive sampling using a 3D marine observation platform. Data assimilation experiments indicate that adaptive sampling data from a 3D mobile observation platform based on evolutionary deep reinforcement learning improves the accuracy of marine environment numerical prediction systems.

Keywords: marine environment observation; evolutionary learning; reinforcement learning; path planning; deep learning

check for updates

Citation: Zhang, J.; Liu, Y.; Zhou, W. Adaptive Sampling Path Planning for a 3D Marine Observation Platform Based on Evolutionary Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2023**, *11*, 2313. https://doi.org/10.3390/jmse11122313

Received: 2 November 2023
Revised: 28 November 2023
Accepted: 5 December 2023
Published: 7 December 2023

1. Introduction

Adaptive sampling is an important component of marine environment monitoring, which plays an important role in improving the accuracy of marine environment numerical prediction systems [1–3]. The limited coverage and high cost of direct observations make it unrealistic to carry out large-scale, long-term observations, and limited resources are the main obstacle to the development of complex regional marine observation and prediction technologies. Adaptive sampling of regional ocean data using mobile observation platforms (MOPs) in a complex and dynamic underwater setting with limited resources is a challenge in the field of marine monitoring [4,5].

Marine observation path-planning technology has been advanced in recent decades. Due to the complexity and temporal variability of the marine environment, as well as the diversity of autonomous underwater vehicle (AUV) planning tasks [6,7], adaptive sampling with MOPs requires efficient path-planning technology to ensure the smooth completion of tasks. Algorithms currently applied in path planning include the Dijkstra [8], Bellman Ford [9], Floyd–Warshall [10], A-star [11], Dynamic Programming [12], Artificial Potential Field [13], and Linear Quadratic [14] algorithms. Heuristic algorithms have also been applied, including the Genetic [15], Ant Colony [16], and Particle Swarm Optimization [17] algorithms.

Many researchers have used neural networks (NNs) in the path planning field to deal with nonlinearities and uncertainties in the environment [18]. Recently, they have been widely applied to path planning problems [19,20]. As can be seen, 2D path planning based on NNs is not a novelty [21]. For instance, Tamar et al. [22] proposed the Value Iteration Network (VIN) algorithm to approximate the Bellman update using a convolutional neural network (CNN). Ref. [23] was the first introduced the bioinspired neural network (BNN)

method to solve the path planning problem of mobile robots in 1998. Compared with other NN algorithms, BNN algorithms do not need to be learned [24]. The dynamic activation value function of each neuron cell can be solved through information transfer between neurons. Godio et al. [25] described an approach based on bio-inspired neural networks that can solve the coverage planning problem for unmanned aerial vehicle formations exploring critical areas. An approach that combines a bio-inspired neural network and the potential field was suggested in [26] to address the safety issue of autonomous underwater robot path planning in dynamic and uncertain situations. In this method, a bio-inspired neural network uses the environment to determine the best course for an autonomous underwater robot. The path of the bio-inspired neural network is modified by the potential field function such that the autonomous underwater robot can avoid obstructions [27]. The experimental findings demonstrated that the strategy strikes a balance between autonomous underwater robot safety and path logic. The intended routes can accommodate the need for navigation in dynamic and unpredictable surroundings. It should be noted that although 3D path planning plays a vital role in autonomous robots, it is rarely handled by NNs because of their drawbacks of high complexity, hyperparameters, and non-optimization [28].

With the development of artificial intelligence algorithms, applied reinforcement learning (RL) has been introduced to path planning and navigation. Wang et al. [29] proposed a distributed deep reinforcement learning (DRL) algorithm for unmanned surface vehicle (USV) formations based on the learning of two key abilities, adaptability and scalability, such that the formation can arbitrarily increase the number of USVs or change the formation's shape. A path-planning strategy based on DRL and collision-avoidance functions was found to solve path-planning problems associated with unmanned craft in uncertain environments [30]. A continuous hybrid model-free RL method based on a deterministic policy gradient [31] has been applied to adjust the framework of SMC parameters to control the course of AUVs. A model-based RL method for organizing unmanned surface vessels (USV) when searching for multiple moving targets has been proposed [32]. Zheng and Liu [33] improved the deep deterministic policy gradient (DDPG) method, added a mean field network to maximize multi-agent return, and achieved good crowd-evacuation path-planning results in a crowded simulation system. Muse and Wermter [34] applied actor-critical algorithms to platform-independent robot control for navigation tasks in complex environments. Lachekhab and Tadjine [35] fused fuzzy and actor-critical algorithms, added heuristic methods to improve the robustness of the system, and achieved the smooth navigation of a robot.

There has also been progress in multi-agent path planning. Etinkaya et al. [36] proposed a multi-agent path-planning method based on DRL to provide real-time solutions for path-planning problems. For meal delivery and the virtual service market, this method enables such services to quickly meet different needs. For real-time solutions of the dynamic MAPF problem, multi-agent path optimization has been applied [37] with a decentralized multi-agent reinforcement learning framework based on a multi-step pre-tree search strategy to improve decision-making efficiency. This algorithm can be extended to a wide range of multi-agent environments within an acceptable response time. Multi-agent navigation in a dynamic environment has important practical implications for the deployment of large-scale robot fleets.

Recent studies have integrated RL and evolutionary reinforcement (ER) algorithms, combining their respective advantages to develop new algorithms. For example, Liu et al. [38] proposed a decentralized and locally observable multi-agent path-planning method based on an ERL algorithm. The method learns the local planning strategy in a hybrid dynamic environment to improve the stability and performance of multi-agent training. Qiu et al. [39] combined a traditional optimization algorithm and RL methods to produce a new multi-agent collision-avoidance framework where the agent learns whether to adopt a navigation strategy or autonomous action to avoid obstacles through a deep neural network with intensive learning at each step. In airport management or warehouse automation, a problem occurs where an agent is assigned a new goal immediately after reaching the previous

goal. To solve this problem, Sartoretti et al. [40] proposed a distributed RL framework by introducing RL and imitation learning. In this algorithm, the agent learns a completely decentralized strategy and plans the path online and in real time in observable environments.

The observation path planning of an MOP is a sequential decision optimization problem with multiple constraints, involving the sampling of environmental data in selected marine areas based on the marine environment's numerical-prediction field and the assimilation of sampling data to verify the sampling effect. In this process, it is necessary to model the path-planning process of ocean MOPs and to clarify data acquisition and result evaluation methods. In the present study, an RL algorithm is used to solve the problem of adaptive observation path planning with MOPs. RL is a type of sequential decision-making algorithm that learns through interaction with the environment. Compared with traditional optimization algorithms, RL has the characteristics of intelligence and autonomy. In theory, RL can serialize the path point selection of an MOP, although it has the disadvantage of low sampling efficiency. Evolutionary neural networks (ENN) use an evolutionary algorithm to update the parameters of a neural network [41,42]. Combined with RL, this type of network can further improve the ability of the RL algorithm to interact with the environment, improving sampling efficiency and enabling its application to the adaptive observation path planning of MOPs [43,44]. Notably, in recent years, much effort has been devoted to improving prediction accuracy by combining data assimilation and machine learning techniques [45], and good performance has been achieved [46–48]. However, this is not the focus of this paper and will not be expounded upon here.

To summarize, RL is currently used for the path planning of carriers where the focus is mainly on navigation problems, i.e., the avoidance of obstacles, finding the shortest path, and minimizing energy consumption. There is no way to plan an observation path based on the marine environment field, and although RL path planning has developed rapidly, problems remain, such as effective reward function design and exploration utilization balance. The current development trend is to integrate RL with other methods to improve the efficiency of algorithm exploration and accelerate convergence.

The traditional RL algorithm has the disadvantages of low sampling efficiency and poor robustness. Taking advantage of the adaptability of the ENN to dynamic environments, RL and ENN can be combined for adaptive path planning of MOPs based on the ERL algorithm. The coupling of data assimilation methods based on EAKF aids the assimilation of adaptive sampling and results in verifying the effect of a mobile observation path that is planned on the basis of the ERL algorithm and the analysis and prediction ability of the coupled environment numerical prediction system.

To address the shortcomings of low sampling efficiency and poor robustness in the traditional RL algorithm, this study applied the adaptability of an ENN to dynamic environments, combining it with RL for the adaptive observation path planning of MOPs through coupling data assimilation methods using EAKF to assimilate adaptive sampling results. The project included the following aspects.

1. The RL algorithm was applied to sample the path planning of MOPs in a 3D dynamic ocean environment, and research on coupled modeling of ocean observations incorporating a priori environmental information was carried out. We took advantage of the fact that RL algorithms enable direct interaction between MOPs and the marine environment to solve the problems that arise when using heuristic algorithms, such as the difficulty of modeling the tight coupling between the environmental information and the observation process, the difficulty of solving the optimal observation paths, and the low efficiency of observation.
2. The ERL algorithm was introduced to overcome the low sampling efficiency and robustness of traditional RL in MOP path planning, and an adaptive path planning method for MOPs based on the ERL algorithm was designed. We conducted an analysis of the fusion of the evolutionary algorithm and two strategic gradient-reinforcement learning algorithms (DDPG and Soft Actor-Critic with maximum entropy (SAC)), termed EDDPG and ENSAC. Path-planning simulation results based

on the ENSAC, EDDPG, SAC, and DDPG algorithms in a 3D environment field were compared, and the advantages of the ERL algorithm were highlighted.

3. By conducting data assimilation experiments with the sampling results of the four algorithms, we verified that the ENSAC and EDDPG are more effective than the SAC and DDPG, and in particular, the ENSAC is able to effectively improve the observational efficiency and analytical forecasting capability of marine environmental elements.
4. Moreover, single-platform, dual-platform and five-platform observation experiments were conducted, and the results were assimilated. The assimilation results show that an increase in the number of platforms improves the accuracy of numerical predictions of the marine environment, but the effect diminishes with more than two platforms.

The remainder of this paper is organized as follows. Section 2 describes the adaptive sampling mathematical model of the MOP and the RL method. Section 3 introduces the adaptive sampling method of MOPs based on ERL, including the simulation environment model, action space, state space, return function design, and agent model. Section 4 describes the simulation experiment and result analysis. Finally, the conclusions and suggestions for future work are provided in Section 5.

2. Methods

2.1. Adaptive Model of the 3D Ocean MOP

Self-adaptive mathematical modeling of the observation path of an MOP refers to the process of self-adaptive observation in a specific observation area. An ocean MOP network comprises one or more MOPs. The MOP used here can perform observation tasks alone or as part of a homogeneous or heterogeneous marine environment mobile observation network. Areas with high temperature gradients in the background field affect the analysis and prediction ability of the coupled environmental numerical prediction system, and the target system here is the temperature gradient in the regional ocean.

To improve the analysis and prediction ability of the coupled environmental numerical prediction system, the observation path of the marine MOP must be optimized according to field information. Here, the area 124° E–129° E and 16° N–21° N was selected as the marine observation area. In regional marine environmental observation tasks, path planning for MOPs is carried out on a large scale, so that the MOP can be considered an observation point for the whole observation sea area [49].

Figure 1 provides a schematic diagram of adaptive sampling of MOPs. The path of the ith MOP from $S_i(x_i, y_i, z_i)$ to $S_{i+1}(x_{i+1}, y_{i+1}, z_{i+1})$ is expressed as follows:

$$\begin{cases} x_{i+1} = x_i + v_{i+1} * t * \cos\theta_{i+1} \\ y_{i+1} = y_i + v_{i+1} * t * \sin\theta_{i+1} \\ z_{i+1} = z_i + v_{i+1} * \cos\phi_{i+1} \\ v_{i+1} = v_i * \kappa \end{cases} \tag{1}$$

where θ_{i+1} is the heading angle of the MOP in the $X-Y$ plane at path point $(i+1)$, ϕ_{i+1} is the heading angle of the MOP in the $X-Y$ plane at path point $i+1$, v_{i+1} is the speed at path point $(i+1)$, κ is the variation coefficient of velocity (= 1 here), and t is the time step.

The objective function of the adaptive sampling path of the MOP can thus be modeled as follows:

$$\max f = f(T) = \sum_{i=1}^{N} \Delta T_i$$
$$\text{s.t.} \begin{cases} d = d(t_i) \\ v_l \le v_i \le v_u \\ \theta_l \le \theta_i \le \theta_u \\ \varphi_l \le \varphi_i \le \varphi_u \end{cases} \tag{2}$$

where d is the endurance constraint of the MOP; v_l, θ_l, and φ_l are the lower limits of the speed and heading angle of the MOP; and v_u, θ_u, and φ_u are the upper limits.

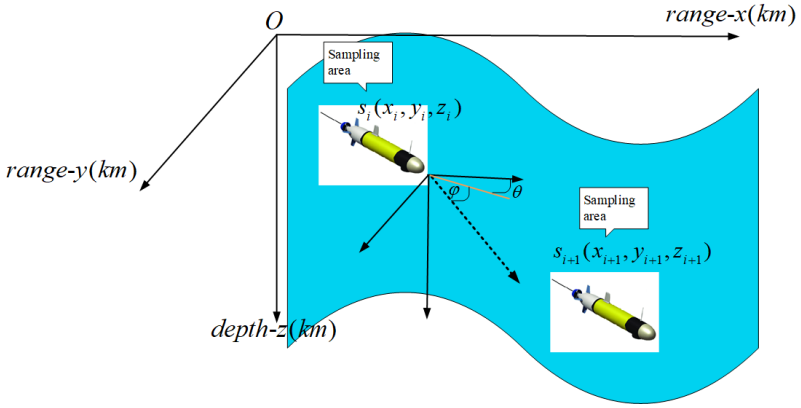


Figure 1. Schematic diagram of adaptive sampling by the observation platform.

The DRL algorithm was applied to observation path planning for a 3D MOP in a continuous-state space and used to simulate MOP performance in complex observation tasks. To solve the observation task, the observation problem was modeled as a Markov decision process [50]. An MDP is generally defined by $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{P}$ is the state transition probability matrix, $\mathcal{R}$ is the reward function, γ is the discount factor, and $\mathcal{A}$ represents a finite set of actions. The next state of the agent, s_{t+1}, is related only to the current state, s_t. The process of moving from one state to another is termed a policy π. The policy function is defined as $\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$. In the training process, the goal of the agent is to maximize the cumulative $\mathcal{R}$.

Value functions and the Bellman equation are important formulae in solving Markov decision processes. The state value function represents the expected return obtained by following policy π from state s, defined as follows:

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] \tag{3}$$

The action value function represents the expected return obtained by the agent performing action a on the current state s when following π, defined as follows:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] \tag{4}$$

The relationship between the state and action value functions is expressed as:

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) q_\pi(s, a) \tag{5}$$

The Bellman expectation equations of the state and action functions are as follows:

$$q_\pi(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s') \tag{6}$$

$$q_\pi(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a' \mid s') q_\pi(s', a') \tag{7}$$

In this observation task, the observation state of the marine MOP is modeled as a multi-stage Markov decision-making process. At each stage, the deep learning dataset includes marine environment prediction data from the coupled environmental numerical analysis and prediction system, allowing environment analysis and prediction for the next 5 days, the data of which can be used as background field prior information for the observation path planning of the MOP. The time step of the agent is 6 h, i.e., the MOP observes the marine environment every 6 h. The agent obtains information for the environmental field to

be tested by interacting with the environmental background field to obtain the return value of the agent. After obtaining the environmental field information, the agent records the current state information and performs actions (according to the forward angle and speed of the agent), observing the next point to be measured. At the end of each observation, the background field is updated according to prediction information. The agent repeats this process as an observation round over 20 sampling points (i.e., 20 observation rounds) to complete the observation strategy. RL is intended to yield an optimal observation strategy such that the MOP carries out observation sampling with continuous interaction with the dynamic background field to accumulate the optimal long-term return.

2.2. Deep Reinforcement Learning

RL is a type of machine learning [51] that integrates relevant knowledge in statistics, control, psychology, and computers. Due to the benefits of RL in sequential decision making, it is widely used in many fields of application. RL is similar to human learning and is based on interactions and trial and error between the agent and environment to achieve autonomous learning. In interactions with the environment, the agent acts first, and the environment updates the next state and gives reward feedback. The agent trains the strategy according to the reward to obtain more rewards and selects actions according to the strategy, entering a cycle. In each cycle, the agent can learn strategies from the results of the interactions with the environment to maximize rewards and stores the strategies in a certain representation form such that RL can realize autonomous learning. The general framework of RL is shown in Figure 2.

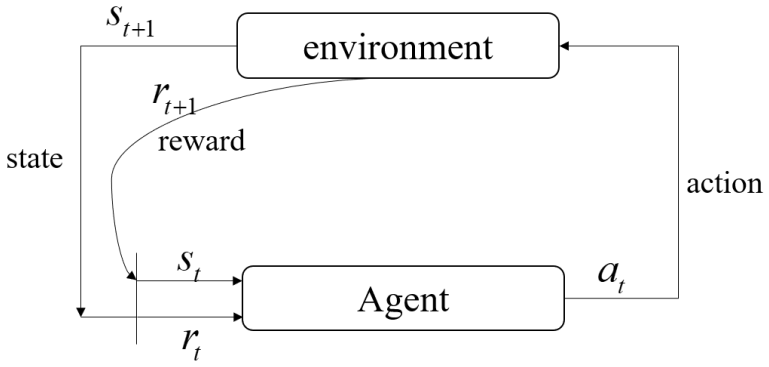


Figure 2. General framework of reinforcement learning.

Mapping from state to action is strategy $\pi(a \mid s)$. The state of the agent at time t is s_{t+1}, and action a_t's execution is selected in the action set of this state using the strategy function $\pi(a_t \mid s_t)$. The environment gives the reward r_t, and the state changes to s_{t+1}.

In the actual environment, the states and actions of practical problems are complex and cannot be solved by traditional RL. To overcome this problem, the deep learning representation ability must be fully exploited so the agent can perceive the complex environmental state and establish more complex action strategies.

DRL combines the respective advantages of deep learning and RL to establish an artificial intelligence system [17,52] . It uses the powerful data expression ability of deep neural networks in RL. For example, the value function can be approximated by a neural network to realize end-to-end optimization learning.

At present, most DRL models use both strategy and value networks in approximating strategy and value functions. The actor-critic algorithm combines strategy and value networks. Silver [53] extended the concept of a policy gradient to a deterministic policy gradient and proposed a deterministic policy gradient algorithm to reduce data variance and to improve the convergence of the algorithm. Based on the deterministic gradient function, the DDPG algorithm was proposed. This algorithm can be applied to both continuous and action spaces. Although it is an off-line strategy to improve sampling efficiency, its training is unstable with poor convergence, so it is difficult to adapt to

different complex environments [54] . Haarnoja et al. [55] proposed a more stable off-line SAC algorithm, which is a type of maximum-entropy RL. The algorithm uses an actor to represent the strategy function, overcoming the difficulty of solving the strategy function in continuous space, and is a very efficient model-free RL algorithm, with a framework as shown in Figure 3.

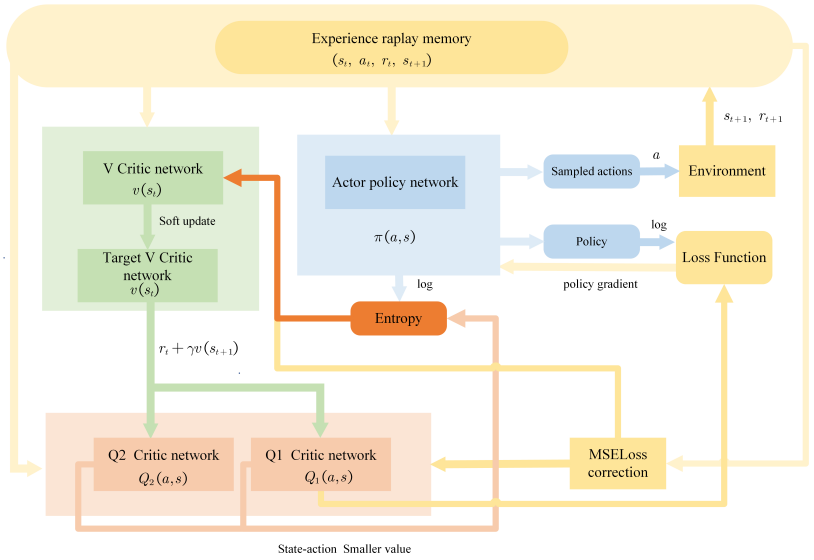


Figure 3. SAC algorithm framework.

This study focused on the adaptive sampling of MOPs, which is a continuous-action task. The DDPG and SAC algorithms suitable for continuous-action space tasks were selected to solve the adaptive sampling problem for MOPs. Here the focus is mainly on improving the SAC algorithm, with the DDPG algorithm being used for comparison.

Entropy is a measure of the degree of randomness of a random variable. In RL, entropy is used to represent the randomness of strategy π in state s. Maximum-entropy RL involves maximizing the cumulative reward when the strategy is random, and is defined as follows:

$$\pi^* = \arg\max_{\pi} \sum_{t} \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi}\left[\gamma^t (r(\mathbf{s}_t,\mathbf{a}_t) + \alpha\mathcal{H}(\pi(\cdot \mid \mathbf{s}_t)))\right] \tag{8}$$

where $\mathcal{H}(\pi(\cdot \mid \mathbf{s}_t))$ is the entropy function and α is the regularization coefficient, which controls the importance of the entropy. The larger the α, the more exploratory it is, which is conducive to subsequent strategy learning, reducing the possibility of poor local optimization.

The soft Bellman equation is as follows:

$$Q_{\text{soft}}(\mathbf{s}_t,\mathbf{a}_t) = r_{(\mathbf{s}_t,\mathbf{a}_t} + \gamma\mathbb{E}_{\mathrm{S}_{t+1}\sim p_{\mathrm{s}}}[V^*_{\text{soft}}(\mathbf{s}_{t+1})] \tag{9}$$

where the state value function is:

$$V^*_{\text{soft}}(\mathbf{s}_t) = \mathbb{E}_{\mathrm{a}_t\sim\pi}[Q^*_{\text{soft}}(\mathbf{s}_t,\mathbf{a}_t) - \alpha\log\pi^*_{\text{MaxEnt}}(\mathbf{a}_t \mid \mathbf{s}_t)] \tag{10}$$

In the SAC algorithm, there are two action value functions, Q. When using a Q network, a network with a small Q value is selected to alleviate the problem of the overestimation of Q. The loss function of action value function Q is as follows:

$$L_Q(\phi) = \mathbb{E}\left[\left(Q(S_t,A_t) - r(S_t,A_t) - \gamma\mathbb{E}_{S_{t+1}}\left[V_\phi(S_{t+1})\right]\right)^2\right] \tag{11}$$

Because an SAC algorithm is an off-line strategy algorithm, R is the data collected previously by the strategy. The loss function of strategy π is obtained from KL divergence, as follows:

$$L_\pi(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\pi_\theta}\left[\alpha \log \pi_\theta(a \mid s) - Q_\phi(s,a)\right]\right] \tag{12}$$

For a continuous-motion space environment, the re-parameterization technique is used to ensure the sampling process of a Gaussian distribution is derivable, and the loss function is modified as follows:

$$L_\pi(\theta) = \mathbb{E}_{s\sim\mathcal{D},\epsilon\sim\mathcal{N}}\left[\alpha \log \pi_\theta(f_\theta(s,\epsilon) \mid s) - Q_\phi(s, f_\theta(s,\epsilon))\right] \tag{13}$$

The SAC algorithm also provides automatic adjustment of the regularization parameter, α, as follows:

$$L(\alpha) = \mathbb{E}_{a\sim\pi_\theta}[-\alpha \log \pi_\theta(a \mid s) - \alpha\kappa] \tag{14}$$

where κ is a super parameter that can be understood as the target entropy. This updating method is termed the automatic entropy adjustment and employs the dual form of the original strategy optimization problem under the constraint that the average entropy of each step is at least κ.

2.3. Evolutionary Reinforcement Learning

The RL algorithm has achieved good results in path planning, but many problems remain, such as low exploration efficiency and convergence being greatly affected by super parameters. The high sampling efficiency and robustness of the evolutionary algorithm enable improvement of exploration efficiency by considering the agent as a population individual and performing crossover and mutation operations on it, and the strategy algorithm based on gradient update and the heuristic neural evolutionary algorithm improve the robustness of the RL algorithm.

The evolutionary algorithm is a feature-search algorithm whose abilities include the generation of new solutions, solution change, and solution selection. With many candidate solutions, these operations result in new solutions and the retaining of better solutions according to probability, which is indicated by the fitness of the solution; the higher the fitness, the better the solution quality and the greater the probability of being selected. The quality of the retained solution increases with multiple iterations. An ENN was used in this study. In the algorithm, the neural network is combined with the evolutionary algorithm, and the deep neural network is regarded as the individual in the evolutionary population. Crossover and mutation operations between individuals in the population are then the crossover and mutation operations of the neural network. Crossover between neural networks refers to the exchange of weights at the same neuron between different individuals, while genetic variation is the random disturbance of the weight parameters of the neural network. A schematic diagram of the ENN is shown in Figure 4. The main concepts of the ERL algorithm are as follows. (1) The use of multiple strategy algorithms allows us to form an algorithm set (here, each strategy algorithm is termed a 'learner'). The advantage of using multiple learners is that each learner explores the whole action space, improving exploration efficiency and the vision of the underlying Markov decision-making process. In addition, the use of multiple learners together greatly reduces the destabilizing effects of differences in super parameters or random seeds relative to single-learner exploration, thereby improving algorithm robustness. (2) A strategy-generator actor is used in the strategy algorithm as an individual to form a population with an evolutionary function. The reward obtained by each actor in the environment is evaluated as its fitness, data generated in the evaluation process are added to the memory pool, and actors of the population are cross-mutated according to the principle of the ENN to retain actors with high fitness. (3) Learner and population actors interact regularly, with the former considered part of the population and participating in the cross-mutation process of the population. The evolutionary strategy gradient algorithm is formed by integrating the exploration advantages of the evolutionary algorithm and RL learning ability.

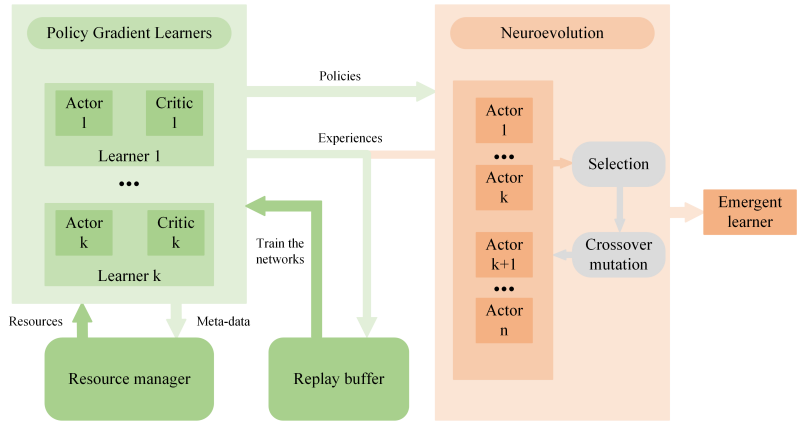


Figure 4. Schematic diagram of evolutionary neural network.

3. ERL Adaptive Sampling Path Planning for 3D Ocean MOPs

ERL was applied to adaptive sampling path planning for a 3D MOP, verifying that adaptive sampling results improve the prediction accuracy of regionally coupled environmental data assimilation and numerical prediction systems. The overall process is shown in Figure 5.

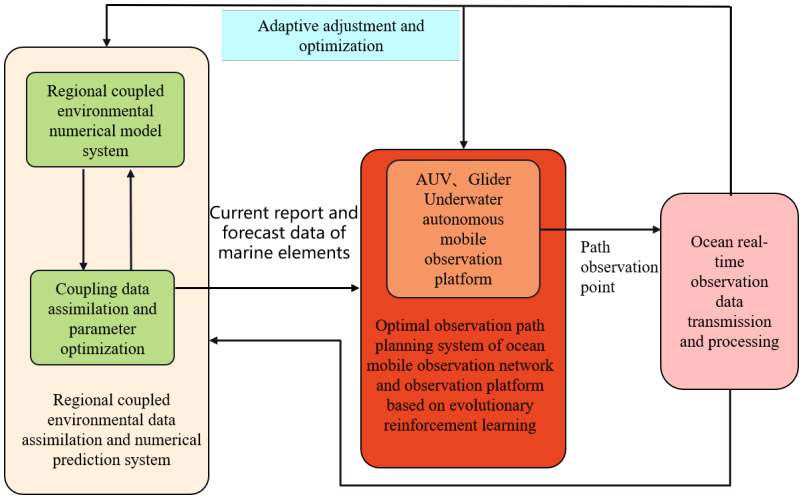


Figure 5. Adaptive sampling process of a mobile observation platform.

3.1. Acquisition of Marine Environmental Background Fields and Verification of Observations

The marine environment background field is an a priori environment for the adaptive observation path planning of the MOP. MOP adaptive observation path planning must first output the marine environment prediction information of the area to be measured through the coupled numerical prediction system as prior information pertaining to the background field. Then, marine environment sampling data are obtained along the planned observation path, and assimilation estimation and analysis are undertaken through the coupled data assimilation method based on a set of adjusted Kalman filters to update the marine environment analysis and prediction information. To update the observation path of the MOP, this process is looped to realize the adaptive planning and adjustment of the observation path of the MOP network.

Based on the ocean numerical model [52] used here, the Princeton Ocean Model(POM) involves a completely physical process and is a general operational physical oceanography model. This model simulates atmospheric forcing and the characteristics of sea surface temperature and currents of complex oceans. The POM was therefore selected as the background model in this study.

The coupled environment numerical analysis and prediction system based on the POM was first selected to obtain the environment analysis and prediction information of the sea area to be monitored over the following 5 days. This information is used as the background field of prior information for the observation path planning of the MOP. The energy conservation characteristics of the POM mean it has advantages in the simulation and evolution of atmospheric, ocean, and land temperature data. The horizontal resolution of the study area was improved to 1/6° by applying four-layer nesting technology, and the vertical profile was divided into 15 layers. The area 124°–129° E/16°–21° N was selected as the ocean observation area. The temperature gradient was taken as the observation target for the design of the adaptive observation scheme of the regional ocean MOP. In addition, with the POM adjusted by nesting technology, an adjusted Kalman filter data assimilation method was used for the assimilation of observation data.

3.2. Simulation Environment Modeling

In our MOP adaptive observation path planning process in a dynamic environment field, initial marine environment background data were first obtained, and initial sampling points were selected according to constraints. In the selection of subsequent sampling points, the points must be selected according to the obtained background data (Figure 5).

The data used in the dynamic environment field included 20 groups of marine environment time-gradient field data, including the Rankgauss gradient field. Five sets of data are shown in Figure 6 out of the 20 sets of data. Environment construction in the adaptive observation path planning of the MOP based on RL must include the reward function and the state and action spaces in the dynamic environment field. The reward function is the weighted sum of the spatial gradient of the environmental field to be measured, including the time gradient of the environmental field, measurement constraints of the MOP, and obstacle avoidance constraints. The state of the environment includes global and local, time-gradient, spatial-gradient, obstacle, and current location field information. The agent of the dynamic environment field provides speed and heading.

Compared with the traditional RL algorithm, all actors in the ERL algorithm interact simultaneously with the population to obtain empirical data, so it is necessary to build a parallel running environment. Here, multi-process technology was used to explore the actors of the agent and evolutionary population in parallel, making full use of hardware resources to increase the speed of evaluating the individuals of the evolutionary population and obtaining actor exploration data. The use of a multi-process resource manager to synchronize exploration data from multiple processes avoids conflict caused by the reading and writing of data. Communication methods used in Python multi-process programming generally include pipes, signals, message queues, semaphores, and shared memory, and the manager module of multi-process library multiprocessing is advanced and encapsulated. With multi-process programming, the above basic communication methods can be ignored, and direct use can be made of the data type list and Dict Namespace with secure data communication between multiple processes.

3.2.1. Data Collection and Processing

RL is an artificial intelligence algorithm for sequential decision optimization in which agents and the environment learn interactively. The optimized extraction of features from original data can speed up the convergence of the algorithm and model. A reasonable data preprocessing method is conducive to planning the optimal adaptive observation path of the MOP. Original data cannot reflect the relative positional relationship between points, which is meaningless for the task of path planning. Therefore, the original data were reorganized into evenly distributed marine environment field data according to the coordinated data information. Figure 7 shows a data schematic diagram of the original background field of the 3D marine environment to be measured in the study area (124°–129° E/16°–21° N).

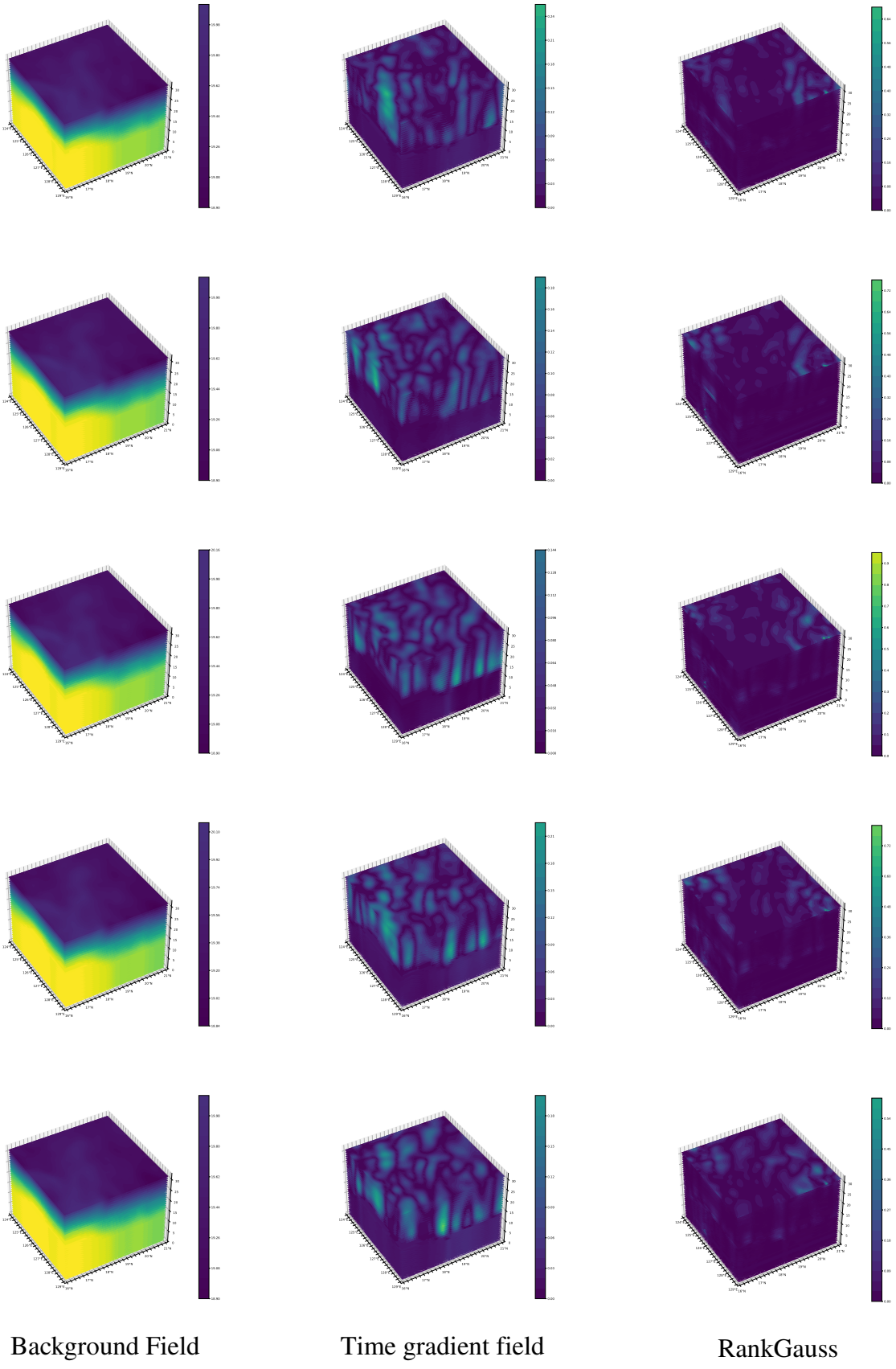


Figure 6. The processing of 3D marine environmental data fields.

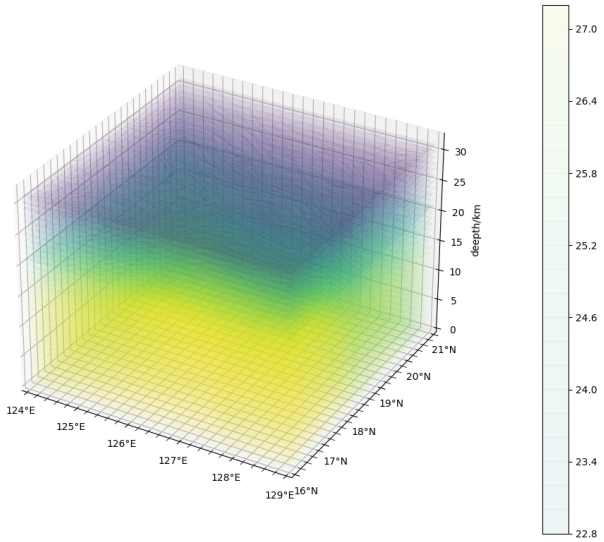


Figure 7. The data schematic diagram of the original background field of the 3D marine environment.

When using DRL to plan the observation path of the MOP, it was first necessary to obtain in advance a set of marine environment numerical prediction field data at a fixed time. Training could then include a round of decision-making processes by the agent based on this dataset. The preprocessing of marine environment prediction field data affects the path-planning results of MOPs. To improve the generalization of different field data by agents, it is necessary to eliminate differences between field datasets as much as possible. In this study, the RankGauss data preprocessing method [56] was used to process marine field data.

In the RankGauss data standardization method, data are not Gaussian, and their sorting information is retained. The overall process in RankGauss normalization is as follows. First, all data are sorted into groups over a range including all field data. To restore the final data to grid data, the position of each data point must be recorded while sorting. Second, the sorted data scale is converted to a $[-1, 1]$ format. Third, data from the original relative size relationship are converted to the size relationship determined by the location. Fourth, the original data distribution is converted to a normal distribution, with only the relative data size relationship retained. The environment after processing is shown in Figure 6 (third column).

3.2.2. State Space

The main task of the MOP is to observe areas with steep temporal and spatial gradients in seawater temperature, taking note of measurement attribute and obstacle avoidance constraints. Marine environment background field data are a type of grid data, and positions in the grid represent the positional relationship with the actual environment. The spatial gradient of seawater temperature represents the difference between data at a certain point and the surrounding environment. Therefore, it was assumed that the neural network of the agent can treat the background field as the basis for decision making with spatial gradient information. The time gradient indicates the change in temperature at a certain point with time. It was assumed that the time gradient field of the marine environment can be used as the basis for time-gradient decision making. In addition, considering the balance between the rewards of the overall and step tasks, the local environmental field in a certain area around the current observation platform should also be considered part of the state. Finally, to deal with obstacle avoidance, obstacle information in the observation area is also considered part of the state. The state of the environment includes mainly global

and local marine environment, time-gradient and obstacle field information, and position information regarding the MOP (Figure 8).

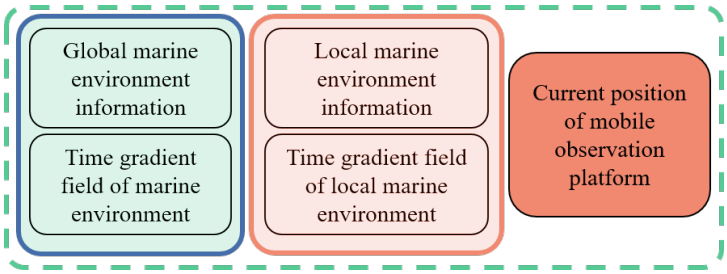


Figure 8. State space design.

3.2.3. Action Space

All actions performed by the agent constitute the environmental action space. A reasonable design of this space aids the agent in taking action and accelerating convergence. The exploration scope of the agent affects whether the algorithm can achieve global optimization, so it cannot make the action space large and comprehensive or make it too simple. Here, when the MOP plans the path for the sampling area, it must consider its speed and heading. Therefore, the design of the action space should include reasonable speed and heading ranges.

Specific values for heading and speed were continuously output. The SAC, PG, and DDC decision-making algorithms were applied to the network, and actions taken by the MOP agent included heading and speed.

The design concept for the action space is shown in Figure 9. The decision-making area of the agent was formed by limiting the ranges of heading and speed.

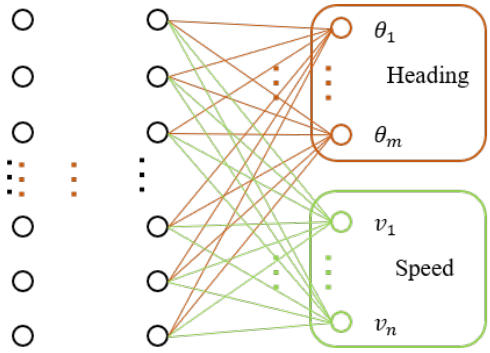


Figure 9. Agent action design.

3.2.4. Reward Function

The goal of RL is to maximize the cumulative expected reward. The reward function affects the training speed and direction of agents. Multiple factors must be considered when sampling with the MOP, such as the ocean temperature gradient of the area to be studied, measurement attributes of the MOP, and obstacle information. Therefore, when designing the reward function, it is necessary to determine and integrate the weight of each factor so that the agent model converges more rapidly. In this study, the reward function was set as the weighted sum of the space–time gradient of the environmental field to be measured to enable the agent to accurately collect target samples while considering the measurement constraints of the MOP and obstacle avoidance constraints.

The main task of the ocean observation network is to collect sensitive information about the regional ocean. Here, we consider mainly locations with large temperature changes. With limited observation resources, the design of optimal acquisition paths minimizes the waste of resources and improves sampling efficiency. In designing the reward function, we considered changes in the time and space gradients of the environmental

field at the sampling point of the MOP. The measurement attributes of the MOP itself affect its observation range, and here, we focus mainly on the measurement range, time step, and endurance mileage of MOPs. The MOPs must not collide when sampling, so the location of the MOP and the locations of unknown obstacles must be considered when designing the reward function. Regarding the collision avoidance constraint, when the distance between the MOP and an obstacle or another MOP is twice the step size, punishment is incurred. The reward function considered here is shown in Equation (15), where the first term is the sum of the time and space gradients and the second term is the sum of the measurement range, endurance mileage, and time step.

$$R = \sum_{\text{grad}} \text{grad}(R_1) + \sum_{\text{meas}} \text{plat}(R_2) \tag{15}$$

3.3. Regional Marine Environment Mobile Observation Network Model

The ERL model algorithm is quite different from that of general RL as follows. (1) The RL algorithm generally contains only one agent for learning the optimal strategy by interaction with the environment. ERL includes multiple agents learning at the same time. By evaluating the learning effect, each agent is allocated computing resources. Finally, the agent with the best learning effect is selected to make up for the poor robustness of the RL algorithm caused by the problem of parameter initialization. (2) In addition to intelligence, the ERL algorithm includes an ENN, and an evolutionary algorithm is used to update the population in exploring the environment to obtain diversified data and enhance the exploration efficiency of agents.

The adaptive sampling process of the MOP based on the ENSAC algorithm is shown in Figure 10.

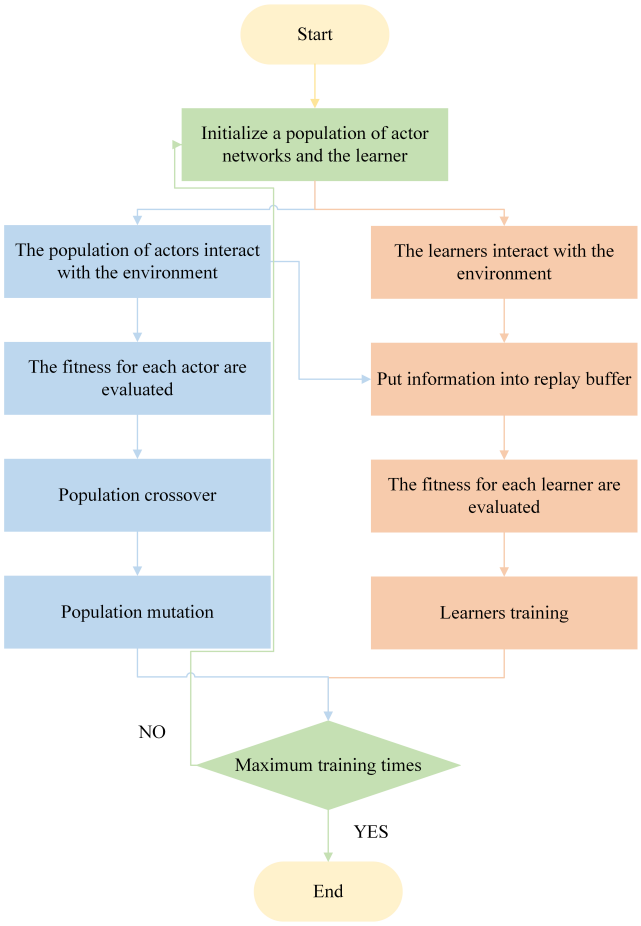


Figure 10. Schematic diagram of evolutionary reinforcement learning algorithm.

4. Experiment and Analysis

The experiment was run under the Microsoft Windows 10 system of the Anaconda configuration module, which used an Intel Core i9 10900K Ten-Cores 3.7 GHz processor and an NVIDIA GeForce RTX3080 graphics card. The software environment used to develop the algorithmic models was Python 3.8 combined with PyTorch.

4.1. Parameter Setting

Four algorithms based on DDPG, EDDPG, SAC, and ENSAC were applied to verify the effectiveness of MOP observation path planning. The hyperparameters for algorithmic models were initially based on references, i.e., based on the hyperparameters of the SAC algorithm primarily referenced [57–59], while the hyperparameters of the ENN network mainly referenced [43,60]. Moreover, the learning rate controls the size of each parameter update. Too high a learning rate can cause problems, such as too many updates and unstable convergence; conversely, it can make convergence slow and lead to lower learning performance. The Adam optimizer with gradient clipping at 10 and a learning rate of $1e^{-3}$ was used for the actor learning rate and critic learning rate. The discount factor determines how much the agent focus on future rewards and was set to 0.9. The crossover probability mainly affects the search performance and was set to 0.01 after many experiments and adjustments. The mutation probability mainly prevents the model from falling into the local optimal solution while maintaining the diversity of the population and was usually set to 0.2. The proportion of elites in the population was set to 0.2. The batch size is the number of samples randomly selected from the experience playback buffer each time. Larger values indicate that more samples are used in training, and the model will be updated more consistently, but too large a value will cause a computational burden and a long training time. In this experiment, the batch size was set to 64 based on the memory resources of the computer used. After the experiments, results tended to be stable after 8000 sessions of agent training, so the total number of training rounds was set to 10,000 to verify the performance of different algorithms in path planning. After repeated experiments and historical experience, the main parameter settings of the algorithm are shown in Table 1.

Sea surface temperature field data over the next 5 days were output as initial data of the experiment through the regional coupled environmental numerical prediction system (Section 2). There were 100 experimental groups for the observation area of 124°–129° E/16°–21° N, and the observation time interval of the MOP was 6 h.

Table 1. Algorithm's main hyperparameter settings.

Parameter	Value	Description
minibatch size	64	Make the gradient descent direction more accurate
buffer size	10,000	Buffer capacity
discount factor	0.9	Attenuation coefficient
rollout_size	3	Size of learner rollouts
init_w	1	Whether the neural network parameters are initialized
actor_lr	$1e^{-3}$	Actor learning rate
critic_lr	$1e^{-3}$	Critic learning rate
noise_std	0.1	Gaussian noise exploration std
ucb_coefficient	0.9	Exploration coefficient in UCB
pop_size	7	Population size
elite_fraction	0.2	Proportion of elites in the population
crossover_prob	0.01	Probability of crossing
mutation_prob	0.2	Probability of variation

4.2. Adaptive Sampling Results, Data Assimilation Results, and Analysis of Observation Platform

The DDPG, EDDPG, SAC, and ENSAC algorithms were used to observe path coordinate points in the static environment field with the highest reward function for data assimilation. The specific steps involved the input of path coordinate points into the data

assimilation system and the calculation of the root-mean-square error (RMSE) between the true and predicted values. The results are shown in Table 2, where bold data indicate the algorithm with better data assimilation results in a given simulation environment.

Table 2. RMSE comparison of sampling and assimilation results of observation path of platform, where $RMSE_{ramdom} = 0.18201$.

Platform	*ENSAC*	*EDDPG*	*DDPG*	*SAC*
single	**0.16539**	0.17408	0.17938	0.16775
dual	**0.12013**	0.13619	0.13052	0.13284
five	**0.16952**	0.17350	0.17213	0.17151

Reward values for the four adaptive observation paths in the sampling area are shown in Figure 11, where red represents the ENSAC algorithm, purple represents the EDDPG algorithm, green represents the SAC result, and orange represents the DDPG algorithm.

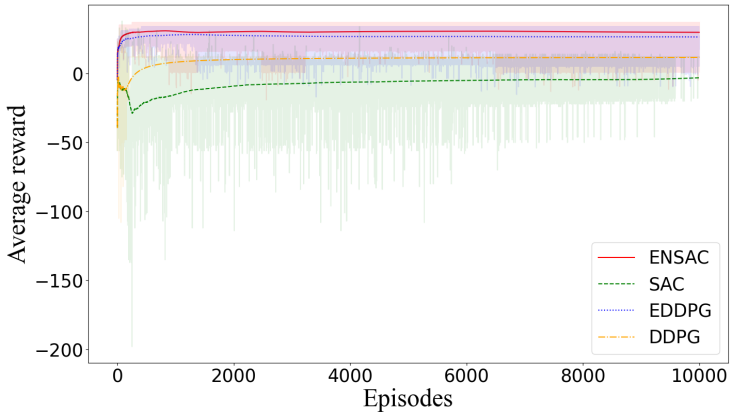
In the single ocean environment observation platform experiment, the reward function curves of the four algorithms indicate that the reward values of the ENSAC and EDDPG algorithms were much higher than those of the SAC algorithm. The observation data of MOPs thus improve the prediction accuracy of ocean data prediction systems after observation path planning. In the adaptive observation path planning of MOPs in dynamic environments, the observation results of the SAC and DDPG algorithms based on the ERL were better than those of the DDPG algorithm based on the SAC algorithm. The results indicate that the strategy gradient algorithm based on the ERL is suitable for the adaptive observation of MOPs in dynamic environments. The adaptive observation result of the strategy gradient algorithm fused with the ERL is generally superior to the assimilation result of the other two algorithms and randomly selected path points, which indicates that the adaptive observation path planning of MOPs based on the strategy gradient algorithm of the fusion algorithm enables more effective observations.

Analysis of the number of platforms indicates that the degree of improvement in the double-platform observation experiment is greater than that in the single-platform and five-platform experiments, which indicates that an increase in the number of platforms improves prediction accuracy, but with more than two platforms, the effect diminishes.

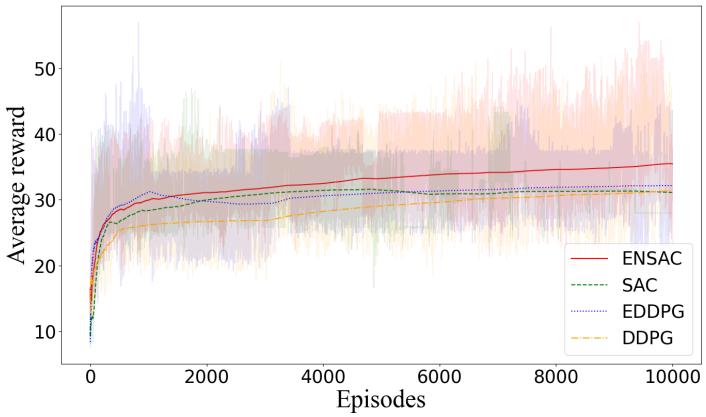
From the above results, we conclude that the strategy gradient algorithm based on ERL is more suitable for the adaptive observation path planning of MOPs in dynamic environment fields than the traditional strategy-gradient algorithm. The evolutionary algorithm improves the sampling efficiency of RL and the exploration efficiency of an intelligent body. The adaptive observation path planning of MOPs based on ERL also improves the analysis and prediction ability of the coupled environmental numerical prediction and DDPG algorithms, enabling more rapid convergence of the reward curves of the ENSAC and EDDPG algorithms. This indicates that integration of the ERL and strategy-gradient algorithms improves the agent's ability to explore the environment and accelerates convergence of the algorithm. Figure 11 also indicates that although the reward curve of SAC is lower than that of DDPG, the reward curve of the ENSAC algorithm is higher than that of the EDDPG algorithm after integration of the ERL algorithm. The ERL algorithm thus has a stronger role in the improved SAC algorithm than the DDPG algorithm.

After the above discussion, it can be concluded that the strategy-gradient algorithm based on an evolutionary algorithm is more suitable for the adaptive observation path planning of mobile observation platform in a dynamic environment field than the traditional strategy-gradient algorithm. The evolutionary algorithm can effectively improve the sampling efficiency of reinforcement learning and the exploration efficiency of the intelligent body. The adaptive observation path planning of mobile observation platform based on evolutionary reinforcement learning can effectively improve the analysis and prediction ability of the coupled environmental numerical prediction system's algorithm and the DDPG algorithm, and the reward curves of the ENSAC algorithm and the EDDPG algorithm can approach convergence speed faster. This shows that the integration of the

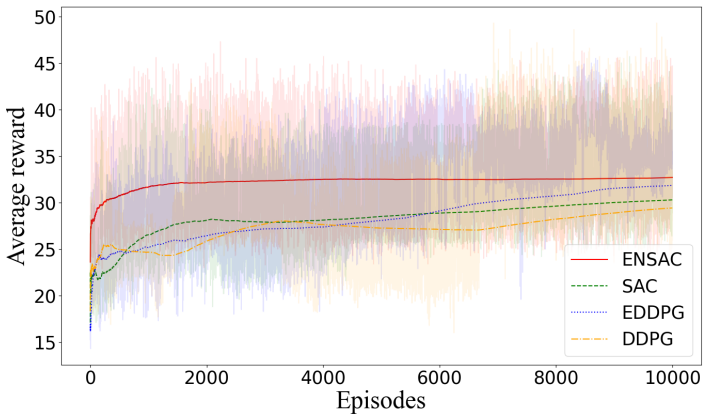
evolutionary algorithm and strategy-gradient algorithm can effectively improve the agent's ability to explore the environment and accelerate the convergence speed of the algorithm. At the same time, it can be seen from Figure 11 that although the reward curve of SAC is lower than DDPG, the ENSAC algorithm after integrating the evolutionary algorithm is higher than the EDDPG algorithm. This shows that the role of the evolutionary algorithm in the improved SAC algorithm is greater than that of the DDPG algorithm.



(a) single platform



(b) dual platform



(c) five platform

Figure 11. Comparison of average rewards of ENSAC, EDDPG, SAC, and DDPG in single-platform, dual-platform and five-platform experiments.

In the double-platform marine environment observation experiment, before 1500 training sessions, the average reward value of the EDDPG algorithm was higher. After 1500 training sessions, the average reward of the ENSAC algorithm exceeded that of the EDDPG algorithm and gradually increased until the end of training. The EDDPG algorithm fluctuated widely during the training process and did not reach its optimum output. Although the DDPG algorithm output increased throughout the training process, its average reward value did not reach the optimal state. The SAC algorithm performed similarly to the EDDPG algorithm. In the adaptive sampling experiment of five MOPs, the ENSAC algorithm reached the optimal state after 1000 training sessions and remained stable until the end of training. However, the EDDPG, DDPG, and SAC algorithms fluctuated, especially in the first 4000 training sessions.

The above experimental analysis indicates that in the four comparative experiments, the hybrid algorithm maintains stable efficiency in the marine environment adaptive observation task. Therefore, the hybrid algorithm combining the ERL and RL algorithms improves the observation efficiency of agents in marine environment observation.

Adaptive sampling paths of the experiments with single, dual, and five MOPs are shown in Figure 12 based on the ENSAC algorithm and showing views from the front, side, and top. It is clear that the MOP conducts adaptive observation in areas of large slope change, and the planned path meets the requirements for sensitive sea areas, i.e., targets with large temperature differences.

The four algorithms (ENSAC, EDDPG, SAC, and DDPG) were used to assimilate sampled data of the observation path coordinate points with the highest reward values. The specific steps were to input the path coordinate points into the assimilation system and to calculate the RMSEs of the true and predicted values. The results are shown in Table 2, with bold data representing the algorithm with better data assimilation results under the given simulation environment.

The average RMSE of results for single, dual, and five platforms increased, respectively, by 9.13%, 33.99%, and 7.55% for the ENSAC algorithm; 4.35%, 25.17%, and 4.88% for the EDDPG algorithm; 1.44%, 28.28%, and 5.50% for the DDPG algorithm; and 7.83%, 27.01%, and 6.25% for the SAC algorithm.

These results indicate that observation data of MOPs improve the accuracy of ocean data prediction systems after observation path planning. With the adaptive observation path planning of MOPs in dynamic environments, the observation results of the SAC and DDPG algorithms based on the ERL algorithm were better than those of the DDPG algorithm based on the SAC algorithm. The strategy-gradient algorithm based on the ERL algorithm is thus suitable for the adaptive observation of MOPs in dynamic environments. In general, the adaptive observation result of the strategy-gradient algorithm fused with the ERL algorithm is superior to the assimilation results of the other two algorithms with randomly selected path points. The adaptive observation path planning of the MOP, based on the strategy-gradient algorithm of the fused algorithm, enables more effective observation. Regarding the number of platforms, the degree of improvement with double-platform observations is greater than that for single- and five-platform observations. An increase in the number of platforms thus improves the accuracy of numerical prediction of the marine environment, but the effect diminishes with more than two platforms.

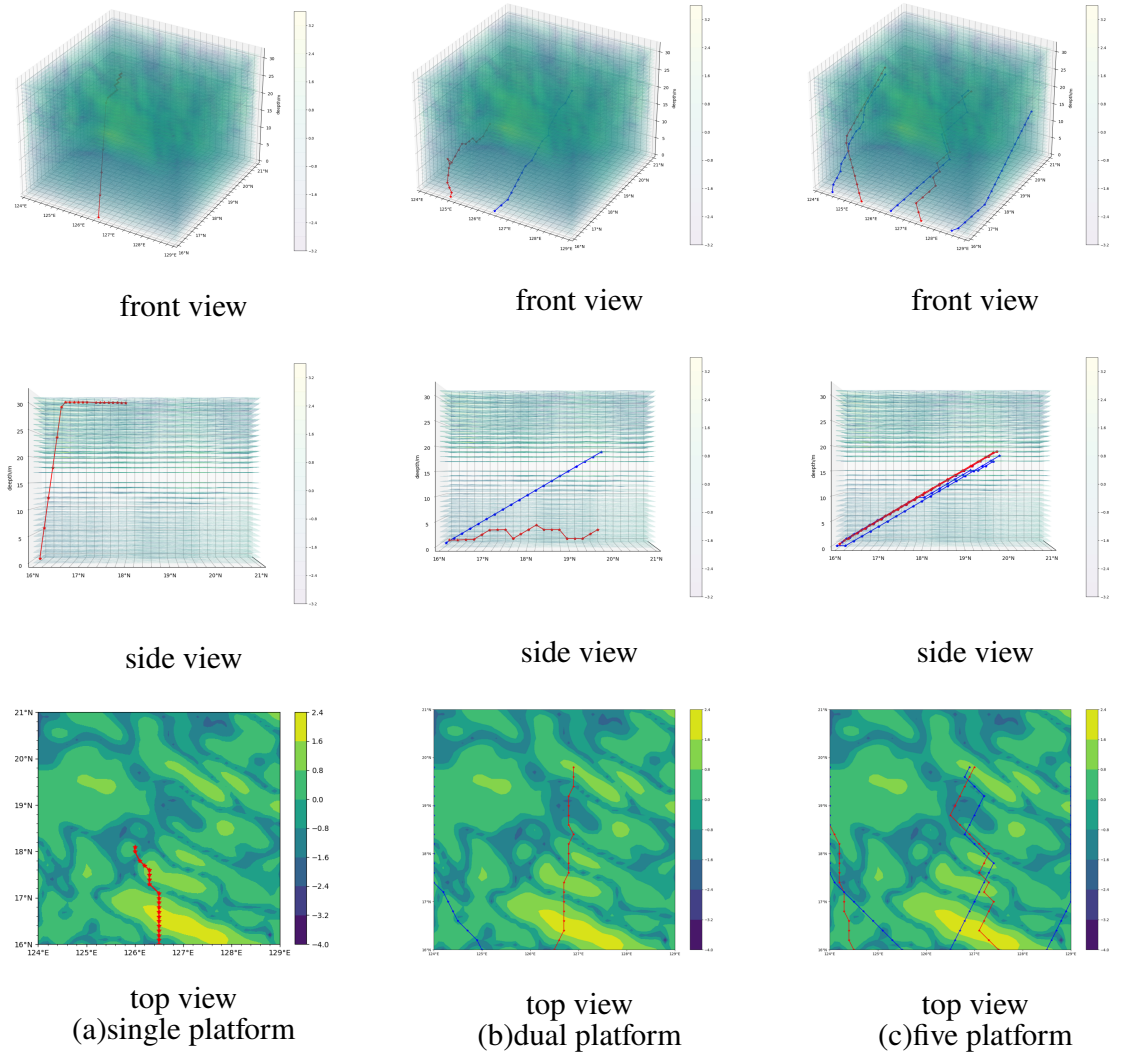


Figure 12. Adaptive observation path map for single platform, dual platforms, and five observation platforms using ENSAC.

5. Conclusions and Future Work

Due to the wide range of 3D dynamic ocean areas, the variety of marine environmental elements over time and space, and the limited observation resources that can be utilized, how to efficiently complete observation path planning is a key issue in ocean observation research. When heuristic algorithms are used for the path planning of MOPs, they face the challenges of difficultly in modeling the tight coupling between environmental information and the observing process and difficulty in solving for the optimal observing paths, which limits the observing efficiency of ocean observing platforms. Aiming at the above problems, this paper applies RL algorithms to the sampling path planning of MOPs in a 3D dynamic ocean environment. Using the advantage that RL algorithms can realize direct interactions between MOPs and the ocean environment for autonomous learning, we conducted research on the coupled modeling of ocean observation incorporating a priori environmental information.

On the other hand, the ERL algorithm was introduced to overcome the low sampling efficiency and robustness of traditional RL in MOP path planning, and an adaptive path planning method for MOPs based on the ERL algorithm was designed. Simulation results for adaptive path planning for MOPs based on two strategy algorithms and evolutionary learning fusion were discussed. Path-planning simulation results based on the ENSAC,

EDDPG, SAC, and DDPG algorithms in a 3D environment field were compared, and advantages of the ERL algorithm were highlighted. The sampling results of the four algorithms were assimilated. Compared with random sampling, the sampling results obtained through MOP path planning improved the accuracy of the data prediction system, verifying that the ENSAC and EDDPG algorithms used here are more effective than the SAC and DDPG algorithms. The results indicated that the ENSAC and EDDPG algorithms improve sampling efficiency and RL convergence. The reward-function and data assimilation results of the ENSAC algorithm were better than those of the EDDPG algorithm, indicating that the fusion of the ERL and RL algorithms based on a maximum-entropy strategy achieves an improvement over the deterministic strategy algorithm. In addition, we conducted observation experiments on a single platform, dual platform, and five platforms and assimilated the experimental results. The assimilation results show that an increase in the number of platforms improves the accuracy of numerical predictions of the marine environment, but the effect diminishes with more than two platforms.

There is still much work to do in the future. Firstly, the proposed method will be used to carry out practical experiments to verify the feasibility of the algorithms in practical applications and to enhance the robustness of the findings. In addition, further validation of the reliability and generalization ability of the model performance is needed. Secondly, whether the adaptive observation of more MOPs would improve the analysis and prediction capability of the coupled-environment numerical prediction system remains to be studied. Meanwhile, the effects of the fusion of the evolutionary algorithm, other strategies, and various RL algorithms also require further study.

Author Contributions: Conceptualization and methodology, J.Z. and Y.L.; writing—original draft preparation, J.Z.; visualization, J.Z.; investigation and data curation, Y.L.; resources, funding acquisition, and software, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Laboratory of Marine Environmental Information Technology (MEIT), the NSFC (Nos. 41676088), and the Fundamental Research Funds for the Central Universities (Nos. 3072022YY0401).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors thank Shuo Yang and Chunwang Yang for their full support of and suggestions for the early version of this manuscript.

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

References

1. Berget, G.E.; Fossum, T.O.; Johansen, T.A.; Eidsvik, J.; Rajan, K. Adaptive sampling of ocean processes using an auv with a gaussian proxy model. *IFAC-PapersOnLine* **2018**, *51*, 238–243. [CrossRef]
2. Stankiewicz, P.; Tan, Y. T.; Kobilarov, M. Adaptive sampling with an autonomous underwater vehicle in static marine environments. *J. Field Robot.* **2021**, *38*, 572–597. [CrossRef]
3. Zhang, B.; Sukhatme, G.S.; Requicha, A.A. Adaptive sampling for marine microorganism monitoring. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 2, pp. 1115–1122.
4. Ezeora, O.S.; Heckenbergerova, J.; Musilek, P. A new adaptive sampling method for energy-efficient measurement of environmental parameters. In Proceedings of the 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016; pp.1–6.
5. Fossum, T.O. Adaptive Sampling for Marine Robotics. Ph.D. Thesis, Institutt for Marin Teknikk, Trondheim, Norway, 2019.
6. Vu, M.T.; Le, T.H.; Thanh, H.L.N.N.; Huynh, T.T.; Van, M.; Hoang, Q.D.; Do, T.D. Robust Position Control of an Over-actuated Underwater Vehicle under Model Uncertainties and Ocean Current Effects Using Dynamic Sliding Mode Surface and Optimal Allocation Control. *Sensors* **2021**, *21*, 747. [CrossRef] [PubMed]
7. Vu, M.T.; Thanh, H.L.N.N.; Huynh, T.T.; Do, Q.T.; Do, T.D.; Hoang, Q.D.; Le, T.H. Station-Keeping Control of a Hovering Over-Actuated Autonomous Underwater Vehicle under Ocean Current Effects and Model Uncertainties in Horizontal Plane. *IEEE Access* **2021**, *9*, 6855–6867. [CrossRef]

8. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D. Optimal path planning of an unmanned surface vehicle in a real-time marine environment using Dijkstra algorithm. In Proceedings of the 12th International Conference on Marine Navigation and Safety of Sea Transportation (TransNav 2017), Gdynia, Poland, 21–23 June 2017; pp. 399–402.
9. Parimala, M.; Broumi, S.; Prakash, K.; Topal, S. Bellman–Ford algorithm for solving shortest path problem of a network under picture fuzzy environment. *Complex Intell. Syst.* **2021**, *7*, 2373–2381. [CrossRef]
10. Solichudin, S.; Triwiyatno, A.; Riyadi, M. A. Conflict-free dynamic route multi-agv using dijkstra Floyd-warshall hybrid algorithm with time windows. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 3596. [CrossRef]
11. Martins, O.O.; Adekunle, A.A.; Olaniyan, O.M.; Bolaji, B.O. An Improved multi-objective a-star algorithm for path planning in a large workspace: Design, Implementation, and Evaluation. *Sci. Afr.* **2022**, *15*, e01068. [CrossRef]
12. Mokrane, A.; Braham, A. C.; Cherki, B. UAV path planning based on dynamic programming algorithm on photogrammetric DEMs. In Proceedings of the 2020 International Conference on Electrical Engineering (ICEE), Istanbul, Turkey, 25–27 September 2020; pp. 1–5.
13. Lin, Z.; Yue, M.; Wu, X.; Tian, H. An improved artificial potential field method for path planning of mobile robot with subgoal adaptive selection. In *Intelligent Robotics and Applications, Proceedings of the 12th International Conference, ICIRA 2019, Shenyang, China, 8–11 August 2019*; Proceedings, Part I 12; Springer: Berlin/Heidelberg, Germany, 2019; pp. 211–220.
14. Putro, I.E.; Duhri, R.A. Longitudinal stability augmentation control for turbojet UAV based on linear quadratic regulator (LQR) approach. In Proceedings of the 7th International Seminar on Aerospace Science and Technology—ISAST 2019, Jakarta, Indonesia, 24–25 September 2019; Volume 2226.
15. Wang, H.; Fu, Z.; Zhou, J.; Fu, M.; Ruan, L. Cooperative collision avoidance for unmanned surface vehicles based on improved genetic algorithm. *Ocean Eng.* **2021**, *221*, 108612. [CrossRef]
16. Han, G.; Zhou, Z.; Zhang, T.; Wang, H.; Liu, L.; Peng, Y.; Guizani, M. Ant-colony-based complete-coverage path-planning algorithm for underwater gliders in ocean areas with thermoclines. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8959–8971. [CrossRef]
17. Hu, L.; Naeem, W.; Rajabally, E.; Watson, G.; Mills, T.; Bhuiyan, Z.; Raeburn, C.; Salter, I.; Pekcan, C. A multiobjective optimization approach for COLREGs-compliant path planning of autonomous surface vehicles verified on networked bridge simulators. *IEEE Trans. Veh. Technol.* **2019**, *21*, 1167–1179. [CrossRef]
18. de Castro, G.G.R.; Pinto, M.F.; Biundini, I.Z.; Melo, A.G.; Marcato, A.L.M.; Haddad, D.B. Dynamic Path Planning Based on Neural Networks for Aerial Inspection. *J. Control Autom. Electr. Syst.* **2023**, *34*, 85–105. [CrossRef]
19. Kim, W.S.; Lee, D.H.; Kim, Y.J.; Kim, T.; Lee, H.J. Path detection for autonomous traveling in orchards using patch-based cnn. *Comput. Electron. Agric.* **2020**, *175*, 105620. [CrossRef]
20. Terasawa, R.; Ariki, Y.; Narihira, T.; Tsuboi, T.; Nagasaka, K. 3d-cnn based heuristic guided task-space planner for faster motion planning. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 9548–9554.
21. Rehder, E.; Naumann, M.; Salscheider, N.O.; Stiller, C. Cooperative motion planning for non-holonomic agents with value iteration networks. *arXiv* **2017**, arXiv:1709.05273.
22. Tamar, A.; Wu, Y.; Thomas, G.; Levine, S.; Abbeel, P. Value iteration networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2154–2162.
23. Luo, M.; Hou, X.; Yang, J. Multi-robot one-target 3d path planning based on improved bioinspired neural network. In Proceedings of the 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, Chengdu, China, 14–15 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 410–413.
24. Ni, J.; Yang, S. X. Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments. *IEEE Trans. Neural Netw.* **2011**, *22*, 2062–2077. [PubMed]
25. Godio, S.; Primatesta, S.; Guglieri, G.; Dovis, F. A Bioinspired Neural Network-Based Approach for Cooperative Coverage Planning of UAVs. *Information* **2021**, *12*, 51. [CrossRef]
26. Cao, X.; Chen, L.; Guo, L.; Han, W. AUV global security path planning based on a potential field bio-inspired neural network in underwater environment. *Intell. Autom. Soft Comput.* **2021**, *27*, 391–407. [CrossRef]
27. Qin, H.; Shao, S.; Wang, T.; Yu, X.; Jiang, Y.; Cao, Z. Review of Autonomous Path Planning Algorithms for Mobile Robots. *Drones* **2023**, *7*, 211. [CrossRef]
28. Liu, L.X.; Wang, X.; Yang, X.; Liu, H.; Li, J.; Wang, P. Path planning techniques for mobile robots: Review and prospect. *Expert Syst. Appl.* **2023**, *227*, 120254. . [CrossRef]
29. Wang, S.; Ma, F.; Yan, X.; Wu, P.; Liu, Y. Adaptive and extendable control of unmanned surface vehicle formations using distributed deep reinforcement learning. *Appl. Ocean Res.* **2021**, *110*, 102590. [CrossRef]
30. 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]
31. Wang, D.; Shen, Y.; Wan, J.; Sha, Q.; Li, G.; Chen, G.; He, B. Sliding mode heading control for AUV based on continuous hybrid model-free and model-based reinforcement learning. *Appl. Ocean Res.* **2022**, *118*, 102960. [CrossRef]
32. Miao, R.; Wang, L.; Pang, S. Coordination of distributed unmanned surface vehicles via model-based reinforcement learning methods. *Appl. Ocean Res.* **2022**, *122*, 103106. [CrossRef]

33. Zheng, S.; Liu, H. Improved multi-agent deep deterministic policy gradient for path planning-based crowd simulation. *IEEE Access* **2019**, *7*, 147755–147770. [CrossRef]
34. Muse, D.; Wermter, S. Actor-critic learning for platform-independent robot navigation. *Cogn. Comput.* **2009**, *1*, 203–220. [CrossRef]
35. Lachekhab F.; Tadjine, M. Goal seeking of mobile robot using fuzzy actor critic learning algorithm. In Proceedings of the 2015 7th International Conference on Modelling, Identification and Control (ICMIC), Sousse, Tunisia, 18–20 December 2015; pp. 1–6.
36. Çetinkaya, M. Multi-Agent Path Planning Using Deep Reinforcement Learning. *arXiv* **2021**, arXiv:2110.01460.
37. Zhang, Y.; Qian, Y.; Yao, Y.; Hu, H.; Xu, Y. Learning to cooperate: Application of deep reinforcement learning for online AGV path finding. In Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, Auckland, New Zealand, 9–13 May 2020; pp. 2077–2079.
38. Liu, Z.; Chen, B.; Zhou, H.; Koushik, G.; Hebert, M.; Zhao, D. Mapper: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 11748–11754.
39. Qiu, H. Multi-agent navigation based on deep reinforcement learning and traditional pathfinding algorithm. *arXiv* **2020**, arXiv:2012.09134.
40. Sartoretti, G.; Kerr, J.; Shi, Y.; Wagner, G.; Kumar, T.S.; Koenig, S.; Choset, H. Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2378–2385. [CrossRef]
41. Risi, S.; Togelius, J. Neuroevolution in games: State of the art and open challenges. *IEEE Trans. Comput. Intell. Games* **2015**, *9*, 25–41. [CrossRef]
42. Stanley, K.O.; Miikkulainen, R. Evolving neural networks through augmenting topologies. *Evol. Comput.* **2002**, *10*, 99–127. [CrossRef]
43. Khadka, S.; Tumer, K. Evolution-guided policy gradient in reinforcement learning. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2018; Volume 31.
44. Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv* **2017**, arXiv:1703.03864.
45. Cheng, S.; Quilodrán-Casas, C.; Ouala, S.; Farchi, A.; Liu, C.; Tandeo, P.; Fablet, R.; Lucor, D.; Iooss, B.; Brajard, J.; et al. Machine learning with data assimilation and uncertainty quantification for dynamical systems: A review. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 1361–1387. [CrossRef]
46. Farchi, A.; Laloyaux, P.; Bonavita, M.; Bocquet, M. Using machine learning to correct model error in data assimilation and forecast applications. *Q. J. R. Meteorol. Soc.* **2021**, *147*, 3067–3084. [CrossRef]
47. Tang, M.; Liu, Y.; Durlofsky, L.J. A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *J. Comput. Phys.* **2020**, *413*, 109456. . [CrossRef]
48. Cheng, S.; Prentice, I.C.; Huang, Y.; Jin, Y.; Guo, Y.K.; Arcucci, R. Data-driven surrogate model with latent data assimilation: Application to wildfire forecasting. *J. Comput. Phys.* **2022**, *464*, 111302. . [CrossRef]
49. Hu, Y.; Wang, D.; Li, J.; Wang, Y.; Shen, H. Adaptive environmental sampling for underwater vehicles based on ant colony optimization algorithm. In *Global Oceans 2020: Singapore–US Gulf Coast*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–9.
50. White, C.C., III; White, D. J. Markov decision processes. *Eur. J. Oper. Res.* **1989**, *39*, 1–16. [CrossRef]
51. Sutton, R.S. *Reinforcement Learning, a Bradford Book*; Bradford Books: Bradford, PA, USA, 1998; pp. 665–685.
52. Mellor, G.L. Users guide for a three-dimensional, primitive equation, numerical ocean model (June 2003 version). *Prog. Atmos. Ocean. Sci.* **2003** . Available online: https://www.researchgate.net/publication/242777179_Users_Guide_For_A_Three-Dimensional_Primitive_Equation_Numerical_Ocean_Model (accessed on 1 November 2023).
53. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 387–395.
54. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
55. Haarnoja T.; Zhou A.; Hartikainen K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P. Soft actor-critic algorithms and applications. *arXiv* **2018**, arXiv:1812.05905.
56. Jahrer, M. Rankgauss. Available online: https://github.com/michaeljahrer/rankGauss (accessed on 1 November 2023).
57. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 25–31 July 2018; pp. 1861–1870.
58. Wang, Z.; Sui, Y.; Qin, H.; Lu, H. State Super Sampling Soft Actor–Critic Algorithm for Multi-AUV Hunting in 3D Underwater Environment. *J. Mar. Sci. Eng.* **2023**, *11*, 1257. [CrossRef]
59. Wang, Z.; Lu, H.; Qin, H.; Sui, Y. Autonomous Underwater Vehicle Path Planning Method of Soft Actor–Critic Based on Game Training. *J. Mar. Sci. Eng.* **2022**, *10*, 2018. [CrossRef]
60. Pourchot, A.; Sigaud, O. CEM-RL: Combining evolutionary and gradient-based methods for policy search. *arXiv* **2018**, arXiv:1810.01222.

61. Van Hasselt , H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
62. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.