Comprehensive Ocean Information Enabled AUV Path Planning via Reinforcement Learning

Meng Xi, Jiachen Yang, *Senior Member, IEEE*, Jiabao Wen, Hankai Liu, Yang Li, Houbing Herbert Song, *Senior Member, IEEE*

***Abstract*—The path planning of the Autonomous Underwater Vehicle (AUV) has shown great potential in various Internet-of-Underwater-Things (IoUT) applications. Although considerable efforts had been made, prior studies are confronted with some limitations. For one thing, existing work only uses the ocean current simulation model without introducing real ocean information, having not been supported by real data. For another, traditional path planning algorithms have strong environment dependence and lack flexibility: once the environment changes, they need to be re-modelled and re-planned. To overcome these challenges, this paper proposes COID, an AUV path planning scheme exploiting comprehensive ocean information and reinforcement learning, which consists of three steps. First, we introduce the comprehensive real ocean data including weather, temperature, thermohaline, current, etc., and apply them into the regional ocean modeling system to generated reliable ocean current. Next, through well-designed state transition function and reward function, we build a 3D grid model of ocean environment for reinforcement learning. Furthermore, based on the framework of Double Dueling Deep Q Network (D3QN), COID integrates local ocean current and position features to provide state input and uses priority sampling to accelerate network convergence. The performance of COID has been evaluated and proved by numerical results, which demonstrate efficient path planning and high flexibility for expansion into different ocean environments.**

***Index Terms*—Internet-of-Underwater-Things (IoUT), Autonomous Underwater Vehicle (AUV), path planning, 3D grid model, reinforcement learning**

I. Introduction

THE ocean is of great importance to the ecological environment and human society. It affects the global climate and ecosystem, and is rich in natural resources such as biology, minerals and energy. Therefore, human beings have been researching efficient, intelligent and environment-friendly ocean exploration technology [1], [2]. Owning to the large detection range, strong maneuverability, long endurance, and high intelligence, the Autonomous Underwater Vehicle (AUV) stands out and has become an effective tool for various Internet-of-Underwater-Things (IoUT) applications which possess high military and civil values [3]. Recently, quantities of researches on AUV have emerged [4]–[6], of which the prominent problem is the path planning of AUV.

As an important research field in control discipline, path planning mainly contains two aspects: environment modeling and path planning algorithm. Environment modeling refers to the process of patterned representation of environment elements, so as to generalize the real environment and facilitate research. An excellent environment model not only fits the physical needs, but also adapts to the path planning algorithm and promotes the convergence. Typical environment modeling methods include grid method [7], visibility graph [8], and Voronoi diagrams [9]. The visibility graph method transforms the path planning problem into mathematical geometry. It builds a unique geometric model for each environment and thus lacks flexibility. The paths generated by Voronoi diagrams are not smooth and the modeling process is complex as well. By contrast, the grid method divides the environment into regular grids, which is easy to implement and expand, and is suitable for various algorithms. Therefore, path planning studies usually adopt the grid method.

The path planning algorithm is to search for an optimal path between the start and terminal points, and can be divided into two categories: traditional algorithms and intelligent bionic algorithms based on biology. The former finds the optimal path by path searching or sampling, such as Dijkstra algorithm [10], A* algorithm [11], Rapidly exploring Random Tree (RRT) [12], and RRT* [13]. Although these algorithms are convenient to implement, when the problem scale expands, the large search range will cause low efficiency and insufficient global optimization. The latter leverages the behavior of biological groups, biological structure or biological evolution mechanism. Typical algorithms include particle swarm optimization [14], ant colony optimization [15], genetic algorithm [16], neural network-based algorithm [17], etc. Compared with traditional algorithms, the intelligent bionic algorithms have obvious advantages in dealing with complex environment. However, they still have deficiencies in strong environment dependence and easy to fall into local optimum.

With the accelerated development of artificial intelligence, machine learning techniques provide more possibilities for solving the underwater problems in IoUT [18]. As a new paradigm of machine learning, Reinforcement Learning (RL) technology makes people surprised as soon as it comes into our sight. The mechanism, imitating human from scratch to

This work was partially supported by National Natural Science Foundation of China (No. 61871283), Foundation of Pre-Research on Equipment of China (No.61400010304), and Major Civil-Military Integration Project in Tianjin, China (No.18ZXJMTG00170). *(Corresponding author: Jiachen Yang.)*

Meng Xi, Jiachen Yang, Jiabao Wen and Yang Li are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mails: {ximeng, yangjiachen, Wen_Jiabao}@tju.edu.cn, liyang328@shzu.edu.cn).

Hankai Liu is with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: hkliu@tju.edu.cn).

Houbing Herbert Song is with the Security and Optimization for Networked Globe Laboratory (SONG Lab), Embry-Riddle Aeronautical University, Daytona Beach, FL, 32114, USA (e-mail: h.song@ieee.org).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3155697, IEEE Internet of Things Journal

master a skill, endows RL with high intelligence, flexibility and adaptability. At present, RL has been applied in many fields, such as robot control [19], recommendation system [20], and automatic driving [21]. Furthermore, some RL based path planning methods have emerged and shown great potential in overcoming the dependence on environment [22]. Most studies are based on Q-learning algorithm [23] and Deep Q Network (DQN) algorithm [24]. The Q-learning algorithm establishes a table to store the values of all the existing state-action pairs in the environment, which then will be read by inquiries. As a result, it is only suitable for smaller environments and fewer states. When the environment expands, it will bring memory burden, reduce efficiency, and even fail to converge. Although DQN uses neural network to estimate the values and overcomes the disadvantages of Q-learning, its updating mechanism leads to overestimation, resulting in poor stability.

Generally, the path planning of AUV is confronted with two challenges.

1) Authenticity of Environment Model: There is a trade-off between simplification and authenticity in the environment modeling of path planning. Existing studies abstract the mathematical model from the physical demand to simplify the problem. Some convert the ocean environment into a 2D model, which only considers the 2D horizontal currents but ignores the vertical currents. Some synthesize the 3D ocean current by simple mathematical functions. As a result, the ocean environment models are divorced from reality, and the algorithms lack the support of real data.

2) Effectiveness of Path Planning Algorithm: Due to the complexity of ocean environment, the motion of AUV is affected by many factors, especially the ocean current. An intelligent algorithm should make full use of ocean current to save the time, reduce the energy consumption and shorten the path length. However, most existing algorithms ignore the use of ocean current, resulting in poor flexibility and adaptability, and their paths are usually unstable because of overestimation.

To overcome these challenges, we propose an AUV path planning scheme, COID (Comprehensive Ocean Information D3QN), of which the environment model reduces the gap with the practical applications, and the algorithm provides a flexible and stable path. It introduces the real data of the comprehensive ocean information including weather, temperature, thermohaline, current, etc., and applies the Regional Ocean Modeling System (ROMS) to generate the reliable 3D ocean current data. Then, we establish a 3D grid model for reinforcement learning environment, which elaborates the state transition function and the reward function. The state transition function helps to determine the AUV position under the combined function of ocean current and action policy; the reward function points out the learning goal, guides the neural network updating, and accelerates the learning process. Based on the characteristics of the ocean environment, we design an AUV path planning algorithm. It integrates the local ocean current with position features and transforms the original observation into state input, which helps to take advantages of ocean current to improves the intelligence and adaptability. The proposed algorithm employs the framework of Double Dueling Deep Q Network (D3QN), which effectively avoids the overestimation and improves the stability. Furthermore, priority sampling is used to accelerate network convergence.

TABLE I: Notations

Name	Description
$\mathcal{D}$	A replay buffer based on priority sampling
$\mathcal{N}$	Capacity of replay buffer
p_l	Priority of experience in replay buffer $\mathcal{D}$
θ	Parameters of current network
θ'	Parameters of target network
θ_s	Parameters of state value branch in current network
θ_a	Parameters of action value branch in current network
c	Global counter for the decay of exploration
M	Maximum training episode
t	Number of time step in a single episode
I_{done}	Finish indicator in a single episode
P^t	Position of AUV at the time step t
$\vec{P}^t_{cur}$	Ocean current value of the position P^t
T	Maximum time step in a single episode
ε	Exploration probability of random action
ω_ε	Exploration decay factor
f_c	Update frequency of current network
f_t	Update frequency of target network
N	Update times of current network
m	Size of minibatch
E_i	A piece of experience in replay buffer $\mathcal{D}$
P_j	Sampling probability of experience E_j
α	Weight of priority for the sampling probability P_j
ϖ_j	Weight of sample for loss function
β	Impact factor of weight ϖ_j
ω_β	Weight increase factor
γ	Discount factor for reward
η	Learning rate for update
δ_j	Temporal Difference error (TD-error)

The main contributions of this paper are as follows:

1. For the first time, we introduce real data into the ocean environment model for AUV path planning, which effectively narrows the gap with practical applications. We utilize ROMS and comprehensive ocean information to generate the 3D ocean current data, which makes up for the lack of authenticity of existing work. Compared with the existing models, it is more reliable and has higher practical value.

2. An RL environment with well-designed state transition function and reward function is established, which accurately summarizes ocean environmental characteristics and guides the network update. This RL environment provides real data support and is conducive to the convergence of the algorithms.

3. We propose an RL based AUV path planning algorithm, which integrates the comprehensive ocean information to assist AUV planning path. The utilization of ocean current helps to improve adaptability and expand to robust environments, and the employment of the D3QN framework overcomes the overestimation to ensure better stability.

The notations in the text are listed in Table I and the rest of this paper is organized as follows. In Section II, we illustrate the background knowledge of ocean current models and D3QN. Then, the details of our method are described in Section III. In Section IV, we carry out a series of experiments and analyze their results. Some related studies are introduced in Section V. Finally, we summarize the work of this paper and discuss the future directions in Section VI.

II. Background

A. Ocean Current Modeling

The ocean environment is a 3D space with uncertainty, heterogeneity and variability, and the motion of AUV is influenced by many factors, especially the ocean current. Following the ocean current, AUV can shorten time, improve speed, save energy, and vice versa. Therefore, a complete ocean environment model for AUV path planning requires the ocean current. A great number of researches on ocean current model have emerged, among which the most common method is the mathematical function fitting. For example, Chen *et al.* [25] introduce the typical non-circulating ocean current

$$P_{cur}(x,y,t) = 1 - \tanh\left(\frac{y - \lambda(t)\cos(\kappa(x-\varphi t))}{\sqrt{1+\left(\rho\lambda(t)\sin(\kappa(y-\varphi t))\right)^2}}\right), \tag{1}$$

$$\lambda(t) = \lambda_0 + k\cos(\omega_0 t + \phi), \tag{2}$$

where $P_{cur}(x,y,t)$ represents the ocean current value of position $P(x,y)$. The velocity is (V_x, V_y), where

$$V_x = \frac{\partial P_{cur}(x,y,t)}{\partial x}, \quad V_y = \frac{\partial P_{cur}(x,y,t)}{\partial y}. \tag{3}$$

For the more complicated eddy current, the velocity components of longitude and latitude are

$$U(x,y) = U_{\max}\cos(x - P_{cur})\sin(y - P_{cur}), \tag{4}$$

$$V(x,y) = -V_{\max}\sin(x - P_{cur})\cos(y - P_{cur}). \tag{5}$$

There are some limitations in the ocean current models generated from mathematical functions. The ocean current is an integrated result of numerous factors, such as climate, temperature, season, and topography. The comprehensive ocean information and physical constraints are not taken into account in these models. The vertical velocity of the 3D ocean current, which is much smaller than the longitude and latitude components, is usually ignored for the sake of simplicity. Generally, in the ocean environment of AUV path planning, the ocean current models usually lack the support of real data and are different from the practical application requirements.

B. Double Dueling Deep Q Network

RL solves problems through interaction, which divides the system into two parts: agent and environment. The agent learns policies, takes actions and receives rewards; the environment changes according to the state transition function, gives rewards or punishments by the reward function. The ultimate goal is to maximize the reward and the agent updates the parameters accordingly. The interaction can be summarized by the Markov Decision Process (MDP) and represented by the five tuple $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{P} >$. $\mathcal{S}$ is the state space, which is a collection of all states; the action space $\mathcal{A}$ is a collection of all actions; $\mathcal{T}$ represents the state transition probability of the environment and $\mathcal{R}$ is the corresponding immediate reward; agent makes the action decision according to the policy $\mathcal{P}$. The combination (s, a, r, s') represents a concrete process of interaction, which is called a piece of experience. Agent interacts with the environment to collect pieces of experience and then samples for learning.

Q-learning is a basic algorithm of RL, which constructs a table (Q table) to store the value of all the state-action pairs, $Q(s,a)$. $Q(s,a)$ represents the reward expectation obtained when action a is taken in the case of state s. Through interaction, Q-learning updates the Q table by Eq. (6) to find the best action for the maximum reward

$$Q(s,a) = Q(s,a) + \eta\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right), \tag{6}$$

where $Q(s',a')$ means the value of next state-action pair; η is the learning rate and γ is the reward discount. Q-learning is only suitable for the problem of discrete state and action space with small dimension, which can be listed in a table.

DQN introduces a neural network to overcome this limitation, which is regarded as a function fitting problem, $Q(s,a;\theta) \approx Q(s,a)$. θ is the parameter of the neural network. At the same time, DQN also introduces other innovations, two neural networks and replay buffer $\mathcal{D}$. These two neural networks, one is called current network $Q(s,a;\theta)$ and the other is target network $Q'(s,a;\theta')$. The current network participates in the information flow of forward propagation and outputs the estimated value of state-action pair. The target network is used to find the maximum value of next state-action pair. For updating, the loss function is calculated by

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

where $E[\cdot]$ represents the expectation. Then, the current network parameter is updated by gradient descent as

$$\nabla_\theta Loss = E[(r + \gamma \max_{a'} Q'(s',a';\theta') - Q(s,a;\theta))\nabla_\theta Q(s,a;\theta)]. \tag{8}$$

The selection of the maximum value of next state-action pair by DQN leads to overestimation.

The design of D3QN solves the problem of overestimation. It improves DQN in two aspects: network structure and update mechanism. In the network structure, D3QN decouples the value of state-action pair into state value and action value. The network parameters become $\theta = [\theta_s, \theta_a]$, in which θ_s and θ_a represent state value branch and action value branch respectively. During updating, in stead of directly outputting the maximum value of the next state-action pair by target network, the current network selects the best action, and then the target network outputs the value of the corresponding state-action pair. The double decoupling effectively improves the learning stability and convergence speed. The performance of D3QN has been proved by many applications, so that it becomes the mainstream algorithm for RL.

III. Methodology

Based on the real data of the comprehensive ocean information, we propose COID, an RL based AUV path planning scheme, of which the detailed technologies are as follows.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3155697, IEEE Internet of Things Journal

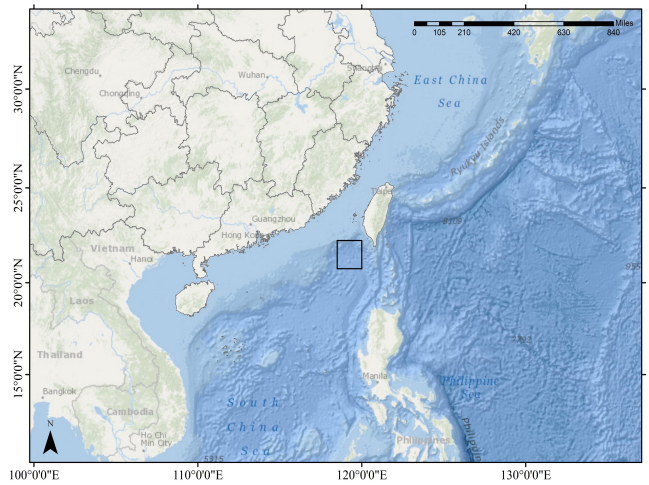


Fig. 1: Topographic map of the South China Sea. The black box identifies the research sea area, $118.5°E$ to $120°E$, $20.75°N$ to $22.25°N$, and $-3600M$ below the surface.

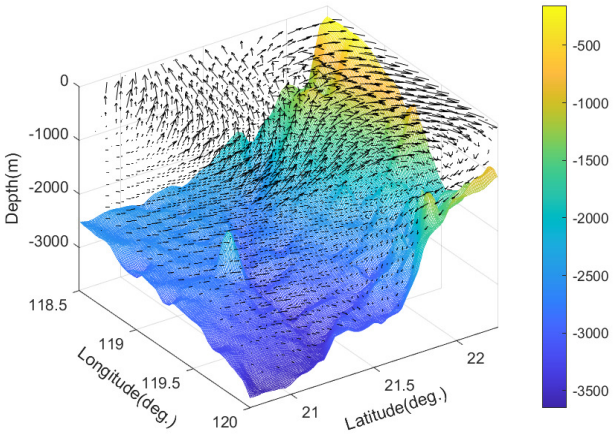


Fig. 2: The visualization of the 3D grid ocean environment model of the research sea area, which is divided into $50 \times 50 \times 50$ grids. The black arrow on each grid indicates the 3D ocean current.

A. Utilization of Comprehensive Ocean Information

To solve the contradiction between simplification and authenticity in the environmental modeling, we introduce the real data of the comprehensive ocean information to narrow the gap with the practical application scenarios, which can effectively improve the reliability of the simulation environment and provide real data support for the path planning algorithms.

As an ocean environment simulation tool, ROMS integrates comprehensive physical and mathematical constraints, which has been widely used in the simulations of hydrodynamics and water circulation. Therefore, we exploit ROMS to generate the ocean current data to overcome the limitations of mathematical functions in environment modeling. It follows the conventions of the Earth System Modeling Framework (ESMF) for model coupling, with a sophisticated initial field, boundary field, forcing field, and generating coupled 3D ocean current data. This paper takes part of the South China Sea as the research object ($118.5°E$ to $120°E$, $20.75°N$ to $22.25°N$, and $-3600M$ below the surface), as shown in Fig.1. For initial field, we input the ocean current and surface information in Feb. 2018 and the thermohaline data in Nov. 2020. The corresponding edge of the initial field and the thermohaline record in Nov. 2013 are set as the boundary field. We add the weather information of Feb. 2018 for the forcing field.

Through ROMS, we obtain the reliable ocean current data, which are then utilized to constructed a 3D grid ocean environment model. The target sea area is divided into $50\times50\times50$ grids, and the ocean current data are interpolated accordingly, so that each grid point P has a corresponding ocean current $\vec{P}_{cur}(u, v, w)$. Specifically, u, v represent the 2D ocean current, along the longitude and latitude. w is vertical current, which is normally smaller. Fig. 2 visualizes this 3D grid model of the target sea area, in which the black arrow of each grid indicates the corresponding 3D ocean current. The black arrow points to the direction, and the length represents the ocean current value $\left\|\vec{P}_{cur}\right\|$.

B. Reinforcement Learning Environment

On the basis of the 3D grid ocean environment model, we design an RL environment, which mainly comprises following two essential modules: the state transition function and the reward function.

1) State Transition Function: The state space $\mathcal{S}$ is a continuous vector of six dimensions, which is the input information of the neural network. The action space $\mathcal{A}$ is represented by a six dimensional discrete vector, which indicates six motion directions. The state transition function generalizes the environmental changes and is the fitting object of neural network. A thoughtful and accurate state transition function helps to improve the performance of the algorithm. In the complex ocean environment, the motion of AUV is affected by the joint action of external conditions and internal motivation. For simplicity, AUV is usually reduced to a mass point and we use grid coordinates uniformly. Suppose the AUV transfers from the position $P(x, y, z)$ to the next position $P'(x', y', z')$. The spatial coordinates are continuous while the value of ocean current is discrete, thus the ocean current of position P needs to be interpolated by its eight adjacent grid points $P^i, i = 1, 2 \cdots 8$. The Euclidean distance between two points is calculated by

$$Dis(P, P') = \sqrt{(x - x')^2 + (y - y')^2 + (z - z')^2}. \quad (9)$$

Then, the value of the ocean current at position P is

$$\vec{P}_{cur} = \sum \vec{P}^i_{cur} Dis(P, P^i) \Big/ \sum Dis(P, P^i). \quad (10)$$

The action $a = [a_1, a_2, a_3, a_4, a_5, a_6]^{\mathrm{T}}$, $a \in \mathcal{A}$ indicates the movement of the AUV in six directions, in which a_1 and a_2 along the direction of longitude, a_3, a_4 along the latitude, and a_5, a_6 along the verticality. The next position $P'(x', y', z')$ is calculated as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3155697, IEEE Internet of Things Journal

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + V_{AUV} \begin{bmatrix} \hbar(a_1, a_2) \\ \hbar(a_3, a_4) \\ \hbar(a_5, a_6) \end{bmatrix} + V_{cur} \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad (11)$$

in which

$$\hbar(p, q) = \mathrm{sgn}(p) - \mathrm{sgn}(q) \quad (12)$$

indicates the direction of motion determined jointly by p and q. Moreover, the ratio of ocean current velocity V_{cur} to AUV velocity V_{AUV} reflects the ocean current intensity

$$I_{cur} = V_{cur}/V_{AUV}. \quad (13)$$

2) Reward Function: In RL, the purpose of the agent is to maximize the expectation of reward, which further influences the direction of the neural network gradient update. Therefore, the reward function is essential and should be carefully designed for specific applications. In our model, the reward function includes the following four items: the distance reward r_{dis}, the ocean current reward r_{cur}, the step reward r_{step} and the goal reward r_{goal}. The distance reward is

$$r_{dis} = Dis(P, P^{Goal}), \quad (14)$$

which calculates the grid distance between the present position and the terminal point. It guides the AUV to approach the target point. The ocean current reward r_{cur} makes use of the ocean current, which helps to encourage the AUV to follow the ocean current to shorten the moving time and reduce energy consumption. This paper expresses the item r_{cur} as

$$r_{cur} = \cos(\sum_{i=1}^{6} (\mathrm{sgn}(a_i)\pi - \phi_i)), \quad (15)$$

which is only related to the direction of the ocean current and ϕ_i represents the angle between the ocean current component and the corresponding plane. The step reward r_{step} is the number of time steps executed in one episode. The goal reward r_{goal} is a huge attraction and only available when the AUV successfully reaches the terminal point. Finally, the reward function is expressed as

$$r = c_1 r_{dis} + c_2 r_{cur} + c_3 r_{step} + I_{done} r_{goal}, \quad (16)$$

where c_i is the weighting factor and I_{done} is the finish indicator. Specifically, $c_1 = -0.7$, $c_2 = 1$, $c_3 = 10^{-3}$, $r_{goal} = 10^3$, and $I_{done} = 0$ or 1. Under the regulation of such reward function, the AUV fully utilizes the ocean current to reach the terminal point as fast as possible.

C. Path Planning Algorithm

One remarkable characteristic of ocean environment is the uncertainty, i.e., the ocean current has a great influence on the movement of AUV. The path planning of AUV puts forward higher requirements for the flexibility of the algorithm. Taking advantage of the ocean current can effectively improve the intelligence and adaptability. Traditional path planning algorithms have limited capabilities, so we design an RL based algorithm, COID.

Algorithm 1: COID Algorithm

Input : Replay buffer capacity $\mathcal{N}$, initial priority $p_l = 1$, current network $Q(s, a; (\theta_s, \theta_a))$, target network $Q'(s, a; (\theta'_s, \theta'_a))$, $\theta'_s \leftarrow \theta_s, \theta'_a \leftarrow \theta_a$.

Global counter $c = 0$
for $n = 1, M$ **do**
  $t = 0$, $I_{done} = False$, start point P^0, initial state s_0
  while $t <= T$ *or NOT* I_{done} **do**
    Integrate the position P^t and the local ocean current $\vec{P}^t_{cur} : s_t \leftarrow \Gamma(P^t, \vec{P}^t_{cur})$
    Estimate action value
    $$Q(s_t, a; (\theta_s, \theta_a)) = q(s_t; \theta_s) + q(s_t, a; \theta_a) - AVE(s_t)$$
    Choose action a_t
    $$a_t = \begin{cases} random\ a \in \mathcal{A}, & \varepsilon_t \\ \arg\max_a Q(s_t, a; (\theta_s, \theta_a)), & 1 - \varepsilon_t \end{cases}$$
    Transfer state according to $\mathcal{T}$
    Receive s_{t+1}, r_t, and I_{done}
    Store the experience (s_t, a_t, r_t, s_{t+1}) into $\mathcal{D}$
    $t \leftarrow t + 1$, $c \leftarrow c + 1$
  end
  Update current network
  if $n\ mod\ f_c\ ==\ 0$ **then**
    for $i = 1, N$ **do**
      for $j = 1, m$ **do**
        Probability $P_j = \frac{p_j^\alpha}{\sum_{p_l \in \mathcal{D}} p_l^\alpha}$
        Weight $\varpi_j = \frac{1}{(\mathcal{N} P_j)^\beta \max \varpi_l}$
        Sample a piece of experience $E_j \sim (P(j), \varpi_j)$
        Find best action by current network $a_{t+1} = \arg\max_a Q(s_t, a; (\theta_s, \theta_a))$
        Calculate update target $y_t = r_t + \gamma Q'(s_{t+1}, a_{t+1}; (\theta'_s, \theta'_a))$
      end
      Calculate loss function
      $$Loss = \frac{1}{m} \sum_{j=1}^{m} (y_t - Q(s_t, a_t; (\theta_s, \theta_a)))^2 \varpi_j$$
      Update parameter $\theta \leftarrow \theta + \eta \nabla_{(\theta_s, \theta_a)} Loss$
      for $j = 1, m$ **do**
        TD-error $\delta_j = y_t - Q(s_t, a_t; (\theta_s, \theta_a))$
        Update priority $p_j \leftarrow |\delta_j| + \chi$
      end
    end
  end
  Update target network
  if $n\ mod\ f_t\ ==\ 0$ **then**
    Copy from the current network $\theta'_s \leftarrow \theta_s, \theta'_a \leftarrow \theta_a$
  end
end

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3155697, IEEE Internet of Things Journal

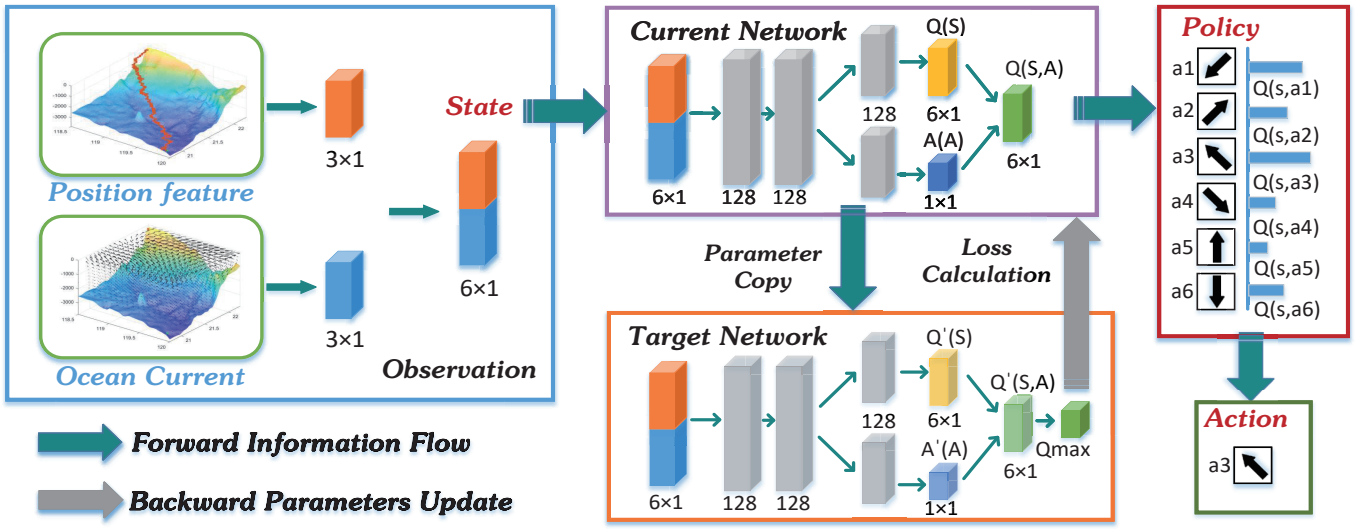


Fig. 3: The pipeline of COID.

At first, COID integrates the local ocean current information $\vec{P}_{cur}(u,v,w)$ and the position feature $P(x,y,z)$ by

$$\Gamma(P,\vec{P}_{cur}) = [x,y,z,\rho,\iota,\sigma]^{\mathrm{T}}. \tag{17}$$

Specifically, it processes the local ocean current information and transforms the spatial coordinate values $\vec{P}_{cur}(u,v,w)$ into relative values (ρ,ι,σ) by

$$\rho = \left\|\vec{P}_{cur}\right\|_2, \tag{18}$$

$$\iota = \arccos(w/\left\|\vec{P}_{cur}\right\|_2), \tag{19}$$

$$\sigma = -\pi/2\,(\mathrm{sgn}(u)-1)\,\mathrm{sgn}(v) + \arctan(v/u). \tag{20}$$

Such feature integration can effectively improves flexibility, because the supplement of ocean current information helps the agent summarize the uncertainty of the environment, so as to master the ability to exploit the ocean current, improving the speed, shortening the time and saving the energy consumption.

There is a problem of exploration and exploitation dilemma in RL, which means that the agent chooses actions based on experience or explores the unknown. Too much dependence on experience may lead to falling into the local optimum and unable to find the global optimum. On the contrary, too much exploration will increase the training time and reduce the effect. To balance this contradiction, we decay the exploration probability of random action by

$$\varepsilon = \varepsilon_{final} + (\varepsilon_{start} - \varepsilon_{final}) \times e^{\frac{-c}{\omega_\varepsilon}}, \tag{21}$$

which ensures a gradual decline with the increase of the global counter c. The initial exploration is $\varepsilon_{start} = 0.9$, final exploration is $\varepsilon_{final} = 10^{-2}$ and exploration decay factor is $\omega_\varepsilon = 8 \times 10^6$. The action is determined by

$$a = \begin{cases} random\ a \in \mathcal{A}, & \varepsilon \\ \arg\max\limits_a Q(s,a;\theta), & 1-\varepsilon \end{cases}. \tag{22}$$

Then, to pursue better learning performance, COID uses the framework of D3QN, which has become the mainstream basic algorithm for discrete task. As the network structure shown in Fig. 3, COID extracts the common feature through two fully connected layers. After that it is divided into two branches: $Q(S)$ and $Q(A)$, which respectively estimate the state value $q(s;\theta_s)$ and the action value $q(s,a;\theta_a)$. Then the value of state-action pair is calculated by

$$Q(s,a;(\theta_s,\theta_a)) = q(s;\theta_s) + q(s,a;\theta_a) - AVE(s), \tag{23}$$

in which

$$AVE(s) = \frac{1}{|\mathcal{A}|}\sum_{a_i \in \mathcal{A}} q(s,a_i;\theta_a) \tag{24}$$

means the average value of action value. This independent structure decouples the state value and action value, which effectively improves the learning reliability and accelerates the convergence speed. To update the network parameters, COID selects the action by the current network

$$a' = \arg\max_a Q(s',a;(\theta_s,\theta_a)). \tag{25}$$

Subsequently, the target network outputs the value of the corresponding state-action pair $Q'(s',a';(\theta'_s,\theta'_a))$. This update mode decouples action selection and value estimation, which solves the problem of overestimation. The loss function is

$$Loss{=}E[(r+\gamma Q'(s',a';(\theta'_s,\theta'_a)) - Q(s,a;(\theta_s,\theta_a)))^2]. \tag{26}$$

The current network updates by

$$(\theta_s,\theta_a) \leftarrow (\theta_s,\theta_a) + \eta\nabla_{(\theta_s,\theta_a)} Loss. \tag{27}$$

According to the task requirements and experimental comparison, we determine that the reward discount $\gamma = 0.99$ and the learning rate $\eta = 10^{-4}$.

In addition, to further accelerate the convergence speed and improve the efficiency, COID employs priority sampling for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3155697, IEEE Internet of Things Journal

network parameter updates. For each update, a minibatch of experience is sampled with the probability

$$P_j = \frac{p_j^\alpha}{\sum_{p_l \in \mathcal{D}} p_l^\alpha}, \quad j = 1, 2, \dots m, \tag{28}$$

where m is the size of minibatch and α is the weight of priority for sample probability. The priorities are updated by the Temporal Difference error (TD-error), which is usually used to solve the sequence prediction problem. When $\alpha{=}0$, Eq. (28) is equivalent to uniform sampling; when $\alpha{=}1$, it becomes greedy sampling which always selects the sample with the highest priority. At the same time, the loss function is rewritten as

$$Loss{=}\frac{1}{m}\sum_{j=1}^{m}(r{+}\gamma Q'\left(s', a'; (\theta'_s, \theta'_a)\right){-}Q\left(s, a; (\theta_s, \theta_a)\right))^2 \varpi_j, \tag{29}$$

where

$$\varpi_j = \frac{1}{(\mathcal{N} P_j)^\beta \max \varpi_l} \tag{30}$$

is the weight of priority sample. $\mathcal{N}$ represents the capacity of the replay buffer. β is the impact factor of weight and expressed as

$$\beta = \min(1, \beta_{start} + \frac{k(1-\beta_{start})}{\omega_\beta}). \tag{31}$$

The initial weight $\beta_{start} = 0.4$, the weight increase factor $\omega_\beta = 10^5$ and k is the sampling times.

To summarize, COID integrates the ocean current information and position feature into the state input of neural network, so that AUV can intelligently take advantages of ocean current for path planning to possess better flexibility. The double decoupling of action value & state value and action selection & value prediction overcomes the problem of overestimation and makes the path more stable and reliable. The priority based sampling speeds up the network convergence and improves the efficiency of the algorithm. The information flow of the algorithm and the network structure are shown in Fig. 3. The specific process of parameter update and iteration is shown in Algorithm 1.

IV. Experiments

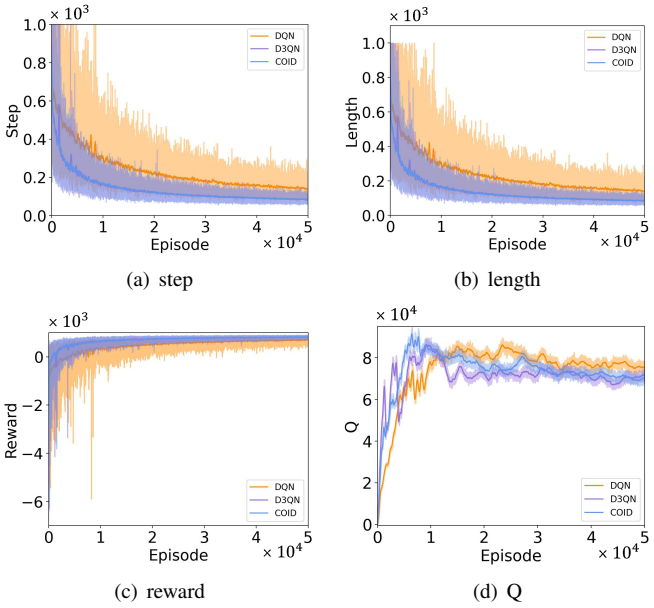
A. Basic Settings and Evaluation Indicators

We take part of the South China Sea as research object ($118.5°E$ to $120°E$, $20.75°N$ to $22.25°N$, and $-3600M$ below the surface). The real data of comprehensive ocean information are from the National Marine Data Center [26], IRI/LDEO Climate Data Library [27], and European Centre for Medium-Range Weather Forecasts [28]. Through ROMS, we generate the reliable ocean current data and interpolate it into the grid of $50 \times 50 \times 50$. As a contrast, we also implement algorithms of DQN and D3QN. All algorithms are set with the same default parameters, which are listed in Table II. The experiments run on the Ubuntu 20.04, with 16G RAM, NVIDIA GTX 3070 GPU, and python 3.7.

To quantify the performance of different algorithms, we use the following two indicators to evaluate the planned path.

TABLE II: Parameter setting

Parameter	Value
Weighting factor c_1, c_2, c_3	-0.7, 1, 10^{-3}
Finish indicator I_{done}	0 or 1
Goal reward r_{goal}	10^3
Initial exploration ε_{start}	0.9
Final exploration ε_{final}	10^{-2}
Exploration decay factor ω_ε	8×10^6
Network width w	2^7
Replay buffer capacity $\mathcal{N}$	2^{17}
Maximum training episode M	5×10^4
Maximum time step T	10^3
Current network update frequence f_c	1
Update times N	2^7
Minibatch size m	2^7
Weight of priority α	0.6
Initial weight β_{start}	0.4
Weight increase factor ω_β	10^5
Discount factor γ	0.99
Learning rate η	10^{-4}
Target network update frequence f_t	4



(a) step (b) length (c) reward (d) Q

Fig. 4: The results of training process.

1) Step: The AUV chooses an action according to the policy at a fixed interval time step. Then it moves to the next position based on the state transition function. The number of steps reflects interaction times and time consumption.

2) Length: For convenience and standardization, this paper uses grid coordinates uniformly. Therefore, we calculate the grid distance between two adjacent positions on the track and measure the total length of the whole path.

B. Effectiveness

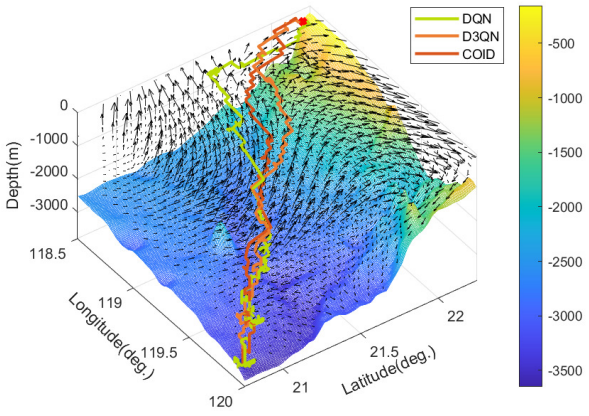
We carry out a validation experiment of effectiveness. Fig. 4 records the training results, in which Fig. 4(a), 4(b) and 4(c) restore the whole process. In the initial stage, the AUV is ignorant and it randomly chooses actions to explore the whole sea area. At this point, the number of time steps is approximately equal to the maximum time step T, and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3155697, IEEE Internet of Things Journal

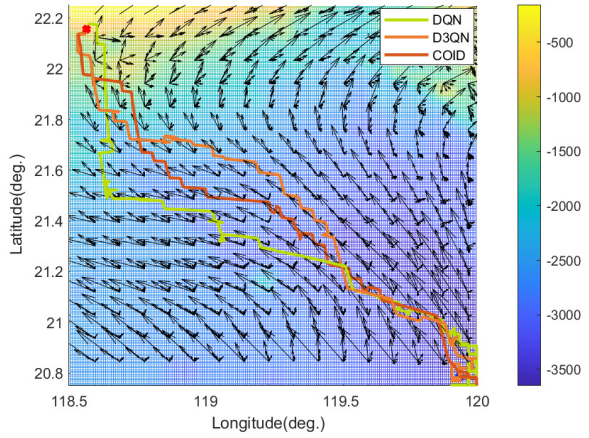
length of the path is quite long. The experience of accidental success brings higher reward, which is stored in the replay buffer. Next, the AUV learns towards the direction of increasing reward and updates the network parameters according to the algorithms mentioned in Section III-C. Finally, with the convergence of the algorithms, the AUV finds a stable and optimal path. The step and path length of COID and D3QN are far less than DQN. Moreover, the variances of D3QN and COID are obviously smaller than that of DQN. Fig. 4(d) tracks the Q value, in which COID maintains a faster and smoother convergence. In addition, its smaller estimate indicates that it overcomes the overestimation as mentioned in Section II-B.

Fig. 5 visualizes the planned paths from the start point $(48, 2, 3)$ to the terminal point $(3, 47, 49)$ in the 3D grid model. To show the details clearly, they are further shown from the vertical view in Fig. 5(b). The path planned by COID is better than the other two, which is shorter and smoother. The reason is that COID masters the characteristics of the ocean current and mostly moves along it, thereby effectively saves time and energy consumption.

Generally, this validation experiment fully proves the effectiveness of COID. For the training process, it is faster and more stable; for the result, it fully exploits the ocean current, which makes the path shorter and smoother.



(a) 3D view



(b) Vertical view

Fig. 5: The planned paths from (119.94, 20.78, -3335) to (118.56, 22.16, -70.97). (The corresponding start and terminal point in the 3D grid model are (48,2,3) and (3,47,49).) The blue and red symbol denote start and terminal point respectively. (a) is the 3D view. (b) is the vertical view.

C. Influence of Ocean Current

Ocean current has a profound impact on the path planning of AUV. The influence can be further divide into the current direction and current intensity. Therefore, we elaborate two series of comparative experiments.

1) Ocean Current Direction: To investigate the influence of ocean current direction, we design six path planning tasks as shown in Table III, which have different start and terminal points. Thereby, these paths are accompanied by different ocean current directions. In particular, Path 1 & 2, 3 & 4, and 5 & 6 exchange their start and terminal points. Such setting ensures that the movement of the AUV at the same position will be with or against the current direction due to the different terminal points.

Table III records the results of the six paths. It can be inferred that most of the currents encountered on the Path 1, 3, 5 follow the motion direction of AUV. On the contrary, the Path 2, 4, 6 are mostly counter the current directions, because they spend more time steps to reach the termianl points and their path lengths are longer. On the whole, the performance of COID is better than that of DQN and D3QN, with fewer time step and shorter path length. Moreover, it is well known that the longer the grid distance between the start and the terminal point, the greater the difficulty of path planning task. For the shorter Path 5 & 6, these three algorithms can achieve similar results. But in the longer Path 1 & 2, COID is obviously better than the other two, which fully proves that our method has sufficient ability to deal with complex ocean current directions. It can adapt to different directions and remain stable no matter how difficult the conditions are.

To further analyze the influence of ocean current direction on AUV path planning, we visualize the Path 1 & 2 planned by COID. The start and terminal points of Path 1 are (119.94, 20.78, -3335) and (118.56, 22.16, -70.97), and Path 2 is the opposite. Fig. 6 shows the 3D trajectories from the vertical view for a clear illustration. The trajectory of Path 1 mostly coincides with the direction of ocean current, so it is smooth and takes fewer time step. Conversely, Path 2 is mainly against the direction of the ocean current, so the AUV has to try its best to overcome the resistance caused by the ocean current. It first moves along the ocean current to the border and then goes against to reach the terminal point, consuming more energy and forming a longer trajectory. We can conclude from the above experiments the ocean currents direction plays an important role in the AUV path planning, and the reasonable utilization of ocean currents can effectively shorten the time and save energy.

2) Ocean Current Intensity: The ocean current intensity is another critical factor that deeply affects the path planning performance. The movement of AUV is determined by the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3155697, IEEE Internet of Things Journal

TABLE III: The results of different ocean current directions.

Path	Start Point	Terminal Point	DQN		D3QN		COID	
			step	length	step	length	step	length
1	(48, 2, 3)	(3, 47, 49)	109.8	123.0732	106.2	123.0600	**106.2**	**122.3556**
2	(3, 47, 49)	(48, 2, 3)	130.8	133.6667	124.4	**133.1060**	**123.8**	133.1505
3	(5, 7, 16)	(47, 48, 45)	102.8	106.6450	101.0	106.1606	**100.8**	**105.8459**
4	(47, 48, 45)	(5, 7, 16)	**109.6**	111.2950	110.0	111.1526	109.8	**110.3932**
5	(4, 20, 19)	(46, 22, 10)	49.0	49.6560	49.0	49.6196	**49.0**	**49.6097**
6	(46, 22, 10)	(4, 20, 19)	53.0	**52.1591**	53.0	52.8208	**53.0**	52.8118

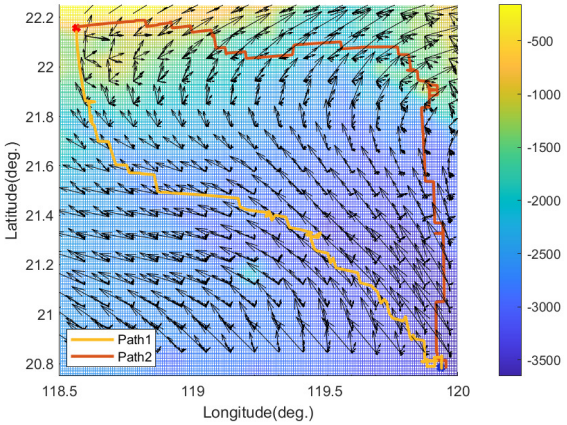


Fig. 6: The influence of ocean current direction.

combined function of action policy and external environment. The ocean current $\vec{P}_{cur}(u, v, w)$ is an environmental uncertainty factor and affects the next position of AUV in the state transition function of Eq. (11). According to Eq.(13), when the ocean current intensity I_{cur} is small, the impact of uncertainty is relatively small. And vice versa, the ocean current will become a non-negligible influence factor. Thus, the larger the current intensity, the greater the difficulty of the path planning task. We conduct the experiments with a wide range of intensities, and the results are recorded in Fig. 7.

Overall, the performance of COID is better than that of DQN and D3QN. As previously analyzed, the current intensity affects the next position of AUV. Therefore, the greater the intensity, the fewer steps are required. Only the step count curve of COID conforms to the gradual decreasing trend of the theoretical analysis. This phenomenon illustrates the good property that COID is sensitive to the ocean current intensity, so that it can make better use of them to shorten the number of steps needed and thus reduce energy consumption. Although the ocean current adds huge uncertainty to the state transition function, comprehensive ocean information assists COID to adapt to the difficult patterns and grasps the skills. The greater the challenge of the path planning task, the more obvious the advantage of COID.

As analyzed above, ocean current brings uncertainty to the environment. To further investigate the effect of current intensity and evaluate the robustness of the algorithm, we elaborate the experiments in Fig. 8. Train the AUVs at a fixed intensity of 0.8, which is a common situation in practical applications, and test under other intensities. The results of COID are generally better than that of DQN and D3QN in terms of the two evaluation indicators. Although the step count curves all decrease as the ocean current intensity increases, which is consistent with the theoretical analysis, the magnitude of COID is significantly greater. This reflects the good adaptability and robustness of COID to the environment, which can cope with a wide range of ocean current intensities. In addition, COID also keeps a relatively good performance on the path length, especially when the ocean current intensity is greater than 0.8.

In conclusion, the current intensity has a significant impact on the time step, and our method can maintain better performance and robustness, even in the most uncertain environments. The integration of the local ocean current information and the position features enriches the input characteristics of neural network, so as to improve the learning ability and adaptability. Therefore, COID can deal with the environmental uncertainty caused by the ocean current intensities, and a well-trained model can be extended to more environments.

V. Related Work

A. Environment Modeling

Environment modeling is the basis of path planning, among which grid method is most convenient and widespread. The grid environment models start from 2D [29], [30] and then expand to 3D [31], [32]. However, part of them have non-negligible limitations that they omit the ocean current, a major factor affecting the path of AUV. Following the ocean current, AUV can save time, improve speed, and reduce energy consumption [33], [34]. Then, a number of institutions and scholars have made efforts to introduce ocean current into the ocean environment. For example, Chen *et al.* [35] introduced a constant ocean current model to represent the effect of ocean current on AUVs. Ma *et al.* [36] added the ocean current constraints into the AUV's path. Yao *et al.* [37] designed a space-variable but time-constant ocean current model and assumed to be known in advance. Chen *et al.* [25] used two mathematical models to generate non-circulating but time-varying ocean currents and stationary but complex spatial

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3155697, IEEE Internet of Things Journal

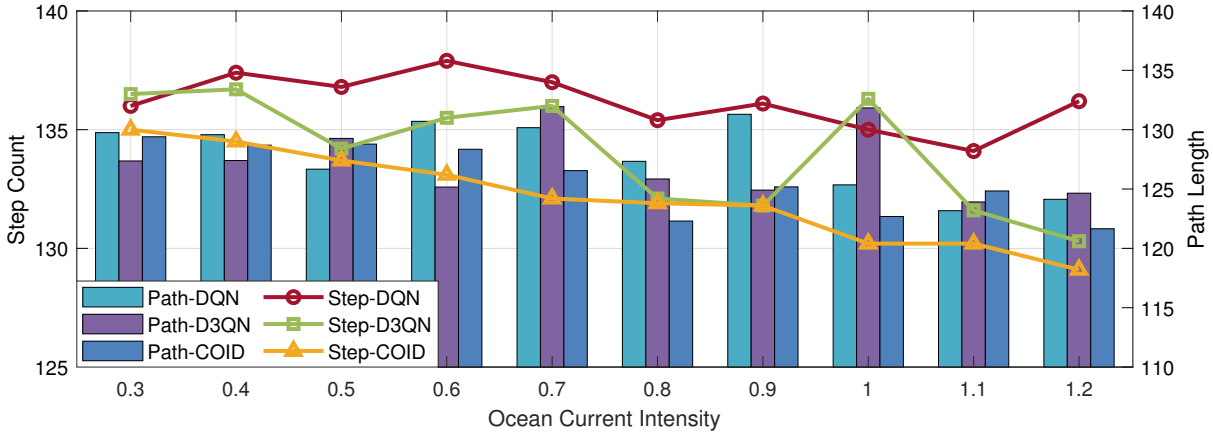


Fig. 7: The influence of ocean current intensity. Each result is from the model trained under the corresponding intensity.

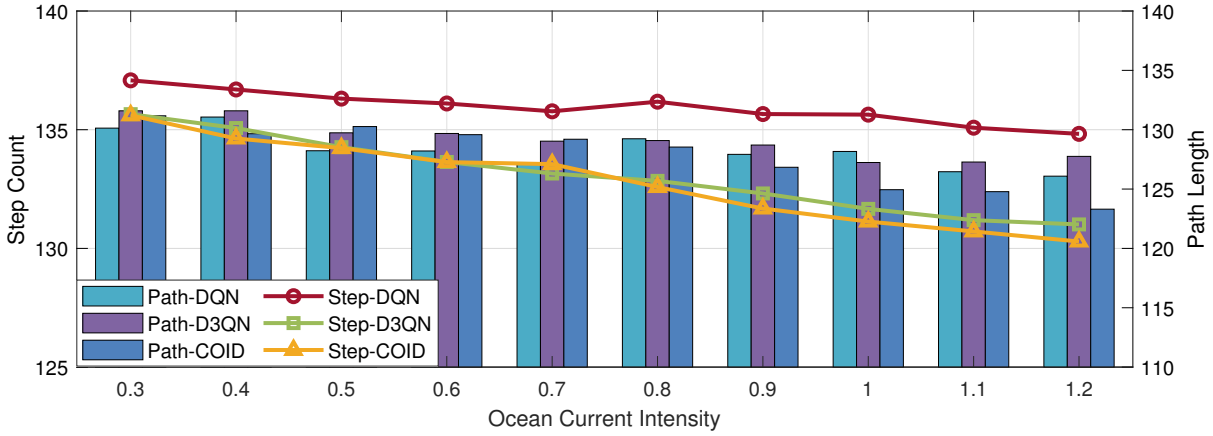


Fig. 8: The influence of ocean current intensity. All results are from the model trained under the ocean current intensity of 0.8.

eddies. Generally, most of these ocean current models are generated by mathematical functions, which lacks the support of real data and physical constraints. To make up for the shortcomings of the existing work, we establish a 3D grid ocean environment model based on real ocean current data, which is more representative and close to actual requirements.

B. Path Planning Algorithm

People are constantly breaking through the bottleneck and improving the path planning algorithm for better performance. For the traditional path planning algorithm, Wang *et al.* [38] introduced the neural network into RRT* algorithm, which solves the problem of sensitive to the initial solution. Xu *et al.* [39] improved the fast marching tree algorithm, using two hybrid search methods to find the optimal solution. Huang *et al.* [40] proposed a planning and tracking framework which used artificial potential field method to assign potential functions. For the intelligent bionic algorithms, Yu *et al.* [41] improved the ant colony optimization algorithm, which uses the A* search for complex environments. Wu *et al.* [42] combined genetic algorithm with other algorithm to improve the performance of global and local search. With the profound development of machine learning [43], [44], RL has attracted a lot of attention, provided solutions for many applications and achieved satisfactory results. In recent years, some scholars have begun to apply it in path planning. For example, Han *et al.* [34] introduced a Q-learning method for multi-AUV path planning. Wang *et al.* [45] proposed a hierarchical deep Q-network to plan the collision avoidance path and approach path. Semnani *et al.* [46] combined Deep Reinforcement Learning (DRL) algorithm with Force-based Motion Planning (FMP), using DRL for time-optimal paths and FMP for collision free paths. Liu *et al.* [47] also used Q-learning based method to adjust the local path of AUV. Generally, the RL based algorithms adopt the end-to-end training mode, which avoids the tedious modeling process and has greater flexibility.

VI. Conclusion and Discussion

In this paper, we propose COID, a comprehensive ocean information enabled AUV path planning scheme. It first introduces real data into the ocean environment model, which makes up for the lack of authenticity in existing work. Subsequently, we elaborately establish an RL environment for AUV path planning, which accurately summarizes the environment characteristics and assists in the algorithm acceleration. Moreover, COID integrates the the local ocean

current with position features and employs the framework of D3QN to enhance the practical value. The experimental results fully prove the superiority of our method. For one thing, the utilization of ocean current information endows COID with intelligence, so that it can find the optimal path and adapt to a variety of environments. For another, the ingenious network structure significantly improves the stability. COID solves several urgent problems in the field of AUV path planning. There still remains some challenges, such as the dynamic and time-varying environment, precise energy consumption model in large-scale environment. We will make unremitting research in the future.

REFERENCES

[1] H. Song, D. B. Rawat, S. Jeschke, and C. Brecher, *Cyber-physical systems: foundations, principles and applications*. Morgan Kaufmann, 2016.

[2] J. Yang, J. Wen, Y. Wang, B. Jiang, H. Wang, and H. Song, "Fog-based marine environmental information monitoring toward ocean of things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4238–4247, 2020.

[3] Y. Sun, H. Song, A. J. Jara, and R. Bie, "Internet of things and big data analytics for smart and connected communities," *IEEE Access*, vol. 4, pp. 766–773, 2016.

[4] Y. Liu, J. Wang, J. Li, S. Niu, and H. Song, "Machine learning for the detection and identification of internet of things (IOT) devices: A survey," *arXiv preprint arXiv:2101.10181*, 2021.

[5] X. Zhuo, M. Liu, Y. Wei, G. Yu, F. Qu, and R. Sun, "AUV-aided energy-efficient data collection in underwater acoustic sensor networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 010–10 022, 2020.

[6] Z. Fang, J. Wang, C. Jiang, Q. Zhang, and Y. Ren, "AoI inspired collaborative information collection for AUV assisted internet of underwater things," *IEEE Internet of Things Journal*, 2021.

[7] H. Yu, K. Meier, M. Argyle, and R. W. Beard, "Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 541–552, 2015.

[8] B. Oommen, S. Iyengar, N. Rao, and R. Kashyap, "Robot navigation in unknown terrains using learned visibility graphs. part I: The disjoint convex obstacle case," *IEEE Journal on Robotics and Automation*, vol. 3, no. 6, pp. 672–681, 1987.

[9] O. Takahashi and R. Schilling, "Motion planning in a plane using generalized voronoi diagrams," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 143–150, 1989.

[10] L. Wenzheng, L. Junjun, and Y. Shunli, "An improved dijkstra's algorithm for shortest path planning on 2D grid maps," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2019, pp. 438–441.

[11] S. Sedighi, D.-V. Nguyen, and K.-D. Kuhnert, "Guided hybrid A-star path planning algorithm for valet parking applications," in *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*, 2019, pp. 570–575.

[12] C.-b. Moon and W. Chung, "Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 1080–1090, 2015.

[13] L. Chen, Y. Shan, W. Tian, B. Li, and D. Cao, "A fast and efficient double-tree RRT*-like sampling-based planner applying on mobile robotic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2568–2578, 2018.

[14] G. M. Nayeem, M. Fan, M. Fan, and Y. Akhter, "A time-varying adaptive inertia weight based modified PSO algorithm for UAV path planning," in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2021, pp. 573–576.

[15] H. Yang, J. Qi, Y. Miao, H. Sun, and J. Li, "A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8557–8566, 2019.

[16] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.

[17] X. Cao and J. Peng, "A potential field bio-inspired neural network control algorithm for AUV path planning," in *2018 IEEE International Conference on Information and Automation (ICIA)*, 2018, pp. 1427–1432.

[18] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, "Machine learning for 6G wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 122–134, 2020.

[19] Z. Wan, C. Jiang, M. Fahad, Z. Ni, Y. Guo, and H. He, "Robot-assisted pedestrian regulation based on deep reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1669–1682, 2020.

[20] Y. Xiao, L. Xiao, X. Lu, H. Zhang, S. Yu, and H. V. Poor, "Deep-reinforcement-learning-based user profile perturbation for privacy-aware recommendation," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4560–4568, 2021.

[21] Y. Zhang, B. Gao, L. Guo, H. Guo, and H. Chen, "Adaptive decision-making for automated vehicles under roundabout scenarios using optimization embedded reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2020.

[22] H. Shi, L. Shi, M. Xu, and K.-S. Hwang, "End-to-end navigation strategy with deep reinforcement learning for mobile robots," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2393–2402, 2020.

[23] C. J. C. H. Watkins, "Learning from delayed rewards," *Ph.d.thesis Kings College University of Cambridge*, 1989.

[24] M. Volodymyr, K. Koray, S. David, A. A. Rusu, V. Joel, M. G. Bellemare, G. Alex, R. Martin, A. K. Fidjeland, and O. Georg, "Human-level control through deep reinforcement learning," *Nature*, p. 529–533, 2015.

[25] M. Chen and D. Zhu, "Optimal time-consuming path planning for autonomous underwater vehicles based on a dynamic neural network model in ocean current environments," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 401–14 412, 2020.

[26] *National Marine Data Center*, http://mds.nmdis.org.cn/pages/home.html.

[27] *IRI/LDEO Climate Data Library*, http://iridl.ldeo.columbia.edu/.

[28] *European Centre for Medium-Range Weather Forecasts*, https://www.ecmwf.int/.

[29] B. C. Shah and S. K. Gupta, "Long-distance path planning for unmanned surface vehicles in complex marine environment," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 3, pp. 813–830, 2020.

[30] J. Wang, Z. Wu, S. Yan, M. Tan, and J. Yu, "Real-time path planning and following of a gliding robotic dolphin within a hierarchical framework," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2021.

[31] G. Han, Z. Zhou, T. Zhang, H. Wang, and M. Guizani, "Ant-colony-based complete-coverage path-planning algorithm for underwater gliders in ocean areas with thermoclines," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2020.

[32] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, 2020.

[33] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, "Stochastic optimization aided energy-efficient information collection in internet of underwater things networks," *IEEE Internet of Things Journal*, p. 1, 2021.

[34] G. Han, A. Gong, H. Wang, M. Martínez-García, and Y. Peng, "Multi-AUV collaborative data collection algorithm based on Q-learning in underwater acoustic sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9294–9305, 2021.

[35] M. Chen and D. Zhu, "A workload balanced algorithm for task assignment and path planning of inhomogeneous autonomous underwater vehicle system," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 4, pp. 483–493, 2019.

[36] Y.-N. Ma, Y.-J. Gong, C.-F. Xiao, Y. Gao, and J. Zhang, "Path planning for autonomous underwater vehicles: An ant colony algorithm incorporating alarm pheromone," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 141–154, 2019.

[37] P. Yao, Z. Zhao, and Q. Zhu, "Path planning for autonomous underwater vehicles with simultaneous arrival in ocean environment," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3185–3193, 2020.

[38] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural RRT*: Learning-based optimal path planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.

[39] J. Xu, K. Song, D. Zhang, H. Dong, Y. Yan, and Q. Meng, "Informed anytime fast marching tree for asymptotically optimal motion planning,"

IEEE Transactions on Industrial Electronics, vol. 68, no. 6, pp. 5068–5077, 2021.

[40] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, and C. Hu, "A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1376–1386, 2020.

[41] X. Yu, W.-N. Chen, T. Gu, H. Yuan, H. Zhang, and J. Zhang, "ACO-A*: Ant colony optimization plus A* for 3-D traveling in environments with dense obstacles," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 617–631, 2019.

[42] Y. Wu, S. Wu, and X. Hu, "Cooperative path planning of UAVs amp; UGVs for a persistent surveillance task in urban environments," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4906–4919, 2021.

[43] S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, "Erratum to: industrial internet of things," in *Industrial Internet of Things*. Springer, 2017, pp. E1–E1.

[44] G. Dartmann, H. Song, and A. Schmeink, *Big data analytics for cyber-physical systems: machine learning for the internet of things*. Elsevier, 2019.

[45] J. Wang, Z. Wu, S. Yan, M. Tan, and J. Yu, "Real-time path planning and following of a gliding robotic dolphin within a hierarchical framework," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2021.

[46] S. H. Semnani, H. Liu, M. Everett, A. de Ruiter, and J. P. How, "Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3221–3226, 2020.

[47] B. Liu and Z. Lu, "Auv path planning under ocean current based on reinforcement learning in electronic chart," in *2013 International Conference on Computational and Information Sciences*, 2013, pp. 1939–1942.

Meng Xi received the B.E. degree in the Internet of Things Engineering from Tianjin University, Tianjin, China, in 2018, where she is currently pursuing the Ph.D. degree in Information and Communication Engineering. Her research interests include reinforcement learning and marine information processing.

Jiachen Yang received the B.S., M.S., and Ph.D. degrees in Communication and Information Engineering from Tianjin University, Tianjin, China, in 2002, 2005, and 2009, respectively, where he is currently a Professor with the School of Electronical and Information Engineering. From 2014 to 2015, he was a Visiting Scholar with the Department of Computer Science, School of Science, Loughborough University, U.K. In 2019, he was a Visiting Scholar with Embry-Riddle Aeronautical University. He is the Leader of the Laboratory of Stereo Visual Information Processing, Tianjin University. His research interests include image processing, artificial intelligence and information security. In these areas, he has published more than 100 technical articles in refereed journals and proceedings, including IEEE Transactions on Neural Networks and Learning System, IEEE Transactions on Cybernetics, IEEE Transactions on Industrial Informatics, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, and IEEE Transactions on Broadcasting. He is also on the editorial boards of IEEE Access, Sensors, and Multimedia Tools and Applications and held special issue on IEEE Transactions on Industrial Informatics, IEEE Access, and Sensors as a Lead Guest Editor.

Jiabao Wen is received the Ph.D. degree from the School of Electrical and Information Engineering, Tianjin University, Tianjin, China, in 2021. He is currently working in the School of Electrical and Information Engineering, Tianjin University. His major research interests include AUV path planning, ocean data processing, edge computing, and Internet of Things.

Hankai Liu received the B.E. degree in Internet of Things Engineering and the M.S. degree in Electronic and Communication Engineering from Tianjin University, Tianjin, China, in 2018 and 2021, respectively, where he is currently pursuing the Ph.D. degree in Computer Science and Technology. His current research interests include wireless sensing and pervasive computing.

Yang Li received his M.S. degree in Electrical Engineering from Dalian University of Technology in 2016, and then worked as a Lecturer at Shihezi University. From 2019, he began to pursue his Ph.D. degree at Tianjin University, China. His current research interests include image processing, deep learning, few-shot learning, quality assessment and Internet of Things.

Houbing Herbert Song (M'12 - SM'14) received the Ph.D. degree in Electrical Engineering from the University of Virginia, Charlottesville, VA, in August 2012, and the M.S. degree in Civil Engineering from the University of Texas, El Paso, TX, in December 2006.

In August 2017, he joined the Department of Electrical Engineering and Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL, where he is currently an Assistant Professor and the Director of the Security and Optimization for Networked Globe Laboratory (SONG Lab, www.SONGLab.us). He has served as an Associate Technical Editor for IEEE Communications Magazine, an Associate Editor for IEEE Internet of Things Journal, IEEE Transactions on Intelligent Transportation Systems, and IEEE Journal on Miniaturization for Air and Space Systems. He is a senior member of ACM and an ACM Distinguished Speaker. He is a recipient of 5 Best Paper Awards (CPSCom-2019, ICII 2019, ICNS 2019, CBDCom 2020, and WASA 2020). He is the editor of 8 books, the author of more than 100 articles, and the inventor of 2 patents (US & WO). His research interests include cyber-physical systems, Internet of Things, cybersecurity and privacy, edge computing, artificial intelligence, machine learning, big data analytics, unmanned aircraft systems, connected vehicle, smart and connected health, and wireless communications and networking. His research has been sponsored by federal agencies (including US Department of Transportation, National Science Foundation, Federal Aviation Administration, US Department of Defense, and Air Force Research Laboratory) and industry. His research has been featured by popular news media outlets, including IEEE GlobalSpec's Engineering360, USA Today, U.S. News & World Report, Fox News, Association for Unmanned Vehicle Systems International (AUVSI), Forbes, WFTV, and New Atlas.