

## Research Article

# Learning-Based Path Planning Algorithm in Ocean Currents for Multi-Glider

Wei Lan <sup>1</sup>, Xiang Jin <sup>1</sup>, Xin Chang <sup>1</sup>, Tianlin Wang <sup>2</sup>, and Han Zhou <sup>3</sup>

<sup>1</sup>Naval Architecture and Ocean Engineering College, Dalian Maritime University, Dalian 116026, Liaoning, China

<sup>2</sup>School of Marine Sciences, Sun Yat-sen University, Zhuhai 519082, Guangdong, China

<sup>3</sup>The 714 Institute of CSSC, Beijing 100101, China

Correspondence should be addressed to Xin Chang; [xin.chang@dlnu.edu.cn](mailto:xin.chang@dlnu.edu.cn)

Received 14 October 2021; Revised 20 January 2022; Accepted 7 February 2022; Published 11 March 2022

Academic Editor: Maia Angelova

Copyright © 2022 Wei Lan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In practical application studies of glider formations, ocean currents are a major influencing factor in their path planning. The purpose of this paper is to solve the path planning problem of glider formations in time-varying ocean currents and establish gliders, glider formation, and ocean current models based on existing data. The Doc-CNN architecture is tailored to conform to the operation and environment characteristics of gliders in practical application. After experiments, the algorithm of the improved architecture can be used in the path planning task of glider formation. The algorithm architecture is compared and tested on two datasets, grid maps and ocean maps. Doc-CNN has an advantage over architecture without being tailored to glider characteristics and makes full use of known global information and local information collected by gliders themselves. The results show that the path planning problem of glider formation in ocean currents can be solved by using Doc-CNN.

## 1. Introduction

After years of development, the technology of underwater glider (UG) is gradually becoming mature [1], and many types of equipment have achieved industrialization and large-scale production [2]. It has been widely used in many marine applications [3], including marine environment monitoring [4], a marine scientific investigation [5], marine resource exploration, and performing missions in deep or dangerous waters that are difficult to reach by human beings [6]. The application of gliders is an important part of the ocean observation network and integrated ocean observation system [7].

UG formation applications have broad application prospects in a large-scale marine survey [8]. Many studies on the area coverage capability of underwater unmanned formation focus on the deployment of fixed nodes in the network [9], which usually adopts the way of discharge to increase the coverage by the ocean current. The marine environment in which these vehicles live is a highly dynamic system with high spatiotemporal variability [10]. Most

gliders also lack power and usually travel at a relatively slow speed, in most cases at or below the speed of the current. Whether in some coastal or deep-sea applications, the influence of ocean currents is considerable and is even the main cause of the influence on the glider path.

For glider applications, path planning is a necessary capability. Path planning is one of the key technologies in the field of underwater vehicles. It refers to finding a collision-free path from the initial state to the target state according to certain evaluation criteria in the water environment containing obstacles. To a certain extent, path planning technology marks the level of intelligence of underwater vehicles [11]. Compared with other types of vehicles, the underwater mission environment is a large-scale three-dimensional (3D) space, and the complexity of the underwater environment increases the difficulty of planning. Meanwhile, the position, energy consumption, posture, and motion constraints should also be considered [12].

For traditional underwater path planning, the shortest path can be found, but it lacks flexibility [13]. There are also a membrane evolutionary artificial potential field

(memEAPF), combined with membrane computing, genetic algorithm, and artificial potential field (APF) [14], and a hybrid path planning algorithm based on membrane pseudobacterial potential field (MemPBPF) [15]. These combinations have better performance than the original APF in terms of path length. Intelligent path planning methods mainly apply modern artificial intelligence technology to the path planning of underwater vehicles, which roughly includes swarm intelligence methods and machine learning methods. Path planning methods are based on swarm intelligence paths, such as genetic algorithm [16], ant colony algorithm [17], and biogeography optimization algorithm [18]. Among them, the energy consumption of Autonomous Underwater Vehicle (AUV) in a large range of environments is considered through planning in the underwater environment, and the energy consumption term of AUV dynamic estimation is added to the genetic algorithm cost function [19]. Ant colony evolution was improved to solve the problems of low search efficiency, long search time, and insufficient ocean current information in global path planning [20]. The underwater vehicles planning system is designed based on the BBO algorithm, and simulation experiments are carried out [21]. The results show that BBO-based planning algorithms have significant potential in real time.

With regard to the late research studies, generally available ocean current prediction sources provide a series of data at discrete time points, but such dynamic prediction is difficult to be applied to motion planning. Problems involving time-varying costs are notoriously difficult to solve. The time-dependent shortest path (TDSP) problem is the simplest general form, and it is known to be NP-hard [22]. Many variables will need to be defined if gliders are to perform dynamic path planning in ocean currents [23]. Dynamic path planning sometimes makes it better to defer action for a while (waiting in place or floating downstream). This is non-FIFO (first-in-first-out) attribute, and efficient algorithms for non-FIFO TDSPs have been gradually used in the past two years [24]. Dynamic planning, for example, in the study of planning in time-dependent flow fields [25], demands accurate or regular predictions of the ocean current environment due to the negative influence of dynamic flow in the marine environment on navigation performance in the ocean. However, the study [25] assumes that ocean currents of various depths have the same velocity, which clearly does not correspond to reality. Such dynamic programming is not really useful when the overall system's prediction capacity is restricted or when the oceans are unfamiliar.

At present, underwater path planning is more about transforming the path planning problem into a search problem or energy optimization problem; the real-time requirement is not high. Machine learning, as a hotspot of actual planning problems, usually describes path planning as Markov Decision Process (MDP) problem, which provides data for training and learning. It can be divided into a neural network (NN), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL). For the neural network, in the path planning of underwater gliders, sensor data is

usually taken as network input and behavior action as network output, and the network model is obtained through training [26]. After the problem is upgraded to 3D space, the AUV path planning problem is studied considering different flow rates [27]. Reinforcement learning is a kind of unsupervised machine learning, and its learning can be regarded as a trial evaluation process [28]. Considering the impact of ocean current on the energy consumption of underwater gliders, the design considers the cost function of ocean current to optimize the path planning in the 3D environment [29]. Compared with RL, which focuses on solving learning problems, deep learning networks can extract abstract features from large-scale data to cope with increasingly complex task environments. DRL can also be used to plan the obstacle avoidance of underwater vehicles [30].

It is of great significance to study path planning algorithms that can get the corresponding navigation strategies directly from vehicles without manual intervention and the navigation strategies can be obtained with as few operation resources as possible. Therefore, it is of great significance to study machine learning algorithms that can be used to solve real-time dynamic programming problems without environment dependence when the planning time is reduced after the algorithm is formed.

Convolutional Neural Networks (CNNs) and Deep Reinforcement Learning (DQN) can be adopted for path planning in a dynamic environment [31]. Inspired by the advanced performance of Deep Convolutional Neural Networks (DCNNs) in visual feature representation and learning, the image-based path planning learns planning directly from the original image. A novel DCNN architecture Value Iteration Network (VIN) can realize path planning based on initial images [32]. The superior feature representation and learning capability of DCNN make it possible to fit large-scale datasets in more convolution layers and larger parameter space compared with CNN [33]. It can be used in large-scale datasets and multiparameter environments, such as ocean environments. DCNN also has the disadvantage of requiring a lot of computing resources. By studying the direction of DCNN and comparing VIN, the Deep Convolutional Neural Network (DB-CNN) algorithm, which has a dual-branch structure and can achieve higher efficiency and precision of path planning, is proposed [34]. The path planning based on the initial state is realized, which is better than the traditional global path planning method based on environment mapping.

Path planning for gliders frequently requires them to be feasible and in line with their own abilities to move the path rapidly. After a large number of improved algorithms and some relatively complex algorithms, although we can get a better path in simulations, there is sometimes inadequate time or computation capacity to utilize complicated algorithms in applications, such as the limited of independent action in gliders, or the actual case where the sensors carried by each glider in the formation are not exactly the same. The advanced and practical characteristics of algorithms need to be balanced.

In order to accommodate the conditions of glider formation use, such as movement constraints, and the

limitation that the glider does not necessarily know the global current conditions at each point in time during operation, the algorithm should first characterize and learn the initial state of the glider formation and the depth characteristics of time-varying ocean current data. Then, according to these depth characteristics, we determine the optimal path for the current situation. Because these features are similar to the data characteristics of the dual-branch CNN network, the dual-branch algorithm utilized by the glider formation in ocean current is improved in this study. The planning of the glider formation is realized from the initial global data and the subsequent local data collected. A Deep Convolutional Neural Network (Doc-CNN) for the ocean current environment is designed and adapted to the path planning of existing and future gliders in ocean currents, which can achieve higher efficiency and accuracy of glider formation path planning in the ocean current environment.

## 2. Preliminaries

This section introduces the preparatory knowledge in this paper. It includes the Markov Decision Process (MDP) and knowledge of neural networks.

**2.1. Markov Decision Process.** It is assumed that the gliders in ocean current have MDP. A standard MDP for sequential decision making is composed of state space  $\mathcal{S}$ , action space  $\mathcal{A}$ , transfer probability distribution  $P$ , and reward function  $R$ .  $\mathcal{S}$  represents the set of descriptions of the possible states of an agent in the environment,  $\mathcal{A}$  represents the set of descriptions of possible actions an agent may take in the environment.  $R: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ , the received reward from the environment for taking action in a certain state.  $P: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ , an agent in a certain state will choose its actions according to  $P$  and then move from one state to another. The quad  $\langle \mathcal{S}, \mathcal{A}, P, R \rangle$  is to find the best strategy  $P$  to perform a series of actions in the environment so that the agent can complete the given task with the best turn return. The discounted factor is  $\gamma$ , in which the policy is represented by  $\pi: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ .  $\gamma \in [0, 1]$  represents the discounted factor of reward, and the optimal strategy is defined as

$$\pi^* = \arg \max_{\pi} E_{s_0} \left[ \sum_{i=0}^{+\infty} \gamma^i r_i | \pi, P \right]. \quad (1)$$

To measure the expected accumulative rewards  $\mathcal{S}$  of  $s_t$  and  $(s_t, a_t)$ , the state value function and the action value function are defined as

$$V(s_t) = E \left[ \sum_{i=t}^{+\infty} \gamma^{i-t} r(s_i, a_i) | \pi, P, s_t \right], \quad (2)$$

$$\begin{aligned} Q(s_t, a_t) &= E \left[ \sum_{i=t}^{+\infty} \gamma^{i-t} r(s_i, a_i) | \pi, P, s_t, a_t \right] \\ &= r_t + E[V(s_{t+1}) | \pi, P, s_{t+1}]. \end{aligned} \quad (3)$$

Equation (2) is substituted into (1); the optimal strategy is derived as

$$\begin{aligned} \pi^* &= \arg \max_{\pi} E_{s_0} [V(s_0)] \\ &= \arg \max_{\pi} E_{s_0, a_0} [Q(s_0, a_0)]. \end{aligned} \quad (4)$$

However, since both state value function and action value function are unknown, it is impossible to determine  $\pi^*$  through (4) directly. Therefore, the value functions of MDP have to be estimated precisely so that the optimal policy can be found.

As both state value function and action value function are unknown,  $\pi^*$  is impossible to determine through (4). Therefore, the value function of MDP has to be accurately estimated in order to find the optimal policy.

**2.2. Value Iteration Network.** Value iteration is a typical method for value function estimation for solving the MDP problem [32].  $V_k(s)$  is the estimated state value function at step  $k$ ;  $Q_k(s, a)$  is the estimated action value function for each state at step  $k$ .  $\pi_k$  is utilized to represent the deterministic policy at step  $k$ . When  $i = 0, 1, \dots$ , the value iteration process can be expressed as

$$\begin{aligned} \pi_k(s_i) &= \arg \max_{a_i} Q_k(s_i, a_i), \\ V_{k+1}(s_i) &= Q_{k+1}(s_i, \pi(s_i)) \\ &= r_i + E_{s_{i+1}} [V_k(s_{i+1})]. \end{aligned} \quad (5)$$

However, since it is difficult to determine the explicit representation of  $\pi_k$ ,  $Q_k$ , and  $V_k$  when the dimension of  $s_t$  is high. Value Iteration Network (VIN) is designed to approximate this process successfully. A planning module is added to the generic policy representation.

VIN has some advantages. The reward function and the transfer function are parameterized and can be derived. The solution of auxiliary policy is introduced to make policy have more generalization ability. The attention mechanism is introduced in policy solving. The design of the Value Iteration Module is equivalent to a CNN network, and the network can be updated by Error Back Propagation (BP) algorithm.

Global information can be passed through VIN to various states in the final value function layer. This architecture performs well in learning to plan tasks. However, VIN also has defects. It takes a lot of time to train such a cyclic convolutional neural network, especially in the value of iteration time.

**2.3. DB-CNN for Value Function Estimation.** DB-CNN architecture is a new architecture for value function estimation, which is improved from DCNN architecture and consists of reprocessing layer and two branches: branch one for global feature representation and branch two for local feature representation. DB-CNN is a dual-branch DCNN architecture for global path planning from raw images.

Therefore, for DB-CNN, the policy of global path planning can be written as

$$a_t = \pi(s_t) = \arg \max_a \widehat{Q}(s_t, a), \quad (6)$$

where  $\widehat{Q}(s, a)$  is an estimate of  $Q(s, a)$ .

Given the expert dataset  $\{(s_i, y_i)\}_{i=1}^{i=N}$  of global path planning, the cross-entropy training loss of  $L_2$  norm is as follows:

$$L(\alpha) = -\frac{1}{N} \sum_{i=1}^N Y_i \log(F_\alpha(s_i)) + \lambda \|\alpha\|_2, \quad (7)$$

where  $\alpha$  is the parameter of DCNN,  $F_\alpha: s_t \rightarrow \text{softmax}[\widehat{Q}(s_t, a=0), \dots, \widehat{Q}(s_t, a=n)]^T$  is the DCNN designed for value function estimation, and  $n$  is the number of strategies.  $N$  is the number of training samples,  $Y_i$  is the one-hot vector form of  $y_i$ , and  $\lambda$  is the hyperparameter to adjust the influence of  $L_2$  norm on the loss function.

The reprocessing layer consists of two convolution layers, each of which is followed by a max-pooling layer. The reprocessing layer is to filter out the noise and compress the raw data into features. Then, the global path planning is changed from single-point data to  $l_1 \times l_2$  planning on the region, which improves efficiency. Branch 1 is composed of convolutional layers, residual convolutional layers, max-pooling layers, and fully connected layers, representing all the original data set global characteristics relative to the target. Branch 2 is composed of convolutional layers and residual convolutional layers, representing the local features related to the current location. Compared with VIN, the depth of DB-CNN is reduced, and the global path planning problem is transformed into designing and training a DCNN for value function estimation. Both global information and local information are effectively retained and represented.

### 3. Model Description

In this paper, the multiglider path planning in ocean current into MDP is defined as  $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R \rangle$

**3.1. Path Planning Model.** The state space of  $\mathcal{M}$  is denoted as  $\mathcal{S} = \{\mathcal{C}, \mathcal{G}, \mathcal{X}\}$ , consisting of  $\mathcal{C} = \{\mathcal{C}_{\text{int}}, \mathcal{C}_t\}$ ,  $\mathcal{G} = \{(g_t^1, g_t^2, g_t^3)\}$ , and  $\mathcal{X} = \{(x_t^1, x_t^2, x_t^3)\}$ .  $\mathcal{C}$  is the ocean current environment space of gliders, and it is composed of two parts.  $\mathcal{C}_{\text{int}} \in \mathbb{R}^3$  is all the ocean current information that can be obtained by gliders in the initial state at the initial time. And  $\mathcal{C}_t \in \mathbb{R}^3$  is the ocean current information that can be obtained by gliders at step  $t$  time.  $\mathcal{G}$  is the target area center position at step  $t$ .  $\mathcal{X}$  is the glider locations at step  $t$ . The action space of  $\mathcal{M}$  is denoted as  $\mathcal{A} = \{(a_t^1, a_t^2, a_t^3)\}$ , representing the continuous movement of gliders in the ocean current.

The state transition process in the MDP of this paper is deterministic. It is defined as  $P: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ . State  $s_{t-1}$  gets through action  $a_{t-1}$ ; the state will get into  $s_t$ . It should be noted that, for a given detection task, in each path planning step  $t$ , the initial  $\mathcal{C}_{\text{int}}$  of the glider does not change.  $\mathcal{C}_{\text{int}}$  belongs to all the ocean current information received by the glider formation in the initial situation, which can be

understood as fixed global current information at the initial time. The target position  $(g_t^1, g_t^2, g_t^3)$  of the glider in state  $s_t$  is unchanged, while the position  $(x_t^1, x_t^2, x_t^3)$  of the glider in state  $s_t$  and the ocean current  $\mathcal{C}_t$  around the glider will change at each step.

After action  $a_t$ , gliders in the step  $t+1$  reach the target; the positive reward  $r_t = \phi_1$  ( $\phi_1 > 0$ ). Otherwise, gliders will get a negative reward  $r_t = \phi_2$  ( $\phi_2 < 0$ ). The single-movement heading angle of the glider is within the steering angle  $\beta$  as shown in Figure 1, and the glider obtains a positive reward  $\tau_1$  ( $0 < \tau_1 \ll \phi_1$ ). If the glider forward angle is beyond the heading, we get the negative bonus  $\tau_2$  ( $|\tau_2| > \tau_1$ ). So, the optimal path from the start area to the target area has the maximal accumulative rewards.

According to the practical problems of gliders, in the initial state, each glider in formation will receive all the environmental data by the whole system in the current time. It can be roughly understood as global ocean current information at  $t=0$ . The ocean current is a time-varying environment; it is going to change over time. The expert data set  $\{(s_i, y_i)\}_{i=1}^{i=N}$  may not be the optimal expert path, and the cross-entropy training loss defined by  $L_2$  norm can still be used from equation (8).

**3.2. Model Underwater Glider.** In this paper, the gliders have characteristics of self-examination, communication capabilities on the water, and autonomous control. And the gliders also have the omnidirectional motor ability, there is no restriction on the direction of movement, and it can adjust and control its moving and turning in any direction and respond to change the direction of movement and turning in a short time. For these omnidirectional gliders, athletic ability and control equation are shown [35]. Figure 2 shows the glider motor ability on a small scale.

The gliders can motion in almost all directions. The motion control equation is as follows.

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix}. \quad (8)$$

For this equation,

$$\begin{aligned} \mathbf{M}_{11} &= \text{diag}[m + A_{11}, m + A_{22}, m + A_{33}], \\ \mathbf{M}_{12} &= \begin{bmatrix} 0 & mz_G & -my_G \\ -mz_G & 0 & mx_G \\ my_G & -mx_G & 0 \end{bmatrix}, \\ \mathbf{M}_{21} &= \begin{bmatrix} 0 & -mz_G & my_G \\ mz_G & 0 & -mx_G \\ -my_G & mx_G & 0 \end{bmatrix}, \\ \mathbf{M}_{22} &= [I_{xx} + A_{44}, I_{yy} + A_{55}, I_{zz} + A_{66}], \end{aligned} \quad (9)$$

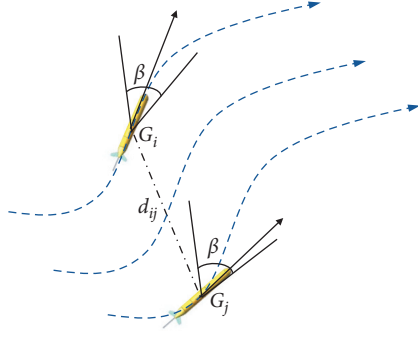


FIGURE 1: Schematic diagram of double glider formation moving in the ocean current.

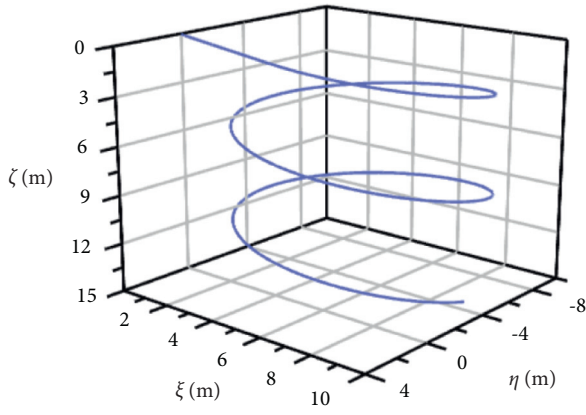


FIGURE 2: The glider spiral trajectory.

where  $m$  is the mass of the glider.  $x_G$ ,  $y_G$ , and  $z_G$  are the barycentric coordinates of the glider.  $F_x$ ,  $F_y$ , and  $F_z$  are the resultant forces of the glider.  $M_x$ ,  $M_y$ , and  $M_z$  are the resultant moment of force.  $A_{11}$ ,  $A_{22}$ ,  $A_{33}$ ,  $A_{44}$ ,  $A_{55}$ , and  $A_{66}$  are added mass.  $I_{xx}$ ,  $I_{yy}$ , and  $I_{zz}$  are the inertia moment around the axes.

In this paper, we consider the model of gliders as a type of omnidirectional gliders with well-steering performance. In this case, the motion of gliders is mainly determined by their own pitch angle  $\theta$  and heading angle  $\psi$ . The gliders dynamics can be modeled as a first-order system with motion pitch angle commands  $\theta_c$  and heading angle commands  $\psi_c$  as follows:

$$\dot{\theta} = -\frac{1}{\tau_\theta}\theta + \frac{1}{\tau_\theta}\theta_c, \quad (10)$$

$$\dot{\psi} = -\frac{1}{\tau_\psi}\psi + \frac{1}{\tau_\psi}\psi_c, \quad (11)$$

where  $\tau_\theta$  and  $\tau_\psi$  are time constants.

The energy consumption of gliders is related to its model, current speed and direction, voyage distance and speed, etc. When the shape and motion characteristics of gliders are determined, the steady-state glide velocity is only related to the net buoyancy and glide angle. During the steady-state motion of gliders, the net buoyancy and glide angle remain basically unchanged. In this paper, gliders are normal

cruising with standard cruising depth and constant depth movement. That means the rate of gliders is a constant value in still water, which can conform to gliders' own motion characteristics and improve the computing speed and simplify the model.

The motion of gliders is often simplified into sawtooth shape trajectories, and the whole voyage can be divided into several cycles. The trajectory of a single cycle is shown in Figure 3, where  $\theta$  is the angle of the pitch,  $\alpha$  is the angle of the attack,  $\xi$  is the angle of the glide, and  $\xi = \theta - \alpha$ .

Normally, the glider's energy consumption mainly comes from the control of the glider, i.e., forces to move and change the mass, and the load consumption, e.g., onboard processor and sensors [36].

$$E_T = E_t + [c](E_f + E_z), \quad (12)$$

where  $c$  is the number of glide cycles, and  $[\cdot]$  is round up to an integer. In this paper,  $E_f$  include the consumption of the buoyancy adjustment module in the diving stage and floating stage.  $E_z$  include the consumption of the adjusting attitude control module and module standby consumption.  $E_t$  mainly includes sensors standby consumption and communication module consumption.

UGs can complete the advancement to the target area under time-varying currents. Avoiding or reducing the overlap working areas with the adjacent gliders during the process, keeping a relatively stable formation, the formation system meets the following conditions:

$$\Delta q_{ijt}(t) = \lim_{t \rightarrow \infty} \|q_{ij}(t) - q_{ij}(t-1)\| \leq 0, \quad (13)$$

where  $\Delta q_{ijt}(t)$  is the position between the  $i$ th and  $j$ th glider at time  $t$  compressing the previous time  $t-1$ .  $q_{ijt}(t)$  is the position between  $i$ th and  $j$ th glider at time  $t$ . The above condition allows the formation system to move toward the demanded area.

**3.3. Ocean Model.** UGs usually work in a complex dynamic ocean environment, and gliders have speeds in water ranging between 0.2 m/s and 0.4 m/s. The influence of ocean current on gliders' motion parameters and trajectory is very huge. For the work motion planning of gliders, the role of currents will inevitably affect the trajectory of gliders [37]. For the path planning techniques in deterministic conditions, ocean currents are ignored because they often have poor performance in practice due to errors in navigation and operative mode in the underwater glider dead reckons [38]. And the ocean current cannot be ignored under the criteria of time or energy consumption in real underwater cases [39].

Ocean currents tend to be different at different depths. In order to reach the destination in optimal time, underwater vehicles need to utilize favorable currents and also avoid adverse currents by diving or rising to appropriate depths [40]. It is important to carry out underwater path planning in the real 3D ocean. For such conditions, predicting ocean currents in a real ocean environment is necessary, but numerically challenging. Oceans are complex dynamic systems, with multiple time scales from seconds to years and

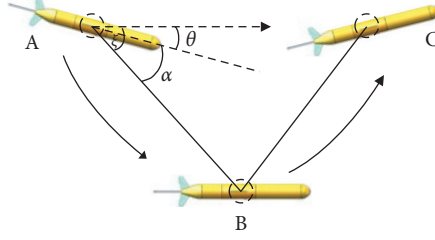


FIGURE 3: Single glide cycle abridged general view.

length scales from millimeters to hundreds of kilometers [10]. Therefore, robust and accurate numerical formats and data assimilation models are needed [41].

The time-varying ocean dataset was established by desensitization data. The image of the data set is shown in Figure 4.

For the 3D environment, the model is constructed by the layered modeling method, and the 3D model can be simplified into several 2D models layered and superimposed. Similar ocean currents are used for the models with different depths.

In this paper, the real ocean currents are assumed to be the same as the flow fields given above at least the characteristics. In order to reduce the calculation amount and difficulty, this paper adopts a two-dimensional ocean current. The glider formation simulates 3D motion in 2D ocean current superposition and presents it in a 2D image. Similar ocean currents are used for the models with different depths.

**3.4. Network Model.** In this subsection, we propose a novel algorithm of multiglider formation path planning in ocean currents, a deep convolutional neural network for the ocean current environment (Doc-CNN). The architecture consists of two branches: branch 1 is used to extract global features, and branch 2 is used to extract local features detected around gliders at different positions, and the local features are used to modify the global path. Previous studies [32] usually designed single-branch DCNNs to extract features. However, since the convolution layers are locally connected, a deeper architecture is required to extract global features of the input image, which increases the computational cost and reduces the convergence speed [34]. Doc-CNN as an algorithm specifically designed to solve the glider formation planning problem takes into account the limitation of gliders' movement ability. During operation, gliders do not necessarily know the global current conditions at each point in time. The double branch is used to extract global and local features, and local features modify the global path results. This architecture is more suitable for glider planning. Doc-CNN consists of a data processing layer, branch 1, and branch 2, as described below.

**3.4.1. Data Processing.** The processing layer consists of a convolution layer and a max-pooling layer to filter the original data noise. As shown in Figure 5, the input data are divided into the initial current dataset  $\mathcal{E}_{\text{int}}$  and ocean at the current point in time  $\mathcal{E}_t$ .  $\mathcal{E}_{\text{int}}$  is the original global data,

representing all current data available to glider formation at the beginning.  $\mathcal{E}_t$ ,  $t$  is a time function starting from 0. Since  $t=0$ ,  $\mathcal{E}_t$  varies with the position of gliders.  $\mathcal{E}_t$  represents the ocean current data collected and detected by individuals in glider formation at different times and positions.  $\mathcal{E}_{\text{int}}$  and  $\mathcal{E}_t$  combined input, in line with the actual use of glider formation.

Compress the original dataset  $\mathcal{E}_{\text{int}}$  into a feature dataset  $\mathcal{E}'_{\text{int}}$ , and the ocean current  $\mathcal{E}_t$  is compressed into a feature dataset  $\mathcal{E}'_t$ . In  $\mathcal{E}'_t$ , during the operation of glider formation, the time-varying current in a global environment cannot be understood. Instead, the current data within the perceived range of each glider can only be collected to form the time-varying current data related to the current time position. After this, the global planning for the dynamic ocean current environment becomes for the feature dataset  $\mathcal{E}'_{\text{int}}$  and the set  $\mathcal{E}'_t$ , rather than the scattered state of the original dataset. This improves the efficiency and conforms to the objective environment of gliders' operation; that is, gliders' state and path depend on global conditions and their own current environments.

**3.4.2. Branch One and Branch Two.** Branch one consists of convolutional layers, residual convolutional layers, and fully connected layers. Notably, the residual convolutional layer not only increases the training accuracy of convolutional neural networks with deep feature representations but also makes them generalize well to testing data.

Doc-CNN needs to represent the depth characteristics of the ocean current environment and can achieve high accuracy in an unknown environment (test dataset), so the residual convolution layer is embedded in Doc-CNN. We represent the depth features extracted by this branch from the original dataset  $\mathcal{E}_{\text{int}}$  as  $f_1 \in \mathbb{R}^D$ , where  $D$  is the dimension of the feature vector.  $f_1$  can be regarded as the global guidance of glider formation in the original dataset, representing the global features of all data points in the dataset  $\mathcal{E}_{\text{int}}$  related to the target position  $(g_t^1, g_t^2, g_t^3)$ .

Branch two consists of convolution layers and residual convolution layers. This branch feature extracts and maps it to  $f_2 \in \mathbb{R}^D$ . The self-convolution neural layer is locally connected, rather than fully connected at this branch.  $f_2$  extracts the local characteristics of ocean current  $\mathcal{E}_t$  at the current time point and position and the processed original dataset  $\mathcal{E}'_{\text{int}}$ . Branch 1 obtained the global feature Fc-2 and estimated the value function of local, as the gliders' local guidance with current state correction.

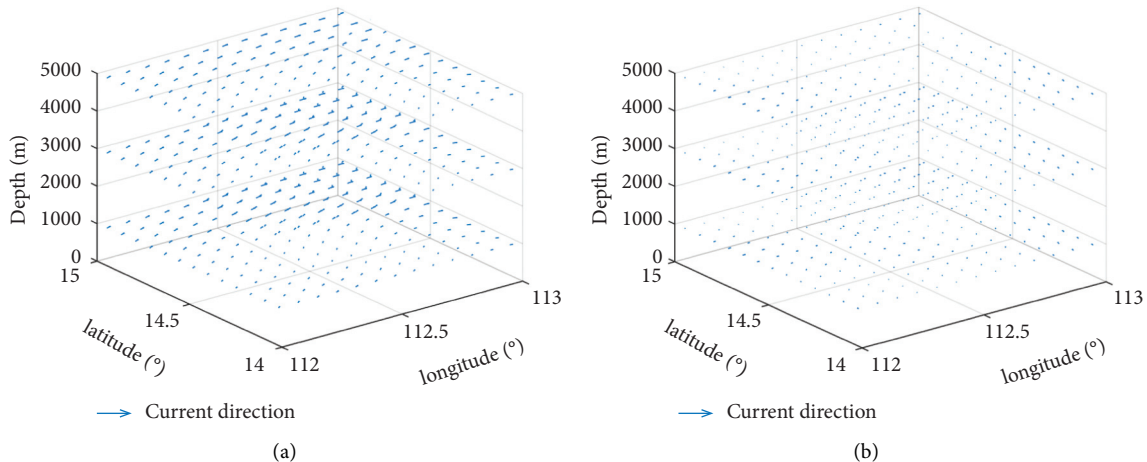


FIGURE 4: Time-varying fields compendious distribution in  $112^\circ E - 113^\circ E$  and  $14^\circ N - 15^\circ N$ .

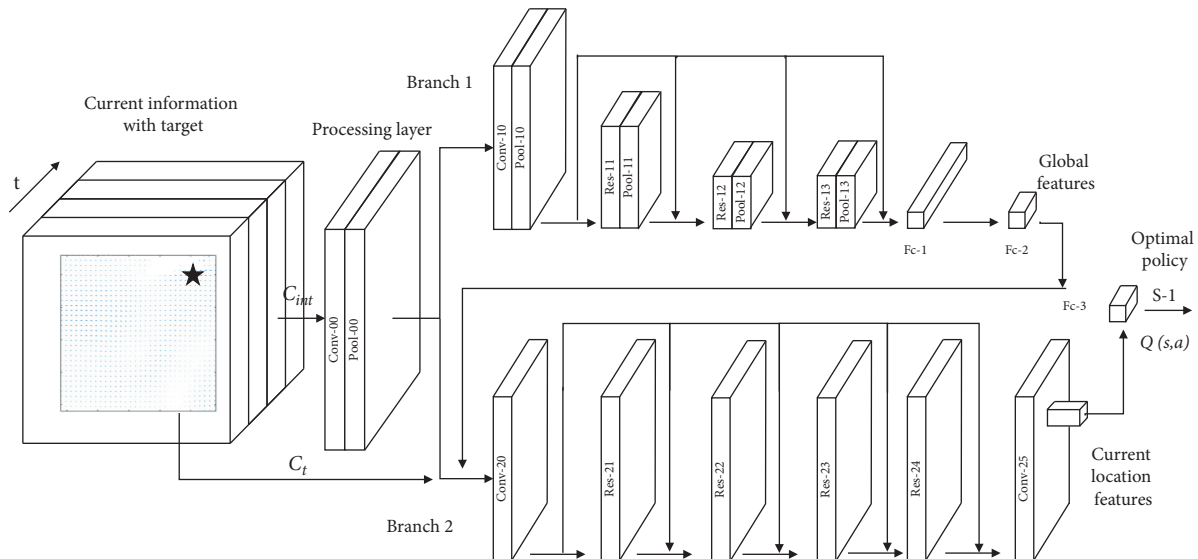


FIGURE 5: Doc-CNN for path planning.

The Doc-CNN is shown in Figure 5, where Conv, Pool, Res, Fc, and S are the abbreviations for convolutional layer, max-pooling layer, residual convolution layer, full-connected layer, and softmax layer, respectively. Compared with VIN and DB-CNN, Doc-CNN is more suitable for gliders application environment, namely, the ocean current environment, and the global information and local current information of the data set are effectively retained and represented. Doc-CNN parameters are shown in Table 1.

Doc-CNN parameters in Table 1 are designed for glider path planning data set of  $128 \times 128$  ocean current environment. Data processing layer parameters and other pooled operation kernel sizes are consulting to existing value functions [32]. We select the kernel size of all residual convolution layers as 3. After comprehensive consideration of training accuracy and calculation cost, we choose kernel size 20. In order to filter the noise of the input data set, we choose the kernel size of the first convolution layer in each

branch to be 5. The number of nodes in the fully connected layers is fine-tuned by experiments.

Doc-CNN can also be trained on other data sets of different sizes. In order to find the optimal kernel size, kernel number, and layer number of Doc-CNN in other datasets, the training accuracy remains stable.

**3.5. Learning-Based Path Planning Algorithm.** The path planning algorithm based on learning based on Doc-CNN is described as a whole, and its working principle is as follows.

In the training phase, the expert data set for global path planning is available, so the training phase is offline. For each training step, batch data is randomly selected (line 3), and the loss  $L_\alpha$  is calculated according to equation (8). At this time, the motion of gliders should conform to its own motion characteristics in equation (9)–(11). We calculate the random gradient  $\nabla L(\alpha)$  and update the learning rate  $\delta$  by

TABLE 1: Parameters of Doc-CNN (example of  $128 \times 128$ ).

Layer	Detail	Parameters
Processing layer	Conv-00	$12 \times 5 \times 5$ kernels with stride 1
	Pool-00	$3 \times 3$ kernels with stride 1
Branch 1	Conv-10	$20 \times 5 \times 5$ kernels with stride 1
	Pool-10	$3 \times 3$ kernels with stride 2
	Res-11	$20 \times 3 \times 3$ kernels with stride 1
	Pool-11	$3 \times 3$ kernels with stride 2
	Res-12	$20 \times 3 \times 3$ kernels with stride 1
	Pool-12	$3 \times 3$ kernels with stride 2
	Res-13	$20 \times 3 \times 3$ kernels with stride 1
	Pool-13	$3 \times 3$ kernels with stride 2
	Fc-1	192 nodes
	Fc-2	12 nodes
Branch 2	Conv-20	$20 \times 5 \times 5$ kernels with stride 1
	Res-21	$20 \times 3 \times 3$ kernels with stride 1
	Res-22	$20 \times 3 \times 3$ kernels with stride 1
	Res-23	$20 \times 3 \times 3$ kernels with stride 1
	Res-24	$20 \times 3 \times 3$ kernels with stride 1
	Conv-25	$12 \times 3 \times 3$ kernels with stride 1
Output layer	Fc-3	4 nodes
	S-1	4 nodes

- (i) Training phase (offline)
  - (1) **for** epoch 1 to  $T$  **do**
  - (2) **for** step 1 to  $M$  **do**
  - (3) Randomly select one batch of training data. To restrict the movement based on equation (9)–(11).
  - (4) Calculate the loss  $L(\alpha)$  and its gradient  $\nabla L(\alpha)$  based on equation (8).
  - (5) Update Doc-CNN  $F_\alpha$ .
  - (6) end for
  - (7) end for
  - Planning phase (online)
  - (8) Receive the initial state  $s_0 = \{(\mathcal{E}_{\text{int}}, \mathcal{E}_0), (g_0^1, g_0^2, g_0^3), (x_0^1, x_0^2, x_0^3)\}$  ( $\mathcal{E}_{\text{int}}$  constant)
  - $\mathcal{E}_t$  change over time and location,
  - $(g_t^1, g_t^2, g_t^3) = (g_0^1, g_0^2, g_0^3)$
  - (9) while  $(x_0^1, x_0^2, x_0^3) \neq (g_0^1, g_0^2, g_0^3)$  **do**
  - (10) Input  $s_t$  into Doc-CNN and output  $F_\alpha(s_t)$ .
  - (11) Using action  $a_t$  based on  $F_\alpha(s_t)$ , equation (7), (9)–(11).
  - (12) Update state  $s_t$  to  $s_{t+1}$ .
  - (13) end while
  - (14) Send path planning results to glider formation.

ALGORITHM 1: Learning-based path planning algorithm.

gradient descent. When all batches of data are used for one training, the training epoch ended. After the number of training epochs reaches the maximum, the phase will stop.

In the planning phase, the glider formation first obtains the initial state  $s_0$ , the initial ocean current data  $\mathcal{E}_{\text{int}}$ , and the current time ocean dataset  $\mathcal{E}_t$  obtained by the whole system, the initial position  $(x_0^1, x_0^2, x_0^3)$  of gliders, and the target position  $(g_0^1, g_0^2, g_0^3)$  (line 8). With  $s_0$  as input, Doc-CNN will output the estimated value function  $F_\alpha(s_0)$  (line 10). The moving direction of the glider  $a_0$  can be determined according to equation (7) and its own motion capacity limitation according to equation (9)–(11) (line 11). The position of

gliders becomes  $(x_1^1, x_1^2, x_1^3)$ , and the status can be updated to  $s_1$  (line 12). Repeat the steps of this plan up to  $(x_t^1, x_t^2, x_t^3) = (g_t^1, g_t^2, g_t^3)$  (line 9) to plan the formation global path.

As shown in Figure 5, considering the original dataset  $\mathcal{E}_{\text{int}}$ ,  $\mathcal{E}_t$ , and target position  $(g_0^1, g_0^2, g_0^3)$ , Doc-CNN can output the position of estimated value  $Q$  through calculation. The entire local feature mapping layer (output of Conv-21) is input directly to layer Fc-3. The time and resource consumption of computing the  $Q$  set values  $\{\widehat{Q}(\{(\mathcal{E}_{\text{int}}, \mathcal{E}_t), (g_0^1, g_0^2, g_0^3), (x_t^1, x_t^2, x_t^3)\}, a_t) | (x_t^1, x_t^2, x_t^3) \in \mathcal{X}, a_t \in \mathcal{A}\}$  is approximately equal to a single  $Q$  value  $\widehat{Q}(\{(\mathcal{E}_{\text{int}}, \mathcal{E}_t), (g_0^1, g_0^2, g_0^3), (x_0^1, x_0^2, x_0^3)\}, a_t)$ .



Lines 10–12 of the online planning phase need to be calculated in the initial stage. For the real-time position feedback data of gliders, global planning is modified through the local feature mapping layer, making the whole path more suitable for the time-varying environment. When multiple gliders at different locations share the same destination, traditional search algorithms have to plan the path of each glider in sequence. Doc-CNN can plan paths for them simultaneously through one forward calculation, which significantly improves efficiency. It can be intuitively understood that branch 1 is a global path planning, and branch 2 is a path correction based on time-varying ocean current at different locations.

Doc-CNN architecture can effectively extract both the global features of UGs formation path planning and the local features of the location at the current time point, so it has good performance in global path planning for gliders in time-varying ocean currents. Due to the local connection characteristics of convolutional layers, the single-branch DCNN has to be deep enough, so that global information such as original data set and target information can be transmitted to each state in the final value function layer. Therefore, their convergence speed is slow and calculation cost is high.

In order to speed up the training speed and reduce the network complexity, Doc-CNN uses a separate branch to compress the original data set, effectively preserving and expanding the global information. Therefore, branch 2 can focus on extracting local features at different times in the operation of gliders, significantly reduce the depth of local features, and modify the path, making the path planning with the characteristics of dynamic programming.

## 4. Experiments and Analysis

The experimental settings, comparison, and analysis of Doc-CNN for UGs planning are as follows. Simulations are run with AMD Ryzen 2600  $\times$  3.6 GHz processor, Nvidia 1660Ti GPU, and 16G memory.

*4.1. Experimental Settings.* Two datasets are used to evaluate the performance of the tailored Doc-CNN for UGs path planning.

The first dataset is maps with obstacles, consisting of 10,000 grid maps with size  $64 \times 64$  (actual size 300 km  $\times$  300 km in realistic application) and random man-made obstacles. Each input contains a grid map with coordinates target position and glider location. The grid map can be regarded as a simplified glider operating environment for evaluating the capability of global path planning algorithms.

The second set is ocean maps with obstacles and ocean currents. The data set is generated by desensitization of real ocean current data and consists of 10,000 map sets with size  $128 \times 128$  (actual size 300 km  $\times$  300 km in realistic application), random man-made obstacles, and vector ocean currents at position points. Each input includes a set of initial time 3D current velocity data, a set of real-time ocean current data around the glider (within 8 km of a single

glider) for location features, a set of target positions, and the gliders' input positions. The path planning algorithm based on Doc-CNN for gliders can also be extended to other similar flow field scenarios. Example data images of two datasets are shown in Figure 6.

For each data set, the output is the optimal direction of movement consistent with the motion capability of gliders. 7/8 pieces of data are randomly selected for training, leaving 1/8 pieces of data for testing. In experiments, the maximum training epoch  $T$  is chosen as 70. The initial value of the learning rate  $\delta$  is generally 0.003, and the attenuation value of the learning rate is generally 0.95.

*4.2. Compared with Other Architectures.* The planning under Doc-CNN is compared with the planning under two other algorithm architectures.

VIN is the state-of-the-art deep neural network structure on path planning with full observations [32]. The iteration number  $K$  in VIN is set to 80. By running VIN on the grid map dataset with increasing  $K$  values ( $K=20,40,60,80,100$ ), the optimal  $K$  value is selected and the  $K$  value with the highest training accuracy is found.

DB-CNN, which is a relatively new network, also has two branches. It consists of reprocessing layer, convolution layer, max-pooling layer, and full connection layer. It does not require prior knowledge and achieves higher accuracy and efficiency in global path planning tasks than existing VIN [34].

The accuracy of global path planning, the success rate of global path planning, and the average path per unit energy are used to evaluate the performance of gliders on the path planning task in the ocean current. Among them, accuracy is defined as the percentage of optimal movement direction predicted by them, the success rate is defined as the percentage of safe path planned by them, and the average path per unit energy is the travel energy efficiency planned by them.

*4.3. Training and Testing Results.* Training performance of all architectures on two datasets is as follows. Training performance of all architectures is shown in Figure 7, Figure 8 and Table 2.

In Table 2, UEP is the unit energy forward path length for the glider. The higher the UEP, the better the gliders' energy utilization, the farther the gliders can voyage with the same amount of energy.

*4.3.1. Training Results Analysis.* As shown in Figure 8, the training accuracy and training loss of Doc-CNN converge faster than the other two algorithms in gliders planning. In Table 2, after 70 trained epochs, Doc-CNN achieved high accuracy and success rate in all datasets, superior to other CNN algorithms. Doc-CNN is applied to the planning of glider formations, which makes the planning of paths more accurate and efficient due to the direct use of the detected  $\mathcal{C}_t$  data and the initial global ocean current data  $\mathcal{C}_{int}$ .

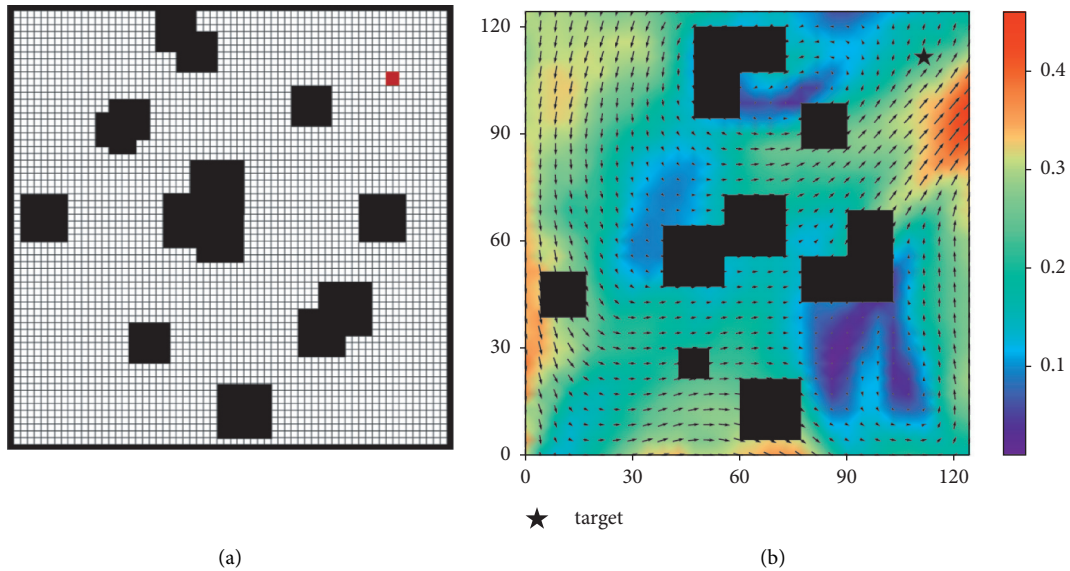


FIGURE 6: Example data images of two datasets. (a) One grid maps of maps dataset. (b) One Ocean data images of Oceans maps of maps dataset.

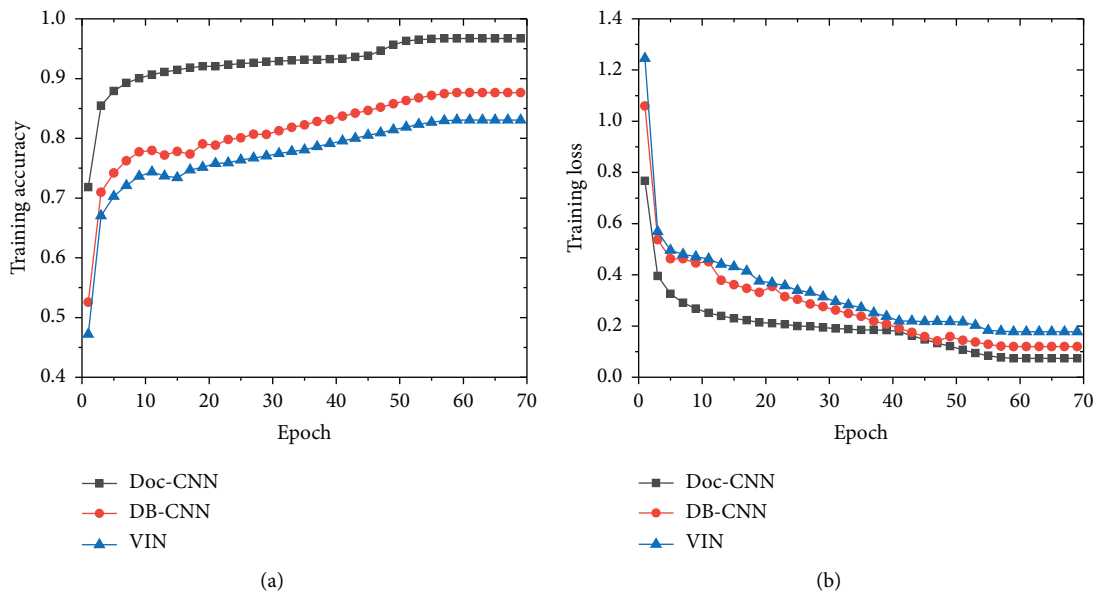


FIGURE 7: Training performance of all architectures on the grid dataset. (a) Training accuracy of the grid dataset. (b) Training loss of the grid dataset.

**4.3.2. Testing Results Analysis.** As shown in Table 2, Doc-CNN maintains excellent global path planning performance on the test data. It is noteworthy that the data in the test data are not exactly the same as those in the training data, which are randomly selected from the dataset, indicating that Doc-CNN is able to plan paths from the unknown data after training. Since gliders work in known or unknown oceans, the known oceans may also have not exactly the same environmental conditions depending on the moment. Therefore, the algorithm needs to plan in both the known environment and the unknown environment to plan the path. Compared to other architectures, Doc-CNN is more effective in actual gliders path planning.

**4.4. Compared with Other Algorithms.** In order to verify the advantages of Doc-CNN in glider path planning, the algorithm is compared with various architectures and algorithms in terms of the number of path cycles, path length, and unit energy path, all of which are indicators of glider path planning. Unit energy path refers to the distance the glider can travel in the horizontal direction per unit of energy. Unit energy path represents the energy utilization efficiency of the glider; the higher the better. In this work, the unit of it is km/Wh [42].

Not all algorithms adapt to the ocean environment and motion constraints of gliders. In this subsection, in addition to the other architectures (DB-CNN and VIN) in the

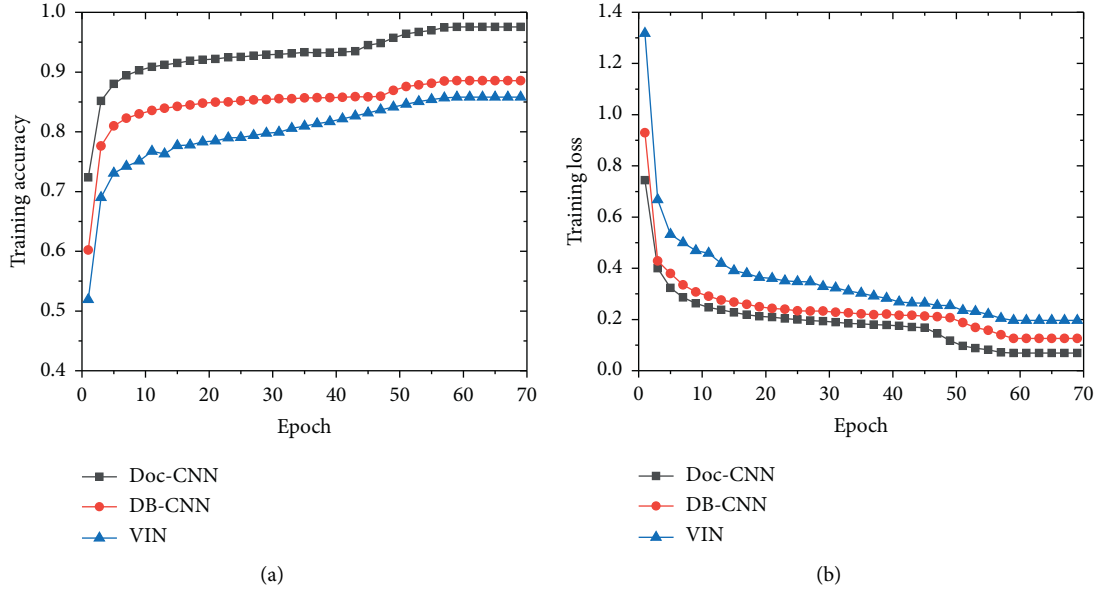


FIGURE 8: Training performance of all architectures on the ocean map dataset. (a) Training accuracy of the ocean map dataset (b) Training loss on the ocean map dataset.

TABLE 2: Accuracy and success rate of all architectures on two datasets.

Dataset	Metrics	Doc-CNN	DB-CNN	VIN
Grid maps	Acc1	<b>96.1%</b>	87.6%	85.8%
	Acc2	<b>90.8%</b>	85.4%	77.4%
	SR1	<b>90.6%</b>	87.6%	69.0%
	SR2	<b>90.0%</b>	83.0%	64.4%
	UEP	<b>0.15</b>	0.12	0.12
Ocean maps	Acc1	<b>97.5%</b>	88.5%	83.0%
	Acc2	<b>92.2%</b>	84.9%	75.6%
	SR1	<b>94.3%</b>	81.0%	48.7%
	SR2	<b>86.5%</b>	72.4%	46.2%
	UEP	<b>0.16</b>	0.12%	0.12%

Acc1: path planning accuracy on training data. Acc2: path planning accuracy on test data. SR1: path planning success rate on training data. SR2: path planning success rate on test data. UEP: unit energy forward path length for the glider. The bold values are the experiment results of the Doc-CNN algorithm proposed in this paper. The bold values of this group data are to strengthen its comparison with other data of algorithms, and it is to show the superiority of the Doc-CNN algorithm.

previous subsection for comparison after the adaptation of ocean environment, the Oci-RRT\* algorithm [42] and the improved artificial potential field method (iAPF) that can adapt to the ocean current are also selected. The DB-CNN, VIN, and APF are redesigned to meet the constraints of UGs' motion and ocean. Simulation experiments are conducted 1000 times. The search step  $L_{step}$  of the OCI-RRT\* algorithm and the step  $L_{step}$  of the iAPF are set as the limit horizontal distance of UGs movement 6.75 km. The analysis of the five algorithms is shown in Figure 9.

Simulation test data are as shown in Table 3.

As shown in Figure 9 and Table 3, in the randomly selected incompletely identical experimental environment, after the algorithms run 1000 experiments, the Doc-CNN is better in the three indicators (number of path cycles, path length, and unit energy path). In addition to the Doc-CNN and Oci-RRT\*, the unit energy path index of the other three algorithms is poor, which conforms to the features that the three algorithms do not optimize for UGs and energy. In

terms of computation time, due to the dual-branch structure of Doc-CNN and DB-CNN, it takes more time. The iAPF is easy to fall into the dead zone and also runs for a long time.

Compared with DB-CNN, VIN, Oci-RRT\*, and iAPF, the average number of cycles of Doc-CNN is reduced by 2.50%, 6.02%, 11.36%, and 14.29%, respectively, and the average path length is reduced by 2.75%, 4.98%, 7.62%, and 9.55%, respectively. The unit energy paths increased by 33.33%, 33.33%, 6.67%, and 45.45%, respectively. The effectiveness and superiority of Doc-CNN are proved.

Although the Doc-CNN is superior to the UGs' problem, it still has its limitations. On the computation time, as shown in Table 3, Doc-CNN is 14.90%, 34.11%, and 17.25% slower than DB-CNN, VIN, and Oci-RRT\*, and 11.67% faster than iAPF. Compared with the traditional CNN, the dual-branch structure of Doc-CNN is complex, and it is easy to generate invalid calculations or makes problems which can be trained with a simple architecture become more complex. For Doc-CNN, compared with VIN, more parameters need to be

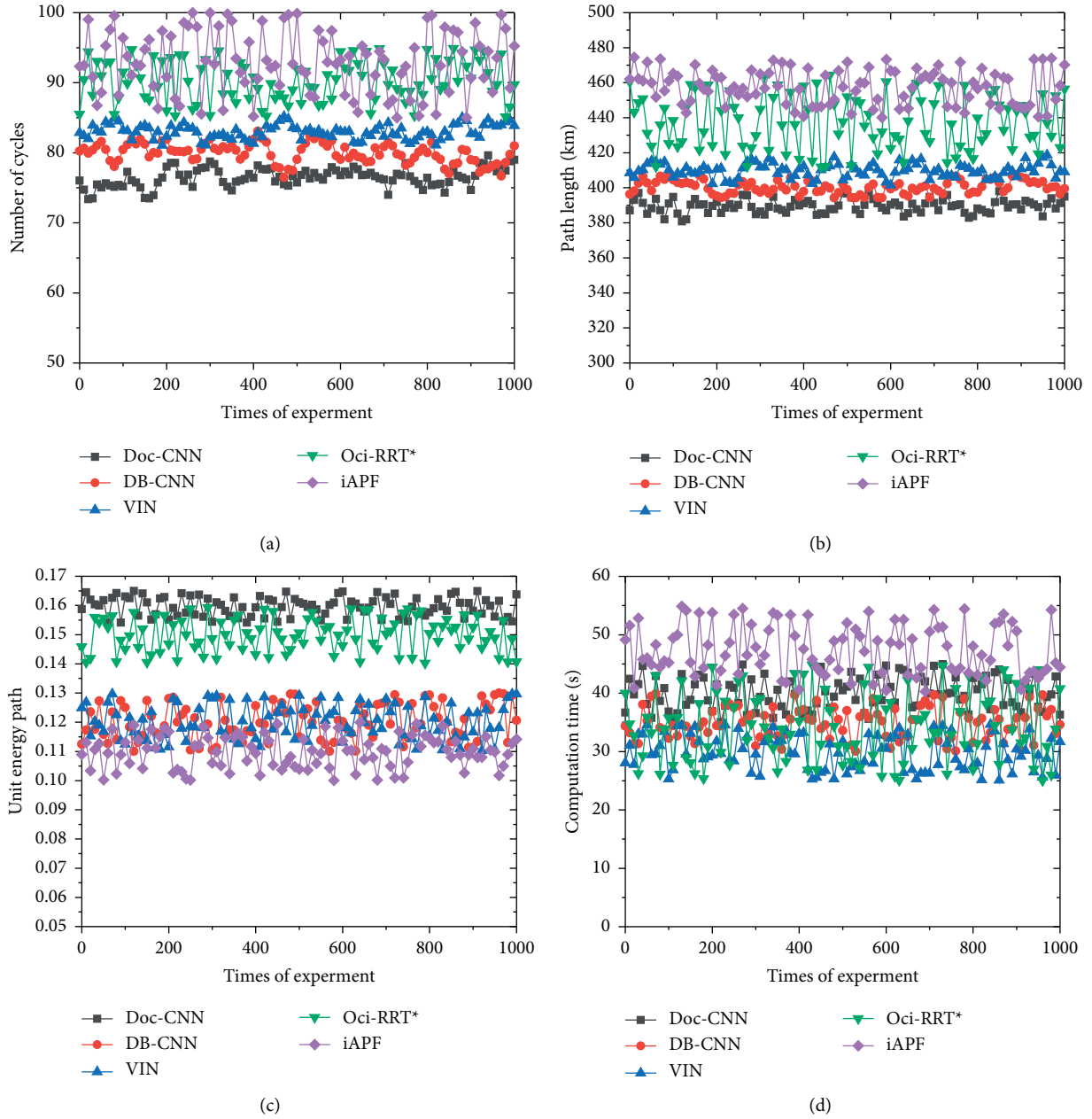


FIGURE 9: Compared to other algorithms. (a) Comparison of the path cycles. (b) Comparison of the path lengths. (c) Comparison of the unit energy paths. (d) Comparison of the computation times.

TABLE 3: Comparison of experimental data of the five algorithms (average of 1000 times).

Methods	Doc-CNN	DB-CNN	VIN	Oci-RRT *	iAPF
Number of cycles	78	80	83	88	91
Path length	389.49	400.50	409.91	421.63	430.61
Unit energy path	0.16	0.12	0.12	0.15	0.11
Computation time	40.1	34.9	29.9	34.2	45.4

adjusted and the range of parameter changes is larger. Under the condition of limited cost, the performance improvement brought by Doc-CNN does not necessarily offset the consumption. Even so, for the planning of UGs, Doc-CNN is also an available and advantageous algorithm.

*4.5. Simulation Results and Discussion.* Figure 10 shows the positions corresponding to the glider formation motion when the Doc-CNN is planning the formation. It contains the shape of the glider formation at each position state. Figure 11 shows the relative distance between individual

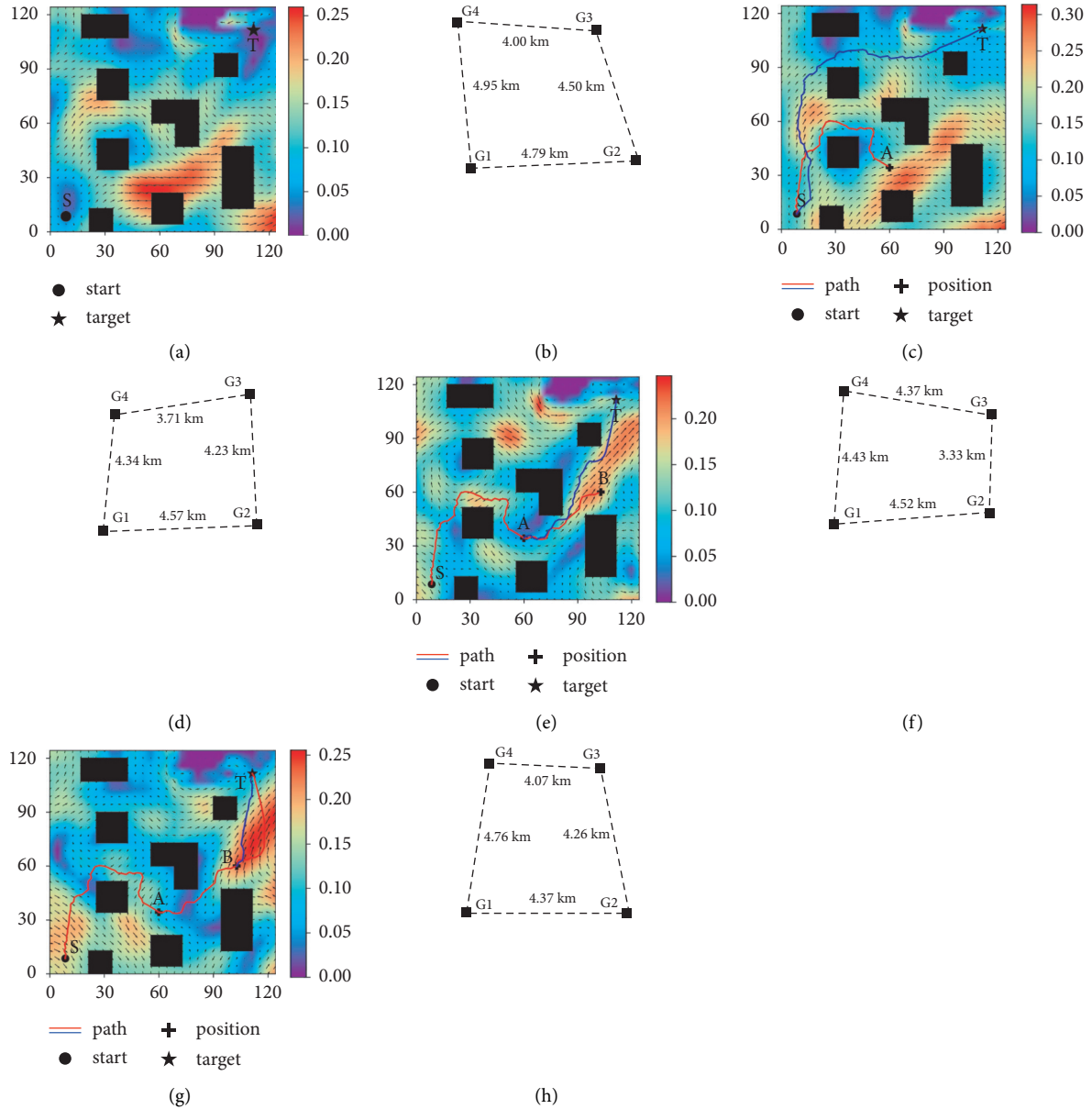


FIGURE 10: The UGs formation path planning on  $128 \times 128$  time-varying ocean maps and the formation shape of UGs in different cycles. (a) State S. (b) Formation Shape in S. (c) State A. (d) Formation Shape in A. (e) State B. (f) Formation Shape in B. (g) State T. (h) Formation Shape in T.

gliders in the horizontal direction during this planning process. This means that the gliders did not collide during the planning process.

Figure 10 illustrates that the Doc-CNN improved for glider problems can be useful for glider planning in time-varying ocean current environment. Each planning is a decision made by gliders in the current state through Doc-CNN. The ocean currents in Figure 10(a), 10(c), 10(e), and 10(g) correspond to the currents at four moments, respectively: the moment of the start point S (ocean current  $\mathcal{C}_{int}$ ), the moments of the intermediate states A and B, and the moment of the arrival at the end point T, where the blue line indicates the path planned by the glider formation to reach the target from state points S, A, and B using the Doc-

CNN algorithm, using the current oceans as a static environment. The blue line is used to compare the ability of the Doc-CNN algorithm to plan based on  $\mathcal{C}_{int}$  and dynamic  $\mathcal{C}_t$ . Figure 10 also shows the shape and the horizontal distance relationship within the glider formation when the formation is in the four states S, A, B, and T. Figure 11 illustrates that the gliders are not colliding within the formation in the planning process.

Figure 12 shows some successful paths planned by Doc-CNN based on the ocean data environment. It can be seen that the path of formation is able to avoid obstacles of different sizes (man-made obstacles) with the guidance of Doc-CNN. In addition, the path trajectory is almost optimal. It is worth noting that prior knowledge of obstacles is only

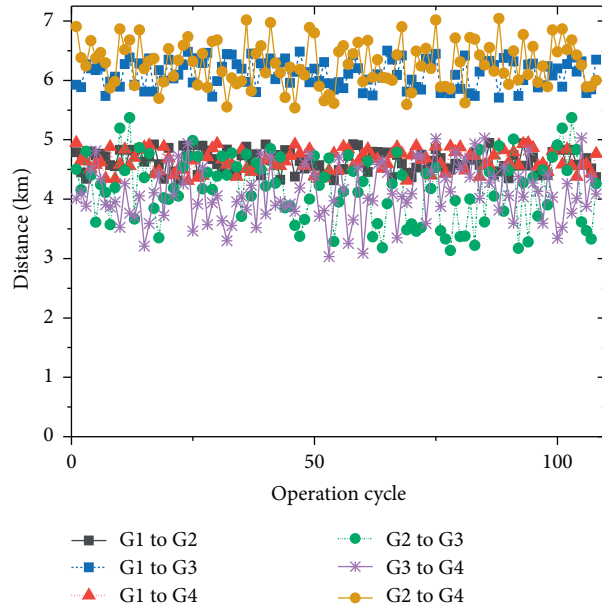


FIGURE 11: Relative horizontal distance between gliders in one task formation.

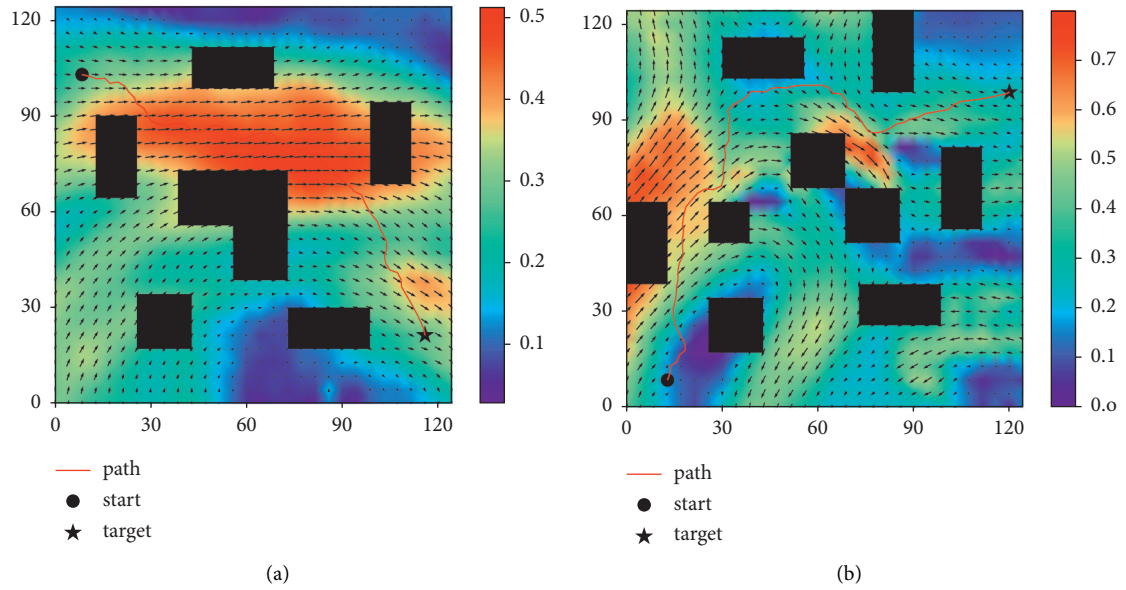
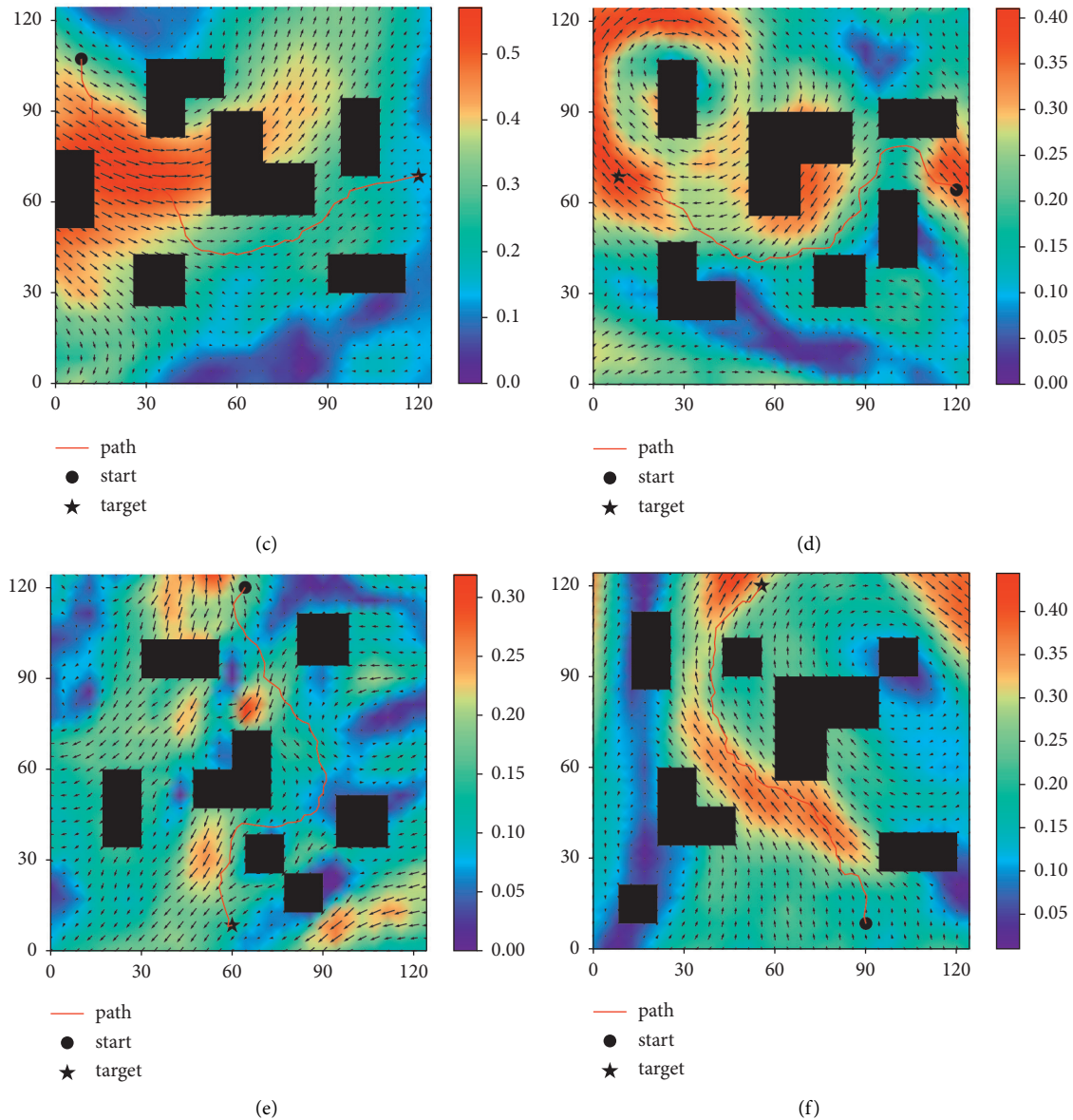


FIGURE 12: Continued.

FIGURE 12: Images of experiments on  $128 \times 128$  ocean maps dataset.

known in  $\mathcal{E}_{\text{int}}$ , and Doc-CNN needs to be trained to learn and understand these deep features of obstacles at the initial moment in order to still have the ability to avoid obstacles during travel, through changes by  $\mathcal{E}_t$ . Therefore, the performance of Doc-CNN is more applicable to the glider planning problem in ocean currents.

Doc-CNN is an algorithmic architecture that is tailored to the individual motion characteristics of gliders and the operational characteristics of glider formations in operation, making full use of known global information as well as local information collected by the gliders themselves. The dual branches structure of Doc-CNN retains the ocean current characteristics of gliders in applications and makes use of them. In summary, Doc-CNN can play a positive role in the path planning of glider formation and maintain its advantages in the other two CNN architectures.

## 5. Conclusions

This work utilizes the DCNN type algorithm approach to solve the path planning problems faced by the glider formation in practical application. In order to solve the planning problem of glider formations in a large-scale ocean current environment, this paper improves the DCNN algorithm as follows. Firstly, the algorithm is improved according to the characteristics of gliders; this architecture directly plans the gliders' path from the environmental data level, without prior knowledge of environmental data, and there is no need to know the global environmental data all the time. Secondly, according to the characteristics of energy consumption of a single glider, the decision that can be used for glider planning is obtained, so that it conforms to the motion characteristics of gliders and optimizes the path at the same time, reducing the path cycles. A large number of

simulation experiments show that the improved algorithm described in this research outperforms the DB-CNN and VIN algorithms and has advantages in energy efficiency and the ability to operate in dynamic environments. The Doc-CNN algorithm proposed in this paper has broad application prospects in the field of glider planning.

The contribution of this research is to explore suitable algorithms for the path planning application of glider formations in the ocean environment, improve the ability of formations to perform path planning tasks, and provide support for the use and rapid deployment of UGs in practical applications. It makes contributions to the path planning of UGs formation. Firstly, an effective planning algorithm is proposed for the rapid deployment of low-speed gliders (taking UGs as an example) in a relatively unfamiliar ocean environment. Secondly, too radical control strategies are reduced. The trained Doc-CNN network can make the formation and internal gliders' energy utilization (unit energy path) better and more conducive to the long-term operation of gliders. Thirdly, due to the improvement of energy utilization, the glider and the formation have enough energy to make effective maneuvers when necessary and sudden. Finally, the paper proves that the class algorithm of DCNN can also be applied to path planning problems in large-scale ocean environments, which lays the foundation for further research.

In future research, in addition to the improvement of the algorithm, the collaborative planning of multiple formations on a large scale will be studied. For example, in the overall ocean current, the environmental information of the formation in the front currents can be used as a reference and prediction basis for the future environment of the rear formation. While strengthening the accuracy of the system environment prediction, the rear formation can be dynamically planned to a certain extent. In this way, the planning of static and dynamic coexistence and full utilization of environmental data in the large formation of gliders is carried out. This situation will bring new challenges to the planning algorithm of gliders. The artificial neural network will also be considered to solve this kind of glider formation planning, simplify the input of the overall problem, and rapidly get the optimal path.

## Data Availability

Ocean current data can be found at <https://www.hycom.org/>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] A. Alvarez, A. Caffaz, A. Caiti et al., "Fòlaga: a low-cost autonomous underwater vehicle combining glider and AUV capabilities," *Ocean Engineering*, vol. 36, no. 1, pp. 24–38, 2009.
- [2] H. Claustre, L. Beguery, and P. L. A. Patrice, "SeaExplorer glider breaks two world records multisensor UUV achieves global milestones for endurance, distance," *Sea Technology*, vol. 55, no. 3, pp. 19–22, 2014.
- [3] S. L. Wood and C. E. Mierzwa, "State of technology in autonomous underwater gliders," *Marine Technology Society Journal*, vol. 47, no. 5, pp. 84–96, 2013.
- [4] J. G. Graver, *Underwater Gliders :dynamics, Control and Design*, Princeton University, Princeton, NJ, USA, 2005.
- [5] N. A. A. Hussain, M. R. Arshad, and R. Mohd-Mokhtar, "Underwater glider modelling and analysis for net buoyancy, depth and pitch angle control," *Ocean Engineering*, vol. 38, no. 16, pp. 1782–1791, 2011.
- [6] S. Zhang, J. Yu, A. Zhang, and F. Zhang, "Spiraling motion of underwater gliders: modeling, analysis, and experimental results," *Ocean Engineering*, vol. 60, pp. 1–13, 2013.
- [7] A. Alvarez and B. Mourre, "Optimum sampling designs for a glider-mooring observing network," *Journal of Atmospheric and Oceanic Technology*, vol. 29, no. 4, pp. 601–612, 2012.
- [8] D.-Y. Xue, Z.-L. Wu, Y.-H. Wang, and S.-X. Wang, "Coordinate control, motion optimization and sea experiment of a fleet of petrel-II gliders," *Chinese Journal of Mechanical Engineering*, vol. 31, no. 1, p. 17, 2018.
- [9] W. Dong, X. Sheng, L. Jia, Q. Ling, and X. Jiang, "Energy constrained multi-platform network underwater detection performance," in *Proceedings of the 9th ACM International Conference on Underwater Networks and Systems, WUWNET 2014*, Rome, Italy, November 2014.
- [10] B. Cushman-Roisin and J.-M. Beckers, *Introduction to Geophysical Fluid Dynamics: Physical and Numerical Aspects*, Academic Press, Cambridge, MA, USA, 2011.
- [11] N. Sariff and N. Buniyamin, "An overview of autonomous mobile robot path planning algorithms," in *Proceedings of 2006 4th Student Conference on Research and Development "Towards Enhancing Research Excellence in the Region"*, pp. 183–188, Shah Alam, Malaysia, June 2006.
- [12] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: a general overview," in *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches* vol. 29, , pp. 3–27, Springer International Publishing, 2015.
- [13] K. Yao, J. Li, B. Sun, and J. Zhang, "An adaptive grid model based on mobility constraints for UAV path planning," in *in Proceedings of 2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE)*, pp. 207–211, Singapore, Singapore, July 2016.
- [14] U. Orozco-Rosas, O. Montiel, and R. Sepulveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Applied Soft Computing*, vol. 77, pp. 236–251, 2019.
- [15] U. Orozco-Rosas, K. Picos, and O. Montiel, "Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots," *IEEE Access*, vol. 7, pp. 156787–156803, 2019.
- [16] J. H. Holland, *Adaptation in Natural and Artificial Systems*, A Bradford Book, Boston, FL, USA, 1992.
- [17] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.
- [18] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [19] K. Tanakitkorn, P. A. Wilson, S. R. Turnock, and A. B. Phillips, "Grid-based GA path planning with improved cost function for an over-actuated hover-capable AUV," in *Proceedings of*



- the 2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pp. 1–8, Oxford, England, October 2014.
- [20] X. Pan, X. Wu, and X. Hou, “Research on global path planning for autonomous underwater vehicle considering ocean current,” in *Proceedings of 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 790–793, Xi’an, China, May 2018.
- [21] S. Mahmoud Zadeh, D. M. W. Powers, and K. Sammut, “An autonomous reactive architecture for efficient AUV mission time management in realistic dynamic ocean environment,” *Robotics and Autonomous Systems*, vol. 87, pp. 81–103, 2017.
- [22] L. Foschini, J. Hershberger, and S. Suri, “On the complexity of time-dependent shortest paths,” *Algorithmica*, vol. 68, no. 4, pp. 1075–1097, 2014.
- [23] B. C. Dean, “Shortest paths in FIFO time-dependent networks: theory and algorithms,” *Rapport technique*, vol. 13, 2004.
- [24] J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, “Efficient optimal planning in non-FIFO time-dependent flow fields,” 2019, <https://arxiv.org/abs/1909.02198>.
- [25] J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, “Hierarchical planning in time-dependent flow fields for marine robots,” in *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, pp. 885–891, Paris, France, May 2020.
- [26] M. Ghatee and A. Mohades, “Motion planning in order to optimize the length and clearance applying a Hopfield neural network,” *Expert Systems with Applications*, vol. 36, no. 3, pp. 4688–4695, 2009.
- [27] S. Li and Y. Guo, “Neural-network based AUV path planning in estuary environments,” in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, pp. 3724–3730, Beijing, China, July 2012.
- [28] B. Liu and Z. Lu, “AUV path planning under ocean current based on reinforcement learning in electronic Chart,” in *Proceedings 2013 International Conference on Computational and Information Sciences (ICIS)*, pp. 1939–1942, Shiyan, Hubei, China, June 2013.
- [29] U. Gautam, R. Malmathanraj, and C. Srivastav, “Simulation for path planning of autonomous underwater vehicle using Flower Pollination Algorithm, Genetic Algorithm and Q-Learning,” in *Proceedings 2015 International Conference on Cognitive Computing and Information Processing (CCIP)*, pp. 1–5, Noida, India, March 2015.
- [30] Y. Cheng and W. Zhang, “Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels,” *Neurocomputing*, vol. 272, pp. 63–73, 2018.
- [31] X. Lei, Z. Zhang, and P. Dong, “Dynamic path planning of unknown environment based on deep reinforcement learning,” *Journal of Robotics*, vol. 2018, Article ID 5781591, 10 pages, 2018.
- [32] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel, “Value iteration networks,” 2016, <https://arxiv.org/abs/1602.02867>.
- [33] J. Gu, Z. Wang, J. Kuen et al., “Recent advances in convolutional neural networks,” *Pattern Recognition*, vol. 77, pp. 354–377, 2018.
- [34] J. Zhang, Y. Xia, and G. Shen, “A novel learning-based global path planning algorithm for planetary rovers,” *Neurocomputing*, vol. 361, pp. 69–76, 2019.
- [35] H. Zhou, T. L. Wang, L. Sun, and W. Lan, “Study on the vertical motion characteristics of disc-type underwater gliders with zero pitch angle,” *Journal of Marine Science and Technology*, vol. 25, no. 3, pp. 828–841, 2020.
- [36] J. J. H. Lee, C. Yoo, R. Hall, S. Anstee, and R. Fitch, “Energy-optimal kinodynamic planning for underwater gliders in flow fields,” in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA)*, pp. 42–51, Sydney, NSW, Australia, December 2017.
- [37] J. Isern-Gonzalez, D. Hernandez-Sosa, E. Fernandez-Perdomo, J. Cabrera-Gamez, A. C. Dominguez-Brito, and V. Prieto-Maranon, “Path planning for underwater gliders using iterative optimization,” in *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1538–1543, Shanghai, China, May 2011.
- [38] A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme, “Risk-aware path planning for autonomous underwater vehicles using predictive ocean models,” *Journal of Field Robotics*, vol. 30, no. 5, pp. 741–762, 2013.
- [39] 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.
- [40] C. S. Kulkarni and P. F. J. Lermusiaux, “Three-dimensional time-optimal path planning in the ocean,” *Ocean Modelling*, vol. 152, 2020.
- [41] P. F. J. Lermusiaux, P. J. Haley, W. G. Leslie, A. Agarwal, O. G. Logutov, and L. J. Burton, “Multiscale physical and biological dynamics in the philippine archipelago: predictions and processes,” *Oceanography*, vol. 24, no. 1, pp. 70–89, 2011.
- [42] W. Lan, X. Jin, T. Wang, and H. Zhou, “Improved RRT algorithms to solve path planning of multi-glider in time-varying ocean currents,” *IEEE Access*, vol. 9, pp. 158098–158115, 2021.