

# Encoder-Decoder Based Deep Reinforcement Learning for Multi-AUV Assisted Data Collection in UWSNs

Jing Xu, Jiarun Tang, Xiao Huang, Xuan Gu, Lanhua Li, Jianhua He, and Wei Liu

**Abstract**—Reliable and timely data collection poses a significant challenge for underwater wireless sensor networks (UWSNs), primarily due to the extremely low data rate of underwater communication and the difficulty in supplying energy to the sensor nodes. In this paper, we explore the use of multiple autonomous underwater vehicles (AUVs) for sequential data collection from sensor nodes distributed over a large area. To facilitate efficient data collection, appropriate cluster heads (CHs) are selected in each region to collect and aggregate data from nearby energy-limited sensor nodes, and then forward the data to the mobile AUVs. Specifically, we investigate how to maximize the value of information (VoI) of the collected data through the joint optimization of CH selection and multi-AUV path planning, while accounting for the energy constraints of sensor nodes. To tackle the formulated VoI maximization problem, we propose a deep reinforcement learning algorithm based on an encoder-decoder architecture. The whole UWSN system is fed into the encoder network, followed by a composite decoder consisting of an AUV selection decoder and a cluster access sequence decoder to generate the cluster access sequence for each AUV. Based on the determined sequences, we further utilize the dynamic programming algorithm to achieve optimal CH selection. Finally, we obtain the sequence of AUVs that access the selected CHs. Simulation results demonstrate that the proposed learning-based approach converges and achieves a higher VoI than existing benchmark algorithms.

**Index Terms**—Autonomous underwater vehicle (AUV), underwater wireless sensor networks (UWSNs), data collection, value of information (VoI), path planning

## I. INTRODUCTION

Underwater wireless sensor networks (UWSNs) play a crucial role in ocean monitoring and resource exploitation, with applications such as marine environment monitoring, surveillance of underwater facilities, and early disaster warning [1]. One of the major challenges for UWSNs is achieving reliable and energy-efficient data collection. Compared to

terrestrial sensor networks, communication between underwater sensor nodes and data sinks often relies on acoustic signals due to the rapid attenuation of radio frequency signals in underwater environments. However, underwater acoustic communication is known for its low data rate, large delay, and high energy consumption limitations [2]. Additionally, replacing or recharging batteries for underwater sensor nodes is extremely difficult. Therefore, there is a strong demand for reliable and energy-efficient data collection methods.

Recent advances in autonomous underwater vehicles (AUVs) have led to growing research interest in using them to assist data collection in UWSNs. A key advantage of using AUVs for data collection is their ability to periodically navigate toward sensor nodes and collect data from them using short-range and high-speed communication links (such as optical or electromagnetic communication) [3]. In this way, data can be collected faster with less energy consumption of the sensor nodes, thereby significantly extending the lifetime of UWSNs. To enhance the efficiency of data collection, a typical approach is to cluster sensor nodes and select cluster heads (CHs) to collect and aggregate data within each cluster [4], [5]. This enables AUVs to collect data from the entire sensing region by visiting only a limited number of CHs, thereby reducing the complexity of AUV path planning.

Existing studies have focused on single AUV deployments in small-scale scenarios. However, the increasing scale of UWSNs and the complexity and low-latency requirements of underwater operational tasks demand collaborative data collection by multiple AUVs [6]. In this paper, we are motivated to investigate the use of multiple AUVs for reliable and efficient data collection in large-scale UWSNs, which typically accomplish node clustering based on the location information of sensor nodes [7]. Therefore, we focus on the path planning of the AUVs, as it significantly affects both their energy consumption and the data collection delay [8]. In addition, selecting CHs to collect and aggregate data from nearby nodes within each cluster has proven to be an efficient approach for transmitting data to AUVs [9]. These observations motivate the joint optimization of CH selection and multi-AUV path planning.

We first formulate the joint optimization problem for the task, taking into account the energy constraints of sensor nodes. Unlike existing research that primarily focuses on reducing data latency and extending the lifetime of sensor

This work was supported in part by EPSRC with RC Grant EP/Y027787/1, UKRI under Grant EP/Y028317/1, Horizon European programme under Grant 101086228, and the Royal Society International Exchanges 2023 under Grant IEC/NSFC/233318. (Corresponding author: Wei Liu.)

J. Xu, J. Tang, X. Gu, W. Liu are with the School of Electronics Information and Communications, Huazhong University of Science and Technology, 430074 Wuhan, China. (e-mail: xujing@hust.edu.cn; tangjiarun@hust.edu.cn; guxuan@hust.edu.cn; liuweili@hust.edu.cn).

X. Huang is with the China Ship Development and Design Center, 430064 Wuhan, China. (e-mail: huangxiao\_88@outlook.com).

L. Li is with the School of Intelligent Systems Engineering, Sun Yat-Sen University, 518107 Shenzhen, China. (e-mail: lilh65@mail.sysu.edu.cn).

J. He is with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K. (e-mail: j.he@essex.ac.uk).

Copyright © 2025 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

networks, we aim to maximize the value of information (VoI), which accounts for both the timeliness and the varying importance of the collected data [10]–[12]. Consequently, we focus on the VoI collected by the system in order to provide a comprehensive consideration of the importance and timeliness of the data. Furthermore, the path planning problem can be viewed as a sequence-to-sequence (Seq2Seq) problem, which can be efficiently addressed using an encoder-decoder architecture. Thus, we propose an encoder-decoder based deep reinforcement learning (EDDRL) algorithm for its solution. The UWSN system serves as input to the encoder network, which is then decoded by a composite decoder comprising an AUV selection decoder and a cluster access sequence decoder. This process allows us to obtain the cluster access sequence for each AUV. Subsequently, CHs are selected by using a dynamic programming method with the predetermined sequences, thereby deriving the selected CHs access sequence for each AUV. Simulation results demonstrate that the proposed DRL algorithm converges efficiently and outperforms the benchmark algorithms in terms of the retained VoI.

Specifically, the main contributions of this paper are summarized as follows:

- 1) We investigate the application of multiple AUVs for reliable and efficient data collection in UWSNs. An optimization problem for joint CH selection and multi-AUV path planning is formulated to maximize the VoI of data collection, while considering the energy constraints of sensor nodes.
- 2) We view the problem of multi-AUV path planning as a sequence mapping problem and design an encoder-decoder architecture to obtain the cluster access sequence for each AUV, followed by a dynamic programming method to obtain the result of optimal CH selection.
- 3) Our proposed EDDRL algorithm is trained end-to-end in an unsupervised manner using the REINFORCE algorithm, and extensive simulations under realistic energy constraints of sensor nodes demonstrate that it consistently achieves higher VoI than the state-of-the-art algorithms.

The remainder of this paper is organized as follows. Section II introduces the related work. The system model and problem formulation are detailed in Section III. The proposed algorithm is elaborated in Section IV. Section V provides simulation results, and Section VI concludes the paper.

## II. RELATED WORK

Underwater communication is significantly influenced by the complexity of the environment. Consequently, there is a growing need to enhance the performance and efficiency of UWSNs. In this context, AUV-assisted data collection schemes in UWSNs have attracted considerable attention. These schemes involve a range of techniques, including node clustering, CH selection and AUV path planning.

Node clustering and CH selection significantly influence communication distance among nodes within clusters, which in turn affects the node energy consumption. Furthermore, the result of CH selection determines the AUV trajectory, thus

underscoring the importance of node clustering and CH selection for data collection in UWSNs. Han *et al.* [13] considered water flow characteristics for hierarchical data collection. In this context, the upper layer of the water flow, being faster, was used for data collection via multi-hop routing, while the lower layer used a neighborhood density-based clustering algorithm for node clustering, with the data collected by an AUV. In [14], Ghoreyshi *et al.* proposed a cluster-based mobile data collection scheme, where node clustering is conceptualized as an optimization challenge, considering both collection latency and node energy consumption. Considering the high energy consumption of CHs, Wang *et al.* [15] developed a high-energy node priority clustering algorithm. This approach selects CHs based on the remaining energy of sensor nodes and the geometry distance among them. Khan *et al.* [16] proposed a scheme in which an AUV is responsible for cluster partitioning and CH selection. The AUV dynamically replaces CHs based on real-time network conditions. In [8], Huang *et al.* employed the K-means clustering algorithm for node clustering and conducted a comprehensive analysis of the selection of initial cluster centers and the value of K. Additionally, the authors proposed that nodes within a specified distance from the AUV trajectory be designated as auxiliary CHs to balance the energy consumption. Qin *et al.* [17] developed an energy-aware clustering protocol based on the improved K-means and introduced a dynamic CH switching mechanism that considers energy load and node survival rate. In a game-theory-based approach, Xing *et al.* [18] introduced a clustering scheme where each node decides to maximize gains according to the Nash equilibrium during the CH selection phase. They introduced a reward mechanism to encourage nodes to make beneficial collective decisions, thereby effectively rotating CHs and balancing node energy consumption. In [19], Guang *et al.* proposed a joint optimized data collection algorithm (JODA) for CH selection, considering both node energy and location. This algorithm selects CHs with higher energy while satisfying the VoI constraint.

AUV path planning not only affects the AUVs' energy consumption but also has a significant impact on the efficiency and quality of data collection. Consequently, the planning of appropriate AUV trajectory is of great importance for the enhancement of UWSN performance. The research on AUV path planning is primarily classified into two categories: single AUV path planning and multi-AUV joint path planning. In order to reduce the data collection delay and increase the collected VoI, Khan *et al.* [20] proposed the utilization of end-to-end data freshness constraints in order to design the AUV movement paths. This approach produces dynamically traversed paths that can improve the overall data freshness of UWSNs. In [21] and [22], Han *et al.* considered setting the AUV path to a fixed elliptical trajectory, with the AUV moving according to the pre-set trajectory and performing periodic trajectory updates. To reduce the number of locations that an AUV needs to visit, a probabilistic neighborhood location-point covering set-based data collection algorithm was designed for AUV in [23], which is suitable for 3D underwater sensor networks. The algorithm enables the AUV to visit a few selected locations by constructing a minimum

probabilistic domain coverage set. Furthermore, Han *et al.* [24] considered that sensor nodes would generally remain stable and proposed the KRR algorithm for data prediction, which can predict the activity of sensor nodes, thereby reducing AUV travel distance. In [25], Wang *et al.* similarly proposed using AUVs and other edge devices to collect and analyze data, with the AUV predicting future data based on historical information, thereby reducing the number of CHs that needs to be visited.

In scenarios where sensor node locations are difficult to obtain, Cheng *et al.* [26] proposed a dynamic trajectory planning algorithm. This algorithm dynamically detects sensor node positions and adjusts the AUV cruising direction, enhancing the efficiency of data collection under energy constraints. Yao *et al.* [27] addressed the AUV minimum time path planning problem with simultaneous arrival constraints. They divided the overall framework into two modules and solved it using a biased min-consensus (BMC) method. The first module plans the trajectory of each AUV, achieving arrival at the destination with the minimum time through the ocean current, while the second module combines the simultaneous arrival collaborative constraints, selecting the path with the maximum sailing time as the benchmark, and re-planning the other AUV paths until the corresponding sailing times close to the benchmark. In [28], Wen *et al.* proposed a reinforcement learning reward rule based on the idea of artificial potential field for path planning, promoting underwater data collection through the matrix completion method. In the event of an uncertain target location, Wang *et al.* [29] proposed a distributed MAPPO optimization scheme to facilitate target search and environment exploration, which can adjust the number of AUVs for each task according to actual demand. On this basis, a target uncertainty map-assisted AUV swarm data collection scheme based on the MAPPO optimization algorithm was proposed [30]. This scheme guides the AUV swarm to prioritize regions with a higher probability of containing sensor nodes through the target uncertainty map and introduces the information age as a reward to achieve a comprehensive exploration of the unknown environment. Song *et al.* [3] proposed a distributed multi-AUV trajectory planning algorithm based on deep reinforcement learning to maximize data collection rate and energy efficiency. In [12], Wang *et al.* proposed a local global DQN approach to plan the paths of multiple AUVs for data collection, with the aim of reducing the overall energy consumption of UWSNs.

Existing works typically assume that node clustering and CH selection have been completed based on factors such as node energy and location, with subsequent efforts focusing primarily on multi-AUV path planning. However, this assumption disregards the potential impact of CH selection outcomes on multi-AUV path planning, resulting in AUV movement distances primarily determined by the pre-selected CHs. Notably, variations in CH selection results can lead to substantial differences in AUV movement distances, consequently impacting the overall system Vol. To address this issue, this paper proposes a joint optimization of CH selection and multi-AUV path planning.

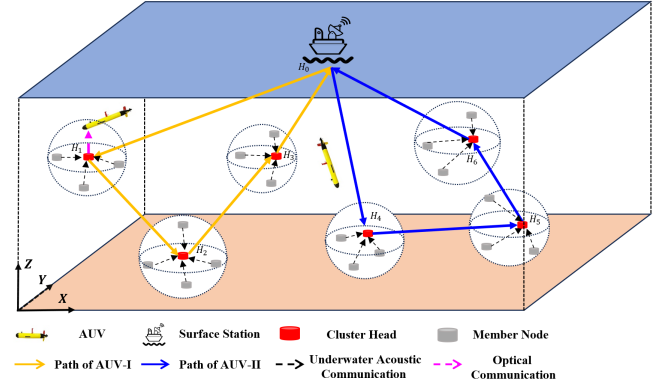


Fig. 1. Example of multi-AUV data collection for UWSNs.

### III. SYSTEM MODEL

In this study, we consider large-scale UWSNs deployed across extensive ocean regions for collaborative data collection using multiple AUVs, as illustrated in Fig. 1. The system comprises a surface station, multiple AUVs, and numerous underwater sensor nodes, whose spatial distribution follows practically motivated rules. Sensor nodes are pre-deployed according to UWSN monitoring requirements, such as long-term environmental observation, underwater infrastructure surveillance, and disaster early warning. These nodes are partitioned into non-overlapping geographical clusters aligned with operational regions. Within each cluster, the CH aggregates data from member nodes prior to AUV arrival. At the beginning of each collection round, all AUVs depart from the surface station and return after completing their assigned CH visits. Under non-emergency conditions, this procedure is periodically repeated to accomplish systematic data collection and offloading.

To improve the efficiency of data collection, all nodes are organized into  $M$  clusters, each of which is designated a CH. For any given cluster  $C_m, m \in \{1, 2, \dots, M\}$ , the CH is designated as  $H_m, H_m \in C_m$ , making the set of CHs represented by  $H = \{H_1, H_2, \dots, H_M\}$ . Before the AUV arrives, CH  $H_m$  collects data from other member nodes through underwater acoustic communication. When the AUV visits  $H_m$ , the CH transmits a data packet to the AUV via an optical communication link. As a result, the energy consumption of the member node  $i$  and the CH  $H_m$  of cluster  $C_m$  can be represented as:

$$\begin{aligned} \mathcal{E}_i &= P_t \frac{D_i}{R(d_i)}, \\ \mathcal{E}_m &= P_r \sum_{i \in \widetilde{C}_m} \frac{D_i}{R(d_i)} + P_t \frac{I_m}{R_o}, \end{aligned} \quad (1)$$

where  $D_i$  is the amount of data collected from the underwater environment by the node  $i$  in cluster  $C_m$ ,  $I_m$  is the total amount of data in the cluster, denoted by  $I_m = \sum_{i \in C_m} D_i$ . The receiving power and transmitting power of the sensor node are denoted by  $P_r$  and  $P_t$ , respectively. The set of member nodes in the cluster  $C_m$  is denoted by  $\widetilde{C}_m$ . The Euclidean distance of node  $i$  from the CH  $H_m$  is denoted by  $d_i$ .  $R(d_i)$

represents the underwater acoustic communication rate [31], while the optical communication rate between the AUV and the CH is denoted by  $R_o$ .

We assume that the number of AUVs is  $U$ , and AUV $_j$  needs to access  $Z_j$  CHs during a data collection task. We further define the binary variable  $A_j^m[\zeta] \in \{0, 1\}$  to indicate whether AUV $_j$  visits the CH  $H_m$  for data collection at step  $\zeta$ , where  $\zeta$  ranges from 1 to  $Z_j$ , with each step corresponding to a single CH visit by AUV $_j$ . A value of 1 represents visitation, while 0 indicates non-visitation. Therefore, the CHs access sequence for all AUVs can be denoted as  $A = \{A_j^m[\zeta], 1 \leq j \leq U, 1 \leq m \leq M, 1 \leq \zeta \leq Z_j\}$ . As mentioned above, the movement of AUVs must satisfy the constraints as follows:

$$\sum_{j=1}^U \sum_{\zeta=1}^{Z_j} A_j^m[\zeta] = 1, \forall 1 \leq m \leq M, \quad (2a)$$

$$\sum_{\zeta=1}^{Z_j} \sum_{m=1}^M A_j^m[\zeta] = Z_j, \forall 1 \leq j \leq U, \quad (2b)$$

where (2a) enforces that each CH is visited by exactly one AUV in each data collection round, and (2b) guarantees that AUV $_j$  accesses all assigned CHs, with the number of visit steps matching the quantity of assigned CHs  $Z_j$ . The trajectory of AUV $_j$  is denoted as  $\mathcal{L}_j = \{\ell_j[0], \ell_j[1], \dots, \ell_j[Z_j], \ell_j[0]\}$ . Here,  $\ell_j[0]$  denotes the coordinates of the surface station, while  $\ell_j[\zeta]$  for  $1 \leq \zeta \leq Z_j$  represents the coordinates of the CH visited by AUV $_j$  at step  $\zeta$ . It is assumed that the power resource allocated for AUVs is sufficient for completing the data collection process, and thus AUV energy constraints are not explicitly modeled in this study. This simplification is motivated by the multi-AUV data collection scenario under consideration, where the collaborative operation of multiple AUVs mitigates the impact of individual energy limitations.

When an AUV accesses the CH  $H_m$  for data collection, following the modeling principles in [10], [11], the data VoI of cluster  $C_m$  at time  $t$  is defined as:

$$V_m(t) = \begin{cases} \beta E_m + (1 - \beta) E_m f(t), & t \geq T_m, \\ 0, & t < T_m, \end{cases} \quad (3)$$

where  $V_m(t)$  denotes the VoI of the data collected by the AUV from the CH  $H_m$  at time  $t$ . Here,  $T_m$  is the data generation time, and  $E_m$  represents the baseline importance of the data at  $t = T_m$ , normalized to a dimensionless utility score. The coefficient  $\beta \in [0, 1]$  is a dimensionless trade-off factor that balances intrinsic importance and timeliness. The decay kernel  $f(t) = e^{-(t-T_m)/\alpha} \in (0, 1]$  is dimensionless and captures the exponential attenuation of information value over time, where  $\alpha > 0$  controls the decay rate.

Once the data is transmitted to the AUV, its VoI decreases over time, and the rate of this decrease is influenced by the subsequent travel time of the AUV. Consequently, it is imperative to strategically devise the path for each AUV in order to minimize the loss of VoI during the data collection. To illustrate, consider the data collected by the AUV $_j$  from the CH at position  $\ell_j[\zeta]$ , which has an initial VoI of  $E_j[\zeta] = \sum_{m=1}^M A_j^m[\zeta] E_m$ . According to (3), when the AUV $_j$

moves to position  $\ell_j[\zeta + 1]$ , the VoI of the data collected at  $\ell_j[\zeta]$  is represented as:

$$V_j^\zeta(T_j^{\zeta+1}) = \beta E_j[\zeta] + (1 - \beta) E_j[\zeta] e^{-\frac{\Delta T_j^{\zeta \rightarrow \zeta+1}}{\alpha}}, \quad (4)$$

where  $T_j^{\zeta+1}$  denotes the time corresponding to the arrival of the AUV $_j$  at position  $\ell_j[\zeta + 1]$ .  $\Delta T_j^{\zeta \rightarrow \zeta+1}$  denotes the difference between the moments when the AUV $_j$  arrives at position  $\ell_j[\zeta]$  and position  $\ell_j[\zeta + 1]$  respectively, which includes the time of the optical communication between the AUV and the CH at position  $\ell_j[\zeta]$ , as well as the time of the movement of the AUV from position  $\ell_j[\zeta]$  to position  $\ell_j[\zeta + 1]$ . The amount of data collected by AUV $_j$  at step  $\zeta$  is denoted as  $I_j[\zeta] = \sum_{m=1}^M A_j^m[\zeta] I_m$ . Thus,  $\Delta T_j^{\zeta \rightarrow \zeta+1}$  is equal to:

$$\Delta T_j^{\zeta \rightarrow \zeta+1} = T_j^{\zeta+1} - T_j^\zeta = \frac{I_j[\zeta]}{R_o} + \frac{d_j[\zeta]}{v_a}, \quad (5)$$

where  $v_a$  denotes the movement speed of the AUV, and  $d_j[\zeta] = \|\ell_j[\zeta + 1] - \ell_j[\zeta]\|$  is the displacement distance of the AUV from position  $\ell_j[\zeta]$  to position  $\ell_j[\zeta + 1]$ . When data collection is complete, the total system VoI received at the surface station is given by:

$$V = \sum_{j=1}^U \sum_{\zeta=1}^{Z_j} \left( \beta E_j[\zeta] + (1 - \beta) E_j[\zeta] e^{-\frac{S_j[\zeta]}{\alpha}} \right), \quad (6)$$

where  $E_j[\zeta]$  denotes the VoI of the cluster visited by AUV $_j$  at step  $\zeta$ , and  $S_j[\zeta] = \sum_{i=\zeta}^{Z_j} \Delta T_j^{i \rightarrow i+1}$  is the remaining travel time until AUV $_j$  returns to the surface station.

The VoI of the data collected by AUVs at each CH depends on the time interval required for the AUVs to return to the surface station. This time interval is affected by the path planning of the AUVs, which is further determined by the selection of CHs in each cluster and the sequence in which the AUVs access these CHs. This paper aims to maximize the VoI of the UWSNs by jointly optimizing the selection of CHs  $H$  and the access sequence of CHs  $A$  for all AUVs. The VoI maximization problem is formulated as follows:

$$\mathbf{P}_0 : \max_{H, A} V(H, A), \quad (7a)$$

$$\text{s.t. } H_m \in C_m, \forall 1 \leq m \leq M, \quad (7b)$$

$$\mathcal{E}_m \leq \mathcal{E}_{\max}, \quad (7c)$$

$$(2a) \text{ and } (2b), \quad (7d)$$

where (7b) indicates the restrictions associated with the CHs, and (7c) denotes the energy constraints for each sensor node within a single round of data collection, where  $\mathcal{E}_{\max}$  represents the maximum allowable energy consumption. Since  $\mathbf{P}_0$  depends on the travel time and communication duration of the AUV with the CH, the cluster access sequence of the AUV needs to integrate considerations of CH coordinates, data volume within the cluster, and the initial VoI of the data.

#### IV. ENCODER-DECODER BASED DEEP REINFORCEMENT LEARNING ALGORITHM

Given that  $\mathbf{P}_0$  involves problems such as path planning and CH selection, it is a classic NP-hard problem that is

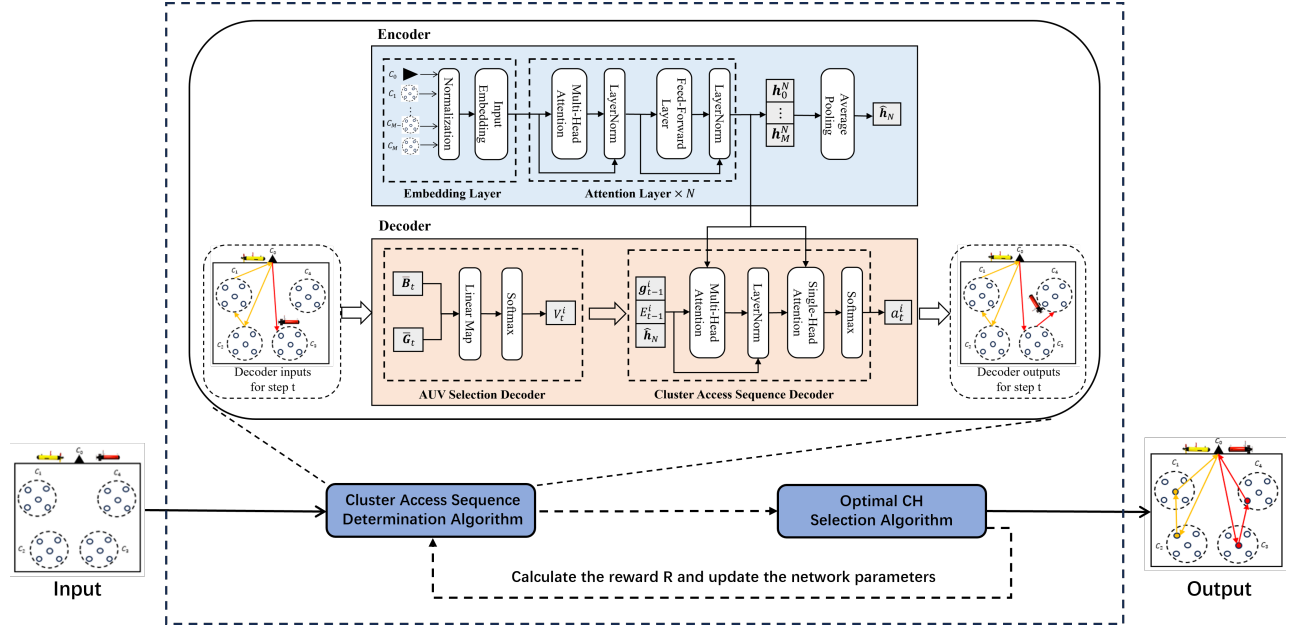


Fig. 2. The proposed algorithm architecture.

challenging to obtain the optimal solution within polynomial time due to its inherent complexity. Path planning problems are typically described as a series of actions from the starting point to the destination. Encoder-decoder architectures, especially Seq2Seq models, are well-suited to handle tasks where both the input and output are sequences. Therefore, we propose an encoder-decoder based deep reinforcement learning (EDDRL) algorithm. Specifically, the algorithm obtains the cluster access sequence of each AUV through a network based on an encoder-decoder architecture. Then, the CHs are selected based on the previously determined sequences through an optimal CH selection algorithm based on dynamic programming. Finally, the algorithm calculates the system rewards based on acquired VoI and updates the parameters of the network using the REINFORCE algorithm. The overall architecture of the algorithm is depicted in Fig. 2.

In the EDDRL algorithm, we model  $P_0$  as a Markov decision process (MDP). The MDP can be represented as a tuple of  $(S, A, R)$ , where  $S$ ,  $A$  and  $R$  denote the environment state space, the action space, and the reward space based on the state and the action, respectively. The environment state consists of the environment encoded by the encoder network and the clusters accessed by each AUV before the current step in the decoder. The action represents the operations in the decoder network at each step based on the current state. Therefore, we consider the cluster access order of each AUV output by the decoder network as the action  $\pi$ . The system VoI is used as the reward, and the policy network parameters are optimized through the gradient method, thereby enabling each AUV to learn a good cluster access sequence  $\pi$  by interacting with the environment.

#### A. Cluster Access Sequence Determination Algorithm

In this paper, the cluster access sequence determination of AUVs is viewed as a Seq2Seq mapping problem, where

the initial sequence contains multiple clusters including the surface station, and the target sequence is the cluster access sequence for each AUV. Notably, determining cluster access sequence of AUVs involves not only task allocation but also path planning. To effectively address the problem, we employ an encoder-decoder based network. Specifically, the encoder implements feature extraction of the input sequence based on a multi-head attention mechanism [32]. The decoder consists of an AUV selection decoder and a cluster access sequence decoder, which enables the network to flexibly allocate tasks among multiple AUVs based on the environmental state and avoid conflicts arising from different AUVs visiting the same cluster.

1) *Encoder*: The encoder is responsible for extracting features from the input data, which mainly contains an embedding layer and multiple encoder layers. Let  $I = \{I_0, I_1, \dots, I_M\}$  denote the input data, where  $I_0 = \ell_0 \in \mathbb{R}^3$  represents the starting point of AUVs, specifically the coordinates of the surface station, which naturally has a dimension of 3. For each cluster  $C_m$ , the input data  $I_m$  includes all the node coordinates, total data volume, and initial VoI. Suppose the number of sensor nodes in cluster  $C_m$  is  $N_m$ , the coordinates of the  $i$ th node in cluster are represented as  $\ell_i^m \in \mathbb{R}^3$ , the initial VoI of cluster is denoted as  $E_m \in \mathbb{R}$ , and the total data volume in cluster is denoted as  $D_m \in \mathbb{R}$ . Therefore, the feature dimension of the input data for the cluster is  $3N_m + 2$ . Given that the number of nodes in different clusters may vary in real environments, it is necessary to unify the dimensions of the input data for different clusters through padding operations, whereby the data is padded to the maximum value  $3N_{\max} + 2$ , where  $N_{\max} = \max_{1 \leq m \leq M} N_m$  is the maximum number of nodes in a cluster. Specifically, for any cluster  $C_m$ , the coordinates of  $N_{\max} - N_m$  nodes are randomly selected from the cluster to be repetitively padded. This approach avoids the introduction of additional data and thus the impact on

network learning. After data padding, the input data, denoted as  $I_m \in \mathbb{R}^{3N_{\max}+2}$ ,  $1 \leq m \leq M$ , is normalized using linear scaling. The normalization formula is as follows:

$$(\tilde{x}, \tilde{y}, \tilde{z}, \tilde{D}, \tilde{E}) = \left( \frac{x}{L_x}, \frac{y}{L_y}, \frac{z}{L_z}, \frac{D}{D_{\max}}, \frac{E}{E_{\max}} \right), \quad (8)$$

where  $x, y, z$  represent the three-dimensional coordinates of nodes, and  $L_x, L_y, L_z$  denote the length, width, and height of the three-dimensional space corresponding to the UWSNs.  $D_{\max}$  represents the maximum data volume sensed by all clusters, and  $E_{\max}$  represents the maximum initial VoI. Note that the input data of the normalized encoder is  $\tilde{I} = \{\tilde{I}_0, \tilde{I}_1, \dots, \tilde{I}_M\}$ , followed by input data embedding via linear projection as follows:

$$h_m^0 = \begin{cases} W_0 \tilde{I}_m + f_0, & m = 0, \\ W_1 \tilde{I}_m + f_1, & 1 \leq m \leq M, \end{cases} \quad (9)$$

where  $W_0 \in \mathbb{R}^{\dim \times 3}$ ,  $W_1 \in \mathbb{R}^{\dim \times (3N_{\max}+2)}$ ,  $f_0, f_1 \in \mathbb{R}^{\dim}$  are learnable parameters. Subsequently, the embedded data  $h_m^0$  ( $0 \leq m \leq M$ ) is fed into  $N$  attention layers for better feature representation. As shown in Fig. 2, each attention layer is the same as the one in the standard Transformer, consisting of a multi-head attention layer and a feed-forward layer, each of which performs residual concatenation and layer normalization. The multi-head attention layer realizes the weighted fusion of different cluster features through the attention mechanism, where multiple heads execute attention computations in parallel and merge the results to obtain cluster feature representations.

Assuming that the input of the  $n$ th attention layer is  $h_n = \{h_0^{n-1}, h_1^{n-1}, \dots, h_M^{n-1}\}$ , where  $h_m^{n-1}$  represents the feature representation of cluster  $C_m$  after going through the previous  $n-1$  attention layers. Then the operation of the  $n$ th attention layer can be represented as:

$$\begin{aligned} Y_n &= \text{MHA}(h_{n-1}, h_{n-1}, h_{n-1}), \\ \bar{Y}_n &= \text{LN}(h_{n-1} + Y_n), \\ h_n &= \text{LN}(\bar{Y}_n + \text{FFN}(\bar{Y}_n)), \end{aligned} \quad (10)$$

where  $\text{MHA}(\cdot)$  represents the multi-head attention calculation,  $\text{FFN}(\cdot)$  represents the feed-forward operation and  $\text{LN}(\cdot)$  represents the layer normalization operation.

After passing through  $N$  encoder layers, the final output  $h_N \in \mathbb{R}^{(M+1) \times \dim}$  of the encoder layers is obtained. Thus, the cluster feature representation for each cluster is denoted as  $h_m^N \in \mathbb{R}^{\dim}$  ( $0 \leq m \leq M$ ). The average of all cluster features, represented as  $\hat{h}_N = \frac{1}{M+1} \sum_{m=0}^M h_m^N$ , serves as the overall graph feature of the UWSN. This value will be used in the decoder stage to construct the context vector for each AUV.

2) *Decoder*: The decoder inputs the cluster feature vectors from the encoder and the state vectors of the AUV before decoding step  $t$ , and outputs the AUV that needs to be moved and the next cluster that the AUV needs to visit. To enable the network to flexibly allocate tasks among multiple AUVs, the decoder includes an AUV selection decoder and a cluster access sequence decoder. The decoding step is to first select an AUV to be moved by the AUV selection decoder, and then

the cluster access sequence decoder chooses the next cluster that the selected AUV needs to visit.

The AUV selection decoder leverages both the current features of each AUV and the historical route features to make informed decisions regarding which AUV to move next. Specifically, in the decoding phase at step  $t$ , the historical route features of AUV <sub>$i$</sub>  are denoted as:

$$\begin{aligned} G_t^i &= [g_0^i, g_1^i, \dots, g_{t-1}^i], \\ G_t &= [\max(G_t^1), \max(G_t^2), \dots, \max(G_t^U)], \\ \bar{G}_t &= \text{FFN}(W_2 G_t + f_2), \end{aligned} \quad (11)$$

where  $G_t^i \in \mathbb{R}^{t \times \dim}$  represents the arrangement of the cluster features visited by AUV <sub>$i$</sub>  before step  $t$ ,  $g_j^i \in \mathbb{R}^{\dim}$  represents the feature vector of the cluster visited by AUV <sub>$i$</sub>  at step  $j$ ,  $W_2$  and  $f_2$  are learnable parameters.  $\text{MAX}(\cdot)$  represents the maximum pooling operation. Similarly, the current features of AUVs can be constructed as follows:

$$\begin{aligned} B_t^i &= [g_{t-1}^i, E_{t-1}^i], \\ B_t &= [B_t^1, B_t^2, \dots, B_t^U], \\ \bar{B}_t &= \text{FFN}(W_3 B_t + f_3), \end{aligned} \quad (12)$$

where  $E_{t-1}^i$  represents the VoI collected by AUV before step  $t$ ,  $W_3$  and  $f_3$  are learnable parameters. Finally, the historical route features  $\bar{G}_t$  and the current features  $\bar{B}_t$  of AUVs are projected linearly and then passed through a softmax function to obtain the probability distribution  $P_t$  for the selection of each AUV.

$$P_t = \text{softmax}(W_4 [\bar{B}_t, \bar{G}_t] + f_4), \quad (13)$$

where  $W_4$  and  $f_4$  are learnable parameters. The AUV selection result  $V_t^i$  for step  $t$  can be determined based on the probability distribution  $P_t$ . Then the cluster accessed by the AUV at step  $t+1$  is determined by the cluster access sequence decoder. Firstly, the characteristics of the cluster in which the AUV is currently located, the VoI of the collected data and the encoding characteristics of the clusters need to be taken into account to construct the feature vector of AUV <sub>$i$</sub>  at step  $t$ , denoted as  $y_t^i$ . A multi-head attention mechanism is used to aggregate the attention-based features of AUV <sub>$i$</sub>  to all clusters, then the attention weight of AUV <sub>$i$</sub>  for selecting each cluster is computed through the single-head attention. The whole process can be expressed as:

$$\begin{aligned} y_t^i &= (W_5 [g_{t-1}^i, E_{t-1}^i] + f_5) + \hat{h}_N, \\ \bar{y}_t^i &= \text{LN}(y_t^i + \text{MHA}(y_t^i, h_N, h_N)), \\ u_{i,m}^t &= \begin{cases} \frac{Q_i(K_m)^T}{\sqrt{\dim_k}}, & b_m = 0, \\ -\infty, & b_m = 1, \end{cases} \end{aligned} \quad (14)$$

where  $Q_i = \bar{y}_t^i W_Q \in \mathbb{R}^{\dim}$ ,  $K_m = h_m^N W_K \in \mathbb{R}^{\dim}$  are query vectors and key vectors, respectively.  $W_5$ ,  $f_5$ ,  $W_Q$  and  $W_K$  are learnable parameters.  $b_m$  denotes whether the cluster  $C_m$  is visited by an AUV. The attention weights of AUV <sub>$i$</sub>  to the cluster  $C_m$  are defined as  $-\infty$  if  $C_m$  has already been visited by AUV in order to prevent the repeated visits. From this, the attention weight of AUV <sub>$i$</sub>  to all clusters is obtained, which is denoted as  $u_i^t = \{u_{i,1}^t, u_{i,2}^t, \dots, u_{i,M}^t\}$ , then the learning

process of  $u_{i,m}^t$  ( $1 \leq m \leq M$ ) is stabilized by the tanh function, and finally the probability of AUV<sub>*i*</sub> visiting each cluster at step  $t$  is calculated by the softmax function as:

$$p_{i,m}^t = \frac{e^{C \cdot \tanh(u_{i,j}^t)}}{\sum_{j=1}^M e^{C \cdot \tanh(u_{i,j}^t)}}, 1 \leq m \leq M, \quad (15)$$

where  $C = 10$  is a scaling coefficient. The next cluster to be visited is determined based on the probability distribution  $p_{i,m}^t$ , upon determining the next cluster to be visited, AUV<sub>*i*</sub> updates the environmental state. Subsequently, leveraging this updated state, it identifies the AUV scheduled to move in the  $(t+1)$ th step and the corresponding cluster for visitation. This decoding process iterates until all clusters have been traversed. Once the decoding process is complete, the cluster access sequence that results from it implicitly determines the number of clusters  $Z_j$  visited by each AUV. Given that the decoder is designed to assign each cluster to exactly one AUV without conflict, the cluster access sequence corresponding to each AUV varies according to the learned policy.

### B. Optimal CH Selection Algorithm

The optimal CH selection algorithm, grounded in dynamic programming, takes as input the AUV cluster access sequences and outputs the corresponding CHs associated with the clusters visited by each AUV. The objective of the algorithm is to derive a path that maximizes the VoI retained by the AUV.

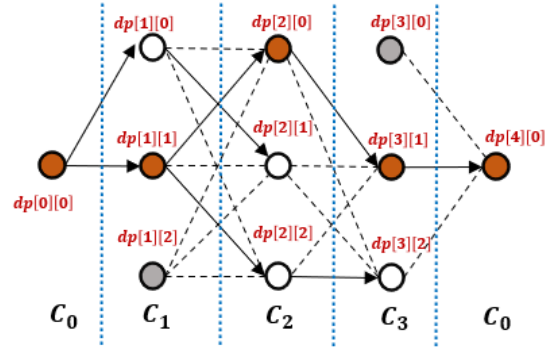
We first analyze the relationship between the VoI when an AUV accesses any adjacent clusters  $C_\zeta$  and  $C_{\zeta+1}$ . Assuming that cluster  $C_\zeta$  has an initial VoI of  $E_\zeta$ ,  $V_n[\zeta]$  denotes the VoI of data retained by the AUV when the AUV visits the  $n$ th sensor node of cluster  $C_\zeta$ . Then, when the AUV moves to the  $m$ th sensor node of cluster  $C_{\zeta+1}$  for data collection, the data VoI is denoted as  $V_{n \rightarrow m}[\zeta+1]$ . Thus,  $V_n[\zeta]$  and  $V_{n \rightarrow m}[\zeta+1]$  can be expressed as:

$$\begin{aligned} V_n[\zeta] &= \sum_{i=1}^{\zeta-1} \beta E_i + \sum_{i=1}^{\zeta-1} (1-\beta) E_i e^{-\frac{\Delta T_{i \rightarrow \zeta}^n}{\alpha}}, \\ V_{n \rightarrow m}[\zeta+1] &= \sum_{i=1}^{\zeta} \beta E_i + \sum_{i=1}^{\zeta} (1-\beta) E_i e^{-\frac{(\Delta T_{i \rightarrow \zeta}^n + \Delta T_{\zeta \rightarrow m}^n)}{\alpha}}, \end{aligned} \quad (16)$$

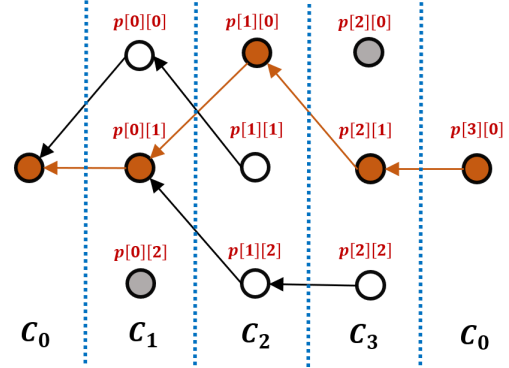
where  $\Delta T_{i \rightarrow \zeta}^n$  denotes the time difference between when AUV starts collecting data from cluster  $C_i$  and reaches the  $n$ th sensor node of cluster  $C_\zeta$ . It is evident that selecting different CHs will make  $\Delta T_{i \rightarrow \zeta}^n$  different, subsequently influencing  $V_n[\zeta]$ ,  $\Delta T_{\zeta \rightarrow m}^n$  denotes the time required for the AUV to collect data from the  $n$ th sensor node in cluster  $C_\zeta$  and move to the  $m$ th sensor node in cluster  $C_{\zeta+1}$ . The first term in  $V_n[\zeta]$  and  $V_{n \rightarrow m}[\zeta+1]$  remains constant over time, while the second term diminishes over time and is affected by the CH selection. We define  $V_{n,\text{var}}[\zeta]$  as decaying VoI of  $V_n[\zeta]$ , which can be expressed as:

$$V_{n,\text{var}}[\zeta] = \sum_{i=1}^{\zeta-1} E_i e^{-\frac{\Delta T_{i \rightarrow \zeta}^n}{\alpha}}. \quad (17)$$

Therefore, during the selection of the CH, it is only necessary to ensure the maximization of decaying VoI to



(a) Dynamic programming process



(b) CH selection process

Fig. 3. Example of optimal CH selection algorithm based on dynamic programming.

guarantee the maximization of the final retained VoI. Similarly,  $V_{n \rightarrow m,\text{var}}[\zeta+1]$  can be defined as decaying VoI of  $V_{n \rightarrow m}[\zeta+1]$ , and its state transition relationship with  $V_{n,\text{var}}[\zeta]$  can be defined as follows according to (16) and (17):

$$V_{n \rightarrow m,\text{var}}[\zeta+1] = e^{-\frac{\Delta T_{\zeta \rightarrow m}^n}{\alpha}} (V_{n,\text{var}}[\zeta] + E_\zeta). \quad (18)$$

It is evident that when sensor nodes  $m$  and  $n$  are determined, maximizing  $V_{n \rightarrow m,\text{var}}[\zeta+1]$  requires maximizing  $V_{n,\text{var}}[\zeta]$ . We define  $V_{n \rightarrow m,\text{var}}^*[\zeta+1]$ ,  $V_{n,\text{var}}^*[\zeta]$  are the maximum values corresponding to  $V_{n \rightarrow m,\text{var}}[\zeta+1]$ ,  $V_{n,\text{var}}[\zeta]$  respectively. Furthermore defining  $V_{m,\text{var}}^*[\zeta+1]$  as the maximum value of the decay over time portion of the data VoI retained by the AUV when it reaches the  $m$ th sensor node of cluster  $C_{\zeta+1}$ , which is expressed as:

$$V_{m,\text{var}}^*[\zeta+1] = \max_{n \in C_\zeta} e^{-\frac{\Delta T_{\zeta \rightarrow m}^n}{\alpha}} (V_{n,\text{var}}^*[\zeta] + E_\zeta). \quad (19)$$

The equation above represents the state transition equation for the decayed VoI of the AUV between cluster  $C_\zeta$  and cluster  $C_{\zeta+1}$ . Since the VoI remains fixed without time decay and is independent of CH selection, the objective of CH selection is to maximize  $V_{\text{var}}^*[z_i+1]$ , which represents the maximum data VoI when the AUV returns to the surface station. Based on (19), the optimal solution  $V_{\text{var}}^*[z_i+1]$  can be obtained through dynamic programming. We define array  $dp[\zeta][n] = V_{n,\text{var}}^*[\zeta]$  to represent the maximum retained decayed VoI when the AUV visits the  $n$ th sensor node in cluster  $C_\zeta$  for data collection. To obtain the selected CH for each cluster, we

---

**Algorithm 1** Optimal CH Selection Algorithm Based on Dynamic Programming
 

---

```

1: Initialize the array  $dp \in \mathbb{R}^{(z_i+2) \times N_{\max}}$ ,  $p \in \mathbb{R}^{(z_i+1) \times N_{\max}}$ ,  $\zeta = 0$ 
2: while  $\zeta \leq z_i$  do
3:   for  $m$  in  $C_{\zeta+1}$  do
4:     if  $x_{\zeta+1,n} = 1$  then
5:       Update  $dp[\zeta+1][m]$  and  $p[\zeta][m]$  according to (20)
6:     else
7:        $dp[\zeta+1][m] = 0$ 
8:        $p[\zeta][m] = -1$ 
9:     end if
10:  end for
11:   $\zeta \leftarrow \zeta + 1$ 
12: end while
13: Initialize  $pre = p[z_i][0]$ , CH set  $H = \{pre\}$ 
14: while  $z_i > 1$  do
15:    $pre = p[z_i - 1][pre]$ 
16:   Insert  $pre$  in the first place of  $H$ 
17:    $z_i \leftarrow z_i - 1$ 
18: end while

```

---

define an array  $p[\zeta][m]$  to store the sensor node in the previous cluster corresponding to  $dp[\zeta+1][m]$ . In this case, we have the equations as follows:

$$dp[\zeta+1][m] = \max_{n \in C_\zeta} e^{-\frac{\Delta T_\zeta^{n \rightarrow m}}{\alpha}} (dp[\zeta][n] + E_\zeta), \quad (20)$$

$$p[\zeta][m] = \arg \max_{n \in C_\zeta} e^{-\frac{\Delta T_\zeta^{n \rightarrow m}}{\alpha}} (dp[\zeta][n] + E_\zeta).$$

Additionally, the selected CH needs to ensure that the energy consumption of the nodes within the cluster satisfies constraint (1). For any node in cluster  $C_\zeta$  serving as the CH, we define binary variable  $x_{\zeta,n} \in \{0, 1\}$  to indicate whether the energy consumption of the nodes within the cluster satisfies the constraint when node  $n$  serves as the CH. In the case of  $x_{\zeta,n} = 0$ , we set  $dp[\zeta][n] = 0, p[\zeta-1][n] = -1$ , which implies that the node  $n$  cannot serve as the CH, when  $x_{\zeta,n} = 1$ , they will be updated as (20). As shown in Fig. 3a, the initial state of  $dp[0][0]$  is 0. The gray nodes indicate that the nodes are unable to serve as the CHs due to not satisfying the energy consumption constraint. Based on (20), the arrays  $dp$  and  $p$  can be updated to determine the CH of each cluster represented by the orange nodes in Fig. 3b. The optimal CH selection algorithm is detailed in Algorithm 1.

### C. Model Training Based on REINFORCE

The parameters of the encoder-decoder based network need to be trained to maximize the system VoI. In this paper, we consider training the network parameters using a Monte Carlo-based REINFORCE algorithm. Given an instance  $s$  of the UWSN, the network parameters are denoted by  $\theta$ , and the network outputs the result of the cluster access sequence for each AUV, which is represented as  $\pi$ . Let  $P_\theta(\pi | s)$  denote the probability distribution of network output actions. According to the REINFORCE algorithm, the gradient of our objective  $\mathcal{L}(\theta)$  is expressed as:

$$\nabla_\theta \mathcal{L}(\theta) = \frac{1}{B} \sum_{i=0}^B [(V(\pi_i | s_i) - V_{bl}(s_i)) \nabla_\theta \log P_\theta(\pi | s_i)], \quad (21)$$

where  $V_{bl}(s_i)$  is the reward obtained from the baseline network, which is used to estimate the average performance of the encoder-decoder based network in determining the AUV cluster access sequences. Before training, the network model parameters  $\theta$  and the baseline model parameters  $\theta_{bl}$  are identical. After training, the parameters of the network model are updated, and if the improvement of  $\theta$  is significant according to a paired t-test (5%), it is copied to  $\theta_{bl}$ . The REINFORCE algorithm is given in Algorithm 2.

---

**Algorithm 2** Network Parameters Training Based on REINFORCE
 

---

```

1: Initialize the strategy network parameters  $\theta$ , baseline model parameters  $\theta_{bl} \leftarrow \theta$ 
2: for epoch = 1, 2, ...,  $N_{\text{epoch}}$  do
3:   for batch = 1, 2, ...,  $N_{\text{batch}}$  do
4:     Obtain batch training data  $D_B$ 
5:     The encoder network computes the cluster features  $h_N$  and  $\hat{h}_N$  for each instance in  $D_B$ 
6:     The decoder network computes the cluster access sequence of AUVs for each instance in  $D_B$ 
7:     Calculate the CH selection results by Algorithm 1
8:     Solve each instance of  $D_B$  via the baseline model and Algorithm 1
9:     Calculate rewards  $V$  and  $V_{bl}$  by (6)
10:    Calculate  $\nabla_\theta \mathcal{L}(\theta)$  by (21)
11:     $\theta \leftarrow \text{Adam}(\theta, \nabla_\theta \mathcal{L}(\theta))$ 
12:  end for
13:  if t-test ( $P_\theta, P_{\theta_{bl}}$ ) < 5% then
14:     $\theta_{bl} \leftarrow \theta$ 
15:  end if
16: end for

```

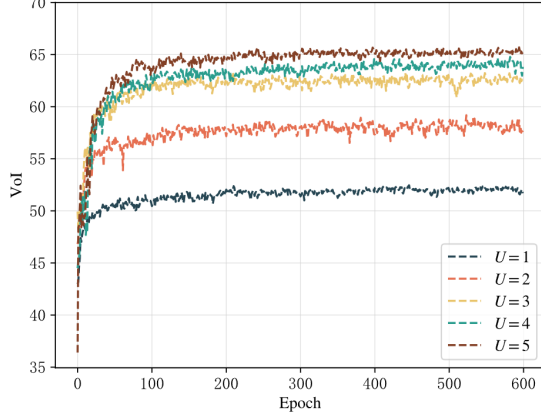
---

### D. Time Complexity Analysis

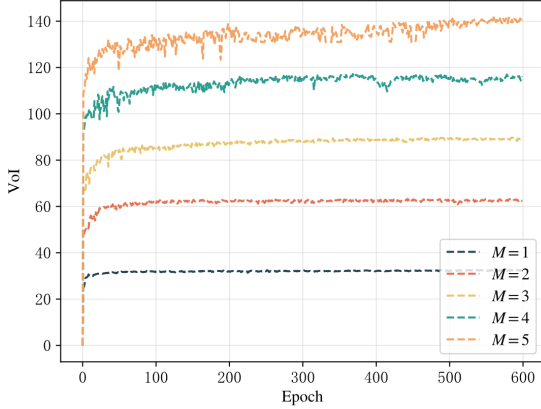
The EDDRL algorithm proposed in this paper consists of two parts. For the cluster access sequence determination algorithm, the time complexity of each attention layer in the encoder depends on the computation of the multi-head attention layer. The standard multi-head attention time complexity is  $\mathcal{O}((M+1)^2 d)$ , where  $d$  is the dimension of the feature vector, and  $M$  is the number of clusters. Similarly, during the decoding stage, each decoding process requires one computation of multi-head attention and one computation of single-head attention. The time complexity of attention computation from AUV<sub>*i*</sub> to all clusters is  $\mathcal{O}((M+1)d)$ , thus the approximate time complexity of a single decoding process is  $\mathcal{O}((M+1)d)$ . Since the decoder needs to decode  $M$  times before all clusters are visited, the time complexity of the decoder is  $\mathcal{O}(M(M+1)d)$ . Consequently, the overall time complexity of the cluster access sequence determination algorithm is  $\mathcal{O}((M+1)^2 d)$ . For the optimal CH selection algorithm, given the cluster access sequence of AUV<sub>*i*</sub>, each sensor node in the cluster needs to undergo state transition by (20), with a time complexity of  $\mathcal{O}(N^2(z_i+1))$ , where  $N$  is the number of sensor nodes in each cluster. Since it is necessary to determine the CHs based on the cluster access sequence of all AUVs, the algorithm's time complexity can be calculated as the sum of the time complexities of all AUVs, denoted as  $\mathcal{O}(\sum_{i=1}^U N^2(z_i+1)) = \mathcal{O}(N^2(M+U))$ . Considering that the number of AUVs is generally much smaller than

TABLE I  
SIMULATION PARAMETERS

| Parameters                               | Value                   | Parameters                         | Value   |
|------------------------------------------|-------------------------|------------------------------------|---------|
| Scale of network                         | 1000 m × 1000 m × 100 m | Sensor node transmit power         | 300mW   |
| Number of clusters                       | 20                      | Sensor node receive power          | 100mW   |
| Number of nodes in each cluster          | 5 – 10                  | Number of encoder attention layers | 3       |
| Initial VoI of data                      | 1 – 7                   | Heads of attention                 | 8       |
| Node data volume                         | 40 – 50 kb              | Learning rate                      | 0.0001  |
| Weight factor $\beta$                    | 0.5                     | Feature vector dimension           | 128     |
| Decay factor $\alpha$                    | 200                     | Learning rate decay                | 0.9995  |
| Speed of AUV                             | 5 m/s                   | Communication rate                 | 20 Mb/s |
| Sensor node energy consumption threshold | 1.2J                    | Training batch size                | 32      |



(a) AUV number



(b) Cluster number

Fig. 4. The convergence of the proposed algorithm under different parameters.

the number of clusters, the time complexity of the algorithm is approximately  $\mathcal{O}(N^2M)$ . Additionally, before selecting the  $x_{\zeta,n}$ , it is necessary to calculate the binary variable  $x_{\zeta,n}$  to determine whether the energy consumption satisfies the constraints. This process requires evaluating the energy consumption situation of each node in the cluster when acting as a CH, with a time complexity of  $\mathcal{O}(N^2M)$ . Thus, the final time complexity of the optimal CH selection algorithm is  $\mathcal{O}(N^2M)$ . Consequently, the time complexity of the EDDRL algorithm is  $\mathcal{O}((M+1)^2d) + \mathcal{O}(N^2M)$ .

## V. PERFORMANCE EVALUATION

### A. Simulation Setup

In the simulation, we consider deploying multiple AUVs with a speed of 5 m/s for data collection in an underwater environment of 1000 m × 1000 m × 100 m, twenty clusters are deployed within, each containing 5 to 10 sensor nodes that are randomly deployed. The initial VoI for data in each cluster is randomly generated between [1, 7], with  $\beta$  set to 0.5 and decay factor  $\alpha$  set to 200. The amount of data perceived by sensor nodes in each round is randomly generated between 40 – 50 kb. The transmission power and reception power of sensor nodes are set to 300mW and 100mW respectively. The maximum energy consumption constraint for nodes during each round is set to 1.2J. During model training, the encoder has 3 attention layers, with 8 heads in the attention, and the feature vector embedding dimension is equal to 128. In addition, the network parameters are trained using the Adam optimizer with an initial learning rate of 0.0001 and a decay of 0.9995 per iteration for convergence. To minimize the random effects of simulation parameters, the results of all algorithms were averaged from 20 experiment repetitions. The detailed parameters are summarized in Table I.

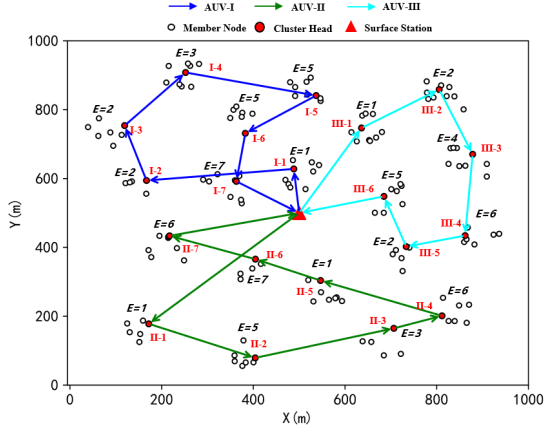
### B. Benchmark Schemes

For comparison, three benchmark schemes are considered:

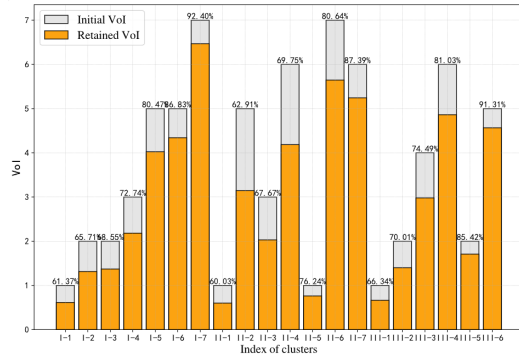
- **LG-DQN [12]:** The node closest to the center of the cluster is selected as the CH, and then each AUV path to access the CH is planned by the DQN algorithm, which sets the reward of reinforcement learning to the total VoI.
- **GAAP [10]:** A greedy strategy is used to select subsequent clusters for each AUV based on the decay of VoI, then the AUV navigates to the nearest node within the selected cluster that satisfies the constraints.
- **SPDC-GA [9]:** Each AUV is responsible for collecting data in a specific region, and a genetic algorithm is used to optimize their path planning and CH selection.

### C. Validity Analysis

We first focus on the convergence of multi-AUV path planning networks with different sensor network parameters. Fig. 4 demonstrates that the proposed algorithm achieves stable convergence. As depicted in Fig. 4a, as the number of AUVs increases from 1 to 5, the convergence of the network



(a) Path planning result

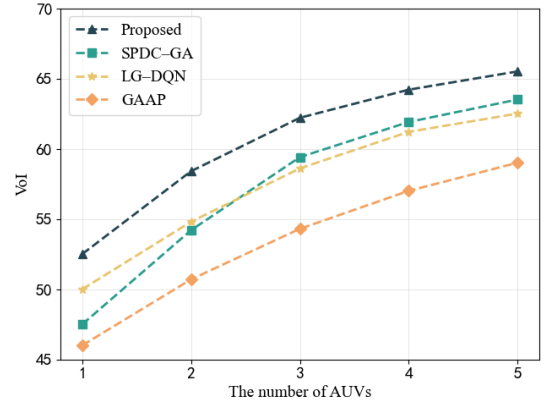


(b) Retained VoI

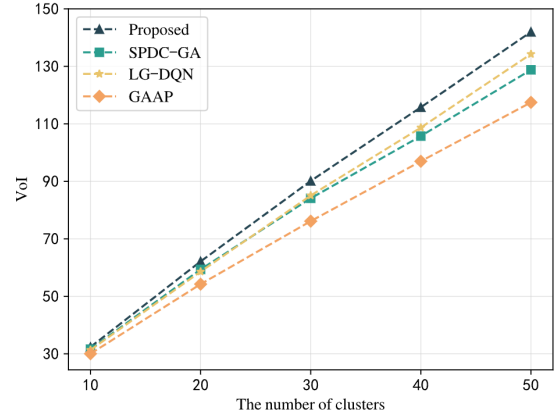
Fig. 5. Example AUV path planning in the 2D plane with the percentage of VoI retained with 3 AUVs.

decelerates. This is because more exploration is required to fully learn the task allocation among AUVs and the path planning. Meanwhile, the VoI retained by AUVs amplifies with the increase in the number of AUVs. This trend stems from the reduction in the number of clusters visited by each AUV, resulting in decreased average distances traveled by each AUV. Fig. 4b shows the variation of system VoI with epoch when the number of clusters increases from 10 to 50. It can be seen that as the number of clusters increases, the VoI retained by the AUVs increases, while the network convergence speed decreases as more exploration is required to learn the path planning of the AUVs.

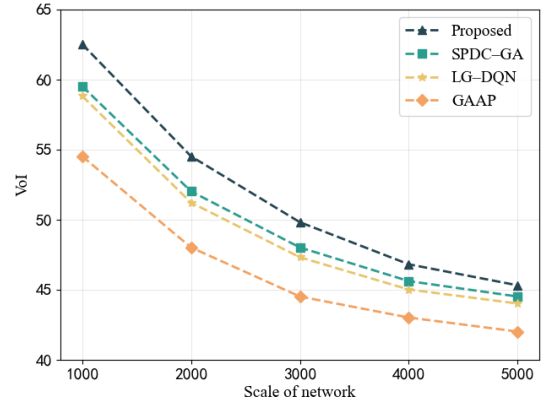
Fig. 5 displays an example of the AUV movement paths planned by our algorithm and the percentage of VoI retained for each cluster in a sensor network scenario. As shown in Fig. 5a, the planned paths for three AUVs are illustrated. In this figure, the clusters are randomly distributed in a square area labeled next to the initial VoI value and the cluster access sequence. It is shown that each AUV tends to visit a similar number of clusters, and the clusters with higher data importance tend to be visited later. The trade-off between the data importance and timeliness is also shown. Additionally, our proposed algorithm guarantees that each cluster complies with the energy constraints after determining AUV cluster access sequences. The selected CH effectively shortens the travel distance of the AUVs, reducing the loss of VoI. Fig.



(a) AUV number



(b) Cluster number



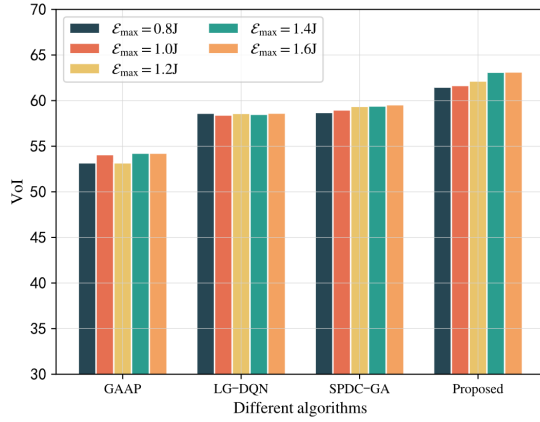
(c) Network scale

Fig. 6. Comparison of collected VoI by different algorithms.

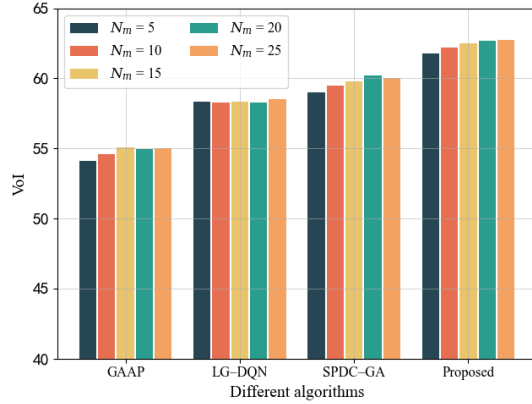
5b shows the retained VoI of each cluster along the AUV movement path. All clusters preserve more than 50% of VoI upon the AUV returning to the surface station. Furthermore, for the last few clusters accessed, each one retains more than 80% of VoI. This supports accessing clusters with higher data importance later in the sequence.

#### D. VoI under Different Strategies

We further evaluate the impact of different parameters on the system VoI. As depicted in Fig. 6a, the VoI retained by



(a) Node energy threshold



(b) Cluster size

Fig. 7. Impact of node energy threshold and cluster size on VoI across different algorithms.

each algorithm increases with the number of AUVs. Notably, our algorithm consistently outperforms others, yielding the highest data VoI. This is due to the efficient learning of AUV cluster access sequences and the selection of optimal CHs in accordance with the sequences. Furthermore, it is evident that the VoI gain retained by each algorithm decreases consistently as the number of AUVs increases. Fig. 6b shows the relationship between the VoI retained by each algorithm for different numbers of clusters. As the number of clusters increases, the VoI collected by each algorithm exhibits an increasing trend. Our algorithm consistently achieves the highest data VoI across different cluster numbers. This is due to the fact that our algorithm utilizes the optimal result of the sampling as the final output, which improves the overall data VoI. As demonstrated in Fig. 6c, the VoI retained by each algorithm exhibits a decreasing trend as the network scale increases. The proposed algorithm consistently achieves the highest data VoI across all examined scales due to its optimal CH selection strategy, which mitigates the loss of VoI in larger networks. In addition, the rate of VoI decline is steeper when the network is small, emphasizing the necessity of carefully delimiting the coverage area of each surface station in practical deployments to meet the desired VoI.

Fig. 7 shows the impact of node energy threshold and cluster

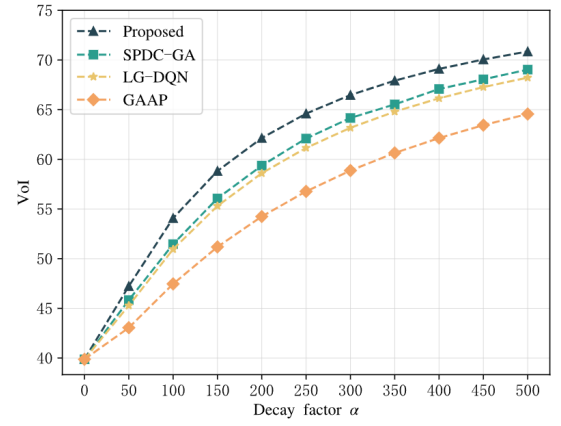
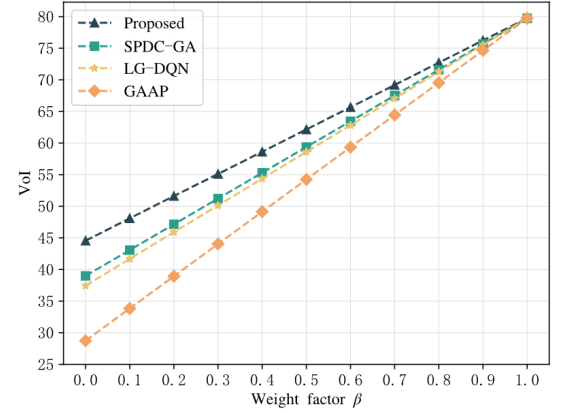
(a) Variation of VoI with  $\alpha$ (b) Variation of VoI with  $\beta$ 

Fig. 8. Relationship between the variation of VoI retained by different algorithms with decay factor  $\alpha$  and weight factor  $\beta$ .

size on VoI across different algorithms, with the number of AUVs set to 3. As depicted in Fig. 7a, the VoI collected by other algorithms fluctuates with the increase of the node energy threshold, while the VoI collected by our algorithm consistently increases with the increase of the node energy threshold, thus demonstrating the validity of our approach. Fig. 7b demonstrates the VoI of the proposed algorithm increases with the cluster size, further validating the effectiveness of the optimal CH selection algorithm based on dynamic programming. Furthermore, the LG-DQN algorithm selects nodes closest to the cluster center as CHs, ensuring that the increasing number of nodes within a cluster exerts minimal influence on the VoI, which remains relatively constant despite variations in cluster size. However, the VoI retained by the GAAP algorithm and SPDC-GA algorithm demonstrates variability in response to alterations in cluster size.

The decay factor  $\alpha$  significantly influences the rate at which VoI diminishes over time. As depicted in Fig. 8a, a larger decay factor leads to a slower VoI decay over time, resulting in the system retaining a higher VoI. Our algorithm consistently achieves higher VoI compared to other algorithms, as evidenced by the figure. Fig. 8b illustrates the relationship between VoI and the weight factor  $\beta$ , which reflects the balance between data importance and timeliness. A

higher weight factor emphasizes the initial importance of data, thereby reducing the impact of time decay on VoI. The figure demonstrates a linear growth in VoI across all algorithms. Notably, when the weight factor is smaller, VoI is more susceptible to time decay, leading to pronounced differences in VoI attainment among algorithms. Our algorithm stands out and achieves VoI enhancements of 14.26%, 18.96%, and 55.06% compared to the SPDC-GA, LG-DQN, and GAAP algorithms, respectively.

## VI. CONCLUSION

In this paper, we investigated the issue of data collection in UWSNs assisted by multiple AUVs. We formulated this problem as a challenging combinatorial optimization problem, aiming to jointly optimizing CH selection and multi-AUV path planning to maximize the system VoI. As the optimization problem is NP-hard, we proposed the EDDRL algorithm to solve it. Specifically, we obtain the cluster access sequence for each AUV through an encoder-decoder based network, after which a dynamic programming method is employed to obtain the optimal CH selection. The network parameters are updated through the REINFORCE algorithm. In the simulation, the effectiveness of the proposed algorithm is validated by network convergence and the results of the solution for an UWSN instance. The performance of EDDRL has been compared to that of SPDC-GA, LG-DQN, and GAAP algorithms. The simulation results demonstrate that our algorithm achieves a higher performance, with VoI improvements of 14.26%, 18.96%, and 55.06%.

## REFERENCES

- [1] T. Qiu, Z. Zhao, T. Zhang, C. Chen, and C. L. P. Chen, "Underwater internet of things in smart ocean: System architecture and open issues," *IEEE Trans. Ind. Informat.*, vol. 16, pp. 4297–4307, Oct. 2020.
- [2] X. Wei, H. Guo, X. Wang, X. Wang, and M. Qiu, "Reliable data collection techniques in underwater wireless sensor networks: A survey," *IEEE Commun. Surveys & Tutorials*, vol. 24, pp. 404–431, Dec. 2022.
- [3] S. Song, J. Liu, J. Guo, B. Lin, Q. Ye, and J. Cui, "Efficient data collection scheme for multi-modal underwater sensor networks based on deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 72, pp. 6558–6570, Dec. 2023.
- [4] L. Liu, Y. Liu, and N. Zhang, "A complex network approach to topology control problem in underwater acoustic sensor networks," *IEEE Trans. Parallel & Distributed Syst.*, vol. 25, pp. 3046–3055, Jan. 2014.
- [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 Things J.*, vol. 7, pp. 10010–10022, Apr. 2020.
- [6] 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 Trans. Veh. Technol.*, vol. 70, pp. 9294–9305, Jul. 2021.
- [7] M. Xi, J. Yang, J. Wen, H. Liu, Y. Li, and H. H. Song, "Comprehensive ocean information-enabled AUV path planning via reinforcement learning," *IEEE Internet Things J.*, vol. 9, pp. 17440–17451, Mar. 2022.
- [8] M. Huang, K. Zhang, Z. Zeng, T. Wang, and Y. Liu, "An AUV-assisted data gathering scheme based on clustering and matrix completion for smart ocean," *IEEE Internet Things J.*, vol. 7, pp. 9904–9918, Apr. 2020.
- [9] Y. He, G. Han, Z. Tang, M. Martínez-García, and Y. Peng, "State prediction-based data collection algorithm in underwater acoustic sensor networks," *IEEE Trans. Wireless Commun.*, vol. 21, pp. 2830–2842, Oct. 2021.
- [10] P. Gjanci, C. Petrioli, S. Basagni, C. A. Phillips, L. Bölöni, and D. Turgut, "Path finding for maximum value of information in multi-modal underwater wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, pp. 404–418, May 2017.
- [11] R. Duan, J. Du, C. Jiang, and Y. Ren, "Value-based hierarchical information collection for AUV-enabled internet of underwater things," *IEEE Internet Things J.*, vol. 7, pp. 9870–9883, May 2020.
- [12] J. Wang, S. Liu, W. Shi, G. Han, and S. Yan, "A multi-AUV collaborative ocean data collection method based on LG-DQN and data value," *IEEE Internet Things J.*, vol. 5, pp. 9086–9106, Oct. 2024.
- [13] G. Han, S. Shen, H. Song, T. Yang, and W. Zhang, "A statistical-learning-based data collection scheme in underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 67, pp. 10671–10682, Aug. 2018.
- [14] S. M. Ghoreyshi, A. Shahrabi, T. Boutaleb, and M. Khalily, "Mobile data gathering with hop-constrained clustering in underwater sensor networks," *IEEE Access*, vol. 7, pp. 21118–21132, Feb. 2019.
- [15] S. Wang, T. L. N. Nguyen, and Y. Shin, "Data collection strategy for magnetic induction based monitoring in underwater sensor networks," *IEEE Access*, vol. 6, pp. 43644–43653, Aug. 2018.
- [16] M. T. R. Khan, S. H. Ahmed, and D. Kim, "AUV-aided energy-efficient clustering in the internet of underwater things," *IEEE Trans. Green Commun. Netw.*, vol. 3, pp. 1132–1141, Jun. 2019.
- [17] C. Qin, J. Du, J. Wang, and Y. Ren, "A hierarchical information acquisition system for AUV assisted internet of underwater things," *IEEE Access*, vol. 8, pp. 176089–176100, Sep. 2020.
- [18] G. Xing, Y. Chen, R. Hou, M. Dong, D. Zeng, J. Luo, and M. Ma, "Game-theory-based clustering scheme for energy balancing in underwater acoustic sensor networks," *IEEE Internet Things J.*, vol. 8, pp. 9005–9013, Feb. 2021.
- [19] X. Guang, C. Liu, W. Qu, and T. Qiu, "A joint optimized data collection algorithm based on dynamic cluster-head selection and value of information in UWSNs," *Vehicular Commun.*, vol. 38, p. 100530, Dec. 2022.
- [20] M. T. R. Khan, Y. Z. Jembre, S. H. Ahmed, J. Seo, and D. Kim, "Data freshness based AUV path planning for UWSN in the internet of underwater things," in *Proc. IEEE GLOBECOM*, Dec. 2019, pp. 1–6.
- [21] G. Han, X. Long, C. Zhu, M. Guizani, Y. Bi, and W. Zhang, "An AUV location prediction-based data collection scheme for underwater wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 68, pp. 6037–6049, Apr. 2019.
- [22] G. Han, X. Long, C. Zhu, M. Guizani, and W. Zhang, "A high-availability data collection scheme based on multi-AUVs for underwater sensor networks," *IEEE Trans. Mobile Comput.*, vol. 19, pp. 1010–1022, Mar. 2020.
- [23] G. Han, H. Wang, S. Li, J. Jiang, and W. Zhang, "Probabilistic neighborhood location-point covering set-based data collection algorithm with obstacle avoidance for three-dimensional underwater acoustic sensor networks," *IEEE Access*, vol. 5, pp. 24785–24796, Oct. 2017.
- [24] G. Han, S. Shen, H. Wang, J. Jiang, and M. Guizani, "Prediction-based delay optimization data collection algorithm for underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 68, pp. 6926–6936, May 2019.
- [25] T. Wang, D. Zhao, S. Cai, W. Jia, and A. Liu, "Bidirectional prediction-based underwater data collection protocol for end-edge-cloud orchestrated system," *IEEE Trans. Ind. Informat.*, vol. 16, pp. 4791–4799, Sep. 2020.
- [26] M. Cheng, Q. Guan, F. Ji, J. Cheng, and Y. Chen, "Dynamic-detection-based trajectory planning for autonomous underwater vehicle to collect data from underwater sensors," *IEEE Internet Things J.*, vol. 9, pp. 13168–13178, Jan. 2022.
- [27] P. Yao, Z. Zhao, and Q. Zhu, "Path planning for autonomous underwater vehicles with simultaneous arrival in ocean environment," *IEEE Syst. J.*, vol. 14, pp. 3185–3193, Sep. 2020.
- [28] J. Wen, J. Yang, W. Wei, and Z. Lv, "Intelligent multi-AUG ocean data collection scheme in maritime wireless communication network," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, pp. 3067–3079, Apr. 2022.
- [29] Z. Wang, J. Du, C. Jiang, Z. Xia, C. Xu, and Y. Ren, "Multi-AUV task scheduling for target hunting and exploration: An AoI-aware DMAPPO approach," in *Proc. IEEE WCNC*, Mar. 2023, pp. 1–6.
- [30] B. Jiang, J. Du, C. Jiang, Z. Han, and M. Debbah, "Underwater searching and multiround data collection via AUV swarms: An energy-efficient AoI-aware MAPPO approach," *IEEE Internet Things J.*, vol. 11, pp. 12768–12782, Nov. 2024.
- [31] X. Hou, J. Wang, T. Bai, Y. Deng, Y. Ren, and L. Hanzo, "Environment-aware AUV trajectory design and resource management for multi-tier underwater computing," *IEEE J. Sel. Areas Commun.*, vol. 41, pp. 474–490, Dec. 2023.
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Jun. 2017, p. 6000–6010.



**Jing Xu** received the B.E. degree in telecommunication engineering and the Ph.D. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2001 and 2011, respectively. He is currently an Associate Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include wireless networks, intelligent sensing and unmanned system, with an emphasis on performance optimization, game theory, and reinforcement

learning and their application in networked systems.



**Jiarun Tang** received the B.E. degree in electronic information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2023, where he is pursuing the master's degree. His research interests include wireless power transfer, unmanned systems, intelligent sensing, and reinforcement learning, with a particular emphasis on its applications in networked systems.



**Xiao Huang** received the B.E. degree in Automation and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2009 and 2018, respectively, where he is currently a senior engineer at China Ship Development and Design Center. His research interests include image processing, pattern recognition and ship intelligence.



**Xuan Gu** received the B.E. degree in electronic information engineering and the M.E. degree in information and communication engineering from Huazhong University of Science and Technology, Wuhan, China, in 2021 and 2024, respectively. His research interests include unmanned systems, intelligent sensing, and reinforcement learning and its applications in networked systems.



**Lanhua Li** is currently a Research Assistant Professor with the School of Intelligent Systems Engineering, Sun Yat-Sen University. She received the B.E. degree and M.E. degree from Hunan Normal University, Changsha, China, in 2013 and 2016, respectively, and the Ph.D. degree in computer applications technology from the University of Chinese Academy of Sciences, ShenZhen, China, in 2021. She was a Postdoctoral Fellow at Sun Yat-Sen University in 2021-2023 and also a Visiting Postdoctoral Fellow at The Hong Kong Polytechnic University (PolyU) in 2022. Her research interests include wireless resource optimization, passive sensing networks, intelligent reflecting surface, backscatter communications, edge AI, UAV-enabled communication and sensing, and semantic communications.



**Jianhua He** received the Ph.D. degree from Nanyang Technological University, Singapore, in 2002. He worked at the University of Bristol, Bristol, U.K., Swansea University, Swansea, U.K., and Aston University, Birmingham, U.K. He is currently a Reader with the University of Essex, Colchester, U.K. He has published more than 150 research papers in refereed international journals and conferences. His main research interests include 5G/6G wireless communications and networks, connected vehicles, autonomous driving, the Internet of Things, mobile edge computing, intelligent transport systems, data analytics, AI, and machine learning. Dr. He was a Steering Committee Member of MobiArch 2021 and a member of the editorial board for several international journals. He is also a Coordinator of EU Horizon 2020 projects COSAFE and VESAFE on cooperative connected autonomous vehicles. He was the Workshop Chair of MobiArch 2020 and ICAV 2021.



**Wei Liu** received the B.E. degree in telecommunication engineering and the Ph.D. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2004, respectively, where he is currently a Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include wireless networking, network measurement, and learning evaluation.