Advanced Framework for Underwater Node Repair via Multi-AUV Based on Multi-Agent Offline Reinforcement Learning

Yimian Ding[*,+], Jingzehua Xu[†,+], Guanwen Xie[*], Gang Li[*], Jingjing Wang[‡], Yong Ren[§]
[*]Ocean College, Zhejiang University, Zhoushan, 316000, China
[†]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518055, China
[‡]School of Cyber Science and Technology, Beihang University, Beijing, 100191, China
[§]Department of Electronic Engineering, Tsinghua University, Beijing, 100083, China

***Abstract*—The Internet of Underwater Things (IoUT) is playing an increasingly important role in various applications such as ocean observation. However, the sensor nodes in IoUT are susceptible to corrosion by seawater over time, leading to eventual damage. Based on above analysis, we propose an advanced framework for underwater node repair utilizing multi-agent offline reinforcement learning (RL). This framework enables path planning for multiple autonomous underwater vehicles (AUVs) based on the perceived external environment, aiming to maximize node repair rates while minimizing the power consumption of the AUVs. We first model the node repair task as a Markov decision process and introduce the multi-agent independent conservative Q-learning algorithm to solve this problem. In addition, the large language model GPT-4 is employed to aid in the design of reward functions, aiming to realize better balance between each optimization objective during the RL training of AUVs. Experimental results demonstrate that the proposed framework exhibits high feasibility and superior performance.**

***Index Terms*—Autonomous underwater vehicles, reinforcement learning, node repair, Internet of Underwater Things, large language model.**

I. Introduction

The Internet of Underwater Things (IoUT) has gained attention for improving ocean exploration and monitoring [1]. However, due to the complexity and variability of the underwater environment, the IoUT networks encounter unique technical challenges compared to terrestrial sensor networks. The harsh underwater conditions make it difficult to replace or recharge node batteries in a timely manner. Additionally, sensor nodes are susceptible to corrosion by seawater over time, leading to eventual damage. The influence of ocean currents can also impair the functionality of underwater IoUT nodes [2]. These factors contribute to node failures, resulting in routing holes, network congestion, and even network breakdowns, thereby hindering the fulfillment of long-term monitoring tasks [3]. Consequently, repairing sensor nodes to maintain their proper functionality is a critical issue that must be addressed in IoUT networks.

Current research focuses on energy-efficient routing protocols, but in densely deployed networks, this can lead to packet collisions and the need for retransmissions [4]. Unfortunately, existing methods can hardly address increasing sensor node energy storage or repairing damaged nodes [5]. In recent years, training multiple Autonomous Underwater Vehicles (AUVs) using reinforcement learning (RL) methods to accomplish related tasks has emerged as a prominent topic in the field of IoUT research. Habob *et al.* [6] employed the actor-critic RL method to effectively address the information collection problem under multiple constraints. Xi *et al.* [7] established a three-dimensional grid model of the RL environment and proposed a D3QN-based AUV path planning scheme that integrates marine information. This environmental model reduces the gap with practical applications, and the algorithm provides a flexible and stable path. Wang *et.al* [8] proposed a multi-agent RL based AUV-assisted node repair (RANR) scheme, which considers limited underwater communication and scheduling between AUVs. However, the methods adopted in these studies is online RL, which often results in low data utilization, poor model convergence, and an inability to avoid the correlation between training data sets, leading to suboptimal training outcomes. In contrast, offline RL can effectively mitigate these shortcomings, offering better training performance and lower training costs.

Traditionally, the design of reward functions for offline RL algorithms has been based on experience and manually crafted. However, this process is time-consuming, labor-intensive, and does not guarantee the optimal design of the reward function [9]. In recent years, inverse RL [10] and preference learning [11] have emerged as popular solutions for designing reward functions. These approaches develop more suitable reward models by leveraging feedback from human preferences. Nevertheless, both methods still demand substantial manpower and data collection, and exhibit poor generalization performance outside the training data. Thanks to the emergence of large language model (LLM), researchers can simply provide the LLM with environment abstractions and task requirements. Subsequently, LLMs can generate effective reward functions for RL training [12].

Based on the above analysis, we develop an advanced framework for underwater node repair via AUVs based on

[+] These authors contribute equally to this work.

multi-agent offline RL. The framework uses the LLM GPT-4 to design reward functions and optimizes multiple objectives such as node repair rate and energy consumption via offline RL, while considering practical constraints and turbulence in ocean environment. The primary contributions of this paper are summarized as follows:

- To the best of our knowledge, this is the first effort in node repair using multiple AUVs facilitated by offline RL. This approach leverages pre-existing expert datasets for offline RL training, thereby minimizing extensive interactions with the environment.
- The node repair task is modeled as a multi-objective optimization task, aiming to minimize energy consumption, avoid collisions, and optimize the node repair rate. Given the complexity of the ocean environment and multi-objective, we introduce the multi-agent independent conservative Q-learning (MAICQL) algorithm, which relies on a decentralized training with decentralized execution (DTDE) model for offline training. And then we utilize the LLM GPT-4 to assist the design of reward functions.
- Extensive simulation experiments indicate that our framework can optimize multi-objective simultaneously, thereby enhancing work efficiency. Compared to other algorithms, the proposed framework demonstrates superior cumulative rewards, enhanced node repair rate, improved energy efficiency and lower collisions number.

The remainder of this paper is organized as follows. Section II elaborates on the development of the system model. Section III describes the problems that need to be addressed, and the detailed content of the reward function for LLM-aided design are elaborated. In Section IV, we introduce the MAICQL algorithm. The simulation results and corresponding discussions are provided in Section V, with the conclusion presented in Section VI.

II. System Model

A. Underwater Nodes Repair Model

The system model of the underwater node repair task is illustrated in Fig. 1, consisting of four main components: common nodes, damaged nodes, AUVs, and sea/land base stations. Common nodes operate normally, while damaged nodes require repairs due to energy issues or environmental damage. Besides, AUVs are deployed for node repairs, which are equipped with necessary tools. Sea/land base stations serve as centers for data management and coordination of repair operations for multi-AUV.

In the repair effort, the AUVs collaborate to identify faulty nodes and transport them to an onshore facility for recharging or maintenance. Concurrently, it is essential to account for the significant impact of ocean turbulence on the motion and energy consumption of AUVs. Therefore, AUVs should avoid turbulent regions during operation to optimize energy utilization.

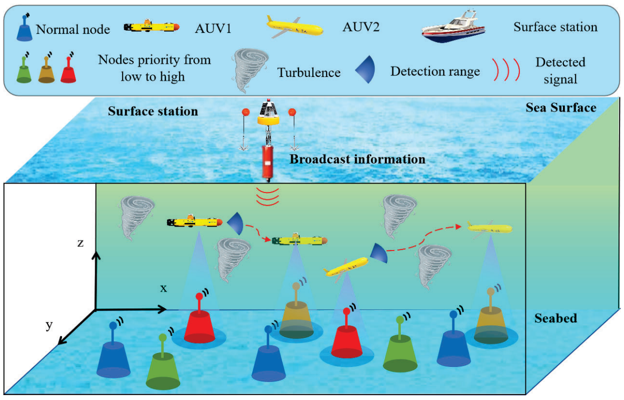

Fig. 1: Multi-AUV assisted underwater nodes repair scenario.

B. Damaged Node Selection Model

In underwater nodes repair task, it is imperative that AUVs prioritize the repair of nodes that exhibit a heightened level of urgency. We define the data transmission rate $\ell_i(t) \in [0, \ell_i^{max}]$ of node i at time t as an indicator to determine the severity of node failure, where ℓ_i^{max} represents the maximum data transmission rate of node i. A smaller value of $\ell_i(t)$ indicates a more severe potential fault in the node, suggesting an urgent need for repair. However, the data transmission rate of a node is influenced by the accumulation ratio $g_i(t)$ of data in the node. The ratio $g_i(t)$ is defined as follows

$$g_i(t) = \frac{\theta_i(t)}{\theta_{max}}, \tag{1}$$

where $\theta_i(t) \in [0, \theta_{max}]$ represent the data storage capacity of node i at time t, and θ_{max} denote the maximum data storage capacity of the node. If $g_i(t)$ is significantly large, it indicates a substantial data backlog at the current time, thereby impeding the data transmission rate.

Taking into account the actual conditions, the priority of node repair is influenced not only by the aforementioned two factors but also by the distance between node i and AUV k. Consequently, we define the repair priority $\mathcal{Q}_i^k(t)$ of a node as follows

$$\mathcal{Q}_i^k(t) = \frac{\theta_{max}}{(\theta_i(t)+\epsilon)(\ell_i(t)+\varepsilon)} - \xi d_i^k(t), \tag{2}$$

where $d_i^k(t)$ represents the relative distance between AUV k and node i, ϵ and ε are constants introduced to prevent calculation errors when $\theta_i(t)$ and $\ell_i(t)$ are equal to zero, and the parameter ξ is a penalty factor. The value of ξ should not be excessively large to avoid the AUV prioritizing the repair of distant but severely faulty nodes. By calculating $\mathcal{Q}_i^k(t)$, the AUV can prioritize the repair of nodes with greater damage severity while considering the relative distance, thereby enhancing the overall efficiency of the system.

C. AUV Energy Consumption Model

During the node repair process, the energy consumption of an AUV is attributed to two primary factors: the energy consumed during the stationary repair of a damaged node (hover energy consumption) and the energy expended while traversing between two damaged nodes. Utilizing principles from computational fluid dynamics (CFD), the resistance of an AUV hovering underwater can be expressed as follows

$$\lambda_j^h = 0.5\rho_s \|\mathcal{V}_{\mathbf{c}}(\mathbf{Y_j^t})\|_2^2 \mathcal{S}_a \Psi_d, \tag{3}$$

The resistance during navigation can be expressed as

$$\lambda_j^n = 0.5\rho_s \|\mathcal{V}_{\mathbf{k}}(\mathbf{Y_j^t})\|_2^2 \mathcal{S}_a \Psi_d, \tag{4}$$

where ρ_s is the density of seawater, $\mathcal{S}_a$ and Ψ_d are the resistance coefficient and front area of the AUV, respectively, $\mathbf{Y_j^t}$ is the position coordinate vector of AUV j at time t, and $\mathcal{V}_{\mathbf{c}}$ and $\mathcal{V}_{\mathbf{k}}$ are the flow velocity and relative velocity at coordinate $\mathbf{Y_j^t}$, respectively. Therefore, the power consumption of AUV j hovering at the $\hbar^{th}$ fault node can be calculated as follows

$$P_j^h[\hbar] = \frac{\lambda_j^h[\hbar] \|\mathcal{V}_{\mathbf{c}}(\mathbf{Y_j^{\hbar}})\|_2^2}{\vartheta}, \tag{5}$$

where ϑ is the electrical conversion efficiency.

Considering the actual situation, when the AUV moves from the $\hbar^{th}$ fault node to the $(\hbar+1)^{th}$ fault node, the relative speeds at different positions vary. Therefore, the speed at a fixed point cannot be used to calculate the energy consumption of the AUV during the moving process. To address this issue, we use the average of the relative velocities at the start point, midpoint, and endpoint of the trajectory to calculate the energy consumption. Taking the process of AUV j moving from the $\hbar^{th}$ fault node to the $(\hbar+1)^{th}$ fault node as an example, the average relative velocity is expressed as follows

$$\overline{\boldsymbol{v}_k}\big(\mathbf{Y_j^{\hbar}}\big) = \frac{\boldsymbol{v}_k\big(\mathbf{Y_j^{\hbar}}\big) + \boldsymbol{v}_k\big(\mathbf{Y_j^{\hbar_m}}\big) + \boldsymbol{v}_k\big(\mathbf{Y_j^{\hbar+1}}\big)}{3}, \tag{6}$$

where $\mathbf{Y_j^{\hbar_m}}$ is the position vector of the middle point of the trajectory. Therefore, the power consumption of the AUV along this motion trajectory is

$$P_j^m[\hbar] = \frac{0.5\rho_s \|\overline{v_k}\big(\mathbf{Y_j^{\hbar}}\big)\|_2^2 \mathcal{S}_a \Psi_d \|\overline{v_k}\big(\mathbf{Y_j^{\hbar}}\big)\|_2^2}{\zeta}. \tag{7}$$

According to the above analysis, it can be concluded that the total energy consumption of AUV j is

$$\mathcal{G}_j = \sum_{i=1}^{M} \sum_{\hbar=1}^{\mathcal{A}_o^{F_j}} \beta_{j,i}[\hbar] P_j^h[\hbar] \mathcal{T}_{j,i}[\hbar] + \sum_{\hbar=1}^{\mathcal{A}_o^{F_j}} P_j^m[\hbar] \mathcal{J}_j^m[\hbar], \tag{8}$$

where M represents the set of all nodes, $\beta_{j,i}[\hbar] = 1$ denotes the event that the AUV j hovers over node i for the $\hbar^{th}$ time, and conversely, $\beta_{j,i}[\hbar] = 0$. $\mathcal{T}_{j,i}[\hbar]$ represents the hovering time over the node, $\mathcal{A}_o^{F_j}$ denotes the set of hovering points of AUV j, and $\mathcal{J}_j^m[\hbar]$ represents the time required to move from the $\hbar^{th}$ node to the $(\hbar+1)^{th}$ node.

III. Problem Formulation

A. Optimization Problem Formulation

In this study, the objective of multi-AUV collaboration is to maximize the net profit of the overall operation. For AUV j, the node repair rate μ_j represents the benefit, while the number of collisions $\mathcal{C}_j$ and the AUV energy consumption $\mathcal{G}_j$ represent the costs. The μ_j is defined as follows

$$\mu_j = \frac{\mathcal{B}_j^c}{\mathcal{B}}, \tag{9}$$

where $\mathcal{B}_j^c$ represents the number of nodes that have been repaired, and $\mathcal{B}$ denotes the total number of nodes that need to be repaired.

Therefore, the net profit from this operation can be expressed as follows

$$\mathcal{P}_r = \sum_{j=1}^{N} (\delta\mu_j - \sigma\mathcal{C}_j) - \sum_{j=1}^{N} \omega\mathcal{G}_j, \tag{10}$$

where N is the total number of AUVs, δ, σ and ω are the contribution factors of the node repair rate, the number of collisions and the AUV energy consumption, respectively.

Based on the above analysis, the optimization problem of node repair can be characterized as follows

$$OP\colon \max_{\beta} \mathcal{P}_r = \sum_{j=1}^{N} (\delta\mu_j - \sigma\mathcal{C}_j) - \sum_{j=1}^{N} \omega\mathcal{G}_j, \tag{11}$$

$s.t.$

$$\sum_{j=1}^{N} \beta_{j,i} = 1, \forall i \in M, \tag{12}$$

$$\sum_{i=1}^{M} \beta_{j,i} = 1, \forall j \in N, \tag{13}$$

where the first optimization constraint specifies that only one AUV can hover over a node at any given time, while the second constraint stipulates that each AUV can hover over only one node at a time.

B. Markov Decision Process and Reward Function Design

Given that the constrained optimization problem is a high-dimensional NP-hard problem, it is challenging to solve directly. Therefore, we first transform it into a Markov decision process (MDP) and then address it using the RL algorithm. In this study, the MDP is represented by the following quintuple

$$\Phi = \{\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{L}, \boldsymbol{\mathcal{R}}, \gamma\}, \tag{14}$$

where $\boldsymbol{S}$, $\boldsymbol{A}$, $\boldsymbol{L}$ and $\boldsymbol{\mathcal{R}}$ represent state space, action space, state transition probability distribution and reward function respectively, and γ is the discount factor.

In the MDP, the AUV relies on the rewards from environmental feedback to evaluate its action strategy. Therefore, designing a reasonable reward function is crucial for the effective training of the AUV. Conventionally, the development of reward function has been manually crafted based on expert experience. However, this approach is both time-consuming

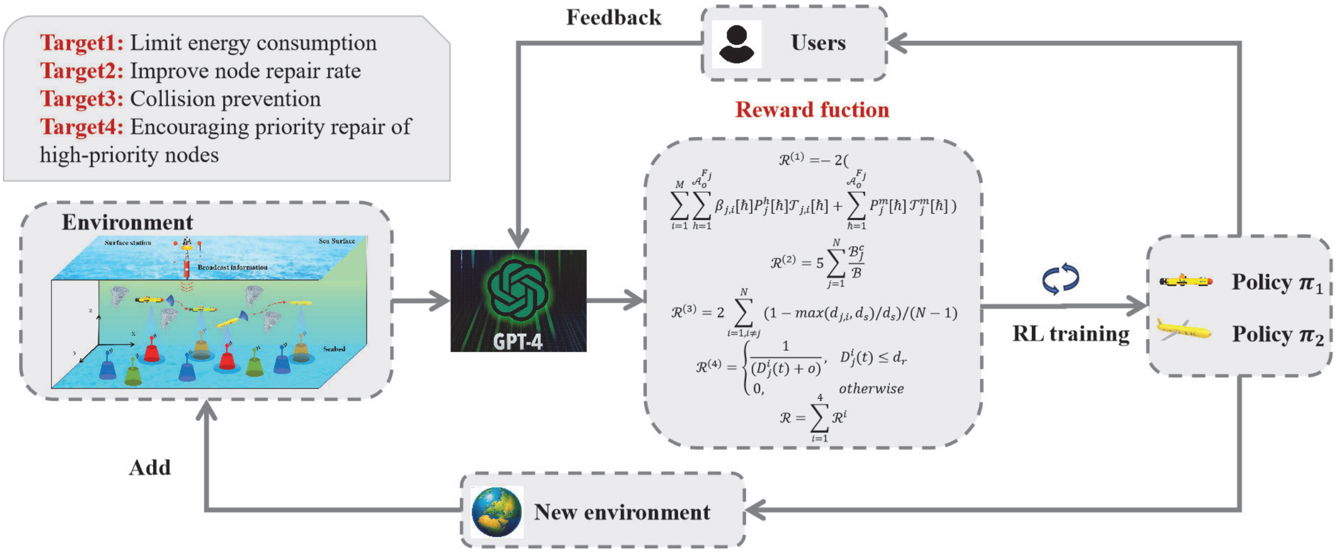
Target1: Limit energy consumption
Target2: Improve node repair rate
Target3: Collision prevention
Target4: Encouraging priority repair of high-priority nodes
Environment
Surface station
Sea Surface
Broadcast information
Seabed
GPT-4
Feedback
Users
Reward fuction
$\mathcal{R}^{(1)} = -2(\sum_{i=1}^{M}\sum_{\hbar=1}^{\mathcal{A}_o^{F_j}} \beta_{j,i}[\hbar]P_j^h[\hbar]\mathcal{T}_{j,i}[\hbar] + \sum_{\hbar=1}^{\mathcal{A}_o^{F_j}} P_j^m[\hbar]\mathcal{J}_j^m[\hbar])$
$\mathcal{R}^{(2)} = 5\sum_{j=1}^{N}\frac{\mathcal{B}_j^c}{\mathcal{B}}$
$\mathcal{R}^{(3)} = 2\sum_{i=1,i\neq j}^{N}(1 - max(d_{j,i}, d_s)/d_s)/(N-1)$
$\mathcal{R}^{(4)} = \begin{cases}\frac{1}{(D_j^i(t)+o)}, & D_j^i(t) \le d_r \\ 0, & otherwise\end{cases}$
$\mathcal{R} = \sum_{i=1}^{4}\mathcal{R}^i$
RL training
Policy π_1
Policy π_2
Add
New environment

Fig. 2: The workflow diagram of designing the reward function based on the LLM GPT-4.

and labor-intensive, and the resulting reward functions are not guaranteed to be optimal.

To address these challenges, we introduced the LLM GPT-4, which generates and shapes dense reward function code based on objective descriptions. After defining the optimization objective, GPT-4 generates initial dense reward function code using environmental information. This dense reward code is then integrated into the RL algorithm for training strategies. Unlike inverse RL, our method can design symbolic rewards with high interpretability. However, given the sensitivity of RL training, the inherent randomness of large language models, and the potential ambiguity in target descriptions, the initial reward function may not be fully suitable for achieving the desired goal. We address this by implementing the trained strategy in the environment and refining the reward function design through manual feedback based on observed outcomes. After multiple iterations of improvement, the designed reward function becomes well-suited to the current training task. The workflow diagram of designing the reward function based on the LLM GPT-4 is presented in Fig. 2. The designed reward function includes the following parts:

1) Limit energy consumption: To ensure that the AUV conserves as much energy as possible and avoids waste during task execution, the reward function should be designed to provide negative feedback when the AUV consumes energy

$$\mathcal{R}^{(1)} = -2(\sum_{i=1}^{M}\sum_{\hbar=1}^{\mathcal{A}_o^{F_j}} \beta_{j,i}[\hbar]P_j^h[\hbar]\mathcal{T}_{j,i}[\hbar] + \sum_{\hbar=1}^{\mathcal{A}_o^{F_j}} P_j^m[\hbar]\mathcal{J}_j^m[\hbar]). \tag{15}$$

2) Improve node repair rate: During a given period, a higher node repair rate corresponds to a higher degree of repair completion

$$\mathcal{R}^{(2)} = 5\sum_{j=1}^{N}\frac{\mathcal{B}_j^c}{\mathcal{B}}. \tag{16}$$

3) Collision prevention: To ensure the safety of multiple AUVs operating concurrently, it is essential to maintain a distance between AUVs that exceeds a certain threshold. Consequently, a negative reward is established to penalize collisions

$$\mathcal{R}^{(3)} = 2\sum_{i=1,i\neq j}^{N}(1 - \max(d_{j,i}, d_s)/d_s)/(N-1), \tag{17}$$

where $d_{j,i}$ indicates the distance between AUV i and AUV j, and d_s indicates the safety distance between the two AUVs.

4) Encouraging priority repair of high-priority nodes: based on previous analysis, the AUV should prioritize the repair of high-priority nodes to enhance repair efficiency. When the distance between the AUV and the target device is less than a specified value, the system provides a positive reward, with the reward increasing as the distance decreases. Therefore, the reward function is defined as follows

$$\mathcal{R}^{(4)} = \begin{cases}\frac{1}{(D_j^i(t)+o)}, & D_j^i(t) \le d_r, \\ 0, & \text{otherwise},\end{cases} \tag{18}$$

where d_r represents the set judgment threshold, $D_j^i(t)$ denotes the distance between the AUV and the target node, and o is a constant introduced to prevent calculation errors when $D_j^i(t) = 0$.

Based on the above analysis, the total reward function of this study is set as follows

$$\mathcal{R} = \sum_{i=1}^{4}\mathcal{R}^{(i)}. \tag{19}$$

IV. Algorithm Design

Offline RL allows the agent to learn from a pre-collected expert dataset, achieving an optimal policy without interaction with environment. Among these offline RL algorithms, CQL

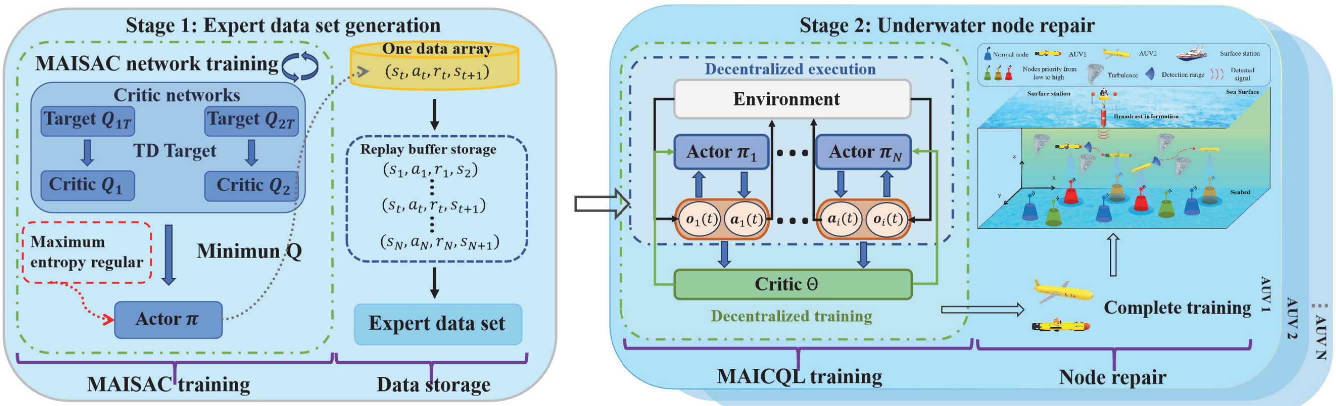

Fig. 3: The framework of our proposed multi-AUV assisted underwater nodes repair based on multi-agent offline RL.

reduces extrapolation error effects by adding constraints to the Bellman equation, keeping function values low for dataset deviations. This study expands CQL to MAICQL, training multi-AUV relying on DTDE model. This approach is applied to perform node repair tasks in a complex ocean environment, with the training dataset generated from a previously trained multi-agent independent soft actor-critic (MAISAC) algorithm [13]. The framework of our proposed multi-AUV assisted underwater nodes repair based on multi-agent offline RL is shown in Fig. 3.

In the MAICQL algorithm, AUV j is equipped with two value functions, Q_{1j} and Q_{2j}, as well as a policy function π_θ^j. Additionally, two target value functions, Q_{1j}^T and Q_{2j}^T, are employed to mitigate overestimation during the update process. For the value functions Q_{1j} and Q_{2j}, the updated equations are as follows

$$
\begin{aligned}
&\varepsilon_1^t \leftarrow \\
&\quad \varepsilon_1^{t-1} - \eta_Q \nabla_\theta \Bigg(\alpha \cdot \mathbb{E}_{s\sim\mathcal{D}} \Bigg[\log \sum_a \exp\left(Q_{1j}^{s_1^{t-1}}(s,a) \right) \\
&\quad - \mathbb{E}_{a\sim\pi_\theta^j(a|s)} \left[Q_{1j}^{s_1^{t-1}}(s,a) \right] \Bigg] \\
&\quad + \frac{1}{2} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\left(Q_{1j}^{s_1^{t-1}}(s,a) - \widehat{\mathbb{B}}^{\pi_\theta} Q_{1j}^{s_1^{t-1}}(s,a) \right)^2 \right] \Bigg),
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
&\varepsilon_2^t \leftarrow \\
&\quad \varepsilon_2^{t-1} - \eta_Q \nabla_\theta \Bigg(\alpha \cdot \mathbb{E}_{s\sim\mathcal{D}} \Bigg[\log \sum_a \exp\left(Q_{2j}^{s_2^{t-1}}(s,a) \right) \\
&\quad - \mathbb{E}_{a\sim\pi_\theta^j(a|s)} \left[Q_{2j}^{s_2^{t-1}}(s,a) \right] \Bigg] \\
&\quad + \frac{1}{2} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\left(Q_{2j}^{s_2^{t-1}}(s,a) - \widehat{\mathbb{B}}^{\pi_\theta} Q_{2j}^{s_2^{t-1}}(s,a) \right)^2 \right] \Bigg),
\end{aligned} \tag{21}
$$

where $\widehat{\mathbb{B}}^{\pi_\theta}$ is the Bellman operator of policy π_θ, $\mathbb{E}$ is the expectation, $\mathcal{D}$ is the state-action space, and α is the entropy regularity coefficient. For the target value functions Q_{1j}^T and Q_{2j}^T, the updated equations are as follows

$$\varepsilon_{1t}^- = \tau \varepsilon_1^t + (1-\tau)\varepsilon_1^{t-1}, \tag{22}$$

$$\varepsilon_{2t}^- = \tau \varepsilon_2^t + (1-\tau)\varepsilon_2^{t-1}, \tag{23}$$

where τ is the soft update coefficient, and the parameter update equation of the policy function π_θ^j is

$$
\begin{aligned}
&\theta_{\rm t} \leftarrow \theta_{\rm t-1} - \\
&\eta_\pi \nabla_\theta \mathbb{E}_{s\sim\mathbb{D}, a\sim\pi_\theta^j(a|s)} \Big[\alpha \log \pi_\theta^j(a|s) - \min(Q_{1j}^{\varepsilon_1}(s,a), Q_{2j}^{\varepsilon_2}(s,a)) \Big],
\end{aligned} \tag{24}
$$

where η_π is the weight coefficient.

V. Simulation Results and Analysis

To evaluate the performance of MAICQL algorithm and proposed framework, this study first conduct experiments to compare MAICQL with MAISAC, behavioral cloning (BC) and generative adversarial imitation learning (GAIL). The experimental results are presented in Fig. 4 and Table I. For clearer visualization, the cumulative reward value in Fig. 4 is adjusted by adding 2000 to the original reward value. The results indicate that the proposed MAICQL algorithm demonstrates superior performance compared to the other three algorithms, evidenced by higher cumulative reward, increased node repair rate, lower energy consumption, and fewer average AUV collisions number. These results validate the efficacy of our advanced framework.

To validate the feasibility of our proposed advanced framework for node repair, two AUVs are trained in both turbulent and non-turbulent environments to perform the node repair task. The simulation results are presented in Fig. 5 and Fig. 6. As depicted in the figures, the AUVs effectively cooperate and efficiently complete node repair tasks regardless of the presence of turbulence. Additionally, in turbulent environments, the AUVs can optimize their paths to avoid turbulent areas as much as possible. These experimental results demonstrate that the proposed method can adapt to varying external conditions and has broad applicability.

TABLE I: Algorithm Performance Comparison

Algorithm	Cumulative Reward	Node Repair Rate	Energy Consumption	AUV Collisions Number
MAICQL	-753	0.93	137.39	1.2
MAISAC	-783	0.90	143.57	2.6
BC	-1053	0.55	156.94	5.5
GAIL	-1276.52	0.39	167.81	5.7

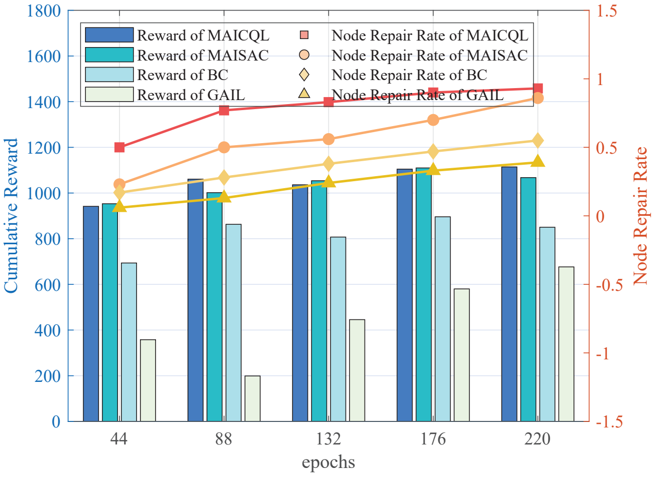

Fig. 4: Cumulative reward of experiments utilizing MAICQL, MAISAC, BC, and GAIL for training, respectively.

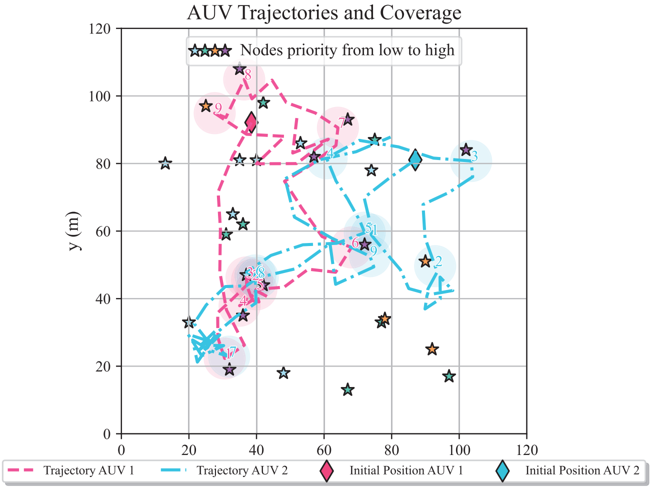

Fig. 5: Trajectories of AUVs for the node repair task in the turbulence-free environment.

VI. Conclusion

In this study, we develop propose an advanced framework for underwater node repair utilizing multi-agent offline RL. The framework optimizes multiple objectives such as node repair rates and energy consumption, considering practical constraints and turbulence in ocean environment. More specifically, we model the problem as a MDP, while proposing the DTDE based MAICQL algorithm to address the challenge of efficiently training multi-AUV to repair the nodes that exhibit varying degrees of damage. Besides, the tailored reward function is designed using the LLM GPT-4. Simulation outcomes demonstrate the superior performance of the proposed framework. Future work will focus on conducting the sim2real experiments in the real ocean environment.

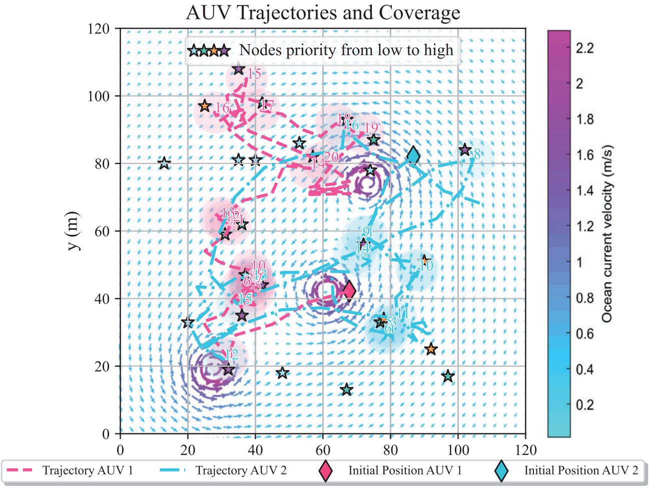

Fig. 6: Trajectories of AUVs for the node repair task in the turbulence environment.

VII. Acknowledgment

This work of Jingjing Wang was partly supported by the National Natural Science Foundation of China under Grant No. 62071268 and No. 62222101, partly supported by the Young Elite Scientist Sponsorship Program by the China Association for Science and Technology under Grant No. 2020QNRC001, and partly supported by the Fundamental Research Funds for the Central Universities. This work of Yong Ren was partly supported by the National Natural Science Foundation of China under Grant 62127801, partly supported by the National Key Research and Development Program of China under Grant 2020YFD0901000. (*Corresponding author: Jingjing Wang*)

References

[1] R. A. Khalil, N. Saeed, M. I. Babar, and T. Jan, “Toward the internet of underwater things: Recent developments and future challenges,” *IEEE Consumer Electronics Magazine*, vol. 10, no. 6, pp. 32–37, 2020.

[2] 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 Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4297–4307, 2020.

[3] K. Y. Islam, I. Ahmad, D. Habibi, and A. Waqar, “A survey on energy efficiency in underwater wireless communications,” *Journal of Network and Computer Applications*, vol. 198, p. 103295, 2022.

[4] T. Hu and Y. Fei, “Qelar: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks,” *IEEE transactions on mobile computing*, vol. 9, no. 6, pp. 796–809, 2010.

[5] R. W. Coutinho, A. Boukerche, L. F. Vieira, and A. A. Loureiro, “Enor: Energy balancing routing protocol for underwater sensor networks,” in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[6] A. A. Al-Habob, O. A. Dobre, and H. V. Poor, “Age-optimal information gathering in linear underwater networks: A deep reinforcement learning approach,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 13 129–13 138, 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 of Things Journal*, vol. 9, no. 18, pp. 17 440–17 451, 2022.

[8] Z. Wang, Z. Zhang, J. Wang, C. Jiang, W. Wei, and Y. Ren, “Auv-assisted node repair for iout relying on multiagent reinforcement learning,” *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4139–4151, 2024.

[9] A. Ng, D. Harada, and S. J. Russell, “Policy invariance under reward transformations: Theory and application to reward shaping,” in *International Conference on Machine Learning*. PMLR, 1999, pp. 278–287.

[10] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, “Maximum entropy inverse reinforcement learning,” in *the National Conference on Artificial Intelligence*. AAAI, 2008, pp. 1433–1438.

[11] K. Lee, L. M. Smith, and P. Abbeel, “Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 6152–6163.

[12] T. Xie, S. Zhao, C. H. Wu, Y. Liu, Q. Luo, V. Zhong, Y. Yang, and T. Yu, “Text2reward: Reward shaping with language models for reinforcement learning,” in *International Conference on Learning Representations*. PMLR, 2024, pp. 1–37.

[13] Z. Zhang, J. Xu, G. Xie, J. Wang, Z. Han, and Y. Ren, “Environment- and energy-aware auv-assisted data collection for the internet of underwater things,” *IEEE Internet of Things Journal*, vol. 11, no. 15, pp. 26 406–26 418, 2024.