

An improved reinforcement learning path planning algorithm for AUVs operating in unknown and unstructured environments

Bhaskar Jyoti TALUKDAR , Kumar Debi PRASAD  and Bikramaditya DAS 

Effective path planning is vital for Autonomous Underwater Vehicles (AUVs) for performing diverse roles, including rescue efforts and logistics operations. In recent times, dynamic path planning has emerged as a prominent research area, enabling AUVs to navigate without the need for previous static information. Researchers are leveraging deep reinforcement learning (DRL), as another trending field of study, to tackle the challenges associated with dynamic path planning. In this study a dynamic sensing and collision avoidance approach is proposed to navigate in an entirely unknown environment with unstructured obstacles. Firstly, we employ Deep Q-Networks (DQN), a distinct domain within DRL, to address the challenge of path planning in a dynamic environment. Initially, established methodologies were implemented such as Double Deep Q-Networks (DDQN) and Dueling Double Deep Q-Networks (D3QN) to develop a model capable of navigating an AUV through environments containing both static and dynamic obstacles. Nevertheless, the previously mentioned techniques exhibit restricted generalization abilities in complex, high-dimensional state environments, frequently leading to inadequate performance in dynamic situations, thus making the more adaptable I-DQN method essential. Subsequently, Improved DQN (I-DQN) is introduced, an enhancement of the original DQN, aimed at further refining the DRL model's performance. Test within a real-time setting is carried out to evaluate the effectiveness of the DRL models against randomly generated starting and destination points. The results obtained indicate that the I-DQN approach outperforms DQN, DDQN, and D3QN

Copyright © 2026. The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (CC BY-NC-ND 4.0 <https://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits use, distribution, and reproduction in any medium, provided that the article is properly cited, the use is non-commercial, and no modifications or adaptations are made

B.J. Talukdar (e-mail: bjt10talukdar@gmail.com) is with Department of Electronics and Telecommunication Engineering, Veer Surendra Sai University of Technology, Burla-768018, Odisha, India.

K.D. Prasad (e-mail: debiprasad.srm@gmail.com) is with Department of Electronics and Communication Engineering, GIFT Autonomous, Bhubaneswar-752054, Odisha, India.

B. Das (corresponding author, e-mail: adibik09@gmail.com) is with Department of Electronics and Communication Engineering, CUPGS, Biju Patnaik University of Technology, Rourkela- 769015, Odisha, India.

This work is supported by Anusandhan National Research Foundation (ANRF), India. The financial assistance from State University Research Excellence (SERB-SURE), India through grant number SUR/2022/004553.

Received 28.02.2025. Revised 2.03.2026.

regarding the path efficiency and the travel duration to reach the destination while avoiding collisions. The stability of the I-DQN algorithm is improved by Prioritized Experience Replay, which carefully selects significant transitions according to their temporal-difference errors, facilitating effective and reliable learning in intricate AUV navigation situations. The proposed strategy can be utilized to improve underwater search and rescue missions in inundated urban regions or areas affected by disasters. The AUVs can maneuver through intricate and unfamiliar underwater settings, skilfully dodging debris and obstructions, to swiftly and efficiently find survivors or evaluate structural damage.

Key words: AUV, path planning, static and dynamic obstacle, DRL, deep Q learning, dueling neural network, ROS

1. Introduction

AUVs are widely utilized robotic systems that serve a diverse array of purposes. AUVs can perform tasks independently. AUVs exhibit remarkable manoeuvrability, allowing them to access remote areas and navigate through narrow, challenging routes, while also being cost-effective to operate [1–4]. The path planning process for an AUV entails several stages of transformations and adjustments in forward angles to achieve a destination based on optimization criteria [5]. AUVs typically function in demanding environments filled with numerous obstacles to avoid. Static obstacles can frequently be assessed and mapped in advance. Typically, a global map is employed to chart the course in the presence of obstacles [6–8]. The primary steps in addressing the AUVs path planning problem include modelling the environment along with determining the route from the starting to the end point.

Planning a path for an AUV in an unstructured setting poses distinct challenges owing to the everchanging and unpredictable characteristics of underwater landscapes [9–11]. In contrast to structured environments, unstructured underwater areas do not have established pathways and are marked by obstacles like rocks, coral reefs, and diverse seabed formations [12–15]. Successful path planning in these scenarios demands that the AUV adjust its course dynamically in real-time, guaranteeing safe navigation while enhancing the route to the intended target.

Recently, some scholarly work has undertaken the task of addressing the challenges of 3D obstacle avoidance for AUVs. Many methods experience challenges because they become ensnared for local optimal solutions, as highlighted by Fan et al. and Zhang et al. [16, 17]. Certain techniques, such as those examined by Sun et al. and Kot et al., encounter significant computational requirements or energy consumption concerns, which could impede their application in real-time scenarios [18–20]. One persistent concern is inadequate real-world verification, with findings from Yan et al., Zhang et al., and Feng et al. demonstrating a gap in different practical evaluations [21–23]. Furthermore, complications emerge in unpredictable underwater settings, as indicated by Zhi et al., along with escalating

prediction errors over extended trajectories, as noted by Zhang et al. [24,25]. Considering these challenges, an Enhanced DQN algorithm can provide numerous advantages for AUV path planning. DQN, as a reinforcement learning method, has the potential to overcome the issue of becoming trapped in local optima by investigating a broader solution landscape [26]. Its capacity to learn from experience may render it more flexible under unpredictable underwater conditions. The learning potential of DQN may also result in improved generalization across a variety of scenarios, addressing the deficiencies in real-world validation observed in some existing methods. The proposed algorithm was validated in both static and dynamic underwater environments. The results from the proposed algorithm demonstrate its effectiveness in the path planning of AUVs within an unstructured setting.

As mentioned in the surveys, DRL techniques hold promise as viable options for AUV path planning and control, and research in this area continues to progress. Most of the existing literature related to AUVs concentrates on implementing value-driven path planning algorithms [27–30]. Nevertheless, the discretization of actions compromises the precision of controlling in real-time. Several studies utilize actor-critic-based approaches. The following is an outline of the research gaps:

- unrealistic assumptions, including the disregard for obstacles, disruptions to the environment, nonholonomic constraints, or dynamic models.
- overlooking essential output trajectory characteristics like practicality and reliability.

Encouraged by the insights highlighted above, this investigation suggests a novel algorithm dedicated to the path planning of AUVs operating within an intricate underwater environment marked by various dynamic and static challenges. The following contributions are uniquely presented in this paper:

- The suggested method utilizes the Enhanced DQN algorithm to execute adaptive path planning in unstructured environments. By combining a DQN with improvements such as prioritized experience replay, doubled Q-learning, and a target network, the algorithm adeptly acquires optimal paths through intricate terrains. This strategy guarantees that the agent can adjust to fluctuating environmental conditions and discover efficient routes in real-time.
- The algorithm features a strong obstacle avoidance strategy by imposing penalties on actions that result in collisions with obstacles or revisiting previously encountered states. A safety margin is established around obstacles to further improve the agent's capacity to avoid collisions. The reward function is crafted to penalize the agent for approaching obstacles and revisiting

the same locations, while simultaneously rewarding progress towards the target. This approach ensures that the agent learns to navigate in an efficient manner, steering clear of potential dangers in the environment.

- The incorporation of regular target network and experience replay synchronization significantly boosts the learning stability and efficiency of the enhanced DQN algorithm. Experience enables the agent to gain knowledge from a varied collection of past experiences by sampling mini-batches from a replay buffer, disrupting the correlation between consecutive experiences. This contributes to stabilizing the learning process and enhances the convergence rate. Furthermore, the target network, which is periodically aligned with the main network, offers stable Q-value targets during training, reducing the likelihood of divergence.

The subsequent sections are delineated as follows. In Section 2, the background of the completed work is elaborated. Traditional path planning techniques such as DQN, DDQN, and D3QN are introduced in Section 3. Section 4 introduces a path planning approach that integrates turnarounds through reinforcement learning. Section 5 showcases the simulation and experimental findings that affirm the proposed method's effectiveness. Section 6 wraps up the paper and explores avenues for future research.

2. Background

2.1. AUV dynamics

The nonlinear motion equation of the AUV as a result of wave disruptions in a six degree of freedom is described here in this section [31–35]. Figure 1 is the

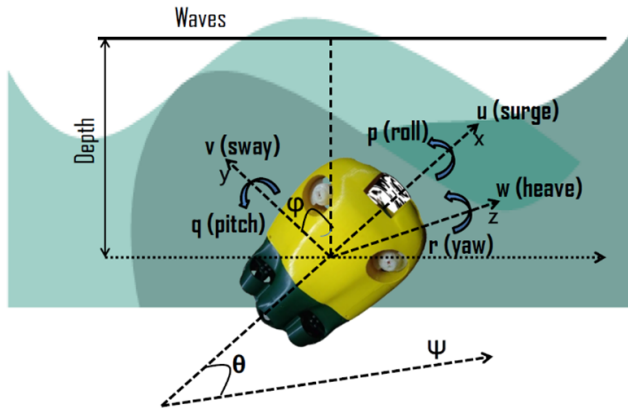


Figure 1: Body fixed frame and inertia of an AUV model

schematic representation of the AUV for body fixed frame. The equations can be described by

$$\dot{X}_{n_{ni}} = j_n (X_{n_{mi}}) v_{ni} , \quad (1)$$

$$\dot{X}_{n_{mi}} = j_n (X_{n_{mi}}) v_{mi} . \quad (2)$$

For different AUVs, $i = 1, 2, 3, \dots, N$, $X_{n_i} = [X_{n_{ni}}, X_{n_{mi}}]^T$ is the vector for the position and orientation of the i -th AUV in the inertial frame, $X_{n_{ni}} = [x_i, y_i, z_i]^T$ are position coordinates and $X_{n_{mi}} = [\varphi_i, \theta_i, \psi_i]^T$ are the orientations of longitudinal, vertical and transverse axes, respectively. $v_i = [v_{ni}, v_{mi}]^T$ are coordinates for the velocity vector in the structure of the AUV fixed frame, $v_{ni} = [u_i, v_i, w_i]^T$ are linear velocities and $v_{mi} = [p_i, q_i, r_i]^T$ are angular velocities [31–35].

$$\dot{v}_{ni} = M_n^{-1} (-c_{ni} (v_{ni}) v_{mi} - D_{ni} (v_{ni}) v_{ni} - g_n (X_{n_{mi}}) + \tau_{ni}) , \quad (3)$$

$$\dot{v}_{mi} = M_m^{-1} (-c_{ni} (v_{ni}) v_{ni} - D_{mi} (v_{mi}) v_{mi} - g_m (X_{n_{mi}}) + \tau_{mi}) . \quad (4)$$

$\tau_i = [\tau_{ni}, \tau_{mi}]^T$ is an input for control in the i -th AUV, $\tau_{ni} = [X_i, Y_i, Z_i]^T$ are the external forces and $\tau_{mi} = [K_i, M_i, N_i]^T$ denoting the instances of exterior pressures operating in both translational and rotational motion on the AUV. A non-singular transformation matrix is represented by $J(\eta)$, which is employed in transforming between Earth-fixed and body-fixed frames [31–35].

2.2. Problem statement

The algorithm must enhance the AUV's trajectory from the initial node S to the target node T while considering obstacle avoidance, path smoothness, depth control, path length efficiency, and overall safety. The solution should strike a balance among these elements to minimize the cost function, leading to an optimal trajectory for the AUV's mission.

2.2.1. Obstacle collision penalty

$$M_{\text{Penalty}} (X_{n_i}) = M_{\text{Path}} (X_{n_i}) + K_{\text{collision}} \sum_i \exp \left(-\frac{|X_{n_{ni}} - q_{\text{obs}}|}{d_{\text{safe}}} \right) , \quad (5)$$

where M_{Penalty} is the total cost function for penalty during a collision [36]. M_{path} is the cost associated with the path itself (e.g., path length, energy consumption, etc.) and $K_{\text{collision}}$ is a positive the weight that establishes the significance of the collision avoidance term. Where, $p_i = (x_i, y_i, z_i)$ position of obstacles and the nearest obstacle configuration (q_{obs}). The safety distance threshold is represented by d_{safe} . The penalty for each configuration q_i along the path is expressed as

$\exp(-d(p_i, q_{\text{obs}})/d_{\text{safe}})$. The penalty term approaches zero when the distance $d(p_i, q_{\text{obs}})$ is significantly greater than d_{safe} , indicating that there is no appreciable penalty for maintaining distance from the obstacles.

2.2.2. Smoothness factor

$$M_{\text{smoothness}}(X_{n_i}) = K_{\text{smoothness}} \sum_i \frac{d^2 \delta \psi_i}{(ds_i)^3}, \quad (6)$$

where $K_{\text{smoothness}}$ is a positive weight used for assessing the significance of the path smoothness term [36]. The diameter or characteristic size of the AUV is denoted by d . ψ_i is the change in heading angle (or curvature) at the i -th point of the reference for the path ds_i . The summation is taken over all waypoints along the planned path, except for the first and last waypoints.

2.2.3. Depth change factor

$$M_{\text{Depth change}}(X_{n_i}) = K_{\text{depth}} \sum_i w_i |z_i - z_{\text{desired}}|^n, \quad (7)$$

where K_{depth} is a positive weight that signifies the depth cost term [36]. w_i is a weighting factor for the i -th point of the reference for the path. The AUV depth (or z -coordinate) at the i -th way-point is denoted by z_i and z_{desired} is the desired depth or depth range for the AUV. $|z_i - z_{\text{desired}}|$ is the definite distinction between the AUV depth and the desired depth at the i -th way-point. A positive constant n determines the rate of increase in the penalty as the depth deviation increases (typically $n \geq 1$). The summation is taken over all way-points along the planned path.

2.2.4. Path length factor

$$M_{\text{path length}} = K_{\text{length}} \times \sum (d_{ij} \times \exp(\alpha \times d_{ij}/R_{\text{turn}})), \quad (8)$$

where:

$$d_{ij} = |X_{n_{n(i+1)}} - X_{n_{ni}}|, \quad (9)$$

where K_{length} is a positive measure of the importance provided to the path length cost term [36]. The distinction between the $(i-1)$ -th and i -th waypoints along the path is d_{ij} . AUV turning radius is R_{turn} , whereas α is the positive constant and determines the rate of increase in the penalty for longer path segments. $X_{n_{ni}} = [x_i, y_i, z_i]^T$ is the i -th AUVs location vector in the inertial frame.

2.2.5. Safety factor

$$M_{\text{safety}} = \sum_{i=1}^N \sum_{j=1}^M (\exp(-\alpha \times d_{ij})) \times (1 + \beta \times v_{oj}), \quad (10)$$

where:

$$d_{ij} = |X_{ni} - X_{oj}|, \quad (11)$$

where d_{ij} signifies the measure of distance in a Euclidean context. The j -th obstacle's coordinates are referred to as X_{oj} . The potential risk or severity related to the j -th obstacle is represented by v_{oj} . The parameters α and β are adjustable factors that modify the weights assigned to distance and risk, respectively. In this instance, N illustrates the total AUV count, while M shows the complete number of obstacles present.

3. Methodology

3.1. Markov Decision Process (MDP)

The mathematical configuration known as a Markov Decision Process (MDP) is employed to portray issues related to decision-making. Figure 2, is the schematic diagram of the action space. The tuple (S, A, P, R, γ) is what defines an MDP, where:

- A : defined set of actions.
- S : defined set of states.
- P : $P(s'|s, a)$ indicates a state transition probability, or the likelihood of changing from state s to state s' following action a .

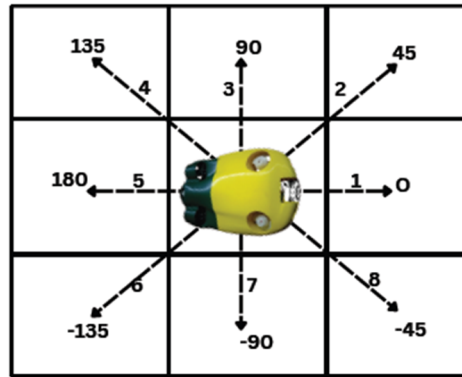


Figure 2: Schematic diagram of action space

- $R: R(s, a, s')$ is a function that rewards that provides the instantaneous benefit obtained during action a 's transition from state s to state s' .
- discount factor $\gamma: \gamma \in [0, 1]$, which stands for the significance of rewards in the future.

3.2. Q-learning

Operating without a model, Q-learning is a technique for RL that seeks to determine the ideal strategy for selecting actions through the Q-function. The Q-value, $(Q(s, a))$, denotes the forecasted total reward for executing action (a) in state (s) while remaining compliant with the optimal policy subsequently. let's align the state and action spaces:

State space:

$$s = (X_{ni}, v_i) \mid X_{ni} = [X_{n_{ni}}, X_{n_{mi}}]^T, \quad v_i = [v_{ni}, v_{mi}]^T, \quad (12)$$

where $X_{n_{ni}} \in R^3$ is the position coordinates, $X_{n_{mi}} \in R^3$ is orientation angles, $v_{ni} \in R^3$ is linear velocities, $v_{mi} \in R^3$ is angular velocities. Action space can be expressed as

$$A = \tau_i = [\tau_{ni}, \tau_{mi}]^T \mid \tau_{ni}, \tau_{mi} \in R^3. \quad (13)$$

The definition of the ideal Q-function is:

$$Q(s, a) = \max_{\pi \in E} \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a, \pi \right], \quad (14)$$

where π is the policy, γ is the discount factor, and r_t is the reward at time t . For Q^* , the Bellman optimality calculation is [37, 38]:

$$Q(s, a) = E \left[r + \gamma \max_{a'} Q(s', a') \mid s, a \right], \quad (15)$$

where following action a in state s , s' is the subsequent state. The temporal difference (TD) error δ is the difference between the current estimate of Q and the target value:

$$\delta = r + \gamma \max_{a'} Q(s', a') - Q(s, a). \quad (16)$$

Q can be updated using the TD error and a learning rate α :

$$Q(s, a) \leftarrow Q(s, a) + \alpha. \quad (17)$$

Substituting the TD error, the Q-learning updating rule is obtained:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]. \quad (18)$$

In the context of AUV control, this update rule would be applied at each time step, where:

- s is the current state (X_{n_i}, v_i)
- a is the control input τ_i
- r is the reward (which could be based on how well the AUV is following its desired trajectory or accomplishing its mission)
- s' is the next state after applying the control input $\max_{a'} Q(s', a')$ is the estimated optimal value of the next state.

This update rule allows the AUV to learn the optimal control policy over time by iteratively improving its estimates of the Q-function based on the rewards it receives, and the state transitions it experiences. A significant drawback of Q-Learning is its difficulty while managing state spaces with large dimensions, thus highlighting the importance of DQN. Figure 3 presents a comprehensive framework for AUV operations that employs an I-DQN for effective decision-making. It employs a feedback system that persistently monitors the surroundings

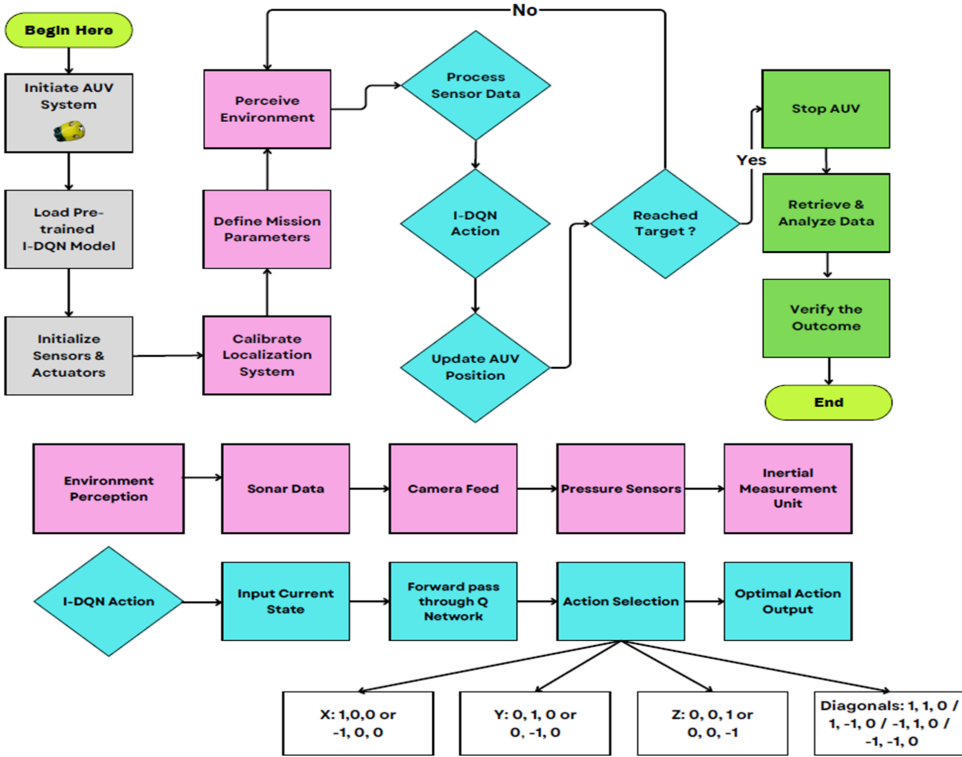


Figure 3: Flowchart of the proposed work

using diverse sensor integration, evaluating the information, and deciding the optimal actions guided by the I-DQN model. The design specifically tackles the difficulties associated with underwater navigation while utilizing reinforcement learning to promote adaptive behavior in both intricate, dynamic settings and stable environments.

3.3. Deep Q Network (DQN)

Figure 4 is the architectural framework of a DQN, typically encompasses the following layers:

- **Input Layer:** The input layer acquires the state representation originating from the environment.
- **Hidden Layers:** The hidden layers are constituted of fully connected (dense) layers, each subsequently followed by a Rectified Linear Unit (ReLU), or other non-linear activation function. These layers play a pivotal role in capturing intricate features and relationships residing within the state space.
- **Output Layer:** The output layer comprises neurons that correspond to the array of potential actions available to the agent. Each neuron generates a Q-value, which signifies the anticipated cumulative reward for executing the associated action within the specified state. The action which yields the highest Q-value is identified as the optimal action.

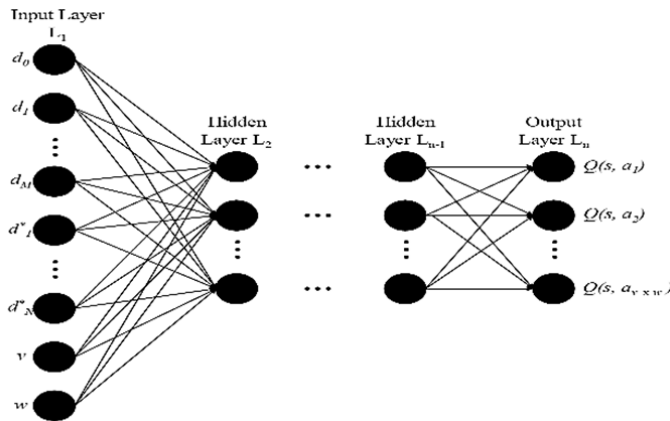


Figure 4: DQN architecture

DQN improves on the conventional Q-learning structure by using deep neural network to accurately estimate the Q-function. The elements that direct the neural network are symbolized by θ . Therefore, the notation $Q(s, a; \theta)$ is used to describe

the Q-value:

$$Q(s, a; \theta) \approx Q^*(s, a), \quad (19)$$

where $Q^*(s, a)$ is the optimal Q-function as defined in the Q-learning section:

$$Q^*(s, a) = \max_{\pi \in E} \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a, \pi \right]. \quad (20)$$

Here, π represents the policy, $\gamma \in [0, 1]$ and r_t is the reward at time t . The DQN is trained through lowering the loss function:

$$L(\theta) = E_{(s,a,r,s') \sim D} [(y - Q(s, a; \theta))^2], \quad (21)$$

where D is the replay memory, and y is the target Q-value defined as:

$$y = r + \gamma \max_{a'} Q(s', a'; \theta^-). \quad (22)$$

Here, θ^- indicates a different target network's parameter, which is periodically updated to improve training stability. The temporal difference error δ as defined, is used in the DQN update:

$$\delta = r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta). \quad (23)$$

The parameters θ are updated using gradient descent, which is an extension of the Q-learning update rule:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right], \quad (24)$$

where α is the learning rate. In the field of AUV navigation in an unstructured environment, the state (s) might summarize the AUVs position and orientation elements, while the action (a) represents the allowable maneuvers (such as moving ahead, executing a leftward turn, or executing a rightward turn). The reward function (r_t) is meticulously crafted to incentivize the AUV to successfully navigate towards the target while simultaneously circumventing potential obstacles. For instance:

$$r_t = \begin{cases} R_{\text{goal}} & \text{if goal is reached,} \\ -R_{\text{collision}} & \text{if collision occurs,} \\ -R_{\text{step}} & \text{otherwise,} \end{cases} \quad (25)$$

where (R_{goal}) , $(R_{\text{collision}})$ and (R_{step}) are constants.

3.4. Double DQN (DDQN)

The DDQN algorithm addresses the overestimation bias present in the standard DQN by dividing the choices of actions from the Q-value evaluation. This is achieved by using two sets of parameters: (θ) for the online network and (θ^-) for the target network. In DDQN, the target value y is modified as follows:

$$y = r + \gamma Q \left(s', \arg \max_{a'} Q(s', a'; \theta); \theta^- \right). \quad (26)$$

The action $(\arg \max_{a'} Q(s', a'; \theta))$ is chosen using the online network parameters (θ) , however, the Q-value is evaluated utilizing the target network parameters (θ^-) . The loss function used to train the DDQN is defined as follows:

$$L(\theta) = E_{(s,a,r,s') \sim D} [(y - Q(s, a; \theta))^2]. \quad (27)$$

DDQN lowers overestimation bias by selecting actions online and evaluating Q-values on the target network, resulting in more steady and accurate learning.

3.5. Dueling Double DQN (D3QN)

In the context of D3QN, the Q-value is partitioned into two distinct evaluative components: the $(A(s, a; \theta, \alpha))$ is the advantage function and state-value function $(V(s; \theta, \beta))$:

$$y = r + \gamma Q \left(s', \arg \max_{\{a'\}} Q(s', a'; \theta); \theta, \right), \quad (28)$$

where $(V(s; \theta, \beta))$ symbolizing the importance of being in a state (s) . $(A(s, a; \theta, \alpha))$ symbolizes the advantage of acting (a) in state (s) . The desired value (y) in D3QN is modified as follows:

$$y = r + \gamma \left[V(s'; \theta^-, \beta^-) + \left(A \left(s', \arg \max_{a'} Q(s', a'; \theta); \theta^-, \alpha^- \right) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s', a'; \theta^-, \alpha^-) \right) \right]. \quad (29)$$

The loss function used to train the D3QN is categorized as follows:

$$L(\theta) = E_{(s,a,r,s') \sim D} [(y - Q(s, a; \theta, \alpha, \beta))^2]. \quad (30)$$

4. Improved DQN (I-DQN) algorithm

I-DQN represents an advanced reinforcement learning algorithm that incorporates a multitude of innovations surpassing the traditional DQN framework, including Distributional RL, Multi-step Learning, Dueling Network Architec-

tures, DDQN, Prioritized Experience Replay, and Noisy Nets. Within the sphere of AUV path planning, I-DQN possesses the capability to proficiently acquire optimal navigation methodologies by leveraging these enhancements to maneuver through the complex and dynamically evolving underwater environment.

4.1. Improved DQN (I-DQN)

I-DQN represents a sophisticated reinforcement learning algorithm that incorporates various improvements over the conventional DQN. Figure 5 is the workflow of the proposed algorithm.

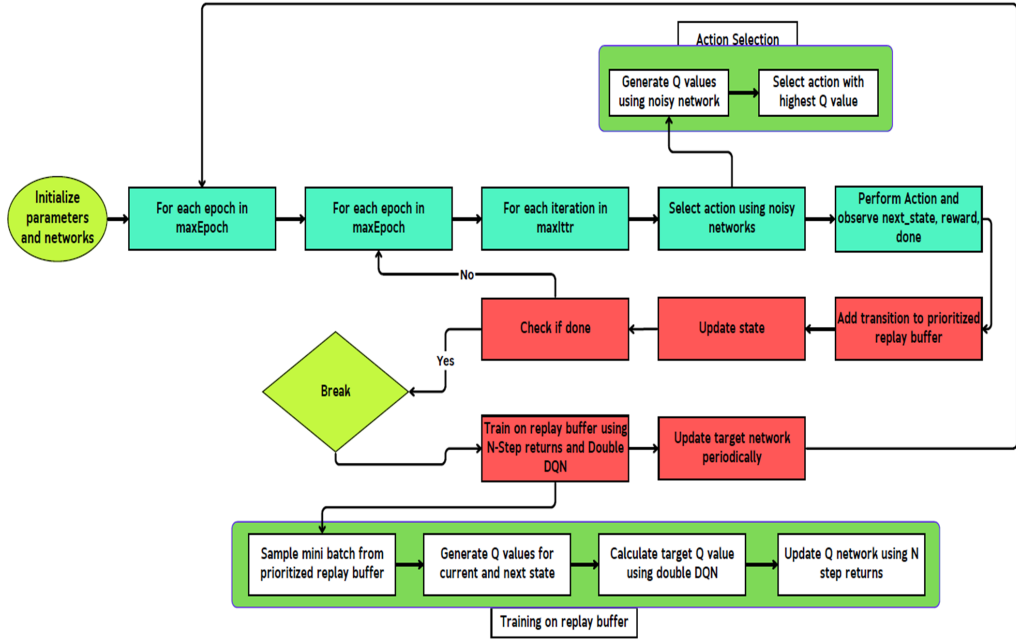


Figure 5: Proposed method workflow

4.1.1. Obstacle avoidance with dynamic obstacles

To improve path planning, the real-time distance between the AUVs location and the time-varying position of each obstacle $q_{\text{obs}}(t)$, enabling collision-free navigation in setting containing dynamic obstacles is necessary. The expression for position of the dynamic obstacle:

$$q_{\text{obs}}(t) = q_{\text{obs}}(t) + v_i t. \quad (31)$$

The position of the dynamic obstacle shifts as time t progresses, according to its speed. Let $q_{\text{obs}}(t)$ denote the changing position of the obstacle. The distance

between the AUV and the obstacle at time t is given by:

$$d_{\text{dynamic}}(X_{ni}, q_{\text{obs}}(t)) = (x_i - (x_{\text{obs}} + v_i t))^2 + (y_i - (y_{\text{obs}} + v_i t))^2 + (z_i - (z_{\text{obs}} + v_i t))^2)^{1/2}. \quad (32)$$

4.1.2. Prioritized experience replay

Prioritized Experience Replay selects significant transitions more often by considering their temporal-difference (TD) error. The likelihood of selecting a transition i is stated as:

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}, \quad (33)$$

where α establishes the degree of prioritizing and p_i is the priority of transition i . The expression for the TD error is:

$$\delta_i = r_t + \gamma \max_{\tau_i} Q(X_{n_{i+1}}, v_{i+1}, \tau_i) - Q(X_{n_i}, v_i, \tau_i). \quad (34)$$

This proposed method guarantees that the educational process emphasizes more impactful experiences, resulting in quicker and more consistent learning.

4.1.3. Exploration-exploitation trade-off

The epsilon-greedy approach is implemented to maintain a balance between exploration and exploitation. The action τ_i is chosen in the following manner:

$$\tau_i = \begin{cases} \text{random action} & \text{with probability } \epsilon, \\ \arg \max_{\tau} Q(X_{n_i}, v_i, \tau) & \text{with probability } 1 - \epsilon. \end{cases} \quad (35)$$

When applied to AUV path planning, this approach enables the AUV to investigate new routes while making use of established effective paths to reach the destination efficiently. AUVs frequently function in continuous environments. The continuous action a_t can be estimated as follows:

$$a_t \approx \arg \max_a Q(s_t, a). \quad (36)$$

4.1.4. Stability and convergence

I-DQN ensures stability during training by using a target Q-Network and Double DQN to reduce overestimation bias. The update rule for Double DQN is given by:

$$Q(X_{n_i}, v_i, \tau_i) \leftarrow Q(X_{n_i}, v_i, \tau_i) + \alpha (r_t + \gamma Q_{\text{target}} - Q(X_{n_i}, v_i, \tau_i)), \quad (37)$$

$$Q_{\text{target}} = Q \left(X_{n_{i+1}}, v_{i+1}, \arg \max_{\tau'} Q \left(X_{n_{i+1}}, v_{i+1}, \tau'; \theta \right); \theta^- \right), \quad (38)$$

where τ' serves as a dummy variable representing all possible actions. This helps in achieving stable and convergent learning.

4.1.5. Hyperparameter tuning

Fine-tuning hyperparameters is vital for the performance of I-DQN. Significant hyperparameters encompass the learning rate α , discount factor γ , epsilon decay rate, and the replay buffer's size. The Q-value function can be altered to integrate the hidden state h_t :

$$Q(X_{n_i}, v_i, \tau_i, h_t) = V(h_t) + \left(A(X_{n_i}, v_i, \tau_i) - \frac{1}{|\mathcal{A}|} \sum_{\tau'} A(X_{n_i}, v_i, \tau') \right). \quad (39)$$

This allows the AUV to maintain a memory of past observations and make informed decisions in partially observable environments.

5. Simulation result

This section showcases the simulation outcomes utilized for training the model and assessing the performance of the proposed method. The conventional approaches employed for comparison with the proposed method include DQN, DDQN, and D3QN. Path planning was executed in the Koel River, located in Rourkela, while considering diverse obstacle-rich areas along with various start and target points to validate the proposed method. Figure 6a illustrates the AUV used and Figure 6b once it is deployed in an underwater environment for distinct scenarios which are examined. Figure 6c–6e, is the experimental site 1, 2 and 3. Obstacles can be both dynamic and static.

The real-time path planning task in a dynamic environment was successfully addressed using the DQN with carefully fine-tuned parameters. The information shown in Table 1 is accurately utilized in the experiments. QGC and Rviz software aid in documenting the AUVs operational status and in the real-time creation of desired trajectories. The information obtained from the IMU and GPS modules of Ardupilot undergo processing and fusion through Kalman filtering, before being transmitted to robot operating system (ROS) via the mavros node. The guiding program node and the path generation program node can both receive the AUV's positional and attitudinal data from ROS, which consequently generates control commands to be relayed to the underlying controller through asynchronous communication. The design of the AUV software control system consists of the Ardupilot, ROS, and Q-Ground Control (QGC), as shown in Figure 7.



(a) AUV used



(b) After deploying the AUV



(c) Map 1



(d) Map 2



(e) Map 3

Figure 6: Visualization of the AUV used and operational scenario

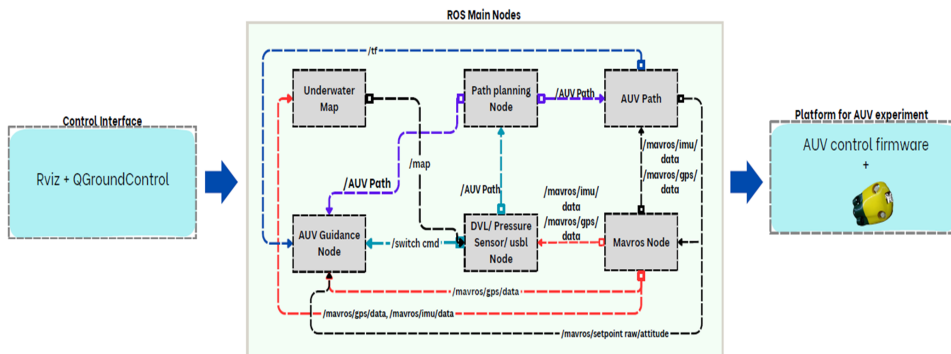


Figure 7: Implementation of the AUV control system

Table 1: Parameters and their values used

Parameter	Value	Parameter	Value	Parameter	Value
maxEpoch	1000	gamma	0.99	epsilonDecay	0.9
maxIttr	900	epsilon	1.0	bufferSize	10000
sampleSize	64	actions	8 possible movements	Reward for reaching target	1000
Reward for hitting obstacles	-100	Reward for moving closer to target	10	Reward for moving away from target	-1
Reward for following reference path	5	Penalty for revisiting the same position	-10	Penalty for being near obstacles	-5
Penalty for each step	-0.1				

5.1. Performance evaluation of proposed method

Figure 8 depicts the nodes within the map 1 that were generated during the training of various reinforcement learning algorithms, such as DQN, DDQN, D3QN, and I-DQN. These algorithms progressively refine the Q-values by relying on the agent's experiences within the environment, resulting in the exploration and mapping of the state space. The unoccupied area where only water and underwater obstacles exist is currently being investigated by the agent. Beyond this area lie the riverbanks, which are distinctly illustrated in Figure 8. The nodes signify the states the agent has encountered, illustrating the effectiveness and extent of the exploration process. The variations in the explored regions underscore the influence of each algorithm's distinct strategy for learning and decision-making, with I-DQN harnessing the advantages of DQN, DDQN, and D3QN to accomplish a more thorough exploration. In Figure 8, we utilized three separate maps for the path planning of the suggested algorithm in conjunction with the alternative RL algorithm. Each evaluation is carried out live, and the quantity and number of obstacle changes for every map. The starting points and destinations are fixed and differ across the maps. The location nearest to the riverbank is selected for the operation and management of the AUV. Figure 9 illustrates the training progress of various agents from the algorithms over 1000 iterations, evaluating the cumulative rewards for each specific agent and the real time visuals of the exploration of the agents are shown in Figure 8. The stability strategies, such as target networks and experience replay, appear to be adequate for tackling stability issues in reinforcement learning.

Figure 10 depict the evolution of Q-values for different reinforcement learning algorithms over the course of 1000 iterations, revealing unique patterns. Figure 10a illustrates convergence towards positive Q-values between 500 and

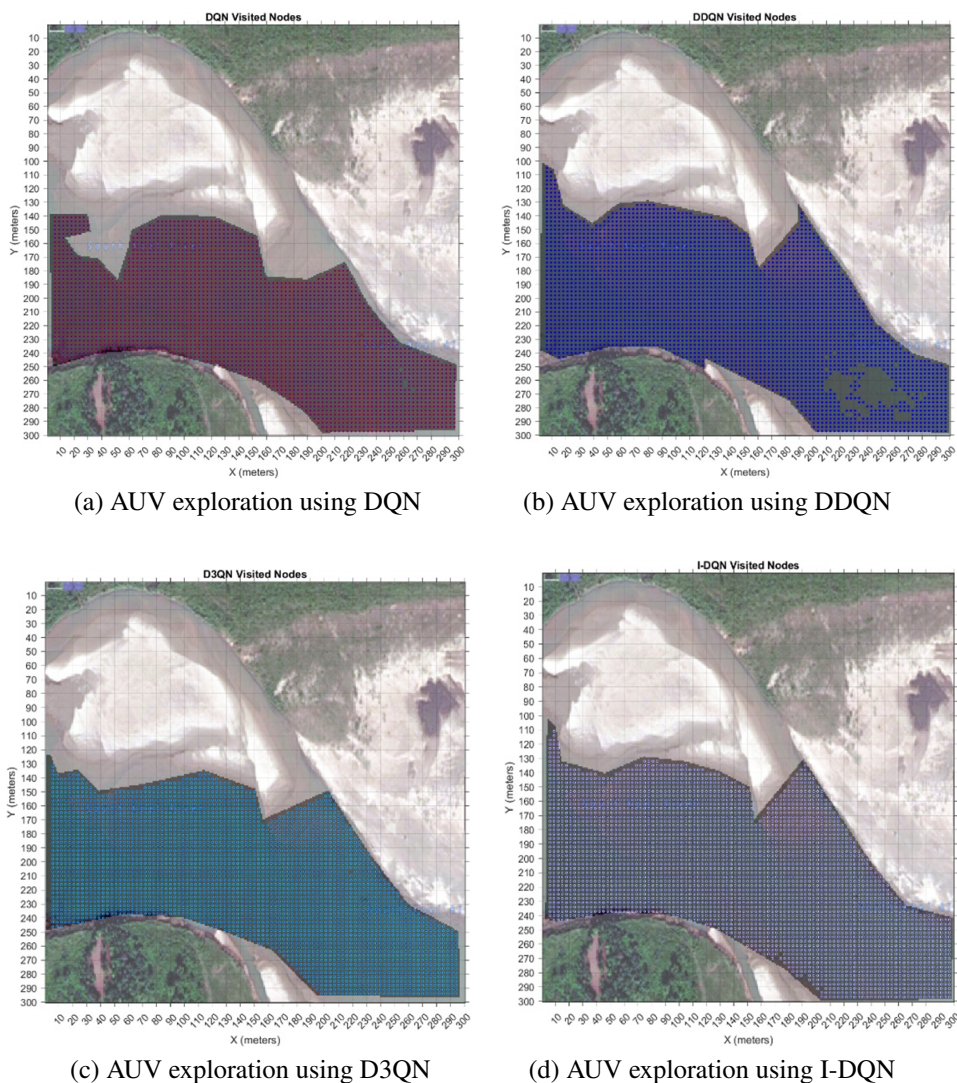
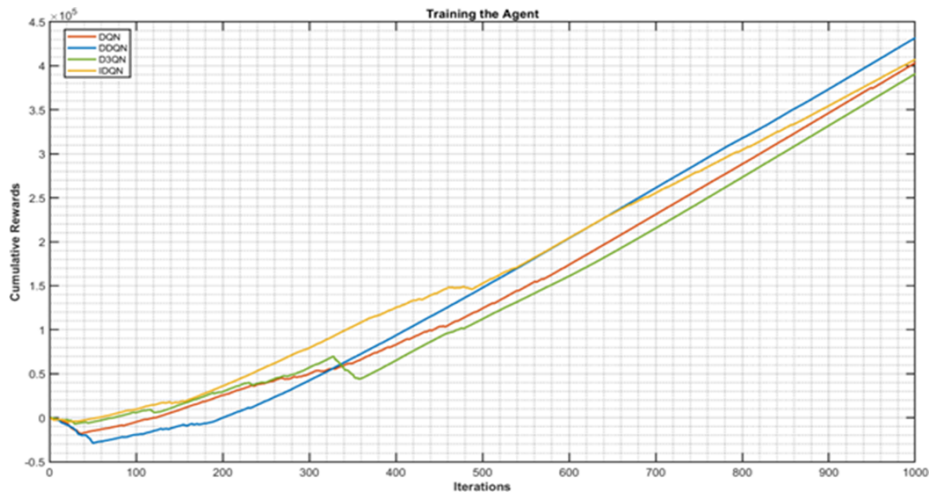


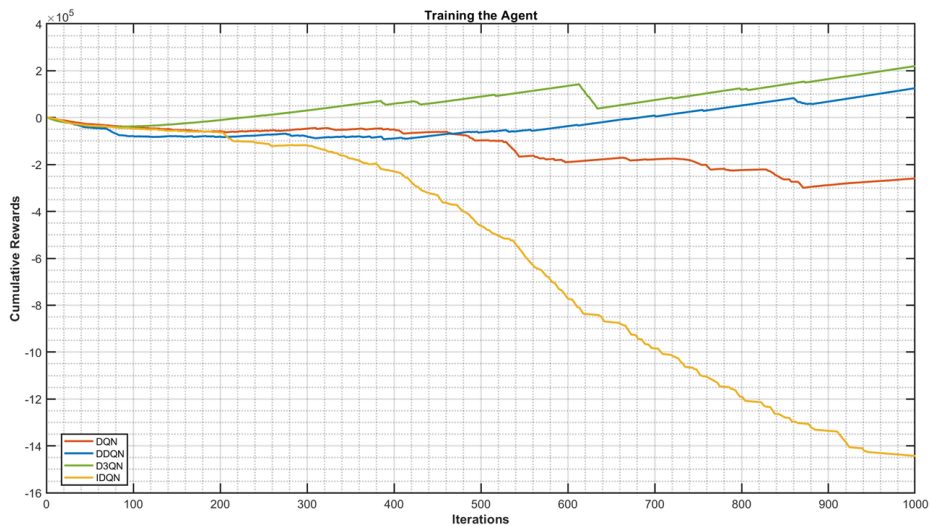
Figure 8: AUV exploration while training in Map 1

600 for all algorithms, while Figure 10b exhibits increased volatility with regular declines into negative values, and 10c demonstrates a pronounced divergence among the algorithms. Positive Q-values signify the agent's anticipation of future rewards, indicating that effective strategies have been acquired, whereas negative Q-values suggest expected penalties or unfavorable outcomes, which may reflect inadequate learning or difficult environments. I-DQN (yellow line) demonstrates a more consistent and stable set of Q-values throughout the iterations when compared to the other algorithms, exhibiting fewer extreme fluctuations. In Fig-

ure 10b and Figure 10c, although I-DQN still shows some variability, it generally converges to higher Q-values more swiftly and sustains a steadier range in the later iterations. This implies that I-DQN might be acquiring a more robust value function, which could result in enhanced and more reliable performance within the specified environments. The diminished volatility in Q-values for I-DQN may suggest better stability in the learning process along with more dependable decision-making by the agent.

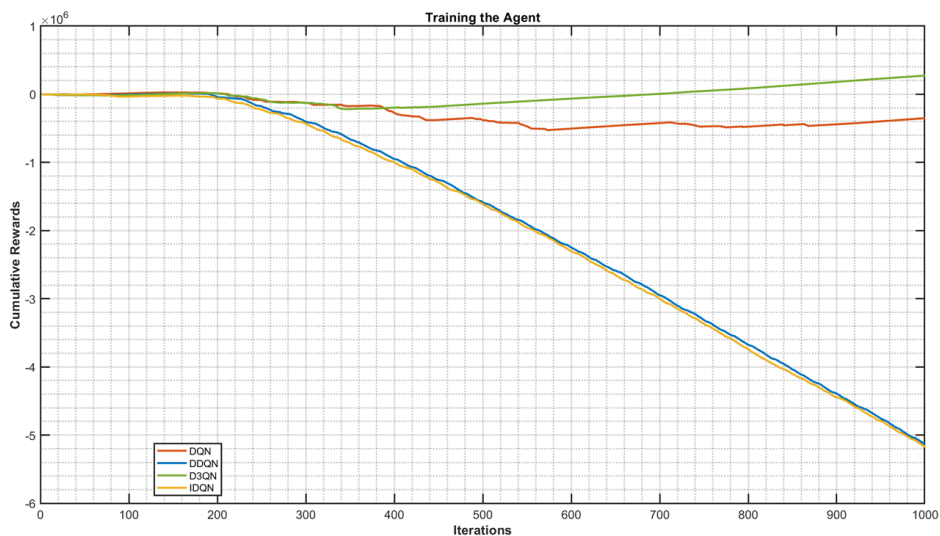


(a) Training graph of AUV in Map 1



(b) Training graph of AUV in Map 2

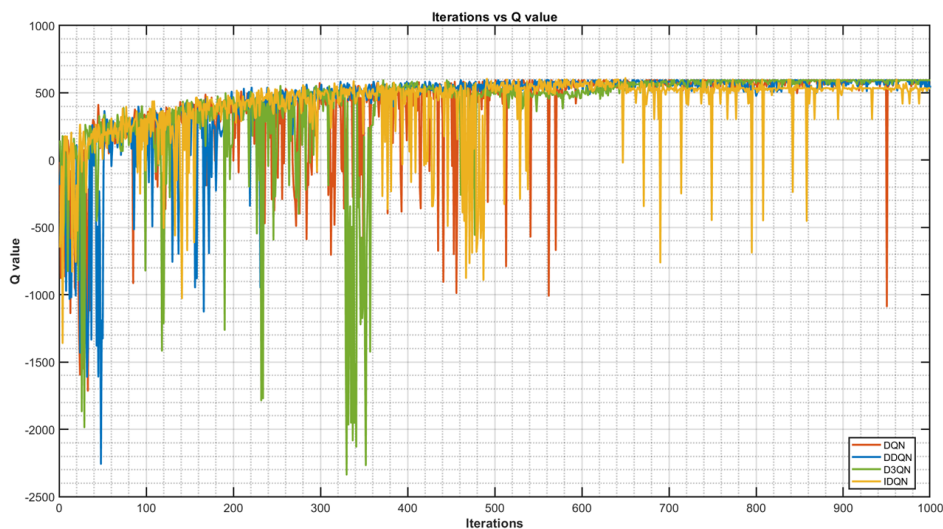
Fig. 9 a, b



(c) Training graph of AUV in Map 3

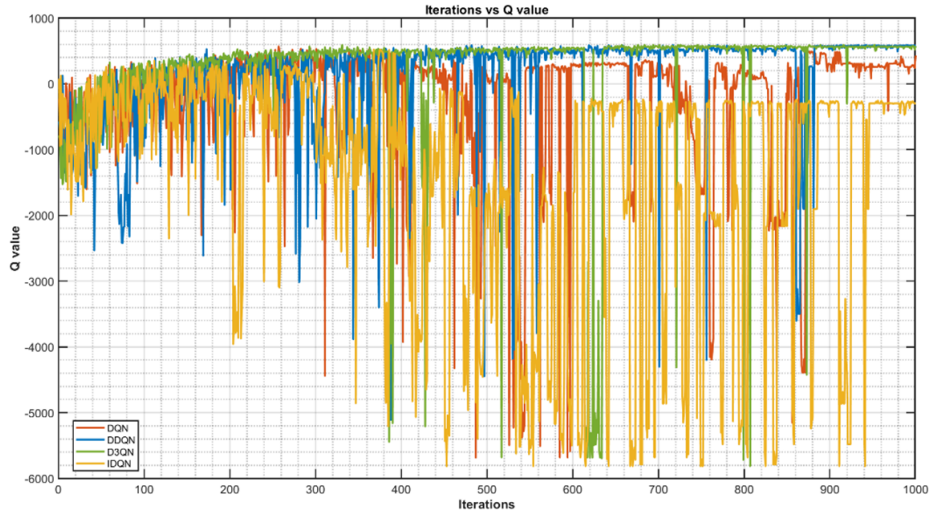
Figure 9: Graphical analysis of the training cost of the AUV across different scenario

These varied outcomes underscore the distinct performance of the algorithms across different contexts, highlighting how anticipated future rewards can vary considerably during the learning process based on the task or setting.

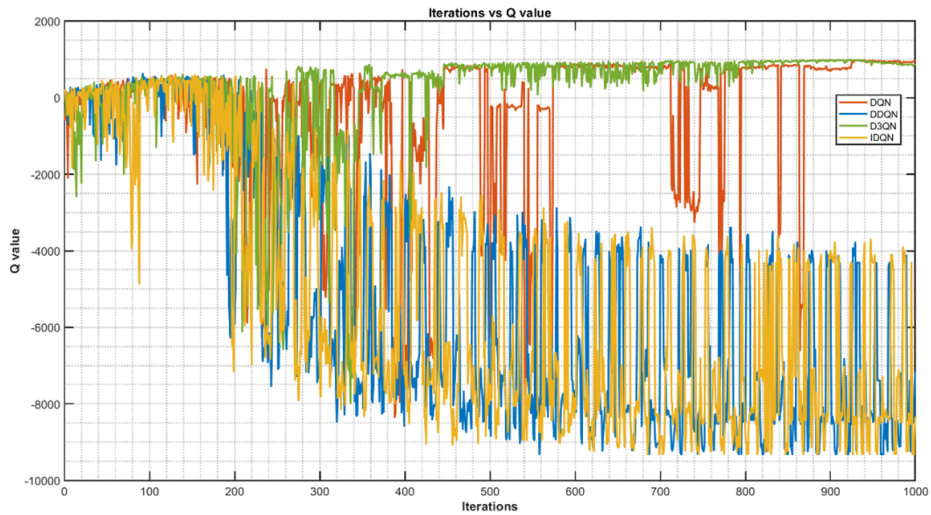


(a) Q-values obtained in Map 1

Fig. 10 a



(b) Q-values obtained in Map 2



(c) Q-values obtained in Map 3

Figure 10: Graphical comparison of Q values

5.2. Performance evaluation of proposed method in static environment

Figure 11, provide a comparative evaluation of various path planning techniques applied to a 300×300 meter terrain map, where each grid unit corresponds to 1 meter. All techniques navigate from a starting point (S) to a target point (T) within this designated area. These findings underscore the differing levels of exploration and efficiency among the various reinforcement learning methodologies for path planning in a complex 300×300 m terrain.

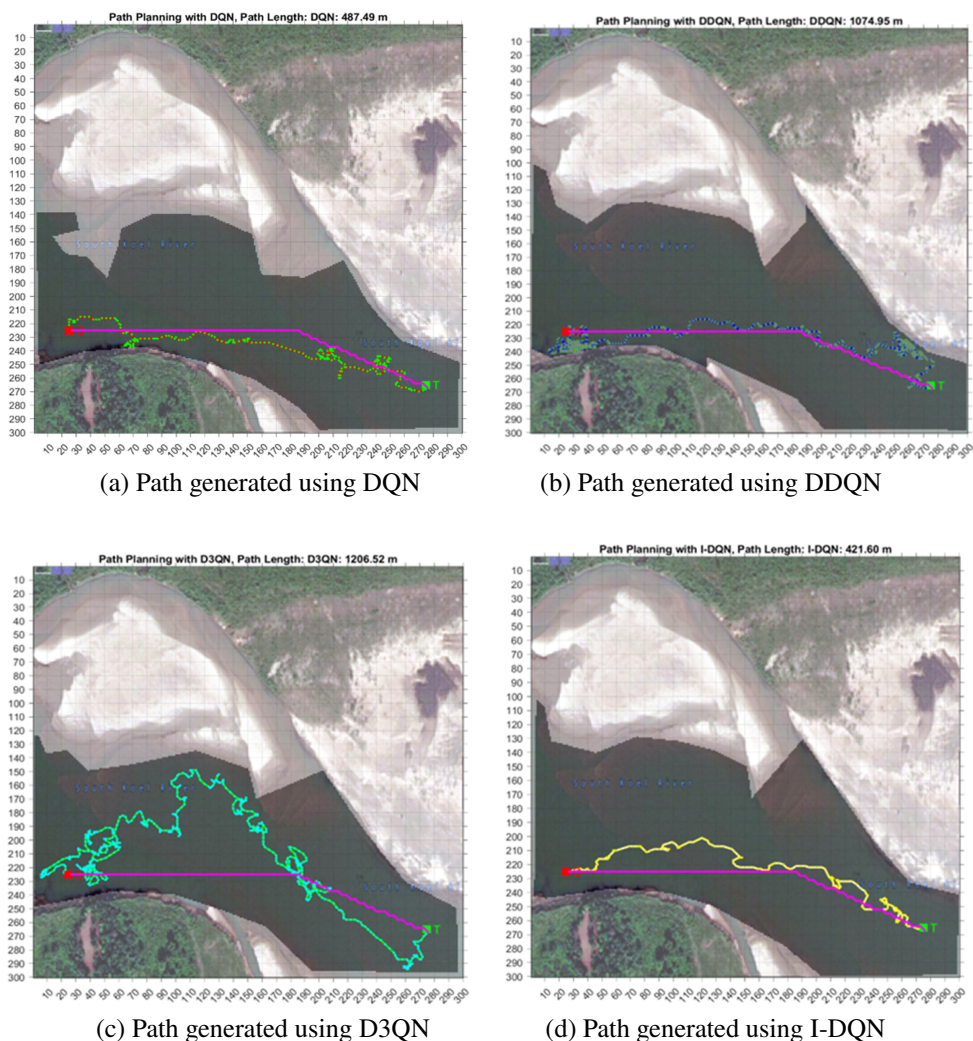


Figure 11: Visualization of path planning in map 1 using 4 RL algorithm

Figure 12 illustrates the approach to path planning utilizing multiple iterations of the different algorithm within a coastal framework. The maps provide an aerial perspective of a shoreline that encompasses cliffs, aquatic environments, and areas of vegetation. Each image highlights a distinct path (represented as a colored line) generated by a specific DQN in map 2. The algorithms endeavor to ascertain the most effective routes while avoiding obstacles and maneuvering through challenging terrains. The pathways exhibit variations in their strategic approaches, with some favoring more direct trajectories along the coastline and others extending further into the aquatic region. The reference trajectory (illus-

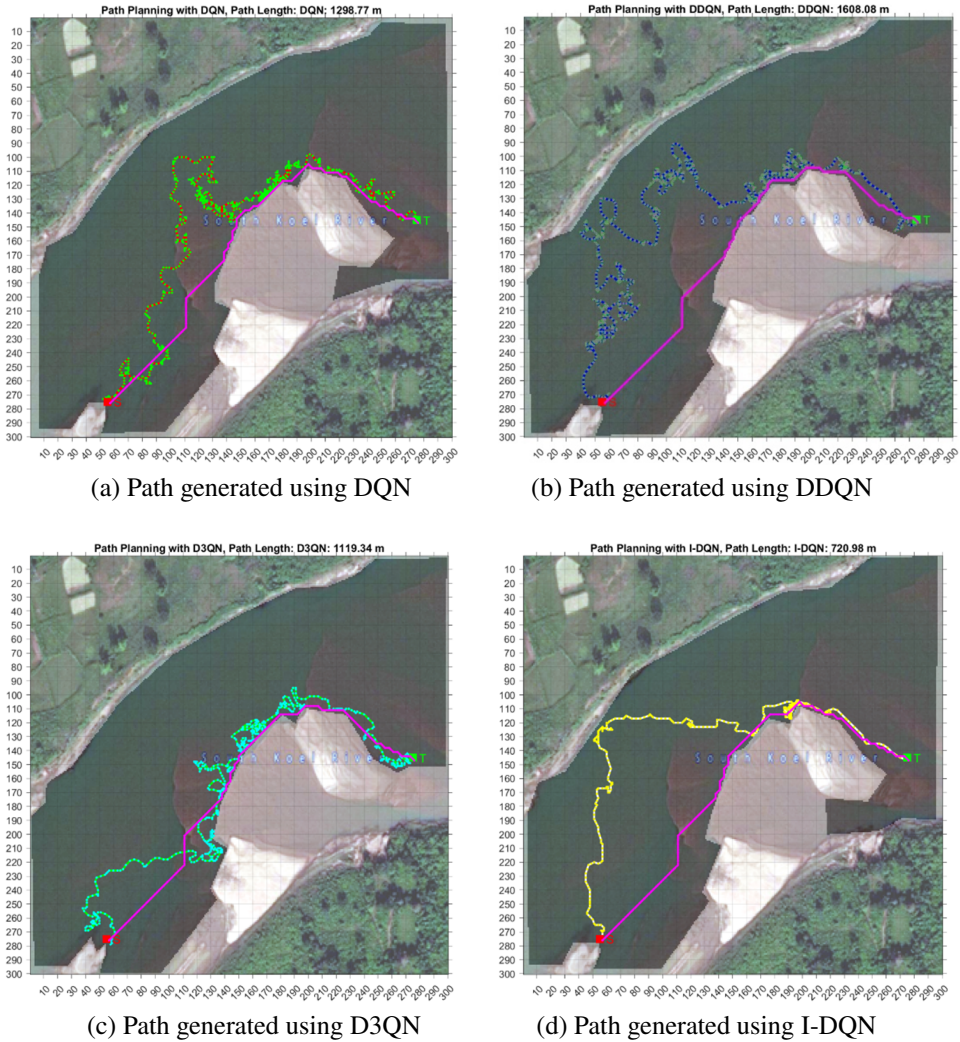


Figure 12: Visualization of path planning in map 2 using 4 RL algorithm

trated in magenta) serves as a benchmark for evaluation, facilitating an analysis of each algorithm's performance in terms of efficiency and obstacle avoidance.

Figure 13 illustrates the results obtained from testing all the algorithms on map 3. From Figure 11, 12, and 13, it is evident that the I-DQN path, indicated in yellow in the fourth image, shows greater efficiency in comparison to the other algorithms. I-DQN consistently achieves the shortest path lengths across all obstacle scenarios, unlike DQN, DDQN, and D3QN. This suggests that the I-DQN may have heightened effectiveness in locating optimal paths, potentially offering improved navigation or routing solutions in contexts where distance re-

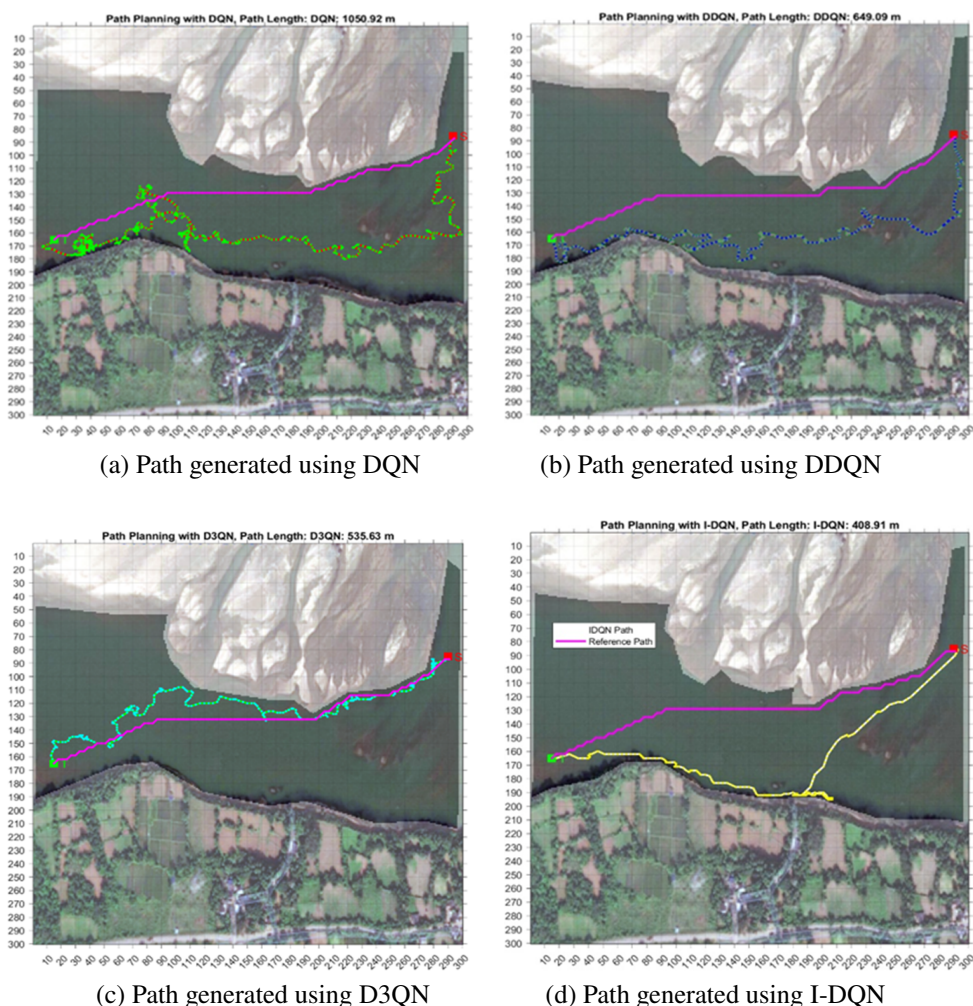


Figure 13: Visualization of path planning in map 3 using 4 RL algorithm

duction is critical. The aerial visualization presented signifies the outcomes of the algorithm within a underwater environment, with the 2D maps serving as visual depictions of the complex underwater landscape. While the actual setting is three-dimensional and submerged, the maps are utilized to convey this information in a 2D format for better understanding. These outcomes display the AUV's planned paths and the position of obstacles, showcasing their navigation through a simulated underwater setting. The traced paths illustrate the algorithm's choices to circumvent underwater obstacles and effectively reach designated targets, thereby highlighting its proficiency in real-time underwater navigation situations. The comparative evaluation of diverse algorithms shown

in Table 2 highlights their performance across different maps, with emphasis on metrics like best, worst, average, and standard deviation. The I-DQN algorithm demonstrates significantly better performance, especially in reducing path length, resulting in a more efficient solution for navigation tasks, particularly in complex environments laden with obstacles and requiring path planning.

Table 2: Comparative analysis of the 4 RL algorithm

Metric	DQN	DDQN	D3QN	I-DQN
Best				
Map 1	240	697	603	204
Map 2	609	337	310	198
Map 3	754	851	638	523
Average				
Map 1	898	999	900	892
Map 2	899	998	888	800
Map 3	900	1000	899	750
Worst				
Map 1, 2, 3	901	901	901	701
Standard deviation				
Map 1	7.35	20.91	14.93	5.93
Map 2	19.93	39.96	64.69	4.41
Map 3	36.73	6.12	38.48	3.82

Table 3, illustrates the path lengths of four algorithms: DQN, DDQN, D3QN, and I-DQN across three different map instances. The I-DQN algorithm consistently records the shortest path lengths in every instance, showcasing its superior efficiency in optimizing paths when compared to the other algorithms. This emphasizes the benefits of I-DQN in efficient navigation and route planning.

Table 3: Path length comparison (m)

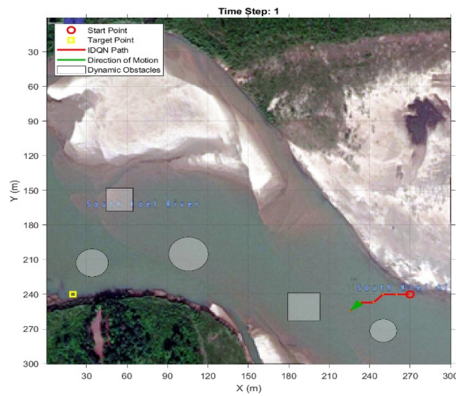
Instances	DQN	DDQN	D3QN	I-DQN
Map 1	1108.59	1206.52	1074.95	421.60
Map 2	1050.92	649.09	535.63	408.91
Map 3	1608.08	1298.77	1119.34	720.98

5.3. Performance evaluation of proposed method in unknown and unstructured environment

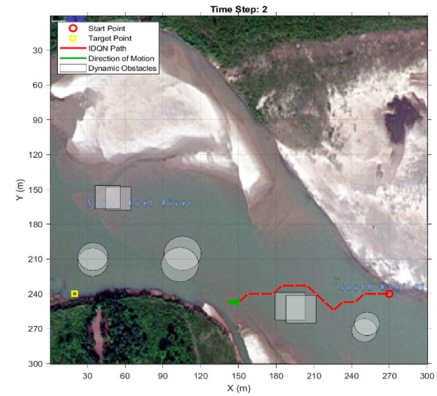
This situation becomes increasingly complex as we acknowledge the unpredictable nature of both static and dynamic obstacles within the variable spatiotemporal current field. The identified obstacles in this context are delineated using circular and square markers in accordance with the configuration of the dynamic impediments. Their positions are determined randomly based on the mean and variance of their uncertainty distribution, with their velocities being directly proportional to the current velocity. The uncertainty associated with both static and dynamic obstacles is linearly propagated in relation to the time updates, resulting in an expansion of the radius of the static obstacles and concurrent alterations in the position and radius of the dynamic obstacles.

The enhancement of I-DQN for path planning in an unstructured environment emphasizes the augmentation of the number of epochs from 1000 to 2000 and iterations from 900 to 3000, thereby facilitating a more thorough acquisition of knowledge regarding intricate scenarios. Concurrently, the reduction of the learning rate from 1.4 to 0.01 fosters a more stable learning experience, permitting the algorithm to adapt proficiently to fluctuating environments without exceeding optimal solutions. These modifications collectively bolster the algorithm's ability to devise efficient pathways while circumventing obstacles within dynamic contexts. Figure 14, 15, and 16 depict the trajectories generated by the proposed methodologies that successfully circumvent dynamic obstacles. Figure 14, 15, 16 (Time Step: 1) illustrates the initial path produced by the proposed methodology. It is imperative to note that the path generated by I-DQN, denoted by the red dashed line, does not collide with the neighboring coast at the commencement of the mission, despite potential visual implications in the 2D representation. The uncertainty associated with the obstacles is distinctly observable in the subsequent Time Steps: 2–4 of Figure 14, 15, 16. The methodology suggested shows a strong potential for ensuring a path that avoids collisions amidst varying obstacles.

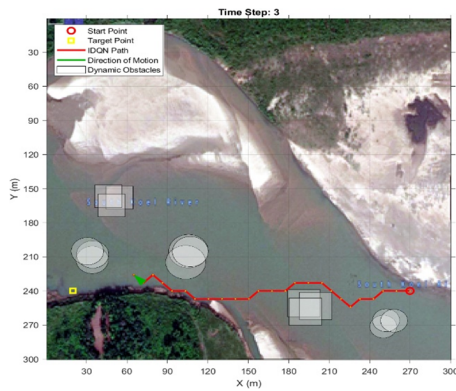
The comparative assessment of various algorithms presented in Table 4 accentuates their performance across a range of maps, with particular focus on metrics such as best, worst, average, and standard deviation. The I-DQN algorithm exhibits markedly superior performance, particularly in the minimization of path length, thereby yielding a more efficient resolution for navigation tasks, especially within an unstructured environment fraught with dynamic obstacles necessitating path planning. Table 5 delineates the path lengths of four algorithms: DQN, DDQN, D3QN, and I-DQN across three distinct map instances. The I-DQN algorithm uniformly records the shortest path lengths in each instance, thereby demonstrating its exceptional efficiency in path optimization compared to the



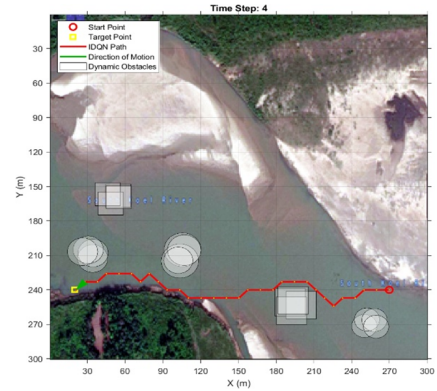
(a) Path generated using I-DQN at time step 1



(b) Path generated using I-DQN at time step 2

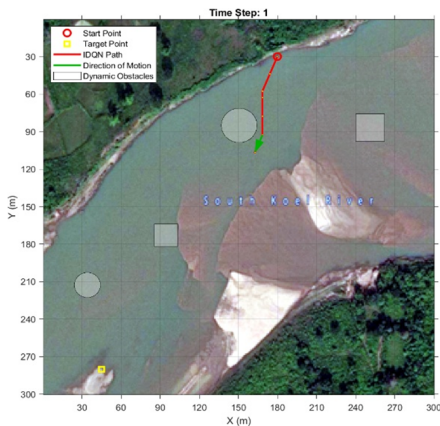


(c) Path generated using I-DQN at time step 3

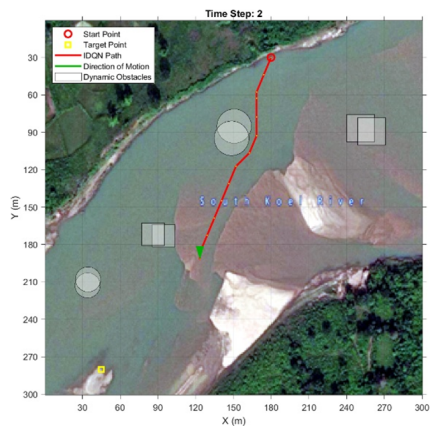


(d) Path generated using I-DQN at time step 4

Figure 14: Path planning in an unstructured environment on map 1 using I-DQN

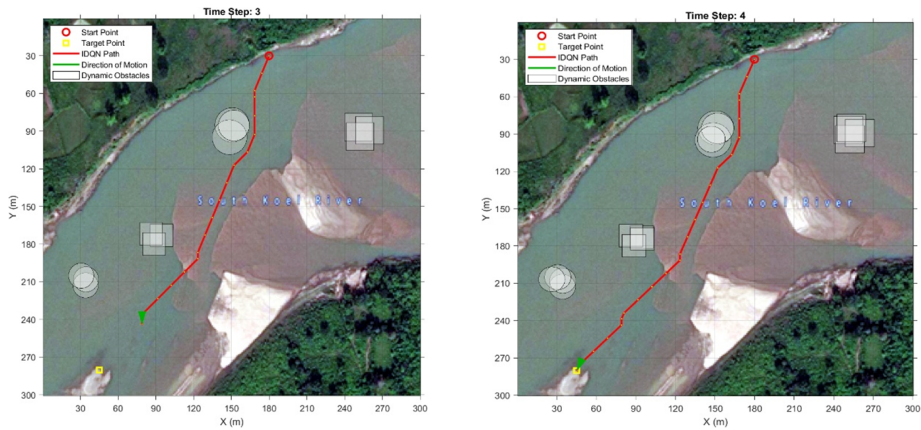


(a) Path generated using I-DQN at time step 1



(b) Path generated using I-DQN at time step 2

Fig. 15 a, b



(c) Path generated using I-DQN at time step 3

(d) Path generated using I-DQN at time step 4

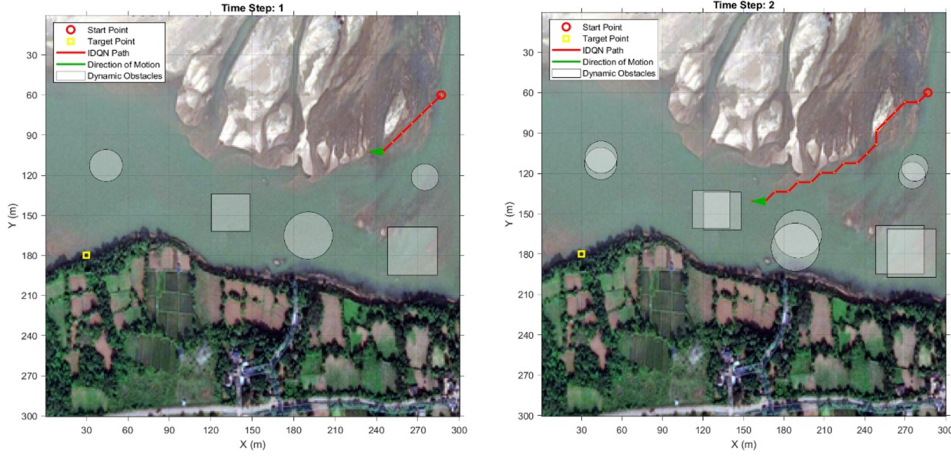
Figure 15: Path planning in an unstructured environment on map 2 using I-DQN

Table 4: Comparative analysis of the 4 RL algorithm in an unstructured environment

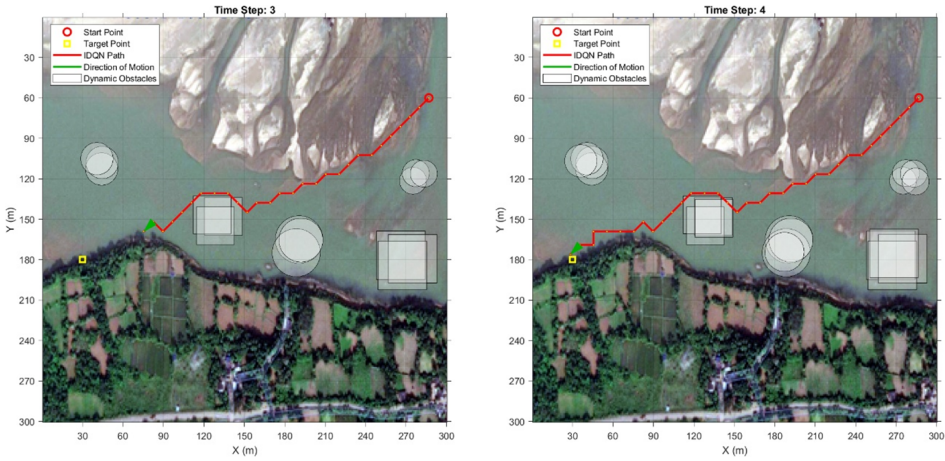
Metric	DQN	DDQN	D3QN	I-DQN
Best				
Map 1	289	715	705	256
Map 2	695	428	389	213
Map 3	789	932	705	589
Average				
Map 1	900	999	915	897
Map 2	936	998	896	856
Map 3	912	999	914	796
Worst				
Map 1, 2, 3	925	925	925	738
Standard deviation				
Map 1	9.48	23.99	15.99	6.97
Map 2	23.98	42.99	76.90	6.49
Map 3	42.86	9.36	49.59	4.96

Table 5: Path length comparison (m) in an unstructured environment

Instances	DQN	DDQN	D3QN	I-DQN
Map 1	563.49	1145.99	1368.75	421.60
Map 2	1113.92	752.23	549.78	517.16
Map 3	1369.59	1786.88	1256.85	769.86



(a) Path generated using I-DQN at time step 1 (b) Path generated using I-DQN at time step 2



(c) Path generated using I-DQN at time step 3 (d) Path generated using I-DQN at time step 4

Figure 16: Path planning in an unstructured environment on map 3 using I-DQN

other algorithms. This underscores the advantages of I-DQN in effective navigation and route planning within unstructured environments or amidst dynamic obstacles.

6. Conclusions

In this study, the I-DQN algorithm is introduced to tackle issues such as lengthy movement paths, excessive overlapping coverage, and heightened energy consumption of the AUV while applying RL algorithms for coverage tasks. When the AUV comes across an area lacking coverage, I-DQN calculates a route

that avoids collisions throughout the entire map, steering the AUV towards the most efficient coverage target cell. This allows the AUV to maneuver along the path effectively and quickly access uncovered regions, leading to more streamlined and smoother trajectories. Additionally, I-DQN adjusts local path planning to reduce overlapping coverage and energy expenditure. The redistribution of the neural network within the workspace significantly reduces the unnecessary computational requirements. Extensive experimental outcomes illustrate that the proposed algorithm accomplishes shorter path lengths during planning in an unstructured environment, reduces redundant coverage, and enhances operational efficiency. It also possesses the ability to escape both static obstacles and dynamic obstacles on the map in real-time, rendering it both practical and flexible. Nevertheless, the proposed algorithm is designed for a single AUV and functions as a centralized algorithm. Given its advantages in energy efficiency and computational performance, along with its broad applicability, future research will explore the deployment of the I-DQN algorithm across multiple AUVs. The proposed technique can be used to better underwater search and rescue efforts in disaster-affected areas or swamped urban zones. The AUVs can navigate through complex and unfamiliar underwater environments, dodging debris and obstacles to quickly and efficiently locate survivors or assess structural damage.

References

- [1] O.A. YAKIMENKO, D.P. HORNER, and D.G. PRATT: AUV rendezvous trajectories generation for underwater recovery. In *2008 IEEE 16th Mediterranean Conference on Control and Automation*, (2008), 1192–1197. DOI: [10.1109/med.2008.4601975](https://doi.org/10.1109/med.2008.4601975)
- [2] A.M. YAZDANI, K. SAMMUT, O.A. YAKIMENKO, A. LAMMAS, Y. TANG, and S.M. ZADEH: IDVD-based trajectory generator for autonomous underwater docking operations. *Robotics and Autonomous Systems*, **92**, (2017), 12–29. DOI: [10.1016/j.robot.2017.02.001](https://doi.org/10.1016/j.robot.2017.02.001)
- [3] S.P. SAHOO, B. DAS, B.B. PATI, F.P. GARCIA MARQUEZ, and I. SEGOVIA RAMIREZ: Hybrid Path Planning Using a Bionic-Inspired Optimization Algorithm for Autonomous Underwater Vehicles. *Journal of Marine Science and Engineering.*, **11**, (2023), 761. DOI: [10.3390/jmse11040761](https://doi.org/10.3390/jmse11040761)
- [4] B. DAS, B. SUBUDHI, and B.B. PATI: Formation control of underwater vehicles using Multi Agent System. *Archives of Control Sciences*, **30**(2), (2020), 365–384. DOI: [10.24425/acs.2020.133503](https://doi.org/10.24425/acs.2020.133503)
- [5] Z. ZENG, K. SAMMUT, L. LIAN, F. HE, A. LAMMAS, and Y. TANG: A comparison of optimization techniques for auv path planning in environments with ocean currents. *Robotics and Autonomous Systems*, **82**, (2016), 61–72. DOI: [10.1016/j.robot.2016.03.011](https://doi.org/10.1016/j.robot.2016.03.011)
- [6] C. CHENG, Q. SHA, B. HE, and G. LI: Path planning and obstacle avoidance for auv: A review. *Ocean Engineering*, **235**, (2021), 109355. DOI: [10.1016/j.oceaneng.2021.109355](https://doi.org/10.1016/j.oceaneng.2021.109355)

- [7] B. DAS, B. SUBUDHI, and B.B. PATI: Co-operative control of a team of autonomous underwater vehicles in an obstacle-rich environment. *Journal of Marine Engineering & Technology*, **15**(3), (2016), 135–151. DOI: [10.1080/20464177.2016.1247636](https://doi.org/10.1080/20464177.2016.1247636)
- [8] B. DAS, B. SUBUDHI, and B.B. PATI: Employing nonlinear observer for formation control of AUVs under communication constraints. *International Journal of Intelligent Unmanned Systems*, **3**(2/3), (2015), 122–155. DOI: [10.1108/ijius-04-2015-0004](https://doi.org/10.1108/ijius-04-2015-0004)
- [9] H. WANG, D. ZHAO, X. BIAN, and X. SHI: Research on autonomous planning for auv in unstructured environment. In *Proceedings of the Intelligent Computing: International Conference on Intelligent Computing, ICIC 2006*, Kunming, China, August 16–19, 2006, Part I 2, Springer, (2006), 586–599. DOI: [10.1007/11816157_71](https://doi.org/10.1007/11816157_71)
- [10] J.D. HERNÁNDEZ, E. VIDAL, G. VALLICROSA, E. GÁLGERAN, and M. CARRERAS: Online path planning for autonomous underwater vehicles in unknown environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, (2015), 1152–1157. DOI: [10.1109/icra.2015.7139336](https://doi.org/10.1109/icra.2015.7139336)
- [11] H. MIN, X. XIONG, P. WANG, and Y. YU: Autonomous driving path planning algorithm based on improved a* algorithm in unstructured environment. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, **235**(2–3), (2021), 513–526. DOI: [10.1177/0954407020959741](https://doi.org/10.1177/0954407020959741)
- [12] B. DAS, B. SUBUDHI, and B.B. PATI: Co-operative control coordination of a team of underwater vehicles with communication constraints. *Transactions of the Institute of Measurement and Control*, **38**(4), (2016), 463–481. DOI: [10.1177/0142331215590010](https://doi.org/10.1177/0142331215590010)
- [13] P. SINGH, B.J. TALUKDAR, B. DAS, and S.K. KHUNTIA: A path planning algorithm for multi-AUVs operating in unstructured underwater terrain environments. *Ships and Offshore Structures*, 1–30. DOI: [10.1080/17445302.2025.2598302](https://doi.org/10.1080/17445302.2025.2598302)
- [14] B.J. TALUKDAR, B.B. PATI, and B. DAS: Path planning algorithm for AUVs operating in unstructured underwater environments. *Transactions of the Institute of Measurement and Control*, (2025). DOI: [10.1177/01423312251376778](https://doi.org/10.1177/01423312251376778)
- [15] I. SEGOVIA RAMIREZ, B. DAS, and F.P. GARCIA MARQUEZ: Fault detection and diagnosis in photovoltaic panels by radiometric sensors embedded in unmanned aerial vehicles. *Progress in Photovoltaics: Research and Applications*, **30**(3), 240–256. DOI: [10.1002/pip.3479](https://doi.org/10.1002/pip.3479)
- [16] J. FAN and L. QU: Innovative differential evolution algorithm with double-layer coding for autonomous underwater vehicles path planning in complex environments, *Ocean Engineering*, **303**, (2024), 117806. DOI: [10.1016/j.oceaneng.2024.117806](https://doi.org/10.1016/j.oceaneng.2024.117806)
- [17] J. ZHANG, Z. ZHU, Y. XUE, Z. DENG, and H. QIN: Local path planning of under-actuated auv based on vadwa considering dynamic model. *Ocean Engineering*, **310**, (2024), 118705. DOI: [10.1016/j.oceaneng.2024.118705](https://doi.org/10.1016/j.oceaneng.2024.118705)
- [18] B. SUN and N. NIU: Multi-auvs cooperative path planning in 3d underwater terrain and vortex environments based on improved multi-objective particle swarm optimization algorithm, *Ocean Engineering*, **311**, (2024), 118944. DOI: [10.1016/j.oceaneng.2024.118944](https://doi.org/10.1016/j.oceaneng.2024.118944)
- [19] R. KOT, P. SZYMAK, P. PISKUR, and K. NAUS: A comparative study of different collision avoidance systems with local path planning for autonomous underwater vehicles, *IEEE Access*, **12**, (2024), 61443–61466. DOI: [10.1109/access.2024.3394569](https://doi.org/10.1109/access.2024.3394569)

- [20] G. BAI, Y. CHEN, X. HU, and J. CUI: Correction redistribution mechanism based on forward-reverse solutions and real-time path dynamic adaptive re-planning for multi-auvs collaborative search. *IEEE Transactions on Intelligent Transportation Systems*, **25**(7), (2024), 7583–7601. DOI: [10.1109/tits.2024.3352902](https://doi.org/10.1109/tits.2024.3352902)
- [21] X. YAN, W. WANG, C. HUANG, and L. LI: A new path planning method for auv based on the navier–stokes equations for ocean currents, *Mathematics and Computers in Simulation*, **222**, (2024), 199–208. DOI: [10.1016/j.matcom.2023.08.030](https://doi.org/10.1016/j.matcom.2023.08.030)
- [22] Y. ZHANG, Y. SHEN, Q. WANG, C. SONG, N. DAI, and B. HE: A novel hybrid swarm intelligence algorithm for solving tsp and desired path-based online obstacle avoidance strategy for auv, *Robotics and Autonomous Systems*, **177**, (2024), 104678. DOI: [10.1016/j.robot.2024.104678](https://doi.org/10.1016/j.robot.2024.104678)
- [23] H. FENG, Q. HU, Z. ZHAO, and X. FENG: Smooth path planning under maximum curvature constraints for autonomous underwater vehicles based on rapidly-exploring random tree star with b-spline curves. *Engineering Applications of Artificial Intelligence*, **133**, (2024), 108583. DOI: [10.1016/j.engappai.2024.108583](https://doi.org/10.1016/j.engappai.2024.108583)
- [24] L. ZHI and Y. ZUO: Collaborative path planning of multiple auvs based on adaptive multi-population pso, *Journal of Marine Science and Engineering*, **12**(2), (2024), 223. DOI: [10.3390/jmse12020223](https://doi.org/10.3390/jmse12020223)
- [25] S. ZHANG, K. LI, R. HE, Z. MENG, Y. CHANG, X. JIN, and R. BAI: Trajectory planning for autonomous driving in unstructured scenarios based on graph neural network and numerical optimization, *arXiv preprint arXiv: 2406.08855*, (2024). DOI: [10.48550/arXiv.2406.08855](https://doi.org/10.48550/arXiv.2406.08855)
- [26] X. CAO, J. PENG, W. LIU, L. REN, and C. SUN: Auv collision avoidance strategy based on fuzzy reinforcement learning, *IEEE Transactions on Intelligent Vehicles*, **10**(3), (2025), 1568–1580. DOI: [10.1109/tiv.2024.3429500](https://doi.org/10.1109/tiv.2024.3429500)
- [27] B. HADI, A. KHOSRAVI, and P. SARHADI: Deep reinforcement learning for adaptive path planning and control of an autonomous underwater vehicle, *Applied Ocean Research*, **129**, (2022), 103326. DOI: [10.1016/j.apor.2022.103326](https://doi.org/10.1016/j.apor.2022.103326)
- [28] B.J. TALUKDAR, A. PATRA, R.N. DASH, and B. DAS: Adaptive Q-learning for Path Planning of AUV in Obstacle-Dense Environments, In *2025 3rd IEEE International Conference on Industrial Electronics: Developments & Applications (ICIDeA)*, Bhubaneswar, India, (2025), 1–6, DOI: [10.1109/ICIDeA64800.2025.10963065](https://doi.org/10.1109/ICIDeA64800.2025.10963065)
- [29] B.J. TALUKDAR, B.B. PATI, and B. DAS: Efficient algorithms for navigation of underwater vehicles with communication constraints. An overview. *Archives of Control Sciences*, **35**(1), (2025), 47–87. DOI: [10.24425/acs.2025.153958](https://doi.org/10.24425/acs.2025.153958)
- [30] S. HAN, J. ZHAO, X. LI, J. YU, S. WANG, and Z. LIU: Online path planning for auv in dynamic ocean scenarios: A lightweight neural dynamics network approach, *IEEE Transactions on Intelligent Vehicles*, **9**(2), (2024), 3782–3795. DOI: [10.1109/TIV.2024.3356529](https://doi.org/10.1109/TIV.2024.3356529)
- [31] T.I. FOSSEN: *Guidance and control of ocean vehicles*, University of Trondheim, Norway, Printed by John Wiley & Sons, Chichester, England, ISBN: 0 471 94113 1, Doctors Thesis, 1999.
- [32] W. REN: Consensus strategies for cooperative control of vehicle formations, *IET Control Theory & Applications*, **1**(2), (2007), 505–512. DOI: [10.1049/iet-cta:20050401](https://doi.org/10.1049/iet-cta:20050401)

- [33] H. YANG and F. ZHANG: Robust Control of Formation Dynamics for Autonomous Underwater Vehicles in Horizontal Plane. ASME. *Journal of Dynamic Systems, Measurement, and Control.*, **134**(3), (2012), 031009. DOI: [10.1115/1.4005507](https://doi.org/10.1115/1.4005507)
- [34] B.D. ANDERSON, C. YU, S. DASGUPTA, and A.S. MORSE: Control of a three-coleader formation in the plane, *Systems & Control Letters*, **56**(9–10), (2007), 573–578. DOI: [10.1016/j.sysconle.2007.04.004](https://doi.org/10.1016/j.sysconle.2007.04.004)
- [35] B. DAS, B. SUBUDHI, and B.B. PATI: Cooperative formation control of autonomous underwater vehicles: An overview, *International Journal of Automation and computing*, **13**, (2016), 199—225. DOI: [10.1007/s11633-016-1004-4](https://doi.org/10.1007/s11633-016-1004-4)
- [36] Z. ZENG, L. LIAN, K. SAMMUT, F. HE, Y. TANG, and A. LAMMAS: A survey on path planning for persistent autonomy of autonomous underwater vehicles, *Ocean Engineering*, **110**, (2015), 303—313. DOI: [10.1016/j.oceaneng.2015.10.007](https://doi.org/10.1016/j.oceaneng.2015.10.007)
- [37] H. PHAM and X. WEI: Bellman equation and viscosity solutions for mean-field stochastic control problem, *ESAIM: Control, Optimisation and Calculus of Variations*, **24**(1), (2018), 437—461. DOI: [10.1051/cocv/2017019](https://doi.org/10.1051/cocv/2017019)
- [38] P. SINGH, B.J. TALUKDAR, R.N. DASH, and B. DAS: An intelligent hybrid algorithm for smooth AUV path generation. In *2025 3rd IEEE International Conference on Industrial Electronics: Developments & Applications (ICIDeA)*, (2025), 1–6. DOI: [10.1109/ICIDeA64800.2025.10963089](https://doi.org/10.1109/ICIDeA64800.2025.10963089)