Contextual Multi-Task Reinforcement Learning for Autonomous Reef Monitoring

Melvin Laux*†, Yi-Ling Liu†, Rina Alo*, Sören Töpper*,
Mariela De Lucas Alvarez†, Frank Kirchner*†, and Rebecca Adam†
* Robotics Research Group, University of Bremen, Bremen, Germany
† Robotics Innovation Center, German Research Center for Artificial Intelligence, Bremen, Germany
Correspondence: melvin.laux@uni-bremen.de

***Abstract*—Although autonomous underwater vehicles promise the capability of marine ecosystem monitoring, their deployment is fundamentally limited by the difficulty of controlling vehicles under highly uncertain and non-stationary underwater dynamics. To address these challenges, we employ a data-driven reinforcement learning approach to compensate for unknown dynamics and task variations. Traditional single-task reinforcement learning has a tendency to overfit the training environment, thus, limit the long-term usefulness of the learnt policy. Hence, we propose to use a contextual multi-task reinforcement learning paradigm instead, allowing us to learn controllers that can be reused for various tasks, e.g., detecting oysters in one reef and detecting corals in another. We evaluate whether contextual multi-task reinforcement learning can efficiently learn robust and generalisable control policies for autonomous underwater reef monitoring. We train a single context-dependent policy that is able to solve multiple related monitoring tasks in a simulated reef environment in HoloOcean. In our experiments, we empirically evaluate the contextual policies regarding sample-efficiency, zero-shot generalisation to unseen tasks, and robustness to varying water currents. By utilising multi-task reinforcement learning, we aim to improve the training effectiveness, as well as the reusability of learnt policies to take a step towards more sustainable procedures in autonomous reef monitoring.**

***Index Terms*—reinforcement learning, AUV, reef monitoring**

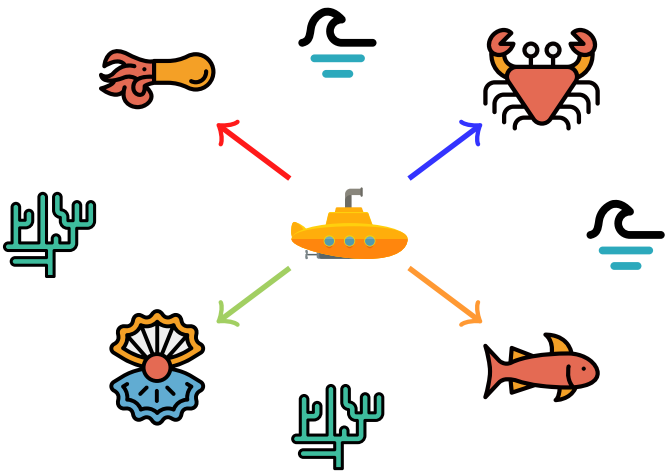


Fig. 1: Traditional single-task RL would require learning individual policies for monitoring different types of organisms, e.g. $\pi_{fish}(a|s)$ and $\pi_{crab}(a|s)$. Instead, we learn a unified multi-task policy $\pi(a|s;c)$ that efficiently reuses shared knowledge between tasks.

I. Introduction

Earth's oceans have been a great source of food and resources throughout human history, however, due to *the blue acceleration*, the conservation and restoration of ocean health becomes an increasingly pressing challenge [16]. With continually rising pollution in the oceans, the need for close and continuous health monitoring of marine ecosystems is a necessary step towards protecting biodiversity [34]. As current restoration and protection efforts mostly rely on cost-intensive ship missions and divers, a key component for boosting the efficacy of marine restoration is the advancement of technological solutions for autonomous habitat monitoring, e.g., by using low-cost, versatile autonomous underwater vehicles (AUVs) [11]. A key challenge in underwater robotics is dealing with highly non-linear, time-dependent, and uncertain hydrodynamics. The movement behaviour of a given AUV is dependent on external factors such as currents, payload, and visibility conditions [6]. While some of these factors can be addressed via traditional control approaches and accurate modelling, a reliable and trustworthy controller requires a certain level of autonomy in order to deal with the remaining uncertainties that cannot be modelled or predicted accurately [10]. These challenges motivate existing work that address the need for control strategies that do not rely exclusively on model-based formulations and are able to deal with complex and changing conditions directly from interaction data. One approach to tackle these types of uncertainties is to use data-driven methods like reinforcement learning (RL). While single-task reinforcement learning often suffers from low sample-efficiency and a tendency to overfit, multi-task reinforcement learning (MTRL) offers a promising avenue towards learning more robust and generalisable control policies. By adopting a context-dependent MTRL approach for learning AUV control policies, we aim to efficiently learn policies that can function reliably under changing conditions and may be reused for different but related monitoring tasks. Hence, in this work, we empirically investigate whether contextual MTRL can be utilised to efficiently learn generalisable and reusable AUV controllers for autonomous underwater reef monitoring. Namely, we aim to answer the following research questions,

- **RQ1**: Are RL-based policies able to solve the reef monitoring task?
- **RQ2**: How do contextual policies compare to mixture-of-expert policies in terms of asymptotic performance, sample-efficiency, and zero-shot generalisation to unseen tasks?

In order to answer these questions, we mathematically formalise the autonomous reef monitoring task to apply reinforcement learning methods. We then evaluate multiple RL approaches to solve the family of tasks in a simulated environment using the high-fidelity underwater simulator HoloOcean [27]. Additionally, we evaluate the contextual RL approach in a simplified version of the task using Minigrid [9]. To the best of our knowledge, this is the first work to apply MTRL for autonomous reef monitoring.

Novel Contribution: Our main contributions are

- a context-dependent formalisation of the autonomous task monitoring scenario for (contextual) MTRL,
- a systematic simulation-based evaluation empirically showing the contextual MTRL applicability to autonomous AUV control.

II. Related Work

The scope of manual reef monitoring is restricted by the physical limits of human divers, who cannot survey vast or deep areas. Thus, AUVs need to perform these tasks with high autonomy and accuracy [26]. Moreover, manual data collection often lacks the consistency needed for long-term studies as human observers introduce subjective errors [23]. To address these challenges, specialised AUVs are increasingly employed for reef monitoring, such as inspection vehicles for coral assessment or larger survey platforms that integrate multi-modal sensing with real-time AI [31]. However, these systems are often challenged by low-visibility conditions and limited maneuverability [31]. Recent work has demonstrated the potential for vision-guided AUVs to map biological hotspots by correlating ecological abundance with reef topography [41]. Scaling such approaches to unseen reef environments, however, remains a challenge. This further highlights the need for more robust control policies that can function reliably under the unpredictable dynamics of varying underwater environments.

RL has emerged as an alternative to traditional control for managing these unpredictable dynamics. Early work by Carlucho et al. [8, 7] has demonstrated that end-to-end deep RL could handle low-level AUV control and position tracking by learning directly from raw sensor data. RL has been applied to specific tasks such as 3D path following [21, 15], navigation under the influence of ocean currents [36, 20], and complex path planning for the Internet of Underwater Things [39]. Furthermore, RL-based obstacle avoidance has addressed the challenge of safe navigation by enabling AUVs to learn collision-free paths through unknown environments [42, 40, 12]. Due to the costs and risks of physical deployment, recent research has focused on using high fidelity physics simulations to develop high precision maneuvers, such as autonomous docking [5, 25]. These works demonstrate that virtual environments provide a safe and systematic way to compare RL architectures prior to their deployment. However, these simulation studies often focus on specific navigation goals, overlooking the requirement for a single controller to manage diverse objectives.

Despite these advancements, most existing RL-based controllers are limited to single-task scenarios, which often leads to overfitting and poor performance in unseen environments. To overcome this, Multi-Task Reinforcement Learning offers a framework for learning versatile policies that generalise across a broad range of monitoring scenarios and environmental settings. To improve data efficiency and stability in such settings, Teh et al. [33] proposed a distilled policy approach. This method captures common behaviours across tasks, preventing the negative interference often seen during joint training. Furthermore, Sodhani et al. [30] demonstrated that providing task metadata as context enables more efficient knowledge transfer across related tasks, which allows the agent to adjust its internal strategy to the specific requirements of a mission. While effective on standardised benchmarks, these frameworks have yet to be applied to the hydrodynamic and perceptual challenges of reef monitoring.

Learning such adaptable policies is essential for achieving zero-shot generalisation, where an agent must succeed in novel situations at deployment time without the need for on-site retraining [17]. As noted by Kirk et al. [17], this adaptability is required to successfully deploy RL in real-world scenarios that are complex and unpredictable. By employing a contextual MTRL approach, we aim to address the scaling challenges identified in recent reef studies [41], enabling a single AUV controller to function reliably across unseen sites and diverse monitoring objectives.

III. Preliminaries

In traditional single task reinforcement learning [32], the agent's task is formalised as a Markov Decision Process (MDP) $\mathcal{T} = (\mathcal{S}, \mathcal{A}, P, R, \mu)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P are the state transition probabilities, R is the reward function and μ is the initial state distribution [3]. The goal of an RL algorithm is then to find an optimal policy $\pi^*(\mathbf{a}|\mathbf{s})$, where any policy $\pi(\mathbf{a}|\mathbf{s})$ is a probability distribution defining the probability of selecting action $\mathbf{a} \in \mathcal{A}$ given any known state $\mathbf{s} \in \mathcal{S}$. Any optimal policy π^* maximises the RL objective $J(\pi, \mathcal{T})$, i.e., the expected discounted return of policy π in task $\mathcal{T}$,

$$J(\pi, \mathcal{T}) = \mathbb{E}_{r_t \sim \pi, \mathcal{T}} \left[\sum_{t=0}^{H} \gamma^t r_t \right], \tag{1}$$

where H is the episode horizon, γ is the discount factor, and r_t is the reward collected at time step t following policy π in task $\mathcal{T}$. Hence, the optimal policy is defined as

$$\pi^*(\mathbf{a}|\mathbf{s}) = \arg\max_{\pi} J(\pi, \mathcal{T}). \tag{2}$$

In multi-task reinforcement learning, the goal is to learn a policy that is able to act optimally in a set of related MDPs

[35]. One common assumption is to assume that the shared structure between the tasks is that all MDPs to be solved use the same state and action spaces. Using this assumption, we can formalise the the MTRL problem as a contextual Markov decision process (CMDP) [13, 22]. A CMDP is a tuple of the form $(\mathcal{C}, \mathcal{S}, \mathcal{A}, \mathcal{M})$, where $\mathcal{C}$ is the context space, $\mathcal{S}$ the state space, $\mathcal{A}$ the action space, and $\mathcal{M} : \mathcal{C} \mapsto \mathcal{T}^{(\mathbf{c})}$ a mapping from context to a specific MDP $\mathcal{T}^{(\mathbf{c})} = (\mathcal{S}, \mathcal{A}, P^{(\mathbf{c})}, R^{(\mathbf{c})}, \mu^{(\mathbf{c})})$, where $P^{(\mathbf{c})}, R^{(\mathbf{c})}, \mu^{(\mathbf{c})}$ are the context-conditioned state transition probabilities, reward functions, and initial state distributions, respectively. It should be noted that defining a suitable context representation is non-trivial for many tasks and can be arbitrarily informative ranging from simple one-hot encodings or random seeds to highly informative task domain knowledge such as physical parameters of the environment [4]. Assuming the context for each task is known, the goal now becomes to find an optimal context-dependent policy $\pi^*(\mathbf{a}|\mathbf{s}; \mathbf{c})$ that maximises the expected discounted return over all context-MDPs $\mathcal{T}^{(\mathbf{c})}$, namely,

$$\pi^*(\mathbf{a}|\mathbf{s}; \mathbf{c}) = \arg\max_{\pi} \mathbb{E}_{\mathbf{c}\sim\mathcal{C}} \left[J(\pi, \mathcal{T}^{(\mathbf{c})}) \right]. \tag{3}$$

Additionally, to evaluate the zero-shot generalisation abilities of policies, it is possible to split the set of tasks induced by the the CMDP into distinct training and test sets of tasks [17].

IV. Contextual MTRL for autonomous reef monitoring

For an AUV to be able to autonomously monitor an underwater habitat, the crucial challenge to navigate its environment to find and detect interesting regions, e.g., different types of organisms, and then navigate towards them. We formalise this task in from of a CMDP, where the agent receives positive rewards for detecting a previously undiscovered marine organism at each time step, while receiving negative rewards for straying too far from the search area. The task of the AUV control policy is the to select an action to maximise the likelihood of finding unseen organisms based on the current observation.

Specifically, we define a high-level representation of the state space $\mathcal{S}$ in terms of the AUV's current location $\mathbf{x} \in \mathbb{R}^2$, rotation $\mathbf{r} \in \mathbb{R}^2$, and velocity $\mathbf{v} \in \mathbb{R}^2$. Additionally, the state contains the total elapsed time $t \in \mathbb{R}_{\geq 0}$ since the beginning of the episode, the number of organisms (both new and previously detected) in its immediate surrounds, as well as the percentage $\mathbf{p} \in [0, 1]^4$ of organisms of each type that still remain undiscovered. In our setting, we consider a scenario where 500 organisms of four different types (represented by different colours, i.e., red, blue, green, black), inhabit the environment and the agent's task is to find all organisms that are flagged as interesting in the given task context. All organisms are considered to be located on the ground. Hence, we omit the z-component of both location and velocity in the state space and let the AUV keep a fixed distance to the seabed. To represent the locally detected organisms, we consider all organisms for each type that are below a given distance threshold d_{max} detected by a simulated detector oracle to focus on the challenge of navigation and control rather than perception. The simulated detector outputs vectors containing the number of total number $\mathbf{l} \in \mathbb{Z}^{16}_{\geq 0}$ and new organisms $\mathbf{n} \in \mathbb{Z}^{16}_{\geq 0}$, in four directions (front right, front left, back right, back left). This representation results in a 44-dimensional state space with states of the form

$$\mathbf{s} = (\mathbf{x}^\top, \mathbf{r}^\top, \mathbf{v}^\top, t, \mathbf{p}^\top, \mathbf{l}^\top, \mathbf{n}^\top)^\top. \tag{4}$$

In our setting, we use a generic torpedo-shaped AUV as is shown in Fig. 5a with one thruster and four fins. The action space is a discrete set of five nominal actions

$$\mathcal{A} = \{\text{forward}, \text{turn right}, \text{turn left}, \text{backward}, \text{no op}\}, \tag{5}$$

which are converted into low-level thruster and fin commands which are then executed at a control frequency of 2 Hz. Specifically, the forward and backward actions keep the fins in a neutral position and set the thruster to 50 or -50 percent of the maximum thrust, respectively. The turn actions are executed by setting the vertical fins to their maximum swing in either direction, i.e., to 45 or -45 degrees, while applying a forward thrust of 50 percent.

As stated, we aim to learn policies that are able to deal with varying water currents and different types of organisms to find. Under the assumption that currents remain constant during a single episode, we consider every different setting of currents to be a new task for the MTRL policy. We represent the currents for a given task as a three dimensional vector $\mathbf{c}_\text{c} \in \mathbb{R}^3$. To indicate which organism types are to be detected by the agent in any given task, we use a 4D binary vector $\mathbf{c}_\text{o} \in \mathbb{B}^4$, where each entry indicates if the corresponding organism type is interesting (vector entry 1) or uninteresting (vector entry 0). For example, the vector $(1, 1, 1, 1)^\top$ represents the task where all types need be found and the vector $(0, 1, 1, 0)^\top$ indicates that only blue and green organisms are of interest. The full context space $\mathcal{C}$ is then the Cartesian product of these two subspaces, namely, $\mathcal{C} = \mathbb{R}^3 \times \mathbb{B}^4$ with

$$\mathbf{c} = (\mathbf{c}_\text{c}^\top, \mathbf{c}_\text{o}^\top)^\top \quad \forall \mathbf{c} \in \mathcal{C}. \tag{6}$$

Based on this context representation, we can now define the context-dependent tasks in terms of reward functions $R^{(c)}$. To avoid overfitting, we keep the reward function simple and define straight-forward success and and failure conditions which each lead to fixed rewards of $r_\text{success} = 1000$ and $r_\text{fail} = -1000$. These specific values were chosen to ensure that successful episodes always lead to positive returns and failed episodes always lead to negative returns, even in the presence of intermediate rewards. The success condition is fulfilled once the agent has detected all organisms of interest. The failure condition is met if the agent strays away too far from its initial position, defined by a maximum allowed distance $d_\text{f} = 10$. To avoid extremely sparse rewards leading to a hard exploration problem, the agent also receives small intermediate rewards $r_\text{n} = 1$ whenever a new organism is detected. In order to keep all tasks on equal scales, we normalise the intermediate rewards by the number of total

organisms of interest. Additionally, we also add a small movement regularisation term $r_{\text{p}} = 0.1$ at each time step to encourage quicker exploration strategies. Combining the intermediate reward and the regularization reward into a joint reward r_{tmp}

$$r_{\text{step}} = \frac{\mathbf{n}^\top \mathbf{c}_{\text{o}}}{\|\mathbf{c}_{\text{o}}\|_1} r_{\text{n}} - r_{\text{p}}. \tag{7}$$

Combining all previously discussed terms, we arrive at the context-dependent reward function,

$$R^{(\mathbf{c})}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) = \begin{cases} r_{\text{success}} & \text{if } \mathbf{p}^\top \mathbf{c}_{\text{o}} = 0 \\ r_{\text{fail}} & \text{if } \|\mathbf{x}_{t+1} - \mathbf{x}_0\|_2 \geq d_{\text{f}} \\ r_{\text{step}} & \text{otherwise} \end{cases}. \tag{8}$$

Using these definitions, we now have a full description of the available context-dependent tasks for our reef monitoring scenario. It should be noted that while the transition distribution is also dependent on the context vector by means of the water currents, it is not necessary to explicitly define them as they are assumed to be unknown in the RL setting. Further note that due to the continuous nature of the current definition, the described CMDP induces an infinite set of tasks to sample from.

V. Experiments

In our experiments, we use both the simplified toy environment as well as the HoloOcean-based simulation to evaluate the suitability of contextual MTRL for the use case of autonomous reef exploration and monitoring via AUVs. Specifically, we aim to answer the research questions:

- **RQ1**: Are RL-based policies able to solve the reef monitoring task?
- **RQ2**: How do contextual policies compare to mixture-of-expert policies in terms of asymptotic performance, sample-efficiency, and zero-shot generalisation to unseen tasks?

To answer these research questions, we evaluate the sample-efficiency, zero-shot generalisation, and final performance of our joint contextual training regime and compare the results against training individual single-task expert policies. As suggested by Agarwal et al. (2021) [1], we report interquartile means (IQM) with bootstrapped 95% confidence intervals (CI) of the undiscounted episode returns to analyse asymptotic performances and sample-efficiency. We use double deep Q-networks (DDQN) [14] as backbone algorithm, implemented using the open-source library rl-blox [18]. To train a contextual version of DDQN, denoted in the following as cDDQN, we concatenate the task context and the observed state as input to the Q-network. All Q-networks in our experiments are represented as multilayer perceptrons (MLPs) [29]. As a baseline, we train individual policies for each separate task from the training set and then build a mixture of experts (MoE), that always selects the corresponding expert for the given task. We then evaluate performance of each policy on both the training set and the test set of tasks.

A. Minigrid Experiments

We initially test the method in a discrete approximation of the autonomous exploration task using Minigrid [9], in which the underwater world is represented as a 2D grid world as shown in Fig. 2. The agent's goal is to navigate to the correctly coloured object based on the given task without running into walls or other coloured objects. The different task contexts consist of a one-hot encoding to indicate which colour of object the agent is expected to collect, resulting in a total of six different contexts

$$\mathcal{C} = \{\text{red}, \text{blue}, \text{green}, \text{yellow}, \text{purple}, \text{grey}\}. \tag{9}$$

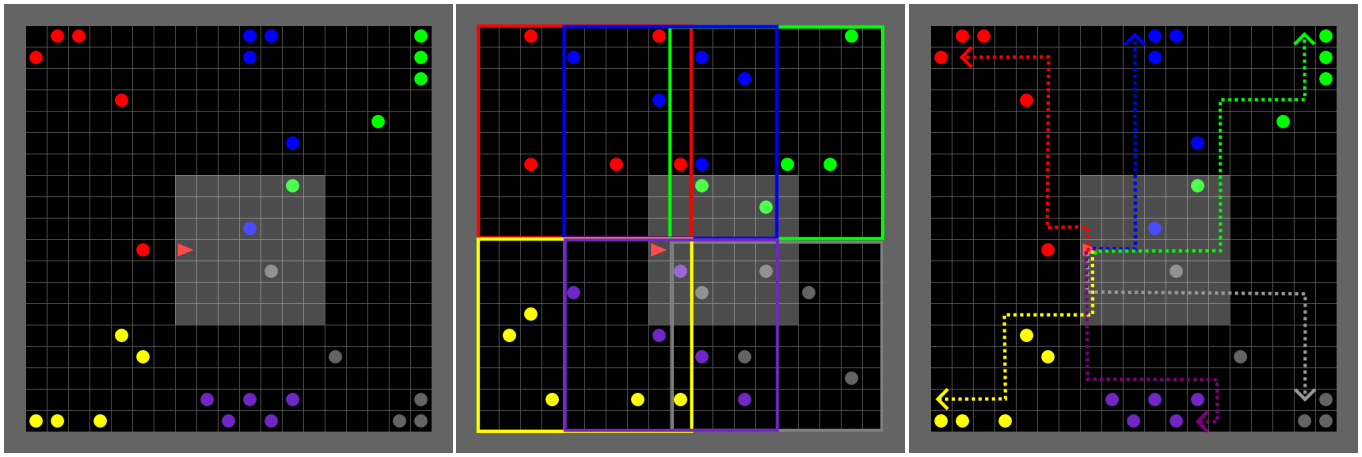
To be able to evaluate the zero-shot generalisation of the trained policies, we split the context set into a training set $\mathcal{C}_{\text{train}}$ containing the red, green, yellow, and grey tasks and a test set $\mathcal{C}_{\text{test}}$ containing the blue and purple tasks. We consider two different settings of the environment, *fixed* and *random*. In the fixed setting, objects are always placed in the same locations, while in the random setting, the locations of the objects are sampled uniformly from colour-specific areas with in the grid at the beginning of each episode. The policies in these experiments are trained for 1M timesteps in the fixed setting and for 5M timesteps in the random setting. Each policy's Q-network is represented as an MLP with two hidden layers of 32 units each. We repeat each experiment for both settings with 10 different random seeds.

Fig. 3 shows the evaluation performance of the trained policies on the training and test sets over time in the fixed setting. On the training tasks, both cDDQN and MoE achieve the same asymptotic performance, however, the MoE policy is able to achieve this with fewer training steps (Fig. 3a). On the unseen test tasks, the performance of cDDQN remains stagnant at initial level, while the performance of MoE significantly drops as training progresses (Fig. 3b). We hypothesise that this effect is caused by overfitting as the cDDQN policy is aware to be in a new context, while MoE overconfidently follows the same strategies in test tasks as in training tasks leading to collisions with incorrect organisms.

Similarly, Fig. 4 shows the performance of the trained policies on both task sets in the randomised setting. Each policy was evaluated on randomly generated grids, reporting the mean total reward on each task over 25 rollouts. As in the fixed setting, both MoE and cDDQN are able to learn policies that avoid collisions and find organisms of the correct type. However, in this more complex scenario, neither algorithm is able to perfectly find every organism of the correct type in every evaluation. As in the previous setting, MoE has a better asymptotic performance on the training set and a better sample efficiency than cDDQN (Fig. 4a), but a much worse zero-shot generalisation to the unseen test tasks (Fig. 4b).

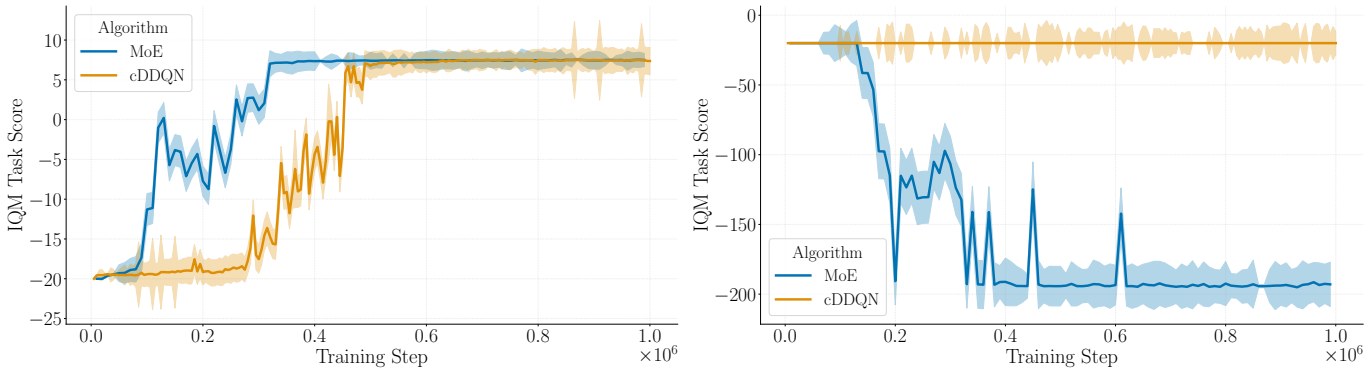
B. HoloOcean Experiments

In our second set of experiments, we use a simulated environment of the reef monitoring task as introduced in the



(a) Fixed minigrid layout (b) Organism areas for random layout (c) Examples policies

Fig. 2: **Minigrid environments.** For preliminary evaluation of the suitability of contextual MTRL for autonomous reef monitoring, we make use of a simplified toy setting based on minigrid. In these environments, the task of the agent is to move towards organisms (represented as circles) of the correct colour, while avoiding moving into walls or incorrectly coloured organisms. The agent receives a small positive reward for moving directly next to a correct organism, a large positive reward for finding all five correctly coloured organisms, and strong negative reward for colliding with the surrounding wall or incorrectly coloured organisms. At each timestep the agent observes its immediate surroundings as highlighted and may choose between moving forward, turning right, or turning left. (a) shows the layout of organisms in the fixed minigrid setup, (b) shows the areas in which the different coloured organisms can be placed in the random setting, and (c) shows example trajectories of final policies trained using cDDQN after 1M timesteps in the fixed setting.

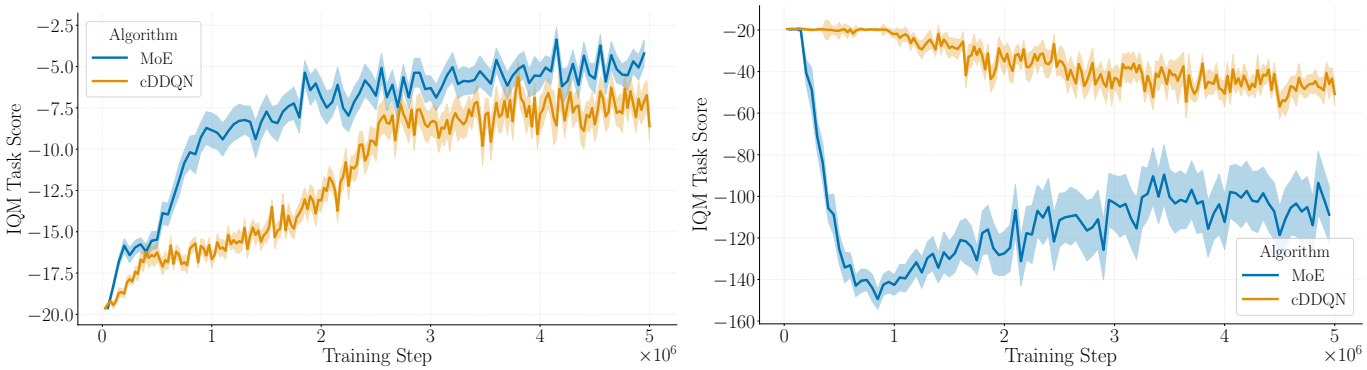


(a) Training set (b) Test set

Fig. 3: **Results Fixed Minigrid.** Interquartile means over 10 random seeds of trained policies on the training set (a) and the test set (b) of the fixed Minigrid environments. Shaded areas show the 95% CI of the IQM. Both MoE and cDDQN are able to solve the training tasks. However, MoE overfits and fails to avoid collisions on the unseen test set, while cDDQN is able to avoid catastrophic failure.

previous section using HoloOcean [27, 28]. Figure 5 shows the used generic AUV, as well as the distribution of possible locations for the different types of organisms. For this set of environments, we again create a training set and a test set of tasks. The training set consists of five different current settings (north, east, south, west, and none) and a single type of organism to detect. In combinations this leads to 20 distinct training tasks. For the test set, we consider four new and unseen task settings (north-east, south-east, north-west, and south-west), each combined with a single organism type. Additionally, we also include the setting with no currents with the goal being to find organisms of any colour, i.e., the context $(0, 0, 1, 1, 1, 1)$, resulting in a total of 17 test tasks. Each policy's Q-network is represented as an MLP with two hidden layers of 128 units each. All policies are trained for a total of 250k timesteps and repeat each experiment for 20 different random seeds.

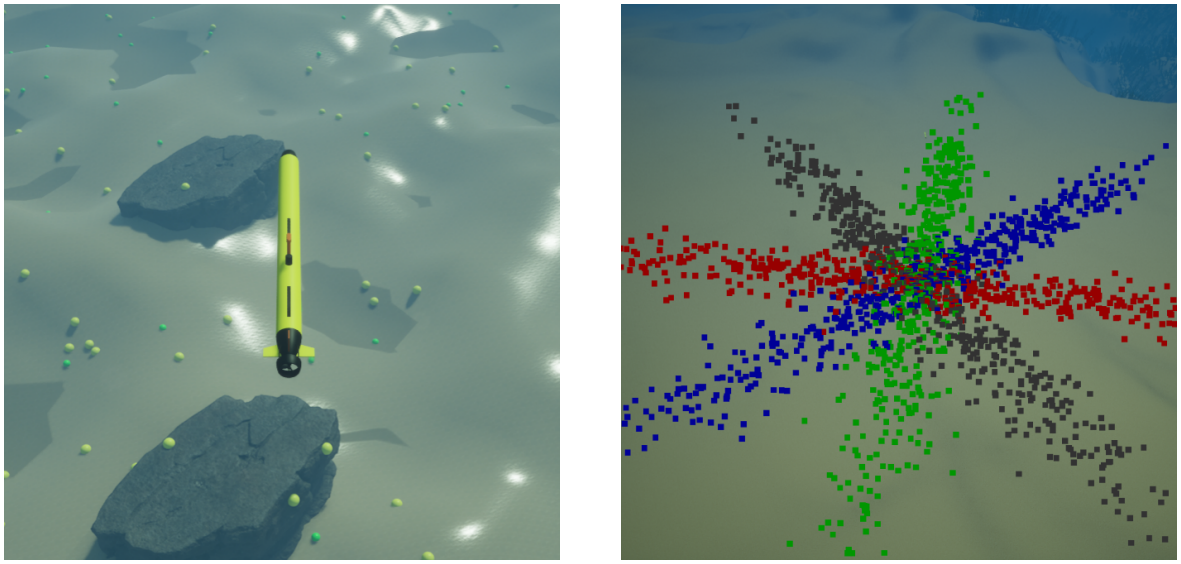
Figure 6 shows the evaluation performance of the trained policies of the HoloOcean experiments on the training and test sets over time. Each policy was evaluated in each context for



(a) Training set

(b) Test set

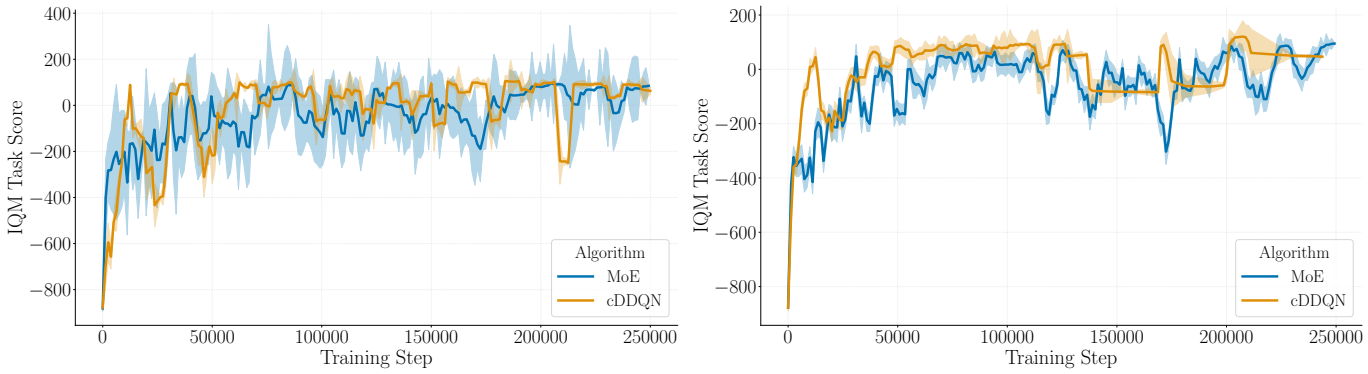
Fig. 4: **Results Random Minigrid.** Interquartile means over 10 random seeds of trained policies on the training set (a) and the test set (b) of the random Minigrid environments. Shaded areas show the 95% CI of the IQM. Both MoE and cDDQN are able to learn policies to correctly find organisms in the training tasks with MoE slightly outperforming cDDQN both in terms of sample efficiency and asymptotic performance. However, as in the fixed setting MoE overfits and fails to avoid collisions on the unseen test set, while cDDQN is able to mostly avoid catastrophic failures.



(a) AUV closeup

(b) Organism distribution

Fig. 5: **HoloOcean environments.** For our main investigation on contextual MTRL for autonomous reef monitoring, we implement a simulated reef environment using HoloOcean. In this environment, the task of the agent is to move towards organisms (represented as circles) of the correct colour, without leaving the search area. The agent receives a small positive reward for detecting to a correct organism, a large positive reward for finding all correctly coloured organisms, and strong negative reward for leaving the search area. A detailed formalisation of the task as a CMDP can be found in Section IV. (a) shows the generic torpedo-shaped AUV used in our experiments, (b) shows an example of the organism distribution within the simulated environment.



(a) Training set

(b) Test set

Fig. 6: **HoloOcean results.** Interquartile means over 20 random seeds of trained policies on the training set (a) and the test set (b) of the random HoloOcean environments. Shaded areas show the 95% CI of the IQM. Both MoE and cDDQN are able to learn policies to correctly find organisms in the training tasks and not leaving the search area. Both approaches are able to transfer to the unseen test tasks with only a small drop in performance. This effect is likely due to the high similarity between training and test tasks.

25 rollouts of which we report the mean total reward. Both cDDQN and MoE achieve a similar asymptotic performance on the training scenarios with no notable difference in sample-efficiency (6a). On the unseen test tasks, the performances of both approaches once again exhibit similar performance and sample efficiency, however on a slightly lower overall level (6b).

C. Discussion

In our extensive experiments, both in Minigrid and HoloOcean, we evaluated the performance, zero-shot generalisation, and sample efficiency of contextual MTRL compared to an MoE approach. Throughout our experiments, we observed similar asymptotic performance of both methods on the known training environments. In terms of sample-efficiency, we observed a slightly faster learning of the training tasks in the Minigrid environments for MoE. However, the zero-shot transfer of these learnt policies to unseen training task led to catastrophic failures, i.e. collisions, likely due to overfitting, while the contextual MTRL method generated more robust policies. In the HoloOcean scenario, we did not observe any notable differences in sample efficiency, zero-shot generalisation, or asymptotic performance. However, it should be noted that when using the MoE approach, multiple networks are trained for every task in the training set. In the HoloOcean experiments this means that the final MoE policies consist of 20 Q-networks, while the cDDQN policy captures all encountered tasks within a single neural network, thus, scaling down the memory requirements of the final policies by a factor of 20. We believe this aspect to be a valuable insight, especially considering generally highly limited computational resources on a real deployed AUV. As task complexity and sensor availability increase in future missions, resource-efficient training algorithms and model will be crucial to equip AUVs with autonomous exploration capabilities.

VI. Conclusion

In this work, we evaluated the suitability of contextual MTRL for autonomous reef exploration and monitoring using AUVs and provided an initial mathematical formulation of a reef monitoring task in terms of a CMDP suitable for learning high-level AUV control policies using MTRL. In our simulated experiments, we observed promising initial results in terms of performance, sample-efficiency, and zero-shot generalisation when compared to a straight-forward mixture of experts approach. Experiments using the HoloOcean simulator indicate that contextual policies may help improve the efficiency and robustness to unseen situations. Notably, contextual policies may be especially promising in highly dynamic environments when limited computational resources are available.

Rather than a ready-to-use system for immediate real-world impact, we see this work as a proof of concept for the usefulnuss of MTRL in AUV-based reef exploration. Based on our findings, we see various potential avenues for further research. For example, we intend to evaluate the MTRL approach more rigorously using more realistic and dynamic environments using HoloOcean, both on real systems and in field tests. Additionally, we aim to further investigate methodological improvements from the existing MTRL literature, e.g., by using more advanced backbone algorithms and model representations [38], online adaptation [37], or exploration strategies [24, 19, 2]. Finally, we also aim to investigate to close the loop between perception and control for AUV monitoring tasks to allow a joint learning of underwater object classifiers and AUV control polices.

Acknowledgments

This work was funded by the German Federal Ministry for the Environment, Climate Action, Nature Conversation and Nuclear Safety (BMUKN) supported by the ZUG under grants 67KIA4036C and 67KIA4036A, and partially supported

by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG) under grant 16ME1010. The authors would like to thank Bilal Wehbe, Yuhan Jin, and Nayari Lessa for their valuable feedback and discussion on this manuscript.

REFERENCES

[1] Rishabh Agarwal et al. "Deep Reinforcement Learning at the Edge of the Statistical Precipice". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 29304–29320.

[2] Susan Amin et al. *A Survey of Exploration Methods in Reinforcement Learning*. Tech. rep. arXiv, Sept. 2021. DOI: 10.48550/arXiv.2109.00157. eprint: 2109.00157.

[3] Richard Bellman. "A Markovian Decision Process". In: *Journal of Mathematics and Mechanics* 6.5 (1957), pp. 679–684. ISSN: 0095-9057. JSTOR: 24900506.

[4] Carolin Benjamins et al. "Contextualize Me – The Case for Context in Reinforcement Learning". In: *Transactions on Machine Learning Research* (Mar. 2023). ISSN: 2835-8856.

[5] Vibhav Bharti et al. "From Simulation to Reality: Deep Reinforcement Learning for Autonomous Underwater Vehicle Docking". In: *OCEANS 2025 Brest*. June 2025, pp. 1–7. DOI: 10.1109/OCEANS58557.2025.11104520.

[6] Jennifer A. Cardenas et al. "A Systematic Review of Robotic Efficacy in Coral Reef Monitoring Techniques". In: *Marine Pollution Bulletin* 202 (May 2024), p. 116273. ISSN: 0025-326X. DOI: 10.1016/j.marpolbul.2024.116273.

[7] Ignacio Carlucho et al. "Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning". In: *Robotics and Autonomous Systems* 107 (2018), pp. 71–86. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2018.05.016. URL: https://www.sciencedirect.com/science/article/pii/S0921889018301519.

[8] Ignacio Carlucho et al. "AUV Position Tracking Control Using End-to-End Deep Reinforcement Learning". In: *OCEANS 2018 MTS/IEEE Charleston*. 2018, pp. 1–8. DOI: 10.1109/OCEANS.2018.8604791.

[9] Maxime Chevalier-Boisvert et al. "Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks". In: *Advances in Neural Information Processing Systems* 36 (Dec. 2023), pp. 73383–73394.

[10] Leif Christensen et al. "Recent Advances in AI for Navigation and Control of Underwater Robots". In: *Current Robotics Reports* 3.4 (Dec. 2022), pp. 165–175. ISSN: 2662-4087. DOI: 10.1007/s43154-022-00088-3.

[11] R. Danovaro et al. "Assessing the Success of Marine Ecosystem Restoration Using Meta-Analysis". In: *Nature Communications* 16.1 (Mar. 2025), p. 3062. ISSN: 2041-1723. DOI: 10.1038/s41467-025-57254-2.

[12] Behnaz Hadi, Alireza Khosravi, and Pouria Sarhadi. "Deep reinforcement learning for adaptive path planning and control of an autonomous underwater vehicle". In: *Applied Ocean Research* 129 (Dec. 2022), p. 103326. ISSN: 01411187. DOI: 10.1016/j.apor.2022.103326. URL: https://www.sciencedirect.com/science/article/abs/pii/S0141118722002589.

[13] Assaf Hallak, Dotan Di Castro, and Shie Mannor. *Contextual Markov Decision Processes*. Tech. rep. arXiv, Feb. 2015. DOI: 10.48550/arXiv.1502.02259. arXiv: 1502.02259.

[14] Hado van Hasselt, Arthur Guez, and David Silver. "Deep Reinforcement Learning with Double Q-Learning". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, Feb. 2016, pp. 2094–2100.

[15] Yu Jiang et al. "Adaptive meta-reinforcement learning for AUVs 3D guidance and control under unknown ocean currents". In: *Ocean Engineering* 309 (Oct. 2024), p. 118498. ISSN: 00298018. DOI: 10.1016/j.oceaneng.2024.118498.

[16] Jean-Baptiste Jouffray et al. "The Blue Acceleration: The Trajectory of Human Expansion into the Ocean". In: *One Earth* 2.1 (Jan. 2020), pp. 43–54. ISSN: 2590-3322. DOI: 10.1016/j.oneear.2019.12.016.

[17] Robert Kirk et al. "A Survey of Zero-shot Generalisation in Deep Reinforcement Learning". In: *J. Artif. Intell. Res.* 76 (2023), pp. 201–264. DOI: 10.1613/jair.1.14174.

[18] Melvin Laux and Alexander Fabisch. *RL-BLOX*. Version 0.4.4. June 2025. DOI: 10.5281/zenodo.15746631. URL: https://github.com/mlaux1/rl-blox.

[19] Melvin Laux et al. "Deep Adversarial Reinforcement Learning for Object Disentangling". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2020, pp. 5504–5510. DOI: 10.1109/IROS45743.2020.9341578.

[20] Artur K. Lidtke, Douwe Rijpkema, and Bülent Düz. "General reinforcement learning control for AUV manoeuvring in turbulent flows". In: *Ocean Engineering* 309 (2024), p. 118538. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2024.118538. URL: https://www.sciencedirect.com/science/article/pii/S0029801824018766.

[21] Dongfang Ma et al. "Neural Network Model-Based Reinforcement Learning Control for AUV 3-D Path Following". In: *IEEE Transactions on Intelligent Vehicles* 9.1 (2024), pp. 893–904. DOI: 10.1109/TIV.2023.3282681.

[22] Aditya Modi et al. "Markov Decision Processes with Continuous Side Information". In: *Algorithmic Learning Theory, ALT 2018, 7-9 April 2018, Lanzarote, Canary Islands, Spain*. Ed. by Firdaus Janoos, Mehryar Mohri, and Karthik Sridharan. Vol. 83. Proceedings of Machine Learning Research. PMLR, 2018, pp. 597–618.

[23] David O. Obura et al. “Coral reef monitoring, reef assessment technologies, and ecosystem-based management”. In: *Frontiers in Marine Science* 6 (SEP Sept. 2019), p. 436982. ISSN: 22967745. DOI: 10.3389/fmars.2019.00580. URL: https://www.un.org/.

[24] Deepak Pathak et al. “Curiosity-Driven Exploration by Self-Supervised Prediction”. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 2778–2787.

[25] Mihir Patil, Bilal Wehbe, and Matias Valdenegro-Toro. “Deep Reinforcement Learning for Continuous Docking Control of Autonomous Underwater Vehicles: A Benchmarking Study”. In: *OCEANS 2021: San Diego – Porto*. Sept. 2021, pp. 1–7. DOI: 10.23919/OCEANS44145.2021.9706000.

[26] Victor J. Piñeros, Alicia Maria Reveles-Espinoza, and Jesús A. Monroy. “From Remote Sensing to Artificial Intelligence in Coral Reef Monitoring”. In: *Machines* 12.10 (2024). ISSN: 2075-1702. DOI: 10.3390/machines12100693. URL: https://www.mdpi.com/2075-1702/12/10/693.

[27] E. Potokar et al. “HoloOcean: An Underwater Robotics Simulator”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA*. Philadelphia, PA, USA, May 2022.

[28] Blake Romrell et al. *A Preview of HoloOcean 2.0*. 2025. arXiv: 2510.06160 [cs.RO]. URL: https://arxiv.org/abs/2510.06160.

[29] F. Rosenblatt. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”. In: *Psychological Review* 65.6 (1958), pp. 386–408. ISSN: 1939-1471. DOI: 10.1037/h0042519.

[30] Shagun Sodhani, Amy Zhang, and Joelle Pineau. “Multi-Task Reinforcement Learning with Context-based Representations”. In: *Proceedings of Machine Learning Research* 139 (June 2021), pp. 9767–9779. URL: http://arxiv.org/abs/2102.06177.

[31] Atif Sultan et al. “Autonomous robotic systems for coral reef monitoring: Review and open research issues”. In: *Ecological Informatics* 92 (Dec. 2025), p. 103511. ISSN: 15749541. DOI: 10.1016/j.ecoinf.2025.103511. URL: https://doi.org/10.25923/wect-ry70.

[32] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[33] Yee Whye Teh et al. “Distral: Robust Multitask Reinforcement Learning”. In: *Advances in Neural Information Processing Systems* 2017-December (July 2017), pp. 4497–4507. URL: http://arxiv.org/abs/1707.04175.

[34] Mine Banu Tekman et al. “Impacts of Plastic Pollution in the Oceans on Marine Species, Biodiversity and Ecosystems”. In: *EPIC3WWF Germany, 221 P.* (Feb. 2022). DOI: 10.5281/zenodo.5898684.

[35] Nelson Vithayathil Varghese and Qusay H. Mahmoud. “A Survey of Multi-Task Deep Reinforcement Learning”. In: *Electronics* 9.9 (Sept. 2020), p. 1363. ISSN: 2079-9292. DOI: 10.3390/electronics9091363.

[36] Chao Wang et al. “AUV Path following Control using Deep Reinforcement Learning under the Influence of Ocean Currents”. In: *ACM International Conference Proceeding Series* (Feb. 2021), pp. 225–231. DOI: 10.1145/3458380.3459041. URL: /doi/pdf/10.1145/3458380.3459041?download=true.

[37] Bilal Wehbe, Marc Hildebrandt, and Frank Kirchner. “A Framework for On-line Learning of Underwater Vehicles Dynamic Models”. In: *2019 International Conference on Robotics and Automation (ICRA)*. May 2019, pp. 7969–7975. DOI: 10.1109/ICRA.2019.8794403.

[38] Bilal Wehbe et al. “Learning of Multi-Context Models for Autonomous Underwater Vehicles”. In: *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. Nov. 2018, pp. 1–6. DOI: 10.1109/AUV.2018.8729823.

[39] Meng Xi et al. “Comprehensive Ocean Information-Enabled AUV Path Planning Via Reinforcement Learning”. In: *IEEE Internet of Things Journal* 9.18 (2022), pp. 17440–17451. DOI: 10.1109/JIOT.2022.3155697.

[40] Jian Xu et al. “A learning method for AUV collision avoidance through deep reinforcement learning”. In: *Ocean Engineering* 260 (2022), p. 112038. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2022.112038. URL: https://www.sciencedirect.com/science/article/pii/S0029801822013683.

[41] Daniel Yang et al. “Robot Goes Fishing: Rapid, High-Resolution Biological Hotspot Mapping in Coral Reefs with Vision-Guided Autonomous Underwater Vehicles”. In: (Feb. 2024). URL: http://arxiv.org/abs/2305.02330.

[42] Jianya Yuan et al. “AUV Obstacle Avoidance Planning Based on Deep Reinforcement Learning”. In: *Journal of Marine Science and Engineering* 9.11 (2021). ISSN: 2077-1312. DOI: 10.3390/jmse9111166. URL: https://www.mdpi.com/2077-1312/9/11/1166.