Reinforcement Learning for Marine Path Planning

Edward Clark[1], Alfie Anthony Treloar[1], Alireza Tamaddoni Nezhad[2], and Alan Hunter[1]

[1]University of Bath, Bath BA2 7AY, United Kingdom
[2]University of Surrey, Guildford, Surrey GU2 7XH, United Kingdom

Contact author: Edward Clark, eprc20@bath.ac.uk

Abstract: This paper investigates the use of reinforcement learning (RL) for marine path planning, focusing on active acoustic imaging with a side scan sonar. This is a high-level path planning problems in simulated marine environments, where the agent must learn to maximize coverage. The agent plans paths to maximize unique area coverage and target detection, considering sensor limitations such as nadir gaps and varying detection probabilities. We employ Proximal Policy Optimization (PPO) as the RL algorithm and systematically analyze the impact of key hyperparameters using Optuna for optimization. The study demonstrates that curriculum learning—progressively training agents from simpler to more complex environments—significantly improves learning efficiency and final performance. Our results highlight the importance of reward function design, state and action space formulation, and hyperparameter selection in achieving robust RL-based path planning. The findings provide practical guidance for deploying RL in real-world marine scenarios and underscore the value of curriculum learning and hyperparameter optimization in overcoming training challenges.

Keywords: *Machine Learning, Reinforcement learning, Path planning, Active acoustic imaging, Passive acoustic detection, Curriculum learning*

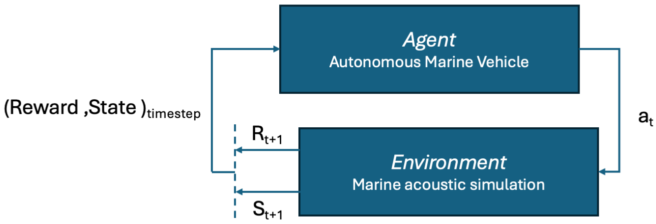


Figure 1: The RL cycle modified from Sutton and Barto [5]. The agent interacts with the environment, taking actions and receiving rewards and an updated state. The agent then updates its policy based on the rewards it receives.

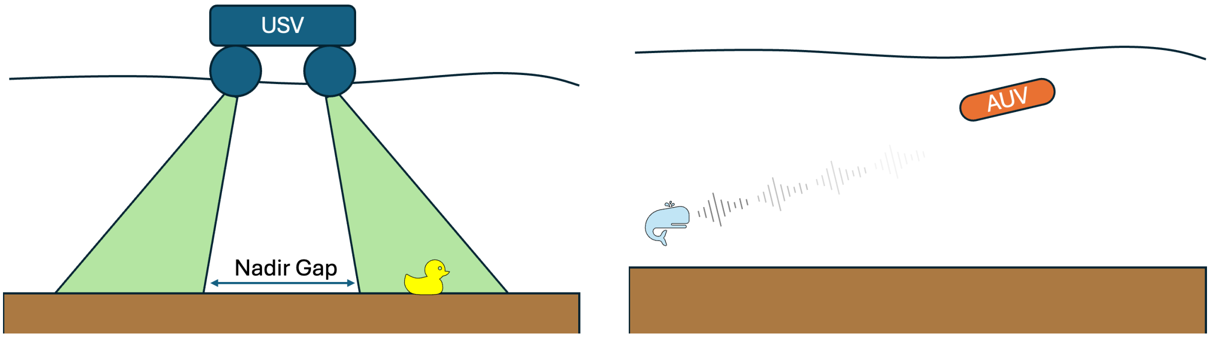
1. INTRODUCTION

Path planning is key to the success of many marine operations, including marine mammal monitoring, seabed investigation and many other tasks. Increasingly these tasks are being performed by autonomous vehicles, which require path planning algorithms to operate effectively. [2, 8, 3] The marine environment is complex and dynamic, with many factors that can affect the success of a path planning algorithm. Reinforcement learning (RL) has been shown to be effective in many path planning tasks, including robotics and autonomous vehicles [1, 3, 7]. It can learn from simulated environments and then transfer this knowledge to real-world scenarios. Marine path planning is expensive to run in the real world, and RL can help to reduce the cost of testing by allowing for simulation-based training. This paper explores the use of RL for marine path planning, with a focus on two case studies: active and passive sonar sensor path planning.

Reinforcement learning (RL) is a type of machine learning that allows an agent to learn from its environment by taking actions and receiving rewards. For a thorough introduction to RL, see Sutton and Barto.[5] An agent follows a policy, which is a mapping from states to actions, and receives a reward signal based on the actions it takes. In our case, the agent is a marine vehicle that is trying to plan a path through an environment to achieve a goal, such as detecting a target or classifying a signal. This is shown in Figure 1, which illustrates the RL cycle. Here a_t is selected by the agent based on its policy π, which maps the current state s_t to an action a_t. It follows that the agents policy is the set of 'instructions' it follows depending on what it observes, to achieve its goal such as turn to port or starboard, dive deeper or ascend. Previous work has looked at the application of RL to passive acoustic detection as seen in Figure 2b[2]. This paper extends the number of marine environments that RL has been successfully applied to for path planning, by investigating active acoustic imaging.

This paper aims to highlight the main considerations when using RL for marine path planning. This paper specifically focuses on PPO (Proximal Policy Optimization) as the RL algorithm, as it has been shown to be effective in many path planning tasks. We demonstrate how important hyperparameters such as batch size, learning rate and discount factor can affect the performance of the RL algorithm. We also explore how curriculum learning can be used to improve the performance of the RL algorithm by gradually increasing the difficulty of the task. Finally, we discuss the results and implications of our findings for future work in marine path planning using RL.

2. ACTIVE ACOUSTIC IMAGING



(a) Active acoustic Imaging *(b) Passive acoustic detection*

Figure 2: Examples of the active and passive acoustic path planning setups. (a) The agent is a USV with a side scan sonar (SSS) mounted on the bottom. The SAS has a fixed range depending on the water depth and a nadir gap where it cannot detect targets. The agent must plan a path to cover the area and detect targets. (b) The agent is an uncrewed underwater vehicle (UUV) with a hydrophone array. The UUV must plan a path to detect targets in the environment.

The active acoustic path planning is a task where an agent must plan to use a side scan sonar (SSS) to detect targets in a simulated environment. This task was developed to train a model (read policy) to deploy on a small (<2m) unmanned surface vessel (USV) to detect small targets and seabed features.

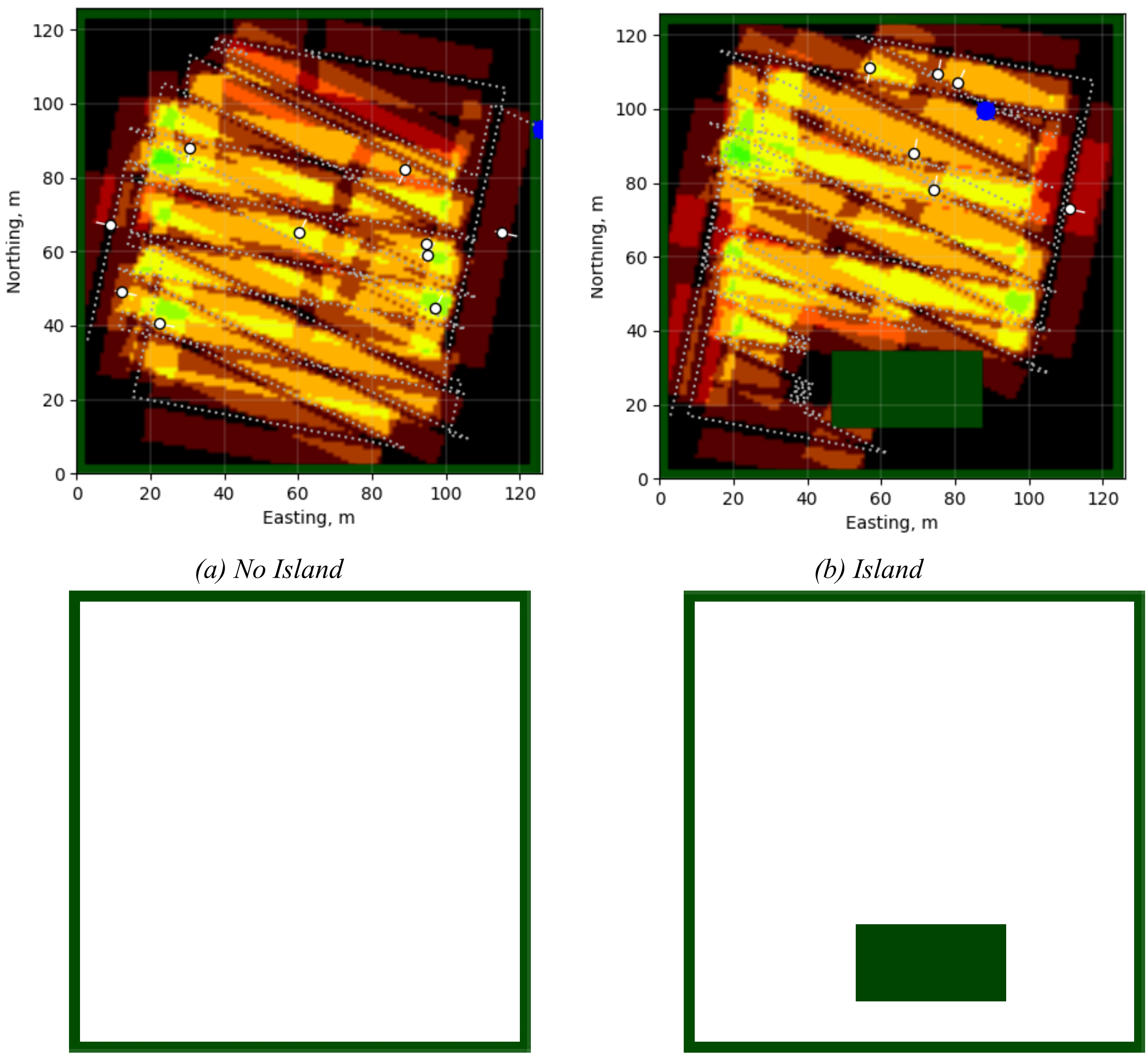
As shown in Figure 2a, the agent is a USV with a SSS mounted on the bottom. As the USV moves past a target it will detect the target with some probability, which is dependent on the target size, seabed type and relative position of the target to the USV. For example a long thin target side on be easily detected but end on it may not be detected at all. The agent must plan a 2D path to cover the area and detect targets, while covering the nadir gap where it cannot detect targets. A typical expert planning such a mission would also try to maximise the number of different angles the SSS sees the target from, to improve detection probability.

2.0.1. REWARD FUNCTION

To encourage the agent to make incremental gains in coverage, we employ an incremental coverage reward function. At each step, the reward is computed as the difference between the current coverage and the coverage at the previous step, i.e.,

$$r_n = \max(0, \text{coverage}_n - \text{coverage}_{n-1}) \tag{1}$$

where coverage_n is the area scanned by the agent at timestep n as a fraction of the total area. This formulation ensures that the agent is rewarded only for newly covered areas, promoting exploration and discouraging redundant actions. Negative differences (i.e., reductions in coverage) are clipped to zero, so the agent is not penalized for revisiting already covered regions but is incentivized to maximize unique coverage over time. This reward structure is particularly effective in environments where the objective is to maximize the area explored or surveyed.



(a) No Island

(b) Island

(c) No Island

(d) Island

Figure 3: Illustration of the two environments and comparison of the planned paths and corresponding coverage maps. The dark green regions indicate the bounds and obstacles in the environment. Coverage is shown by the shaded regions from a scale from red through yellow to green depending on the number of directions it has been scanned from. Contacts are shown as large white dots and the agents location is shown as a blue dot. The path of the agent is shown as a dotted white line. These paths are from the optimal RL learnt policy.

2.0.2. STATE & ACTION SPACE

The agent's state space is a 64x64x3 grid, where each cell represents a 2D area of the environment. The three channels represent the following:

- **Coverage Map:** A map indicating whether a cell has been covered by the agent.
- **Location Map:** A map indicating the location of the agent
- **Occupancy Map:** A map indicating the environment such as bounds and obstacles, where the agent cannot go.

Figure 3 shows an example of the three channels combined into a single image. In reality the maps are larger, but they are downsampled to 64x64 for computational efficiency. Using non square maps is also possible as these are padded before down or upsampling so the trained model can be used in different environments.

The action space is a discrete set of actions that the agent can take at each step. The agent can move in one of any 60 directions, giving it a 6 degree resolution.

3. METHODOLOGY

3.1. RL ALGORITHM

Proximal Policy Optimization (PPO) was used as the RL algorithm for training [4]. PPO is a policy gradient method that has been shown to be effective in many path planning tasks . There are a number of hyperparamters that should be tuned for PPO:

Table 1: Key Hyperparameters of Proximal Policy Optimization (PPO)

Hyperparameter	Description
N steps	Number of steps to run for each environment per update.
Batch Size	Number of samples used for each policy update.
Epochs	Number of updates at each learning phase
Discount Factor (γ)	Determines the importance of future rewards.
Learning Rate	Step size for updating the policy and value networks.
GAE Lambda (λ)	Controls bias-variance tradeoff in advantage estimation.
Clip Range	Limits policy update size to ensure stable learning.
Entropy Coefficient	Encourages exploration by adding entropy to the loss.
Value Function Coefficient	Scales the value loss in the total loss function.
Max Gradient Norm	Maximum value for gradient clipping to improve stability.

3.2. HYPERPARAMETER OPTIMISATION

Optuna is a hyperparameter optimisation framework that allows us to define a search space and an objective function to optimize. The search space is defined as a set of hyperparameters

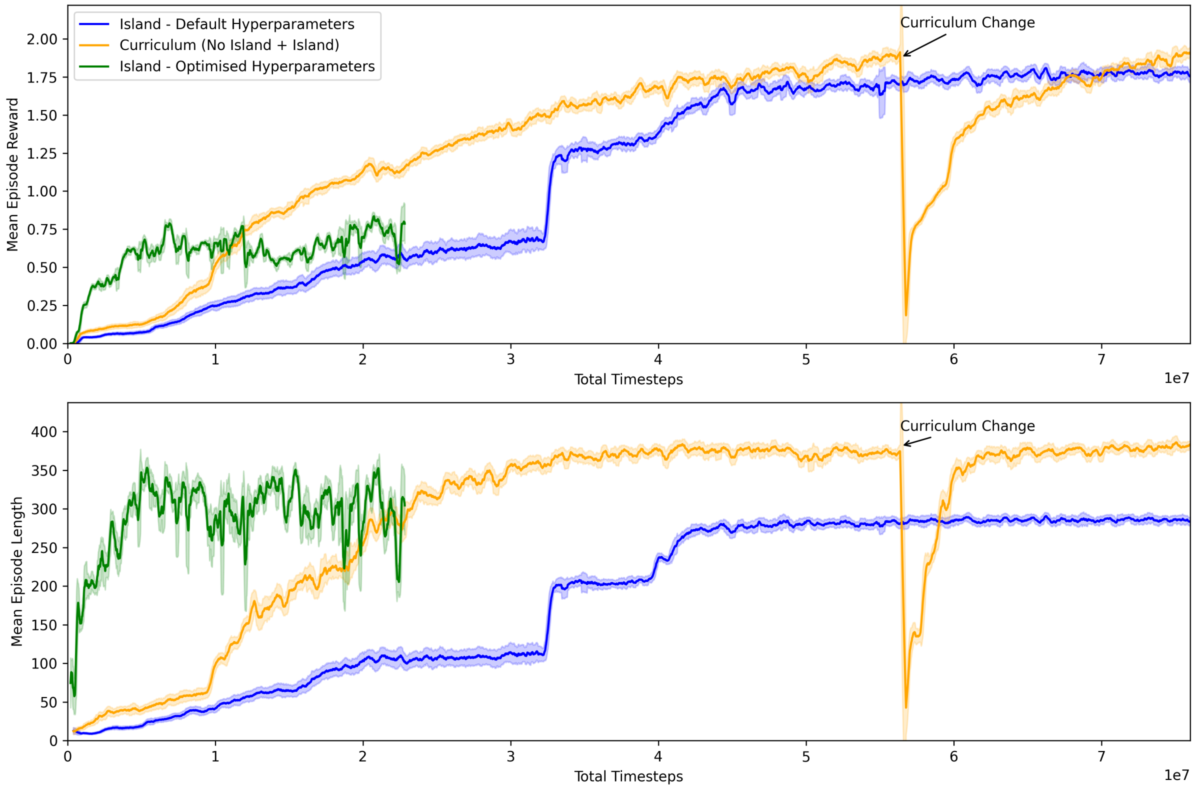


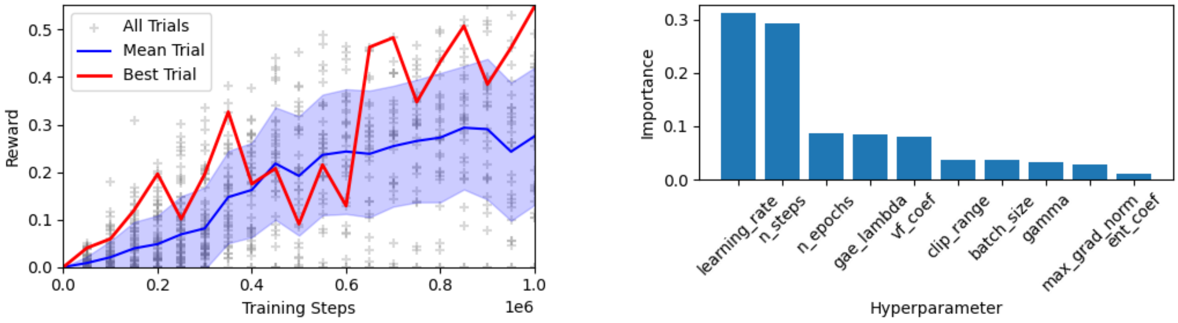
Figure 4: Learning curves for the active acoustic path planning task. For each curve the mean and standard deviation of the reward over a 200 episode rolling window is shown. The blue line shows learning with default hyperparameters in the Island environment. The orange line shows curriculum learning, where the agent is first trained in a simple environment with no islands and then transferred to the more complex environment with an island. The yellow line shows learning with optimised hyperparameters in the Island environment.

and their ranges, as shown in Table 1. The objective function is defined as the average reward over a number of episodes, which is used to evaluate the performance of the RL algorithm.

A TPE (Tree-structured Parzen Estimator) sampler was used to sample hyperparameters from the search space. This is a common sampler used in Optuna that is based on Bayesian optimization. 70 trials were computed which were allowed to run for 1 million steps. Trials were pruned using median pruning after 200,000 steps, which is a common practice to reduce the number of trials that are run to completion. This is where the reward is compared to the median reward of all trials at that step, and if it is below the median, the trial is pruned. The hyper parameter importance is calculated using the PED-ANOVA method [6], which is a method for calculating the importance of hyperparameters based on the variance of the reward.

4. RESULTS & DISCUSSION

The results of the training with and without curriculum learning are shown in Figure 5a. Three results are shown, the first is the performance of the RL algorithm with default hyperparameters in the Island environment. The second is the performance of the RL algorithm with curriculum learning, where the agent is first trained in a simple environment with no islands and then transferred to the more complex environment with an island. The third is the performance of the RL algorithm with optimised hyperparameters in the Island environment. It is clear that curriculum learning improves the performance of the RL algorithm, as it is able to adapt to the new environment and improve its performance relative to simply training in the more complex environment.



(a) Hyper parameter optimisation

(b) PED-ANOVA Hyperparameter importance

Figure 5: Hyper parameter tuning metrics(a) Results from 70 trials of hyper parameter optimisation. Raw data is shown in grey, the mean and standard deviation in blue. The best performing trial is shown in red. (b) Importance of hyperparameters across all trial runs. The importance is calculated using the PED-ANOVA method

The process for determining the best hyperparameters is shown in Figure 5. We see there are a number of trials that gain 0 reward at any stage of training. This implies that incorrect hyperparameter choice can lead to the RL algorithm not learning at all. The best performing trial is shown in red, and the mean and standard deviation of the all trials is shown in blue. Even though the best trial dips below the mean at some points, it is still able to achieve a higher reward than the mean at the end of training.

The hyperparameter importance is shown in Figure 5b. The most important hyperparameters are the the number of steps and the learning rate. The fact that the number of steps is the most important hyperparameter is not surprising, as it determines the diversity of the training data. The learning rate is also important, as it determines how quickly the agent learns from the environment. Too high a learning rate can lead to instability in the training process, while too low a learning rate can lead to slow convergence.

5. CONCLUSION

This paper has explored the use of reinforcement learning for marine path planning, with a focus on active acoustic imaging. We have shown how the environment and the task affect the RL approach, with differences in the state space, action space, reward function and hyperparameters. We have shown that curriculum learning can be used to improve the performance of the RL algorithm by gradually increasing the difficulty of the task. Hyperparameter optimisation is also important for the performance of the RL algorithm, with the number of steps and learning rate being the most important hyperparameters for the active acoustic task.

REFERENCES

[1] Ignacio Carlucho, Mariano De Paula, Sen Wang, Yvan Petillot, and Gerardo G. Acosta. Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. *Robotics and Autonomous Systems*, 107:71–86, September 2018.

[2] Edward Clark, Alan Hunter, Olga Isupova, and Marcus Donnelly. Optimising sensor path planning with reinforcement learning and passive sonar modelling. In *7th Underwater Acoustics Conference and Exhibition, UACE 2023*, pages 411–418, 2023.

[3] Zhaolun Li and Xiaonan Luo. Autonomous underwater vehicles (AUVs) path planning based on Deep Reinforcement Learning. In *2021 11th International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 125–129, December 2021.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017.

[5] Richard S Sutton and Andrew G Barto. Reinforcement Learning: An Introduction. page 352.

[6] Shuhei Watanabe, Archit Bansal, and Frank Hutter. PED-ANOVA: Efficiently Quantifying Hyperparameter Importance in Arbitrary Subspaces, May 2023.

[7] Meng Xi, Jiachen Yang, Jiabao Wen, Hankai Liu, Yang Li, and Houbing Herbert Song. Comprehensive Ocean Information-Enabled AUV Path Planning Via Reinforcement Learning. *IEEE Internet of Things Journal*, 9(18):17440–17451, September 2022.

[8] Changxi You, Jianbo Lu, Dimitar Filev, and Panagiotis Tsiotras. Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robotics and Autonomous Systems*, 114:1–18, April 2019.