Image-based Mapless Navigation of a Hybrid Aerial-Underwater Vehicle using Prioritized Deep Reinforcement Learning

Junior D. Jesus[1*], Victor A. Kich[4†], Alisson H. Kolling[1†], Ricardo B. Grando[1,2†], Rodrigo S. Guerra[1†], Paulo L. J. Drews-Jr[2†]

[1]Centro de Ciencias Computacionais, Universidade Federal do Rio Grande - FURG, Rio Grande, RS, Brazil.
[2]Robotics and AI Lab, Technological University of Uruguay, Rivera, Uruguay.
[3]Intelligent Robot Laboratory, University of Tsukuba, Tsukuba, Japan.

*Corresponding author(s). E-mail(s): dranaju@gmail.com;
Contributing authors: victorkich98@gmail.com; alikolling@gmail.com; ricardo.bedin@utec.edu.uy; rodrigo.guerra@ieee.org; paulodrews@furg.br;
†These authors contributed equally to this work.

Abstract

In recent years, Reinforcement Learning (RL) has made promising progress in several areas, such as control tasks and video games, by using simple, low-dimensional data. However, it struggles when it needs to process more complex, high-dimensional inputs like raw pixel images, offering results that are not as good as those that use information from laser sensors, as many robotics applications demand. This paper introduces a new technique called Contrastive Unsupervised Prioritized Representations in Reinforcement Learning (CUPRL) for mobile robotics. This innovative approach combines RL and Contrastive Learning to effectively handle high-dimensional observations, an area not fully explored. This is crucial for navigating complex environments, especially for hybrid robots, such as the Hybrid Unmanned Aerial-Underwater Vehicles (HUAUVs) that experience strong changes in light when moving between air and water. Our approach excels in taking important information from depth maps and RGB images during training, aiming to improve the ability of RL agents to navigate without a map in the context of HUAUVs. This field has much to be explored. Our tests in a robot simulator show that CUPRL, which uses learning from both RGB and depth images, performs better than current methods that rely only on pixel data. This is especially true for 3D navigation without maps, where we use only RGB images during tests. This proves that CUPRL could be useful for making decisions in HUAUVs. We believe our work not only offers improved solutions for navigation but also encourages further research into the use of high-dimensional data in RL, presenting a more efficient and adaptable method in complex environments compared to earlier strategies.

Keywords: Reinforcement Learning, Autonomous Navigation, Contrastive Learning, Hybrid Aerial-Underwater Vehicle

1 Introduction

Autonomous navigation in ever-changing environments is a significant challenge in robotics. This involves a careful balance of *perception*—understanding the surroundings—and *action*—planning and moving strategically to avoid possible dangers [1]. Sensors such as laser sensors and cameras are essential in enhancing the robot's ability to perceive its surroundings, helping it to avoid hazards and navigate safely.

Hybrid Unmanned Aerial-Underwater Vehicles (HUAUVs) play crucial roles in marine research, oil drilling, and search and rescue missions [2]. These versatile vehicles face unique challenges as they move between air and water environments. Deep Reinforcement Learning (Deep-RL) is a cutting-edge technique that enables these vehicles to navigate autonomously without maps, a method that's been successful in simulations [3–5]. Despite its potential, Deep-RL requires a lot of data and struggles with raw image data, which suggests we need better strategies to meet the demands of operating HUAUVs in varying conditions [6, 7].

To address this issue, we present the Contrastive Unsupervised Prioritized Representations in Reinforcement Learning (CUPRL) approach. This approach makes use of Contrastive Learning to extract meaningful information from depth maps and RGB images during the training phase. Unlike previous efforts that used the same settings for both training and evaluation, our method uses only RGB images during evaluation. This setting shows that our system has learned to skillfully navigate through various scenarios, instead of just memorizing the training environment. It also demonstrates the system's improved accuracy and ability to adapt to various situations within different new environments. Fig. 1 illustrates the CUPRL framework, as well as, including the inputs I_t, I_{t-1}, I_{t-2} and their corresponding depth maps $D(I_t), D(I_{t-1}), D(I_{t-2})$ that are used in networks for training it. A detailed explanation of the CUPRL architecture can be found in Section 3.1.

Our main contributions are:

- Developing a novel Contrastive Learning-based method for 3D mapless navigation evaluated using a HUAUV in transitional environments using only RGB images in the evaluation phase.

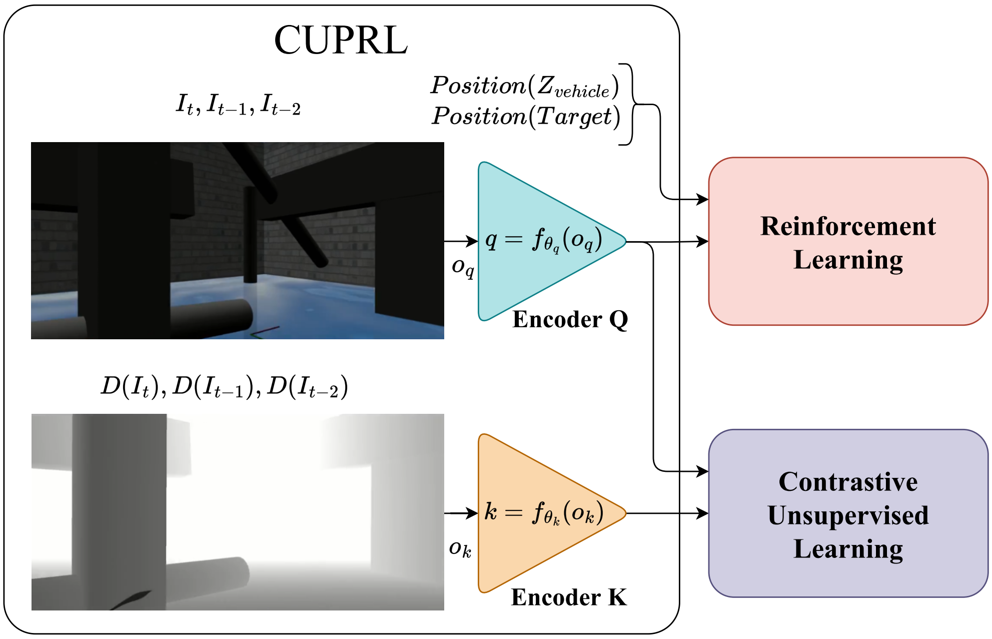


Fig. 1: Overview of CUPRL: The image illustrates the architecture and data flow of the CUPRL model. The model takes as input three consecutive temporal frames of RGB images I_t, I_{t-1}, I_{t-2} and depth $D(I_t), D(I_{t-1}), D(I_{t-2})$. These frames are fed into two encoders: the *query* encoder processes the RGB images, while the *key* encoder processes the depth maps. The encoders extract latent representations q and k from the input data. The latent representation q is then concatenated with target position z-axis and robot position information.

- Demonstrating how our method creates a powerful latent space capable of handling both RGB and depth images, aiding in the development of efficient solutions for navigation issues faced by hybrid vehicles undergoing sudden changes during transitions, thus marking a significant step forward compared to previous efforts.

This paper is structured as follows: Section 2 reviews relevant previous works. Section 3 introduces the proposed methodology. Section 4 reports the evaluation results. Section 5 analyzes the results. Finally, Section 6 presents the conclusions and suggests future research directions.

2 Related Works

Research on navigation without a pre-built map has been thoroughly conducted for mobile robots on land [8]. However, compared to terrestrial robots, there has been less research on autonomous navigation of aerial robots using Deep-RL techniques. In these, the approaches typically abstain from utilizing visual information [9–11], or utilize simplified information without Contrastive Learning [12–15].

A Deep-RL method for landing a stationary or moving base of an Unmanned Aerial Vehicle (UAV) was proposed by Rodriguez *et al.* [12]. Sampedro *et al.*[13] proposed a Deep-RL approach using the Deep Deterministic Policy Gradient (DDPG) algorithm[3] for Search and Rescue tasks in confined environments. In another study, Jesus *et al.*[16] demonstrated the efficacy of Soft Actor-Critic (SAC) algorithm[4] based approaches for robot navigation.

Various studies have investigated the navigation of UAVs, including those by Thomas *et al.* [17], who proposed an RL-based algorithm using self-attention models to control an autonomous UAV. Their work demonstrated the algorithm's effectiveness in completing vehicle navigation even with varying inputs, highlighting its ability to handle noisy or modified state data. Another study by He *et al.* [14] simplified vision information for Deep-RL in UAV navigation and obstacle avoidance by using a Lobula Giant Movement Detector (LGMD). The study achieved an 80% success rate in completing missions in a complex environment.

The work of Grando *et al.* [18] explored the use of Deep-RL for the navigation of robotic systems such as HUAUVs with the use of laser sensors as input. However, using Deep-RL can be challenging due to the need for large amounts of training data and the inefficiency of high-dimensional observations such as raw pixel images [19]. To overcome these challenges, Laskin *et al.* [20] proposed the Contrastive Unsupervised Representations for Reinforcement Learning (CURL) technique, which is capable of extracting useful features from raw images and improving the performance of Deep-RL network control. Jesus *et al.* [21] explore a CURL-based architecture using only depth map images as inputs for the control of a UAV in a simplified 2D navigation context. Although CURL has shown promising results, our work uses only a first-person vision with monocular raw RGB images for navigation in a 3D context in a hybrid environment with air-water transition for the control of a HUAUV.

This work stands out from other related studies by using both depth map and RGB image information in a contrast learning-based approach to address the challenge of high-dimensional observations in the training phase while using just RGB images during navigation in the evaluation phase. It also proposes a prioritized experience replay memory that increases the efficiency of the proposed approach. The use of Contrastive Learning in this work enables the development of a latent space that can relate RGB and depth images, creating an efficient representation to solve navigation problems in complex 3D environments, even when using only RGB images.

3 Methodology

This work proposes an RL-based mapless navigation algorithm using visual and depth information as inputs during training to build a motion policy that avoids obstacles in the environment and performs medium transition. Medium transition is when the vehicle goes from one environment to another, for example going from water to air and air to water. Depth and RGB images are generated and used as inputs to a neural network, which learns to control navigation in these environments through prioritized rewards. The method proposed in this work is called CUPRL. This method combines depth maps and RGB images, a CURL-based network, and prioritized memories for the navigation of a hybrid vehicle. The motion equation for CUPRL is defined as follows:

$$v_t = f^{CUPRL}(I_t), \tag{1}$$

where I_t is the RGB image from the camera, and v_t is the velocities applied to the robot.

To train CUPRL, Equation 1 receives a depth from I_t and information from the pixel observation. The CUPRL neural network extracts information from the depth maps and RGB images and then passes through a SAC network that provides the velocities to the robot.

Fig. 1 shows the inputs of the CUPRL, I_t, I_{t-1}, I_{t-2} or o_q, and $D(I_t, I_{t-1}, I_{t-2})$ or o_k. They are used in the networks for training. $D(I_t, I_{t-1}, I_{t-2})$ is the observation of depth maps that can be estimated from the RGB images I_t, I_{t-1}, I_{t-2}. These observations are processed by their encoders f_{θ_q} and f_{θ_k}, which return the latent space representations q and k. The latent space q, the height of the vehicle in the z axis, and the position of the target in the environment are the information used by the Deep-RL algorithm. For Contrastive Learning, latent spaces q and k are used in the learning of the features of o_q and o_k.

This coordinated training is reflected in the loss function defined as:

$$\mathcal{L}_{\theta_q} = \log\left(\frac{\exp(q^{\intercal} k_{\text{label}})}{\exp(q^{\intercal} k_{\text{label}}) + \sum_{i=0}^{N-1} \exp(q^{\intercal} k_i)}\right), \tag{2}$$

where k_{label} is the recognized label for q in the i-th batch from the replay memory sample with size $N-1$. The dot products $q^{\intercal} k$ are used to effectively determine the similarities between q and the target k values [22]. A key part of this setup is the Polyak-averaging method used to update the θ_k [23].

To capture RGB images and depth maps in a 3D context, where transitions between different media, such as air and water can occur, we decided to use a stereo camera, *e.g* a ZED camera, to generate depth maps in the CUPRL network experiments. A stereo pair can be used for depth estimation both in air and water, as described by Roser *et al.* [24].

3.1 CUPRL Architecture

Three consecutive temporal frames of RGB images and depth are stacked to train the CUPRL model. They are used as inputs to the *query* and *key* encoders, respectively, as illustrated in Fig. 1. The *query* encoder receives the RGB images, while the *key* encoder receives depth maps. Through Contrastive Learning, the encoders learn simultaneously the depth and color characteristics. The input images extract information and obtain the latent spaces q and k. Then, target position and robot position information on the z-axis or $Z_{vehicle}$ are concatenated with the latent space q. The target position information includes the robot's polar distance to the target d_t, the *yaw*, and the *pitch* angles of the vehicle's front relative to the target. The outputs of the CUPRL network are the linear velocities on the x and z axes and the angular velocities that control the vehicle. For the CUPRL, a set of pixel information I is selected, which are three images $100 \times 100 \times 3$, and depth d, which is three images $100 \times 100 \times 1$, both without modifications before being processed.

The neural network architecture employed by CUPRL shares similarities with the SAC network [4]. CUPRL processes inputs through its actor network, which consists of four convolutional layers (serving as the encoder) followed by three fully connected neural network layers to generate output. The choice of layers and nodes aligns with the network structure proposed by Laskin *et al.* [20].

The output values for angular and linear velocity are constrained as follows:

- Angular velocity (z-axis): -0.3 to 0.3 rad/s
- Linear velocity (x-axis and z-axis): 0 to 0.3 m/s

3.2 Reward Function

For tasks involving 3D navigation, the objective is to guide a hybrid vehicle that can travel both on water and air to a designated target point. The reward function formulated for this task is defined as follows:

$$r(s_t, a_t) = \begin{cases} r_{arrive} \text{ if } d_t < c_{d_t}, \\ r_{collide} \text{ if } min_x < c_o, \\ r_{navigating} = c_{navigating}(d_{t-1} - d_t) \text{ if } min_x \geq c_o, \end{cases} \tag{3}$$

where a negative reward ($r_{collision} = -1$) is given if the minimum distance reading from the robot to an object min_x is less than c_o. Here, c_o is equivalent to the distance of $62cm$ from the center of robot to an obstacle or wall that is considered a collision. The c_o follows on the same distance as in the work [21], and it is based on the dimensions of the vehicle [9]. A positive reward and policy to be optimized through RL, thus $r_{arrive} = 1$ was used. A reward is granted if the current distance d_t between the robot and the target is less than the threshold distance $c_{d_t} = 40\ cm$. This condition is defined as 'arriving' at the target point, thereby enabling the robot to visually detect the target. However, it is difficult to obtain the reward r_{arrive} since we are using complex environments. Thus, a reward was defined to encourage the approach between the agent and the target $r_{navigating}$. The previous and current distance of the robot to the target is used to generate this reward. This incentive value is multiplied by a small value $c_{navigating} = 0.1$, aiming to reduce the impact of this reward on the main policy, which is to reach the target in the environment.

3.3 Experimental Setup

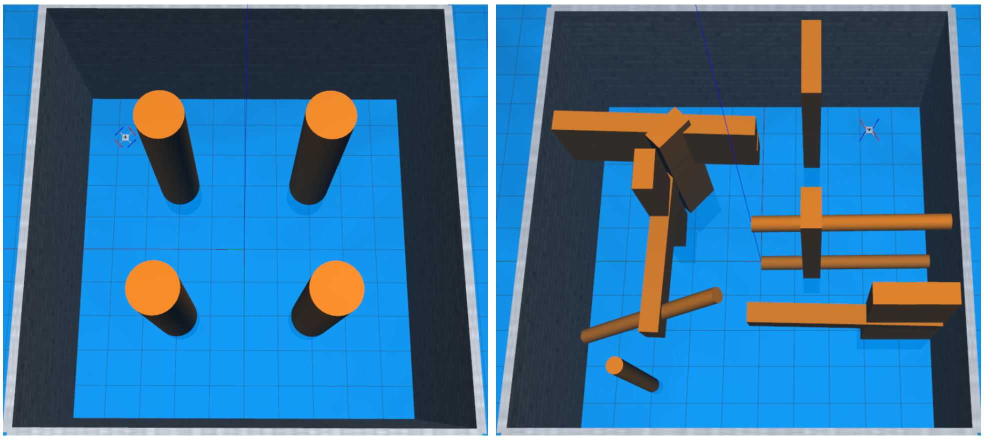
The experiments are conducted in simulation using Gazebo and ROS, with Python being the main programming language complemented by C++ in several parts for efficiency. Neural networks are built using PyTorch, while the OpenCV library facilitates image manipulation. Hydrone robot platform is adopted [25, 26]. Details about the simulation can be found in the work of Grando *et al.* [18].

3.4 Simulated Environments

The training environments designed in Gazebo are illustrated in Fig. 2. These environments were created to demonstrate the methods used in this study, which enable 3D navigation, air-water transitions, and successful arrival to a target while avoiding collisions with walls and obstacles.

The first environment (Fig. 2a) includes four strategically placed obstacles to increase navigational complexity. We employ a Deep-RL technique with a reward function that penalizes collisions with walls or obstacles. Episodes terminate upon collision, assigning a negative reward.

A complex navigation scenario is shown in Fig. 2b. The HUAUV must autonomously navigate this environment, avoiding obstacles and reaching a designated target point.

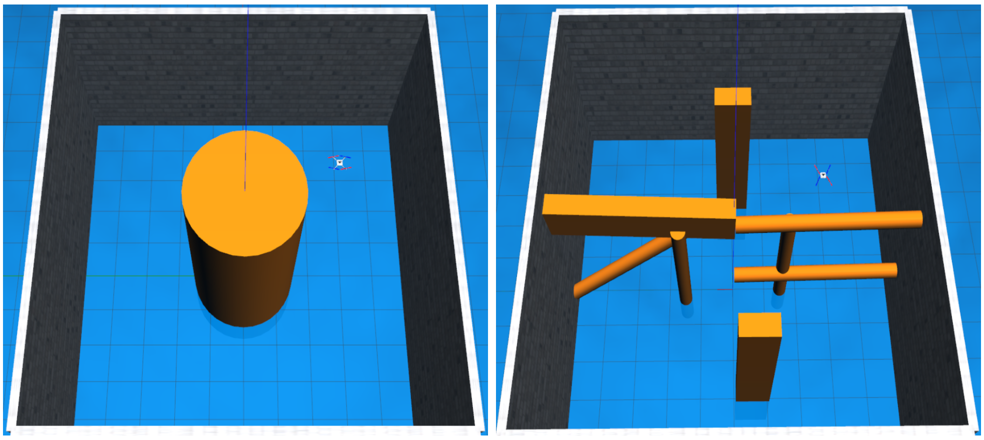


(a) First environment.

(b) Second environment.

Fig. 2: Simulated Environments used for training (Training phase).

We also created two testing environments to evaluate the effectiveness of the proposed technique, as shown in Fig. 3. These environments aim to assess the algorithm's performance after being trained in other environments, to verify whether the networks have learned an effective navigation policy that can be generalized. Fig. 3a presents an environment similar in complexity to the first training environment, where the trained networks will be evaluated. On the other hand, Fig. 3b shows an environment similar to the second training environment, where the networks are also evaluated.



(a) First environment.

(b) Second environment.

Fig. 3: Simulated environment used for evaluation (Testing phase).

4 Experimental Results

We compared the proposed method, CUPRL, with Depth-CUPRL [21], which uses only depth maps as inputs to its networks, with CURL [20] as input for the RGB image, and with a modified version of CURL called CURL (Depth), which uses depth maps as input, but without prioritized memory. In addition, we compared it with a SAC (CNN prio.) network, as implemented in [21], which also incorporates prioritized memory but with convolutional layers. All of the networks evaluated in this study adhered to the architecture presented in Fig. 4. This network architecture is followed by a series of 4 convolutional layers and 4 linear fully connected layers. The vehicle's starting position for training alternates between being on water and in the air. For network evaluation, the initial position of the vehicle is changed for experiments of air-water and water-air transitions.

Navigating in a 3D environment presents unique challenges due to the hybrid vehicle's movement in three dimensions, including the z-axis. Furthermore, a target point is established for the agent to reach training and testing environments. In the first training environment, the target is randomly selected, and the vehicle navigates without colliding with obstacles. The target is alternated between air and water. The alternation between environments is done to force the vehicle's environment transitions to be learned more efficiently. The training episode ends only in case of collision or when the maximum number of time steps is reached.

To train the vehicle in the first environment, a replay memory of 100000 samples is set up for all the trained networks. Each episode consists of $1,000$ time steps t, with an action interval of $0.025ms$ or $40Hz$. The neural networks go through one episode of training, followed by an evaluation of 10 episodes. The evaluation of the networks is conducted using only the deterministic response of the technique, without any noise. This training and evaluation cycle is employed in every method of this work.

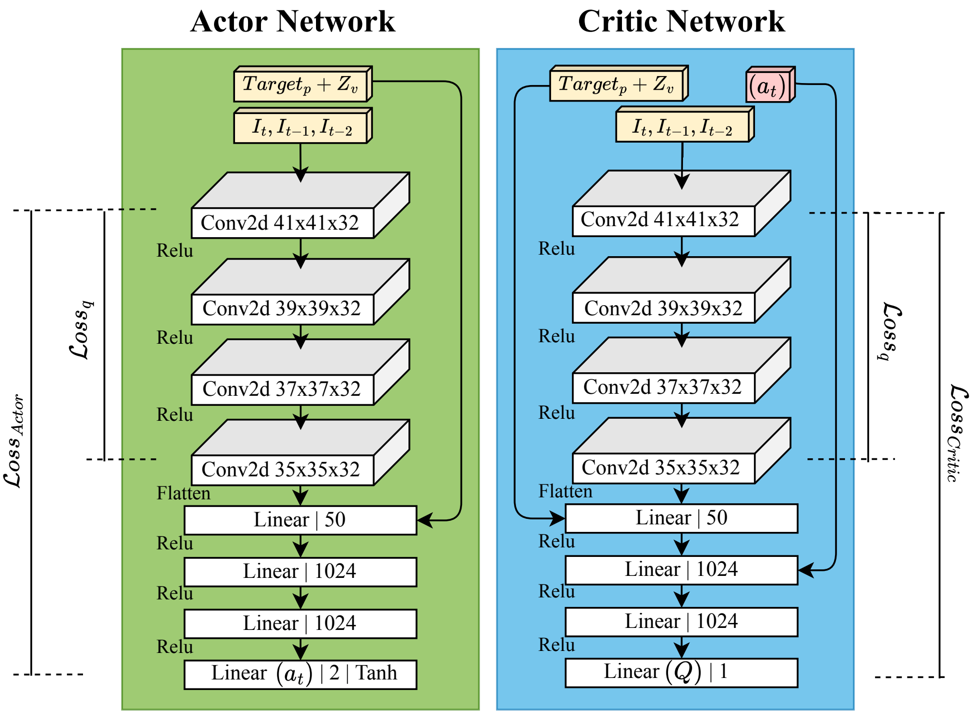
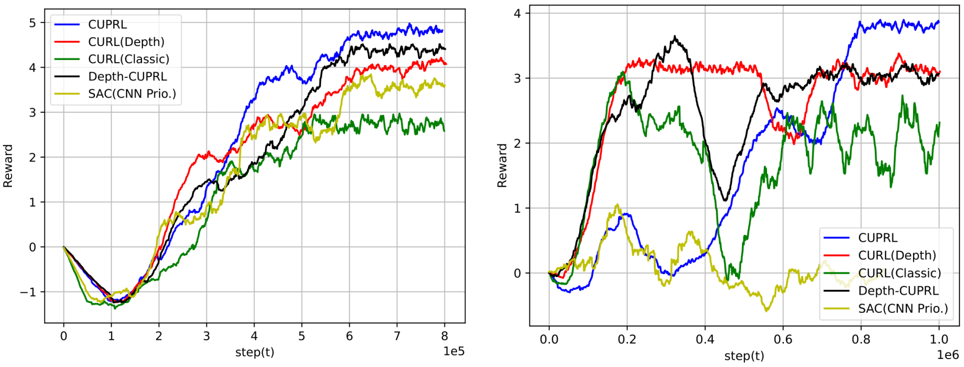


Fig. 4: Proposed CUPRL network architecture.

The results of training in the first environment for the CUPRL and the comparison networks are shown in Fig. 5a. In the initial episodes of neural network training, negative reward values are observed for the actions taken by the agent. The reward acquired in the evaluation episodes indicates the degree of learning of the Deep-RL technique and the average number of targets found during the evaluation. This action is possible because $r_{arrive} = 1$ is the highest value of the reward function, and $r_{navigating}$ is to impact the average reward in an evaluation significantly. All networks were trained for approximately $800,000$ time steps. It can be concluded that the proposed algorithm, CUPRL, achieved the highest average reward in the first training environment. The next highest rewards were obtained by the Depth-CUPRL, CURL (Depth), SAC (CNN prio.), and CURL (Classic) approaches.

The results of the second environment's training are shown in Fig. 5b. All networks were trained for approximately 1 million time steps. Compared to the previous reward functions shown in Fig. 5b, a more unstable reward can be observed for the proposed and the compared algorithms. Although CUPRL had a low average reward in the start of training, at the end it surpassed the average reward acquired compared to other algorithms. Depth-CUPRL and CURL (Depth) had similar results at the end of training. CURL (Classical) can achieve good average rewards despite having an unstable average. On the other hand, SAC (CNN prio.), the only algorithm that does not use Contrastive Learning, could not obtain average rewards greater than 1.



(a) First environment.

(b) Second environment.

Fig. 5: Training moving average of each agent's rewards.

This result indicates that the network could not reach the target point during the SAC (CNN prio.) training. Therefore, it can be concluded that Contrastive Learning presents an advantage in navigating through RGB images and depth maps.

5 Discussion

To better assess the methods and their trained models, we devised two generalization experiments on environments different from the training. The neural networks trained on the first environment were evaluated out throughout 1000 episodes on the environment presented in Fig. 3a, divided into two stages: air-water transition and water-air transition. In the air-water (AW) transition, the vehicle starts at an initial position in the air and aims to reach a target point in the water. In the water-air (WA) transition, the objective is to reach a target point in the air from an initial position in the water. The initial positions of the vehicle and the target position are fixed.

Table 1 shows the results obtained for all evaluated techniques. It is important to note that the information from the *key* encoder in Fig. 1 is used only during the contrastive network training and not during the generalization experiments stage. Therefore, in the case of CUPRL, the evaluation is performed only with RGB images.

Algorithm	**Image**	**Air-water (%)**	**Water-air (%)**
CUPRL [ours]	RGB	**100%**	24.5%
Depth-CUPRL [21]	depth	97.7%	**30.1%**
CURL(Depth) [21]	depth	0%	0%
CURL(Classic) [20]	RGB	0%	15.3%
SAC(CNN prio.) [4]	depth	0%	0%

Table 1: Generalization experiment on the first scenario.

The experimental results presented in Table 1 show that the CUPRL and Depth-CUPRL algorithms achieved the best results. On the other hand, the other evaluated techniques either failed to complete the navigation to the target point or obtained insufficient results, such as the CURL (Classical) in the water-air transition. The air-water transition allowed the proposed networks to perform better. All networks have difficulties in the water-air transition. A common behavior observed in the evaluated techniques, especially those that use only depth maps, was the difficulty in transitioning between media.

The paths shown in Fig. 6 illustrate how the vehicle navigated through the environment during the 1000 evaluation episodes. The trajectory performed is in blue, the initial position of the vehicle is in green, and the target position is in red. The paths taken by CUPRL showed greater stability in its evaluation compared to Depth-CUPRL. The other methods collided several times, as can be observed. However, one case can be seen in Fig. 6c, where CURL (Depth) manages to avoid obstacles and reach the target. However, upon getting close, CURL (Depth) exhibits a hovering behavior, *i.e.*, remaining stationary in the air.

To better assess the trajectories in the generalization experiment, Fig. 7 shows the final median distance between the vehicle and the target at the end of each episode in the first environment AW. In this graph, the red bar represents the median distance, while the error bars indicate the maximum and minimum distances observed for each method.

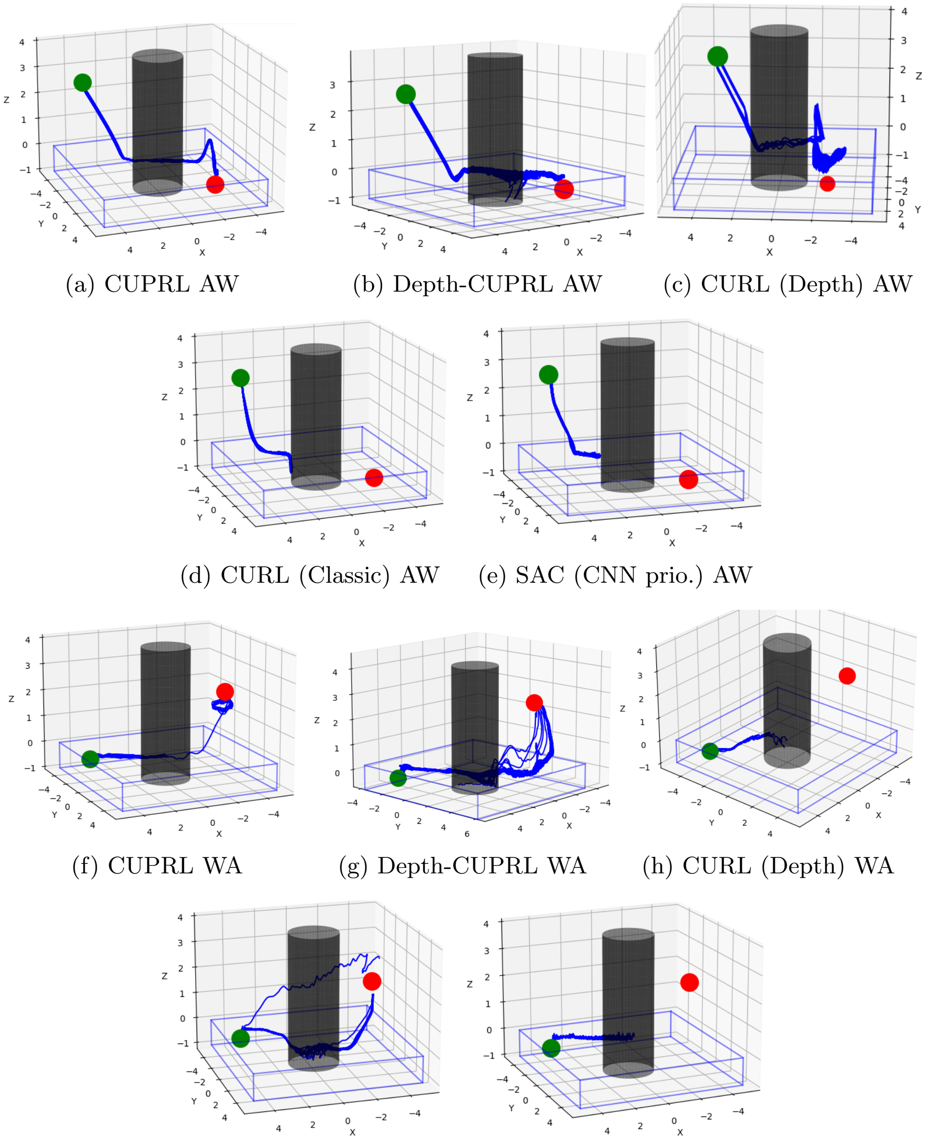
Notably, in Fig. 7, CUPRL consistently maintains a final median distance within 40 cm of the target, and its error bars remain within this threshold as well. This threshold, defined in Equation 3 for c_{d_t}, triggers the r_{arrive} reward, signifying successful arrival at the target. While Depth-CUPRL's median also stays within this threshold, its error bars are larger, and as Table 1 shows, it does not always reach the target. CURL (Depth) approaches the target in terms of median distance, but its error bars reveal that it does not consistently reach the 40 cm threshold. The remaining methods in Fig. 7 exhibit final median distances considerably farther from the target.

In the WA generalization experiments, Fig. 10 shows that in median all methods tested in this work are not close to the target in the end of the experiment. But for CUPRL, Depth-CUPRL, and CURL (Classic) have a minimum close to the thereshold c_{d_t}. This contributes with the Table 1, that shows that some cases in the experiment the methods arrived to the target position in the environment water-air.

In the WA generalization experiments for the first environment, Fig. 8 reveals that, for the median results, none of the tested methods come close to the target by the end of the experiment. However, CUPRL, Depth-CUPRL, and CURL (Classic) demonstrate a minimum distance close to the c_{d_t} threshold. This observation aligns with Table 1, which indicates that these methods did reach the target position in the water-air environment in some instances.

After training the neural networks in the second environment, another set of generalization experiments was conducted on the environment shown in Fig. 3b for the trained model. The evaluation results for all the techniques compared in this study are presented in Table 2. It is important to highlight the complexity of this scenario impacting all methods.

The evaluation in Table 2 shows that the proposed algorithm CUPRL and Depth-CUPRL [21] achieved the best results once again, while the other evaluated techniques



(a) CUPRL AW

(b) Depth-CUPRL AW

(c) CURL (Depth) AW

(d) CURL (Classic) AW

(e) SAC (CNN prio.) AW

(f) CUPRL WA

(g) Depth-CUPRL WA

(h) CURL (Depth) WA

(i) CURL (Classic) WA

(j) SAC (CNN prio.) WA

Fig. 6: Trajectories performed in the generalization experiment on the first scenario for AW and WA transitions.

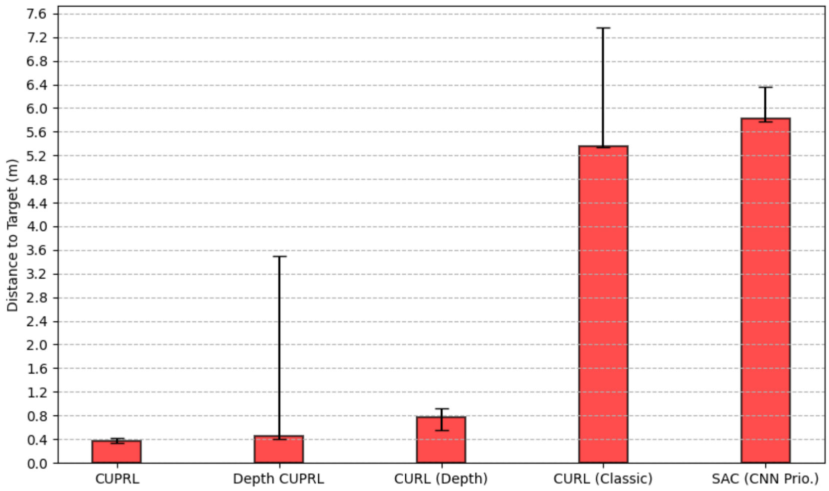


Fig. 7: Final median distance from the vehicle to the target in the generalization experiment on the first environment AW.

Algorithm	**Image**	**Air-water (%)**	**Water-air (%)**
CUPRL [ours]	RGB	**100%**	**14.6%**
Depth-CUPRL [21]	depth	**100%**	0%
CURL (Depth) [21]	depth	0%	0%
CURL (Classic) [20]	RGB	0%	0%
SAC (CNN prio.) [4]	depth	0%	0%

Table 2: Generalization experiment on the second scenario.

failed in all cases. Based on the evaluation results from the first and second environments, the proposed network demonstrated effectiveness in the air-water transition, but none of the evaluated networks achieved good performance in the water-air transition. This limitation can be attributed to the simulation (where the water surface is viewed as an obstacle), reward function, and training methodology used in this study, as the water in the simulation is seen as an obstacle.

The trajectories followed by the robot in the second scenario of evaluation are presented in Fig. 9.

The paths shown in Fig. 9 illustrate how the hybrid vehicle navigated the more complex second environment during 1000 evaluation episodes. CUPRL achieved a 100% success rate in the air-water transition and reached the target in some of the water-air transitions. On the other hand, Depth-CUPRL only succeeded in the air-water transitions. The other techniques evaluated, CURL (Depth), CURL (Classical), and SAC (CNN prio.), all displayed a behavior that generated collisions or hovering

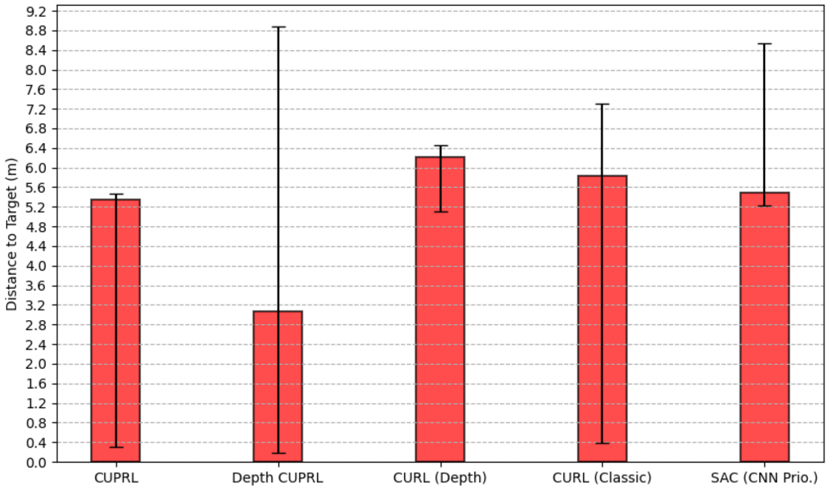


Fig. 8: Final median distance from the vehicle to the target in the generalization experiment on the first environment WA.

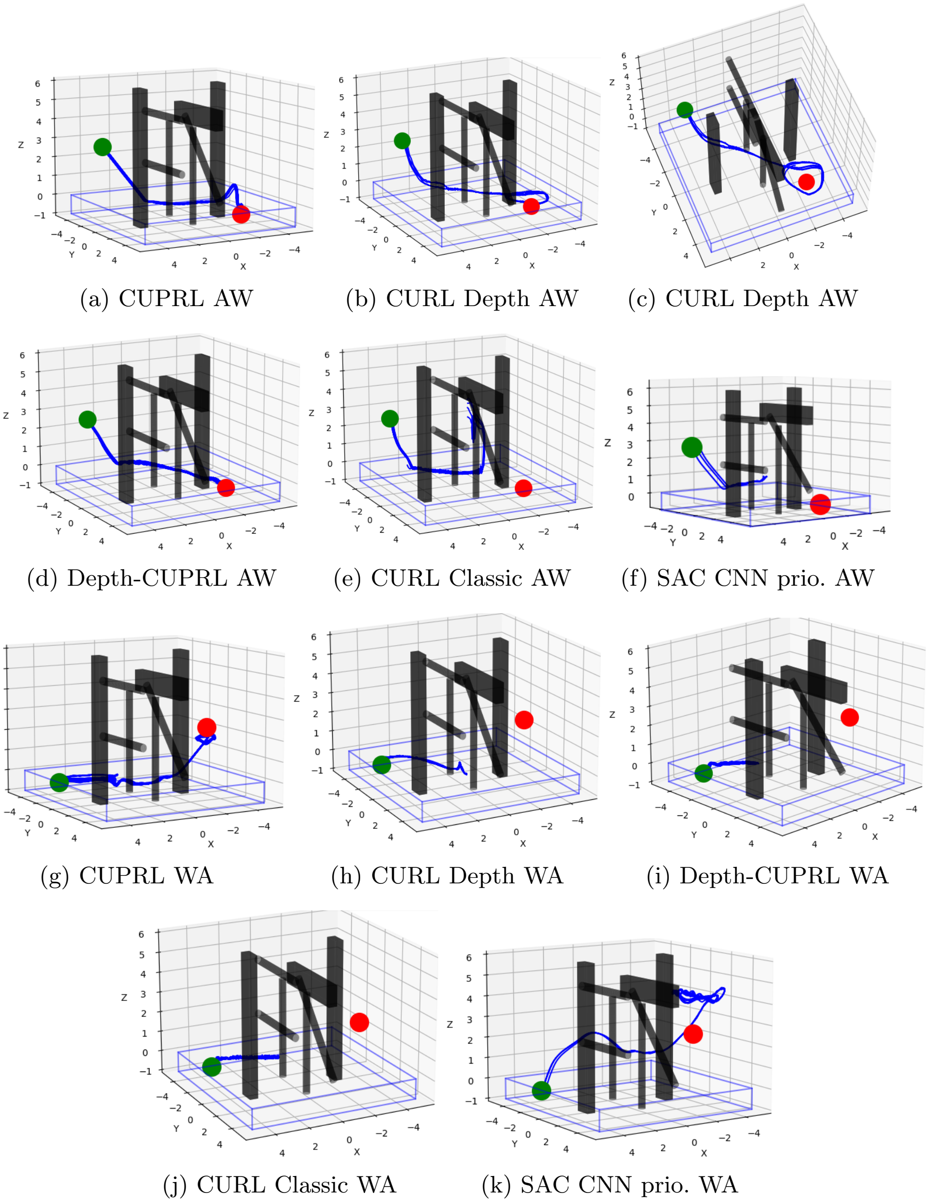
over the water. This hovering behavior can also be observed in the CURL (Depth) network. Another interesting behavior was observed in evaluating SAC (CNN prio.) during the water-air transition. During training, the SAC (CNN prio.) network was not able to receive positive rewards but did not receive negative rewards indicating collisions. The hovering behavior observed near the target could suggest that SAC (CNN prio.) could not understand the optimal policy for navigating to the target.

In the AW generalization experiments for the second environment, Fig. 10 closely mirrors the graph in Fig. 7. The primary distinctions lie in the maximum value for Depth-CUPRL and a slightly increased distance from the target for CURL (Depth). This suggests that the methods maintain consistent results even within a more complex environment.

In the WA generalization experiments for the second environment, Fig. 11 demonstrates that the median results for all methods are distant from the target. However, CUPRL's minimum distance reaches the c_{d_t} threshold, indicating some successful arrivals at the target position, as supported by Table 2. While SAC (CNN prio.) exhibits a median result closest to the target, the trajectories in Fig. 9k reveal a hovering behavior, suggesting that SAC (CNN prio.) struggled to learn optimal actions for this environment.

6 Conclusions

In this work, we proposed a navigation method using only RGB images applied to hybrid vehicles capable of operating in air and water. The motivation behind this



(a) CUPRL AW (b) CURL Depth AW (c) CURL Depth AW

(d) Depth-CUPRL AW (e) CURL Classic AW (f) SAC CNN prio. AW

(g) CUPRL WA (h) CURL Depth WA (i) Depth-CUPRL WA

(j) CURL Classic WA (k) SAC CNN prio. WA

Fig. 9: Trajectories performed in the generalization experiment on the second scenario for AW and WA transitions.

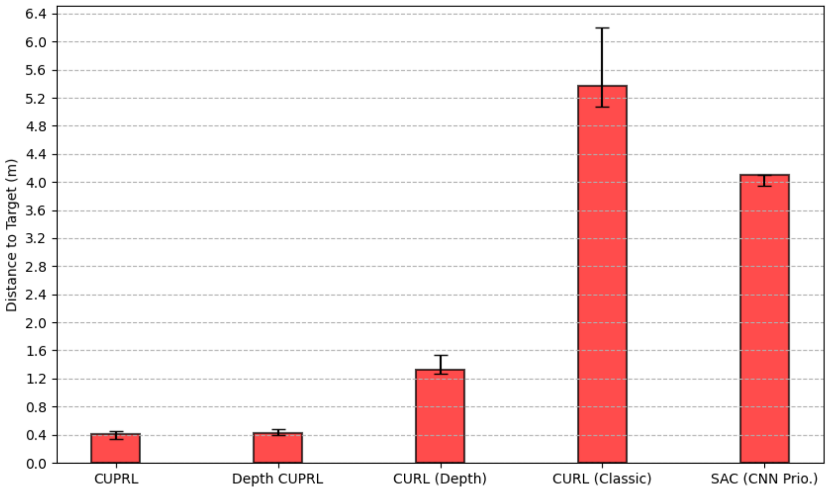


Fig. 10: Final median distance from the vehicle to the target in the generalization experiment on the second environment AW.

approach is that reinforcement learning networks are not sample-efficient in high-dimensional observation spaces [7]. The proposed approach was developed and applied to two tasks in a simulated 3D scenario, showing superior performance compared to other pixel-based approaches, as shown in Fig. 5a and Fig. 5b.

Our approach, called CUPRL, uses RGB images and depth maps as input to train a network that learns latent representations from image sequences. This method is used to navigate the hybrid vehicle and enables it to avoid collisions with objects during the 3D transition between air and water. The main contribution of this work is that the learned representations from the encoder, are extracted from the RGB images and depth maps during training, and just from the RGB images in the evaluation phase. It was found that the use of a combination of Deep-RL and Contrastive Learning resulted in superior performance when compared with other Deep-RL approaches based solely on visual information, even when the proposed method is applied only with RGB image information after training, as is the case with CUPRL.

In this study, we demonstrated that our Contrastive Learning technique can create a useful latent space from RGB and depth images, improving navigation for hybrid underwater and aerial vehicles in complex settings. These vehicles often face challenges due to varying light and contrast in air and water environments as well as dealing with medium transition. We believe our method offers the most effective approach to navigation tasks with visual inputs, and it holds promise for further development in this field.

In future work, we plan to better investigate the transition between water-air medium and explore additional techniques to improve the performance of our proposed

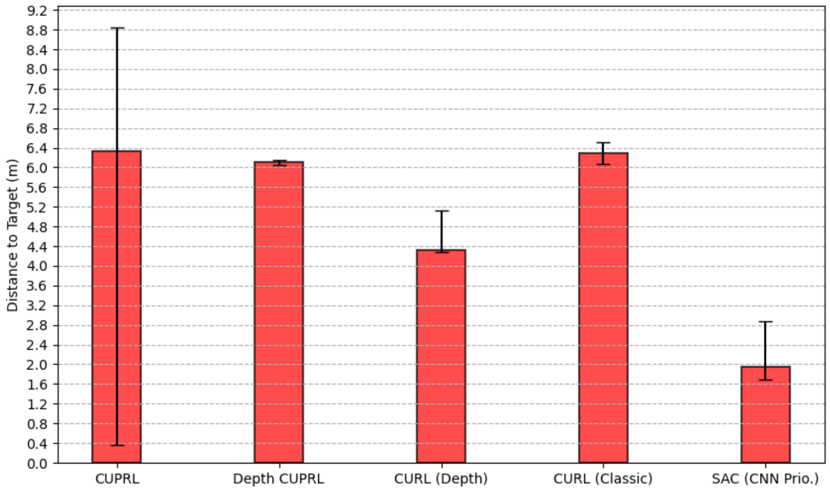


Fig. 11: Final median distance from the vehicle to the target in the generalization experiment on the second environment WA.

approach. We also plan to evaluate the algorithms in a real-world environment by navigating a real Hydrone vehicle.

Supplementary information. The source code has been made publicly available at the following URL: https://github.com/dranaju/cuprl_navigation.

Acknowledgements. The authors would like to thank the VersusAI team. This work was partly supported by the Technological University of Uruguay (UTEC), Federal University of Rio Grande (FURG), University of Tsukuba, FINEP, CNPq, and PRH-ANP 22. The authors also made use of language models for grammatical correction.

Declarations

- Funding:
  - National Council for Scientific and Technological Development (CNPq), Funding Authority for Studies and Projects, or Financier of Studies and Projects (FINEP) and National Agency of Petroleum, Natural Gas and Biofuels (PRH-ANP).
- Conflict of interest/Competing interests:
  - There are no conflict of interest or competing interest.
- Ethics approval and consent to participate:
  - The article has the approval of all the authors.

- Consent for publication:
  - All the authors gave their consent to participate in this article.
- Data availability: Not applicable
- Materials availability: Not applicable
- Code availability:
  - https://github.com/dranaju/cuprl_navigation
- Authors' contribution:
  - **Junior Costa de Jesus** conceived the research, writing the article, designed and program the experiments, collected and processed the test data.
  - **Ricardo Bedin Grando** writing of the article and processed the test data.
  - **Victor Augusto Kich** write the article and processed the test data.
  - **Alisson Henrique Kolling** write the article and processed the test data.
  - **Paulo Lilles Jorge Drews Jr.** conceived the research, writing of the article and discussion of the main ideas of the article.
  - **Rodrigo da Silva Guerra** conceived the research, writing of the article and discussion of the main ideas of the article.

References

[1] Alves, S.F., Rosario, J.M., Ferasoli Filho, H., Rincon, L., Yamasaki, R., Barrera, A.: Conceptual bases of robot navigation modeling, control and applications. Advances in Robot Navigation, 26 (2011)

[2] Guth, F., Silveira, L., Botelho, S., Drews-Jr, P., Ballester, P.: Underwater slam: Challenges, state of the art, algorithms and a new biologically-inspired approach. In: IEEE RAS/EMBS BioRob, pp. 981–986 (2014). https://doi.org/10.1109/BIOROB.2014.6913908

[3] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. In: ICLR (2016)

[4] Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: ICML, vol. 80, pp. 1861–1870 (2018)

[5] Grando, R.B., Jesus, J.C., Kich, V.A., Kolling, A.H., Pinheiro, P.M., Guerra, R.S., Drews-Jr, P.L.: Mapless navigation of a hybrid aerial underwater vehicle with deep reinforcement learning through environmental generalization. In: IEEE LARS/SBR, pp. 1–6 (2022)

[6] Bonatti, R., Madaan, R., Vineet, V., Scherer, S., Kapoor, A.: Learning visuomotor policies for aerial navigation using cross-modal representations. In: IEEE/RSJ IROS, pp. 1637–1644 (2020)

[7] Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behavioral and brain sciences **40** (2017)

[8] Tai, L., Liu, M.: Towards cognitive exploration through deep reinforcement learning for mobile robots. arXiv preprint arXiv:1610.01733 (2016)

[9] Grando, R.B., Jesus, J.C., Kich, V.A., Kolling, A.H., Drews-Jr, P.L.J.: Double critic deep reinforcement learning for mapless 3d navigation of unmanned aerial vehicles. Journal of Intelligent & Robotic Systems **104**(2), 1–14 (2022)

[10] Grando, R.B., Jesus, J.C., Drews-Jr, P.L.: Deep reinforcement learning for mapless navigation of unmanned aerial vehicles. In: IEEE LARS/SBR, pp. 1–6 (2020)

[11] Pinheiro, P.M., Neto, A.A., Grando, R.B., Silva, C.B.d., Aoki, V.M., Cardoso, D.S., Horn, A.C., Drews-Jr, P.L.: Trajectory planning for hybrid unmanned aerial underwater vehicles with smooth media transition. Journal of Intelligent & Robotic Systems **104**(3), 46 (2022)

[12] Rodriguez-Ramos, A., Sampedro, C., Bavle, H., Moreno, I.G., Campoy, P.: A deep reinforcement learning technique for vision-based autonomous multirotor landing on a moving platform. In: IEEE/RSJ IROS, pp. 1010–1017 (2018)

[13] Sampedro, C., Rodriguez-Ramos, A., Bavle, H., Carrio, A., Puente, P., Campoy, P.: A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. Journal of Intelligent & Robotic Systems, 601–627 (2019)

[14] He, L., Aouf, N., Whidborne, J.F., Song, B.: Integrated moment-based LGMD and deep reinforcement learning for UAV obstacle avoidance. In: IEEE ICRA, pp. 7491–7497 (2020)

[15] Li, B., Gan, Z., Chen, D., Sergey Aleksandrovich, D.: Uav maneuvering target tracking in uncertain environments based on deep reinforcement learning and meta-learning. Remote Sensing **12**(22), 3789 (2020)

[16] Jesus, J.C., Kich, V.A., Kolling, A.H., Grando, R.B., Cuadros, M.A.d.S.L., Gamarra, D.F.T.: Soft actor-critic for navigation of mobile robots. Journal of Intelligent & Robotic Systems **102**(2), 1–11 (2021)

[17] Thomas, D.-G., Olshanskyi, D., Krueger, K., Wongpiromsarn, T., Jannesari, A.: Interpretable uav collision avoidance using deep reinforcement learning. arXiv preprint arXiv:2105.12254 (2021)

[18] Grando, R.B., Jesus, J.C., Kich, V.A., Kolling, A.H., Bortoluzzi, N.P., Pinheiro, P.M., Alves Neto, A., Drews-Jr, P.L.J.: Deep reinforcement learning for mapless navigation of a hybrid aerial underwater vehicle with medium transition. In: IEEE

ICRA, pp. 1088–1094 (2021). https://doi.org/10.1109/ICRA48506.2021.9561188

[19] Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R.H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., et al.: Model-based reinforcement learning for atari. arXiv preprint arXiv:1903.00374 (2019)

[20] Laskin, M., Srinivas, A., Abbeel, P.: Curl: Contrastive unsupervised representations for reinforcement learning. In: ICML, pp. 5639–5650 (2020)

[21] Jesus, J.C., Kich, V.A., Kolling, A.H., Grando, R.B., Guerra, R.S., Drews-Jr, P.L.: Depth-CUPRL: Depth-imaged contrastive unsupervised prioritized representations in reinforcement learning for mapless navigation of unmanned aerial vehicles. In: IEEE/RSJ IROS, pp. 10579–10586 (2022)

[22] Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3733–3742 (2018)

[23] Polyak, B.T., Juditsky, A.B.: Acceleration of stochastic approximation by averaging. SICON **30**(4), 838–855 (1992)

[24] Roser, M., Dunbabin, M., Geiger, A.: Simultaneous underwater visibility assessment, enhancement and improved stereo. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 3840–3847 (2014). IEEE

[25] Drews-Jr, P.L., Neto, A.A., Campos, M.F.: Hybrid unmanned aerial underwater vehicle: Modeling and simulation. In: IEEE/RSJ IROS, pp. 4637–4642 (2014)

[26] Horn, A.C., Pinheiro, P.M., Grando, R.B., Silva, C.B., Neto, A.A., Drews-Jr, P.L.J.: A novel concept for hybrid unmanned aerial underwater vehicles focused on aquatic performance. In: IEEE LARS/SBR, pp. 1–6 (2020). https://doi.org/10.1109/LARS/SBR/WRE51543.2020.9307110