PPO-Based Autonomous Swarm Navigation for USVs: A Digital Twin Approach to Defend Brazil's Maritime Domain

André Siqueira Ruela e Marcelo Silva de Souza
Centro de Apoio a Sistemas Operativos - Marinha do Brasil, Rio de Janeiro/RJ - Brazil

***Abstract*— This paper investigates the scalability of Proximal Policy Optimization (PPO) for coordinating unmanned surface vehicles (USVs) in complex asymmetric warfare scenarios. We extend prior single-agent marine navigation research by establishing a configurable multi-agent reinforcement learning framework, capable of supporting more than 5 USVs, systematically evaluating performance across increasing agent populations (1-5 USVs) and environmental complexity levels (0-20 dynamic obstacles). Our *digital twin environment* integrates Crest Ocean System and Dynamic Water Physics 2 to simulate realistic maritime conditions. Experimental results reveal nuanced trade-offs: while Curriculum Learning (CL) significantly accelerates initial training, the baseline PPO achieves higher asymptotic success rates during the training phase. However, in final evaluations on complex multi-agent scenarios, PPO combined with CL demonstrates superior generalization and robustness (e.g., 89.66% mission success in 5-agent/20-obstacle configurations). This work provides critical insights into the interplay of training efficiency, asymptotic performance, and generalization in complex multi-agent Deep Reinforcement Learning (DRL) tasks for autonomous swarm systems.**

***Keywords*— multi-agent reinforcement learning, proximal policy optimization, unmanned surface vehicles**

I. Introduction

Modern asymmetric conflicts, such as the ongoing Russian-Ukrainian war, have underscored the strategic importance of unmanned surface vehicles (USVs) for maritime surveillance, force protection, and precision strike missions [1]. While single-USV operations have proven tactically effective [2], evolving naval doctrine now highlights the role of coordinated USV swarms in overwhelming traditional defenses and expanding the reach of maritime power [3]. This evolution poses critical challenges for the development of robust multi-agent navigation systems capable of operating reliably in dynamic and adversarial marine environments.

Current reinforcement learning (RL) approaches for marine navigation remain predominantly limited to single-agent scenarios [2], [4]. The seminal work of Luo et al. [2] established Long Short-Term Memory (LSTM)-enhanced Proximal Policy Optimization (PPO) as effective for individual USV obstacle avoidance, but their evaluation focused on static environments with only a handful of obstacles. However, operational realities — especially for nations responsible for defending vast maritime domains like Brazil's "Amazônia Azul" — demand scalable solutions that can coordinate multiple USVs through dense obstacle fields, while maintaining formation and mission objectives. This multi-agent setting poses significant challenges for deep reinforcement learning (DRL) methods due to the exponential growth of the state-action space [5].

This work addresses these challenges through three primary contributions: (1) A configurable and scalable multi-agent PPO framework, with experiments scaling up to 10 USVs and focused analysis of 1-5 agent scenarios; (2) Integration of high-fidelity hydrodynamic simulation using Crest Ocean System and Dynamic Water Physics 2, supporting realistic conditions that narrow the gap between simulation and real-world operation [6]; and (3) A systematic evaluation of PPO variants (vanilla, LSTM-enhanced, and curriculum-learned) across a comprehensive 5x5 complexity matrix of agent and obstacle configurations.

Our experimental paradigm extends Luo et al. [2]'s methodology into multi-agent contexts while preserving their core success metrics. In contrast to initial expectations, our evaluation reveals that while Curriculum Learning (CL) significantly accelerates early-stage training, baseline PPO ultimately achieves higher asymptotic performance. Notably, PPO+CL demonstrates stronger generalization in complex multi-agent scenarios, underscoring the nuanced trade-offs between sample efficiency, final performance, and robustness in advanced DRL applications.

This research is not only an advance in autonomous navigation, but also a step toward strengthening the technological autonomy and maritime security of Brazil. By equipping future USV swarms with scalable and adaptive intelligence, this work contributes to the broader mission of defending our national interests across the expansive Brazil's Maritime Domain.

The remainder of this paper is organized as follows: Section II reviews related work in maritime DRL. Section III details the simulation environment and PPO implementation. Section IV presents the experimental matrix and results. Section V discusses implications for real-world deployment, followed by conclusions in Section VI.

II. Related Work

Recent advances in DRL have significantly improved autonomous navigation and control of USVs. Traditional rule-based and potential field methods often lack the adaptability required for complex and dynamic maritime scenarios [4], [7]. DRL, particularly PPO, has shown promise for robust obstacle avoidance and path planning in simulated marine environments [2].

A. S. Ruela, andre.ruela@marinha.mil.br; M. S. Souza, marcelo-silva.souza@marinha.mil.br;

Luo et al. [2] introduced an LSTM-enhanced PPO approach for single-agent USV navigation, demonstrating improved performance in environments with static and dynamic obstacles. However, their work is limited to single-agent scenarios and does not address the scalability of PPO to multi-agent settings or increased environmental complexity.

Multi-agent reinforcement learning (MARL) is gaining traction for coordinated control of multiple autonomous systems. Yu et al. [5] highlighted that PPO, when properly configured, can be surprisingly effective for cooperative multi-agent tasks, although its application in maritime domains remains underexplored. Recent works have also emphasized the importance of realistic simulation environments, such as those built with Unity and advanced water physics engines, to bridge the sim-to-real gap for USV deployments [1], [8].

Additionally, CL has been shown to improve training efficiency and policy robustness by gradually increasing task difficulty [9]. While some studies have explored curriculum strategies for navigation, their integration with MARL and PPO in maritime contexts is still limited.

In summary, prior research [2], [5], [9] has established the effectiveness of DRL and PPO for single-agent USV navigation, but there is a clear gap regarding their scalability to multi-agent and high-complexity environments. This paper addresses this gap by systematically evaluating PPO and its variants across a matrix of agent and obstacle configurations in a high-fidelity maritime simulation, aiming to provide new insights into the adaptability and robustness of PPO for future real-world USV swarm deployments.

III. Methodology

A. PPO Formulation

Our multi-agent PPO implementation builds on the clipped objective function:

$$L^{\text{clip}}(\theta) = \mathbb{E}_t \left[\min \left(r_t \hat{A}_t, [r_t]_{1-\epsilon}^{1+\epsilon} \hat{A}_t \right) \right] \tag{1}$$

where:

$$r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)},$$
$$[r_t]_a^b = \max(\min(r_t, b), a),$$
$$\epsilon = 0.2 \quad \text{(clip range)}$$

where $r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ represents the policy probability ratio, and $\hat{A}_t$ denotes the generalized advantage estimate (GAE) computed across all agents. Thus, $\mathbb{E}_t$ denotes the expectation over timesteps t. The clipping parameter $\epsilon = 0.2$ constrains policy updates to prevent destructive large steps [10].

B. Training Configuration

All PPO agents were trained with a feedforward policy network of 3 hidden layers, each with 512 units, and Swish activations, guided by prior DRL studies [2], [4] and adjusted empirically for stability in our environment. Observations were normalized during training. Unless otherwise stated, the configuration used no memory modules, curiosity, or curriculum. The optimizer employed a batch size of 2048, a buffer size of 20480, and learning rate annealing from an initial value of 3×10^{-4} with linear decay. Table I summarizes the key hyperparameters.

TABLE I: DEFAULT PPO CONFIGURATION PARAMETERS

Parameter	Value
Batch size	2048
Buffer size	20480
Learning rate	3×10^{-4} (linear decay)
Entropy coefficient (β)	0.001 (linear decay)
Clip parameter (ϵ)	0.2
GAE lambda (λ)	0.95
Discount factor (γ)	0.995
PPO epochs per batch	3
Hidden layers	3×512 units
Time horizon	256
Max training steps	15 million

C. LSTM Integration

To handle partial observability in maritime environments, we augment the policy network with LSTM layers:

$$h_t, c_t = \text{LSTM}(o_t, h_{t-1}, c_{t-1}) \tag{2}$$

where o_t represents the current observation vector, h_t the hidden state, and c_t the cell state. The LSTM's gating mechanism enables temporal credit assignment for navigation decisions [11].

D. Curriculum Learning Strategy

Our phased training protocol follows:

$$\mathcal{C} = \{(n_a, n_o)_k | k = 1, ..., K; n_a \uparrow, n_o \uparrow\} \tag{3}$$

where n_a denotes agent count (1-5) and n_o obstacle density (0-20). The CL progresses when agents achieve a success rate greater than 80% for 100 consecutive episodes [9].

E. Simulation Environment

The simulation environment for training the PPO agent is developed using the Unity ML-Agents Toolkit, providing a flexible and efficient framework for training DRL agents in realistic maritime scenarios [12]. Unity ML-Agents connects Unity's powerful real-time 3D simulation to the PyTorch-based RL training pipeline via a dedicated Python API.

Formally, the environment is represented as a Markov Decision Process (MDP):

$$\mathcal{E} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}) \hookrightarrow \text{PyTorch} \tag{4}$$

where:

- $\mathcal{S}$: Observable states (e.g., heading, raycasts)
- $\mathcal{A}$: Actions (rudder, thrust)
- $\mathcal{T}$: Transition dynamics (Unity physics engine)
- $\mathcal{R}$: Reward signal from agent-environment interaction

The arrow ($\hookrightarrow$) denotes integration of these MDP elements into the PyTorch framework via the ML-Agents API. This setup facilitates real-time data exchange for training the agent's policy and value function.

During each episode, Unity provides observations to the agent, which selects an action via the PPO policy. The action is applied in Unity, and the resulting reward and new observation are returned. This loop continues until termination.

To approximate realistic maritime conditions, the simulation incorporates the Crest Ocean Renderer configured for moderate sea states. Wind speed was fixed at 20 km/h, employing the default "WavesBoatWakes" wave spectrum available in Crest's examples and using the Crest Shape FFT for wave synthesis. Although Crest supports extreme conditions up to 150 km/h, the moderate settings chosen ensure training stability and mirror commonly encountered operational scenarios. These settings significantly enhance realism compared to planar or idealized water models; however, the resulting wave fields do not yet match specific Beaufort sea states precisely, and further calibration would be required for strict real-world equivalence.

F. Observation Space

The agent's observation vector $\mathbf{o}_t \in \mathbb{R}^{87}$ combines proprioceptive and exteroceptive inputs, structured into four key components. Table II provides a comprehensive summary.

TABLE II: OBSERVATION SPACE COMPONENTS

Component	Dimensions	Description
Target State	4	• Normalized X/Z displacement • Heading alignment cosine (-1 to $+1$) • Normalized distance scalar
Ship-State	7	• Normalized thrust, speed, acceleration • Rudder angle (%) • Normalized angular velocity • Normalized pitch/roll ($\pm 90° \rightarrow \pm 1$)
Nearest Neighbor	4	• Normalized relative X/Z position • Heading alignment cosine • Collision risk (0 to 1)
Ray Perception	72	• 21 rays $\times$ [distance (norm), one-hot object type (friend/target/obstacle)] • Covers 180° front arc

This structured representation reduces reward engineering overhead compared to related work [2] by shifting complexity into the observation space, enabling greater generalization across navigation tasks. This design enables agents to adapt to a variety of operational scenarios without requiring reward reengineering, thus supporting flexible mission profiles. The standardized observation format also simplifies deployment across heterogeneous USV platforms by maintaining consistent input representations regardless of vessel specifications.

G. USV Vessel Modeling

The simulated USV utilized in this study is based on the Ukrainian maritime drone Magura V5, an unmanned surface vehicle (USV) developed for autonomous maritime operations [3]. Although the current simulation model does not yet represent a verified digital twin, significant efforts were undertaken to replicate its known maneuverability characteristics and hydrodynamic behavior. Model parameters, such as vessel dimensions, thrust, and other dynamics, were carefully calibrated using publicly available data to closely mimic the operational profile of the real-world vessel [3]. The 3D model of the Magura V5, utilized within the Unity simulation environment and illustred by Fig. 1, is publicly available online [13] under the Creative Commons Attribution license.

Fig. 1: 3D model of the Magura V5 USV [13].

Future work will focus on rigorous validation to ensure higher fidelity in representing the vessel's real-world performance, potentially leading to a fully validated digital twin suitable for real-world mission planning and training.

H. Reward Function

A well-designed reward function is essential for shaping effective agent behavior in RL. Our reward system combines several components to promote efficient navigation, continuous progress, and timely arrival at the target. In addition to dense exploration rewards and incremental distance-based incentives, the agent receives a significant goal reward upon successful arrival and is penalized for excessive deliberation. At the end of each episode, an efficiency bonus further encourages optimal route planning and swift completion.

1) Distance-Based Reward: The distance reward incentivizes progressive approach to the target:

$$r_{\text{dist}} = w_{\text{dist}} \cdot \begin{cases} \dfrac{d_{\text{min}}^{(t-1)} - d_t}{d_0}, & \text{if } d_t < d_{\text{min}}^{(t-1)} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where:

- $d_t = \|\mathbf{p}_v^{(t)} - \mathbf{p}_t\|$: Euclidean distance between the vehicle position $\mathbf{p}_v^{(t)}$ and the target position $\mathbf{p}_t$ at time t,
- $d_{\text{min}}^{(t)} = \min(d_{\text{min}}^{(t-1)}, d_t)$: closest achieved distance up to time t,
- $d_0 = \|\mathbf{p}_v^{(0)} - \mathbf{p}_t\|$: initial distance at episode start,
- w_{dist}: distance reward scaling factor.

2) Exploration Reward: To encourage continual movement and mitigate the sparse reward problem, we introduce a dense exploration bonus:

$$r_{\text{explore}}^{(t)} = w_{\text{explore}} \cdot \frac{\|\mathbf{p}_v^{(t)} - \mathbf{p}_v^{(t-1)}\|}{d_{\max}} \tag{6}$$

where:

- $\mathbf{p}_v^{(t)}$: vessel's position in the horizontal plane (X/Z) at time t,
- $d_{\max}$: maximum possible target distance in the environment,
- w_{explore}: exploration reward scaling factor.

3) Goal Reward: Upon successfully reaching the target, the agent receives a terminal reward:

$$r_{\text{goal}} = w_{\text{goal}} \cdot \mathbb{I}_{\text{goal}} \tag{7}$$

where $\mathbb{I}_{\text{goal}}$ is 1 if the target is reached and 0 otherwise, and w_{goal} sets the terminal reward magnitude.

4) Efficiency Bonus: At episode termination, an efficiency bonus is awarded based on the agent's path and time performance:

$$r_{\text{eff}} = w_{\text{eff}} \cdot \eta_{\text{path}} \cdot \eta_{\text{time}} \tag{8}$$

where:

- $\eta_{\text{path}} = \frac{d_{\text{direct}}}{d_{\text{actual}}} \in [0, 1]$: path efficiency (direct distance divided by actual distance traveled),
- $\eta_{\text{time}} = 1 - \frac{t}{t_{\max}} \in [0, 1]$: time efficiency (remaining time as a fraction of maximum episode duration),
- w_{eff}: efficiency bonus weight.

Here d_{direct} is the straight-line Euclidean distance between start and target. Since agents must maneuver around obstacles and respect vessel dynamics, the actual path is always longer, so $\eta_{\text{path}} < 1$ in practice. This bonus encourages agents to find efficient routes and complete tasks quickly.

5) Live Penalty and Collision: To discourage inefficient behavior, the agent incurs penalties for time consumption and unsafe navigation.

a) Live Penalty: A time-dependent penalty is applied at every timestep to encourage faster task completion:

$$r_{\text{live}}^{(t)} = -\frac{w_{\text{live}}}{t_{\max}} \tag{9}$$

where:

- w_{live}: Total penalty if the agent uses all $t_{\max}$ steps
- $t_{\max}$: Maximum allowed steps in the episode

This ensures the cumulative live penalty equals $-w_{\text{live}}$ over a full-length episode. If the episode ends earlier, the accumulated penalty is proportional to the steps used, so the agent receives less than $-w_{\text{live}}$.

b) Collision Penalty: If a collision with an obstacle or another vessel is detected, the agent immediately receives a fixed penalty and the episode terminates:

$$r_{\text{collide}} = -w_{\text{collide}} \cdot \mathbb{I}_{\text{collide}} \tag{10}$$

where $\mathbb{I}_{\text{collide}}$ is 1 if a collision occurs, and 0 otherwise, and w_{collide} sets the terminal penalty magnitude. This terminal penalty promotes cautious navigation and avoidance behavior.

These penalty components complement the sparse terminal rewards and provide additional shaping signals, particularly valuable during early training when successful episodes are infrequent.

6) Reward Function Summary: The complete reward function is composed of both sparse and dense components that guide the agent's learning process. Table III provides an overview of each term and its purpose in the behavior shaping strategy.

IV. Results

This section presents an empirical evaluation of PPO and its variants. Models were trained for 15 million steps using an NVIDIA RTX 3080 GPU. We analyze both the learning dynamics during training and the performance of the final converged policies, which were evaluated over 100 distinct episodes.

TABLE III: REWARD FUNCTION COMPONENTS

Component	Description
r_{dist}	Delta distance to target reward
r_{explore}	Exploration bonus (movement-based)
r_{goal}	Goal reward upon success
r_{eff}	Efficiency bonus (path + time)
r_{live}	Step penalty to encourage faster completion
r_{collide}	Terminal penalty applied upon collision

A. Training Dynamics and Convergence

Fig. 2 allows the comparison of the learning curves of all four PPO variants. The results show distinct learning behaviors. The PPO+CL agent (orange) demonstrates superior sample efficiency, achieving over 80% success rate within approximately 1 million steps, significantly faster than all other variants. However, its performance plateaus and is eventually surpassed by the baseline PPO, which achieves the highest final success rate during training.

The PPO-LSTM agent (green) exhibits the slowest initial learning, requiring nearly 2 million steps to reach a competitive success rate. The PPO-LSTM+CL agent shows moderate learning speeds, similar to the baseline PPO in the initial phases. After approximately 4 million steps, all algorithms converge to a similar performance band of 80-90%, with the primary differences being in their final ranking and stability.

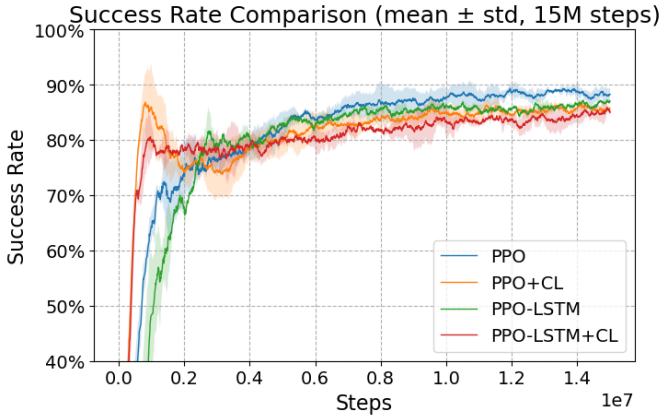

Fig. 2: Success rate during training (mean of runs, smoothed). PPO+CL shows the fastest initial convergence, but baseline PPO achieves a higher final success rate.

To quantify performance differences at convergence, we conducted independent-samples t-tests on the final 100 training records (last 15M steps). The results in Table IV reveal: (1) Baseline PPO significantly outperformed all enhanced variants ($p < 0.05$), with strong superiority over PPO+CL ($t = 4.906, p < 0.0001$) and PPO-LSTM+CL ($t = 5.103, p < 0.0001$); (2) Statistical analysis indicates PPO-LSTM+CL improved over PPO-LSTM alone ($t = 2.633, p = 0.0088$), though both remained below baseline; and (3) No significant difference existed between PPO+CL and PPO-LSTM+CL ($t = 0.613, p = 0.5405$), indicating CL provides similar benefits regardless of LSTM integration. The convergence plot (Fig. 2) uses a smoothing factor 0.98 for visual clarity, which emphasizes long-term trends over short-term variance captured in the statistical analysis.

TABLE IV: STATISTICAL COMPARISON OF SUCCESS RATES

Comparison	t-statistic	p-value
PPO vs. PPO-LSTM	2.190	0.0290
PPO vs. PPO+CL	4.906	< 0.0001
PPO vs. PPO-LSTM+CL	5.103	< 0.0001
PPO+CL vs. PPO-LSTM	-2.250	0.0249
PPO+CL vs. PPO-LSTM+CL	0.613	0.5405
PPO-LSTM vs. PPO-LSTM+CL	2.633	0.0088
All (Kruskal-Wallis H)	27.748	< 0.0001

B. Evaluation of Final Policies

To assess the robustness and generalization of the learned policies, the final models were evaluated over 100 separate episodes for each configuration. Table V presents these results.

TABLE V: SUCCESS RATE FROM 100-EPISODE EVALUATION

Configuration	1A-0O	3A-10O	5A-20O
PPO	100.0%	94.33%	86.79%
PPO + CL	100.0%	98.32%	89.66%
PPO-LSTM	100.0%	92.00%	79.96%
PPO-LSTM + CL	100.0%	91.33%	80.40%

The evaluation results largely align with the observations from the training curves and statistical analyses. In simpler scenarios (1A-0O, where 'A' denotes agents and 'O' denotes obstacles), all algorithms achieved perfect success. As complexity increased, the performance of PPO and PPO+CL remained higher than PPO-LSTM and PPO-LSTM+CL. Specifically, PPO+CL showed very strong performance, even surpassing baseline PPO in the 3A-10O and 5A-20O configurations (98.32% vs. 94.33%, and 89.66% vs. 86.79%, respectively). PPO-LSTM and PPO-LSTM+CL, while showing moderate performance, consistently scored lower than PPO and PPO+CL in complex settings. This indicates that the addition of LSTM, despite its theoretical benefits for partial observability, did not translate into superior performance in these multi-agent environments under the given training conditions.

Fig. 3 illustrates an alternative evaluation run demonstrating the generalization capabilities of the trained agents. The teal lines represent the trajectories of 10 USVs as they navigate a 1 km $\times$ 2 km maritime environment, substantially larger than the 300 m $\times$ 300 m training area. Red capsules denote dynamic obstacles, while the central island acts as a large static obstacle. The aircraft carrier in the background serves as the mission target. Despite the increased scale and complexity—featuring up to 60 obstacles—the agents successfully coordinate to reach the target, highlighting the robustness and adaptability of the learned policy beyond the training distribution.

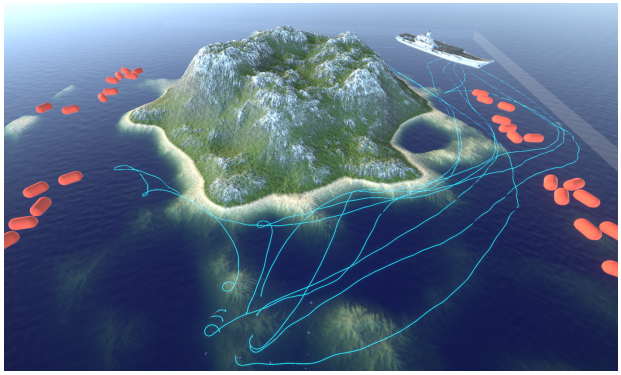

Fig. 3: Agent trajectories (teal) for 10 USVs navigating a 1 km $\times$ 2 km environment with up to 60 obstacles (red) and a central island, targeting the aircraft carrier.

C. Scalability and Failure Analysis

To understand performance degradation, we analyzed the scalability of the baseline PPO. Fig. 4 shows that success rates decline as both agent count and obstacle density increase. The sharpest performance drops occur in scenarios with more than three agents. Fig. 5 reveals the cause: starting from 2 agents, ship-to-ship collisions already account for nearly 80% of terminations, and they remain the dominant failure mode as the number of agents increases. This indicates that multi-agent coordination, not simple obstacle avoidance, is the primary challenge to scalability.

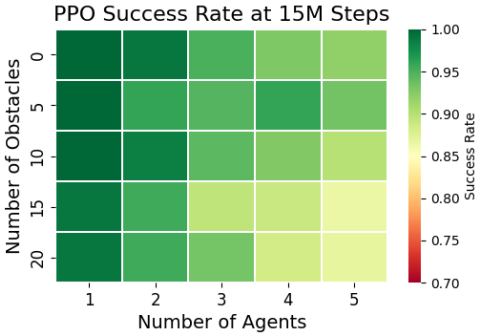

Fig. 4: Success rate of the baseline PPO algorithm at 15M steps across the agent-obstacle matrix.

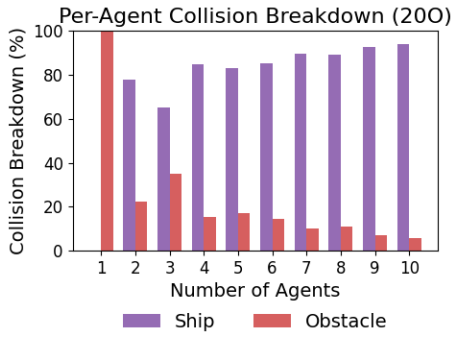

Fig. 5: Breakdown of failure reasons for baseline PPO over 100 episodes in a 20-obstacle environment. Models were not trained with 6 or more other agents.

D. Discussion

The results highlight a trade-off in applying DRL enhancements. CL greatly improves sample efficiency, valuable for resource-constrained projects. Our evaluation shows PPO+CL generalized better than other variants, achieving higher evaluation performance despite lower training scores. CL thus acts as a regularizer, guiding agents

to more robust policies beyond the training distribution. These findings underscore the need for a distinct evaluation phase to assess true policy generalization.

V. Deployment Challenges and Strategic Implications

The strategic value of autonomous USV swarms extends beyond technological advancement — it is a cornerstone for defending Brazil's "Amazônia Azul." Our 7,400-kilometer coastline demands solutions that operate independently across varied maritime conditions to secure sea lanes, protect resources, and deter threats.

a) Current Framework: Our simulation environment demonstrates robust policy learning for multi-agent coordination in complex scenarios, contributing to enhanced responsiveness against potential threats.

b) Deployment Roadmap: Bridging simulation and reality will require:

- **Output Standardization:** Agent outputs must be converted to maritime standards. We propose mapping heading control where $1 \rightarrow 0°$ (true north) and $-1 \rightarrow 360°$, with similar normalization for acceleration.
- **System Integration:** Converting outputs to conventional headings is essential for future naval integration. This standardization enables compatibility with the National Marine Electronics Association (NMEA) 0183 protocol, specifically the HDG (Heading) and HTD (True Heading) sentences, via a conversion layer, ensuring interoperability and rapid integration with command and control systems.
- **Environmental Fidelity:** Future work will prioritize a digital twin calibrated with Navy vessel data and domain randomization for operational sea states.

c) Validation Pathway: Before deployment, all systems will undergo hardware-in-the-loop testing, resilience verification across diverse zones, and incremental trials on prototypes and fleet vessels.

This phased approach ensures technological advancement aligned with national defense priorities, paving the way for a new paradigm in South Atlantic maritime security.

VI. Conclusions and Future Work

This research establishes a foundation for autonomous maritime operations serving Brazil's national security interests. Through comprehensive evaluation of PPO variants across multi-agent scenarios, we demonstrate the complex trade-offs between training efficiency and operational performance crucial for reliable defense systems.

Our findings reveal that while CL accelerates initial training — achieving 80% success rates within 1 million steps — baseline PPO demonstrates superior asymptotic performance. However, PPO+CL exhibits the strongest generalization capabilities, achieving 89.66% success in challenging 5-agent/20-obstacle configurations compared to 86.79% for baseline PPO. These results directly inform training methodology selection for operational USV systems.

The scalability analysis identified multi-agent coordination as the primary challenge, with ship-to-ship collisions accounting for over 80% of failures in dense formations. This emphasizes the critical importance of robust coordination protocols for swarm operations — capabilities essential for defending sea lanes and distributed maritime missions.

Looking ahead, our research focuses on expanding framework capabilities to include dynamic targets, communication constraints, and adversarial scenarios that mirror realistic threat environments. The demonstrated scalability supporting up to 10 agents provides a pathway for larger swarm studies relevant to fleet-scale operations. Complete integration with maritime communication standards ensures deployment aboard existing and future Brazilian Navy platforms without extensive hardware modifications.

By advancing autonomous maritime technologies, the Brazilian Navy reinforces its commitment to technological sovereignty. This work positions Brazil as a leader in maritime autonomy while ensuring naval forces remain prepared for warfare challenges. These capabilities serve national security objectives and contribute to Brazil's emergence as a global leader in autonomous maritime systems.

The path from simulation to operational deployment requires sustained commitment to excellence, but the strategic advantages for defending Brazil's Maritime Domain are vast. As we advance these technologies, we honor our maritime heritage while securing Brazil's future on the seas.

References

[1] D. Menges, A. Von Brandis, and A. Rasheed, "Digital twin of autonomous surface vessels for safe maritime navigation enabled through predictive modeling and reinforcement learning," in *Volume 5B: Ocean Engineering of International Conference on Offshore Mechanics and Arctic Engineering*, 2024.

[2] W. Luo, X. Wang, F. Han, Z. Zhou, J. Cai, L. Zeng, H. Chen, J. Chen, and X. Zhou, "Research on LSTM-PPO obstacle avoidance algorithm and training environment for unmanned surface vehicles," *Journal of Marine Science and Engineering*, vol. 13, no. 3, p. 479, 2025.

[3] Militarnyi, "Ukraine revealed details about the magura v5 maritime drone," 2024, accessed: 2025-06-18. [Online]. Available: https://militarnyi.com/uk/news/ukrayina-rozkryla-detali-pro-udarni-morski-bpa

[4] X. Peng, F. Han, G. Xia, W. Zhao, and Y. Zhao, "Autonomous obstacle avoidance in crowded ocean environment based on COLREGs and POND," *Journal of Marine Science and Engineering*, vol. 11, no. 7, p. 1320, 2023.

[5] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," 2022, arXiv preprint arXiv:2103.01955.

[6] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," *arXiv preprint arXiv:2008.06634*, 2020.

[7] A. A. Vekinis and S. Perantonis, "Aeolus ocean – a simulation environment for the autonomous COLREG-compliant navigation of unmanned surface vehicles using deep reinforcement learning and maritime object detection," 2023, arXiv preprint arXiv:2307.06688.

[8] H. S. Berg, D. Menges, T. Tengesdal, and A. Rasheed, "Digital twin syncing for autonomous surface vessels using reinforcement learning and nonlinear model predictive control," *Scientific Reports*, vol. 15, p. 9344, 2025.

[9] S. Morad, R. Mecca, J. Horgan, and C. Eising, "Autonomous curriculum learning for robot navigation in complex environments," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1234–1247, 2023.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] A. Juliani, V. Walker, C. D. Berrios, H. Teng, T. Haarnoja, E. Vincent, A. Cartland, A. Yadav, B. Stadie, S. Sully *et al.*, "Unity: A general platform for intelligent agents," in *arXiv preprint arXiv:1809.02627*, 2020.

[13] MotionForgeCG, "Ukrainian navy maritime drone 2 - magura v5," 2023, sketchfab, CC Attribution License. Accessed: 2025-06-18. [Online]. Available: https://sketchfab.com/3d-models/ukrainian-navy-maritime-drone-2-a3724696b39f42a8a240b54706a11dd4