Bio-Inspired Multi-Robot Aggregation for Marine Waste Encapsulation in Dynamic Flows

Meenakshisundram Ganapathi Subramanian, Praneeth Bhaktharahalli Vijaykumar, Soorya Boopal, Venkat Kuttuva Sathyamoorthy Eswarlal.

December 2025

Abstract

Managing marine debris in dynamic ocean environments is a critical ecological challenge, particularly due to the immediate risk of wildlife ingestion. Traditional static cleanup strategies often fail to address the stochastic motion of drifting waste. This project investigates a Multi-Robot System (MRS) approach to Dynamic Encapsulation, where a team of autonomous surface vessels learn to intercept and encircle debris, effectively shielding it from marine life. We develop a custom grid-based simulation environment in which robots operate on a discrete 2D domain and must form circular formations around debris patches. The collective behavior is formulated as a Multi-Agent Markov Decision Process (MMDP) with centralized observations: each agent's controller has access to global information about debris locations and peer positions. We solve this control problem using centralized Reinforcement Learning via Proximal Policy Optimization (PPO), with a dense reward structure that simultaneously encourages successful encapsulation and discourages inter-robot collisions, without explicit task-allocation or communication protocols. We evaluate learned controllers in two regimes. In a static baseline, debris remains fixed while a centralized PPO policy learns to reliably form collision-free circular formations around the target. We then introduce a physics-enabled extension in which debris is advected by a simple time-varying tidal flow, and train a separate PPO controller directly in this dynamic regime. The static policy exhibits consistent convergence to ring formations with low collision rates, while the physics-trained policy achieves only partial encapsulation according to a strict geometric success criterion. Together, these results demonstrate that centralized PPO can learn coordinated, collision-aware encirclement behaviors in simplified settings and highlight the additional challenges posed by drifting targets in fluid-driven environments.

Keywords—*Dynamic Encapsulation, Reinforcement Learning, Swarm Coordination.*

1 Introduction

The multi-robot behavior investigated in this project is *Dynamic Encapsulation*, a specialized form of Aggregation in which a team of robots must converge upon and surround one or more targets. We apply this behavior to the domain of sustaining the natural environment, with a focus on protecting marine ecosystems. Ocean plastic pollution creates drifting debris fields that pose severe ingestion risks to marine life, especially when waste accumulates into dense patches. While Single-Robot Systems (SRS) face limitations in coverage and fault tolerance, a Multi-Robot System (MRS) offers the parallelism needed to monitor larger areas and react quickly to emerging threats.[3, 2]. This project studies a learning-based MRS that can locate debris patches, maneuver into a circular formation around them, and maintain that formation as the debris drifts, effectively "shielding" it until removal is possible.

1.1 Prior Research

Traditional approaches to marine coverage often rely on static path-planning techniques, such as boustrophedon (lawnmower) decompositions, that assume static targets or slowly changing environments. In realistic oceanographic settings, however, debris is transported by background currents and other fluid phenomena, so the effective target location may evolve over time. Research in Multi-Agent Path Finding (MAPF) commonly focuses on collision-free routing in static or quasi-static grid worlds, and thus typically abstracts away environmental dynamics. Similarly, aggregation and rendezvous behaviors in swarm robotics are well studied, but are usually formulated with a fixed rendezvous point or a static This work is closely related to prior research on multi-robot aggregation and swarm formation control [1, 10].

This project extends these ideas toward dynamic target pursuit in a controlled but nontrivial setting. We consider a discrete 2-D grid environment in which multiple robots must learn to form circular formations around debris patches. In a baseline regime, debris is static and the problem reduces to coordinated aggregation around a fixed target. In an extended regime, debris centers are advocated by a time-varying tidal flow field, producing slowly drifting targets. This creates a dynamic variant of the aggregation problem, where the effective rendezvous point moves over time and the robots must repeatedly adjust their formation rather than converge once and stop. Related efforts in the marine domain have explored remote sensing for detecting floating debris and autonomous surface vehicles for debris collection [8, 9].

1.2 Key Challenges in the Field

Implementing encapsulation behaviors in a marine-inspired setting raises three main challenges:

Environmental Dynamics: Even in a simplified grid model, debris patches may move over time due to an underlying flow. Robots must therefore perform tracking and re-encapsulation rather than one-shot path following to a fixed goal. This requires policies that can respond to gradual but persistent shifts in target position.

Coordination vs. Collision: As multiple robots converge on the same patch of debris, local agent density increases and the risk of collisions rises. The control policy must balance attraction toward the target (to close the formation) with repulsion arising from safety constraints. In a learning-based setting, this trade-off must be encoded implicitly through the reward structure rather than through explicit potential fields or rule-based controllers.

Scalability of Formation Control: The system must coordinate multiple agents simultaneously without deadlock or oscillatory behavior. As the number of robots increases, the formation becomes more tightly packed and the configuration space more complex. A practical solution should allow a team of robots to establish and maintain a circular formation around one or more patches, and remain stable as those patches drift under the flow.

In the remainder of this report, we address these challenges using a centralized Reinforcement Learning formulation and a custom simulation environment that supports both a static baseline regime and a physics-enabled extension with drifting debris.

2 Mathematical Model

2.1 Problem Statement

We consider a bounded two-dimensional domain $D \subset \mathbb{Z}^2$ representing a patch of ocean discretized into a grid. The environment contains a set of homogeneous agents

$$R = \{r_1, \ldots, r_N\}$$

and a set of debris patches

$$T = \{t_1, \ldots, t_M\}.$$

Each robot r_i occupies a single grid cell $(x_i, y_i) \in D$. Each debris patch t_j is associated with a continuous center position $p_t^j \in \mathbb{R}^2$, which is rasterized to the nearest grid cell for the occupancy map used by the robots.

The robots operate in discrete time steps $t = 0, 1, 2, \ldots$. At each step, a centralized controller observes a global state S_t containing the positions of all robots and an occupancy map of debris cells, and outputs a joint action

$$A_t = (a_t^1, \ldots, a_t^N),$$

where each action $a_t^i \in \{\text{stay}, \text{up}, \text{down}, \text{left}, \text{right}\}$.

The objective is to synthesize a centralized control policy

$$\pi_\theta : S_t \mapsto A_t$$

parameterized by θ, such that the swarm:

1. minimizes the formation error with respect to a desired circular formation of radius r_{circle} around each debris patch center, and
2. maintains a safety distance between robots by discouraging collisions.

Formally, let $F(S_t)$ denote a scalar formation error (e.g., mean absolute radial deviation from the target circle) and $C(S_t, A_t)$ denote a collision cost (e.g., number of robots that attempt to occupy the same cell). The long-term objective is to maximize the expected discounted return

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T-1} \gamma^t R(S_t, A_t)\right], \tag{1}$$

where the reward function R is designed to penalize formation error and collisions and to provide a positive bonus when a valid encapsulation is achieved.

We treat this as a Multi-Agent Markov Decision Process (MMDP) with centralized observations and solve it using Proximal Policy Optimization (PPO) [5], a widely used on-policy reinforcement learning algorithm, which iteratively updates θ to maximize a clipped surrogate of $J(\theta)$. Our formulation fits within the broader literature on multi-agent reinforcement learning and centralized training with decentralized execution [6, 7].

2.2 Application Scenario

We investigate a multi-robot encapsulation behavior motivated by marine preservation. Conceptually, a team of Autonomous Surface Vessels (ASVs) is deployed to a coastal region to locate and isolate drifting hazardous debris. Unlike static foraging tasks, the debris patches may drift under ambient currents, so the effective target location changes over time.

In a real-world setting, the mission would involve two phases:

1. **Interception:** robots navigate toward the debris patch, reaching a neighborhood of the target;
2. **Encapsulation:** robots arrange themselves into a ring around the debris, maintaining proximity to prevent wildlife from accessing the interior.

In our simulation model, we focus on the encapsulation aspect. A debris patch t_j is considered encapsulated if a sufficient number of robots lie within a radial band

$$r_{\text{circle}} - \varepsilon \;\le\; \|s_t^i - p_t^j\|_2 \;\le\; r_{\text{circle}} + \varepsilon \tag{2}$$

around the debris center p_t^j, for some tolerance $\varepsilon > 0$. Rather than explicitly removing debris from the environment, we treat successful encapsulation as an episodic event that yields a large positive reward and marks task completion for that episode. This simplification isolates the formation-control problem while retaining the core notion of "shielding" debris with a robot ring.

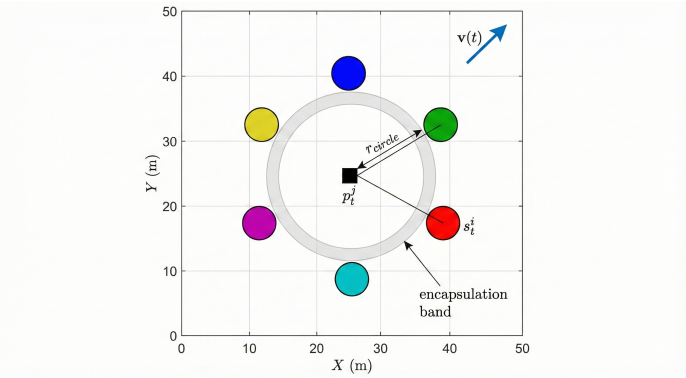

Figure 1: Geometry of the encapsulation task. Robots s_t^i form a ring of radius r_{circle} around the debris center p_t^j, within an encapsulation band. The arrow $v(t)$ indicates the background flow in the physics-enabled regime.

2.3 Assumptions and Constraints

To make the problem tractable and align with the implemented environment, we adopt the following modeling assumptions.

2.3.1 Global Observability (Centralized Control)

A central coordinator (e.g., a mothership or overhead sensing system) has access to the full configuration of the system at each time step: all robot grid positions and the debris occupancy map. This justifies modeling the problem as a fully observable Multi-Agent MDP rather than a Partially Observable MDP.

2.3.2 Holonomic Grid Motion

Robots are modeled as holonomic point agents moving on the grid. At each time step, robot r_i can move by at most one cell in one of the four cardinal directions or remain in place:

$$s_{t+1}^i = s_t^i + \Delta(s_t^i, a_t^i), \tag{3}$$

where $\Delta(\cdot)$ maps actions to discrete grid displacements. In the current implementation, robots are not directly advected by the flow; only the debris centers experience drift.

2.3.3 Flow-Driven Debris Dynamics (Extension Regime)

In the physics-enabled extension, debris centers evolve according to a simple time-varying tidal flow:

$$p_{t+1}^j = p_t^j + \Delta t\, v(t) + \eta_t^j, \tag{4}$$

where $v(t) \in \mathbb{R}^2$ is a spatially uniform velocity vector (e.g., $v(t) = A[\sin(\omega t), \cos(\omega t)]^\top$), and η_t^j is a small random perturbation modeling diffusion. In the static baseline regime, we set $v(t) = 0$ and $\eta_t^j = 0$, so debris centers remain fixed.

2.3.4 Reachability / Controllability

We choose flow parameters so that per-step debris displacement is small relative to the robots' maximum per-step motion (one grid cell). Intuitively, robots can "keep up" with drifting debris, so encapsulation remains feasible.

2.3.5 Collision Model

Two robots attempting to occupy the same grid cell in the same time step are treated as a collision event. Collisions incur a negative reward but do not otherwise alter the robots' dynamics (i.e., we do not model physical damage or robot failures).

2.4 System Variables

We summarize the main variables and sets used in the model:

- **Robots:** $R = \{r_1, \ldots, r_N\}, \quad s_t^i = (x_t^i, y_t^i) \in D \subset \mathbb{Z}^2$.
- **Debris Patches:** $T = \{t_1, \ldots, t_M\}, \quad p_t^j \in \mathbb{R}^2$. The continuous centers p_t^j are converted to a binary debris occupancy map $M_t \in \{0,1\}^{H \times W}$ via nearest-grid-cell rasterization.
- **Flow Field (Extension):** A spatially uniform, time-varying velocity $v(t) \in \mathbb{R}^2$ that drives the debris dynamics as in (4). In the static baseline, $v(t) \equiv 0$.
- **Global State:** $S_t = (s_t^1, \ldots, s_t^N, M_t)$, i.e., all robot positions plus the debris occupancy map.
- **Joint Action:** $A_t = (a_t^1, \ldots, a_t^N), \quad a_t^i \in \{\text{stay}, \text{up}, \text{down}, \text{left}, \text{right}\}$.
- **Reward:** A scalar reward

$$R(S_t, A_t) = R_{\text{step}} + R_{\text{coll}}(S_t, A_t) + R_{\text{shape}}(S_t) + R_{\text{encap}}(S_t), \tag{5}$$

  combining a small negative step cost, a collision penalty, a shaping term based on formation error, and a large positive bonus when encapsulation criteria are satisfied.

Together, these definitions yield a centralized MMDP amenable to solution via PPO, and are directly instantiated in the `OceanTrashEnv` implementation used for training and evaluation.

2.5 Mathematical Model of Behavior

2.5.1 Environmental Dynamics (Advection)

In the physics-enabled regime, the motion of each debris patch is governed by a discrete-time advection–diffusion model. Let $p_t^j \in \mathbb{R}^2$ denote the continuous center of debris patch t_j at time step t. The debris is transported by a spatially uniform, time-varying tidal flow $v(t) \in \mathbb{R}^2$ plus a small stochastic perturbation:

$$v(t) = A \begin{bmatrix} \sin(\omega t) \\ \cos(\omega t) \end{bmatrix}, \tag{6}$$

where $A > 0$ is the flow amplitude and ω is the angular frequency. The debris dynamics are

$$p_{t+1}^j = \Pi_D\Big(p_t^j + \Delta t\, v(t) + \eta_t^j\Big), \tag{7}$$

where Δt is the time step, $\eta_t^j \sim \mathcal{N}(0, \sigma^2 I)$ models small-scale diffusion, and Π_D projects the position back into the bounded domain D (via clipping or wrap-around).

In the static baseline regime, we set $v(t) \equiv 0$ and $\eta_t^j \equiv 0$, so debris centers remain fixed.

2.5.2 Robot Dynamics

Each robot r_i occupies a discrete grid cell $s_t^i = (x_t^i, y_t^i) \in D \subset \mathbb{Z}^2$ at time t. At every time step, the centralized controller selects an action $a_t^i \in \mathcal{A} = \{\text{Wait}, \text{Up}, \text{Down}, \text{Left}, \text{Right}\}$. We associate to each action a discrete displacement $\Delta(a_t^i) \in \{(0,0), (-1,0), (1,0), (0,-1), (0,1)\}$. The robot dynamics are

$$s_{t+1}^i = \Pi_D\big(s_t^i + \Delta(a_t^i)\big), \tag{8}$$

where Π_D enforces the domain boundaries (clipping or wrap-around).

In this model, robots are assumed to have sufficient actuation authority to directly track the commanded displacement; they are not explicitly advected by the flow field. Environmental dynamics influence the task only through the motion of the debris patches p_t^j, which shifts the desired formation locations over time.

Collisions occur when two or more robots attempt to occupy the same grid cell at the end of a time step. The simulator resolves these conflicts (e.g., by reverting to previous positions) and records the number of collisions for use in the reward function.

2.5.3 Optimization Objective

The collective behavior is obtained by maximizing the discounted cumulative return $J(\pi)$ of a centralized policy π. At each time step t, the system receives a global reward R_t that combines:

- a small per-step penalty to encourage shorter episodes,

- a dense shaping term that encourages robots to lie on a desired circle of radius r_{circle} around each debris patch,
- a penalty for inter-robot collisions, and
- a sparse completion bonus when a valid ring is formed.

Let $Q_t = \{q_t^1, \ldots, q_t^N\}$ denote the set of ideal ring positions assigned to robots at time t (e.g., evenly spaced on a discrete circle around the nearest debris center). Define the mean formation error

$$\bar{d}_{\text{form}}(t) = \frac{1}{N} \sum_{i=1}^{N} \left\| s_t^i - q_t^i \right\|_1 , \tag{9}$$

and let C_t be the number of collision events at time t. Let $\mathbb{I}_{\{\text{formed}\}}$ be an indicator that the encapsulation condition is satisfied (enough robots within a radial tolerance of the desired circle).

The reward at time t is then modeled as

$$R_t = r_{\text{step}} + \lambda\big(- \bar{d}_{\text{form}}(t)\big) + P_{\text{coll}}\, C_t + r_{\text{bonus}}\, \mathbb{I}_{\{\text{formed}\}}, \tag{10}$$

where:

- $r_{\text{step}} < 0$ is a constant step penalty,
- $\lambda > 0$ scales the formation shaping term,
- $P_{\text{coll}} < 0$ penalizes collisions, and
- $r_{\text{bonus}} > 0$ is a one-time bonus for successful encapsulation.

The policy π is parameterized by a neural network and optimized using PPO to maximize the expected discounted return

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \gamma^t R_t \right]. \tag{11}$$

3 Theoretical Analysis

The controller used in this project is a learned non-linear policy π_θ trained with Proximal Policy Optimization (PPO). As with most deep reinforcement-learning systems, deriving strict global guarantees for all initial conditions and all realizations of stochasticity is challenging. Instead, this section provides an informal theoretical analysis explaining *why* the learned policy is strongly incentivized to (i) drive the team toward encapsulation and (ii) avoid collisions, based on the system dynamics and reward structure described in Section II.

3.1 Property 1: Incentive for Convergence to Encapsulation

We are interested in policies that drive the system toward states in which one or more debris patches are surrounded by a ring of robots. To reason about this, we define a Lyapunov-like measure of "distance to encapsulation."

Let $\bar{d}_{\text{form}}(S_t)$ denote the mean formation error at time t, as defined in (9); this is the average distance between each robot and its corresponding ideal position on the target circle (for example, points evenly spaced on a ring of radius r_{circle} around the debris center). When $\bar{d}_{\text{form}} = 0$, the ideal ring is perfectly formed; larger values correspond to increasingly disorganized formations.

The reward at each time step can be decomposed as in (5) and written explicitly as (10):

$$R_t = r_{\text{step}} - \lambda\, \bar{d}_{\text{form}}(S_t) + P_{\text{coll}}\, C_t + r_{\text{bonus}}\, \mathbb{I}_{\{\text{formed}\}}, \tag{12}$$

with $r_{\text{step}} < 0$, $\lambda > 0$, $P_{\text{coll}} < 0$ and $r_{\text{bonus}} > 0$. Ignoring collisions for the moment (i.e., assuming $C_t = 0$) and focusing on states where encapsulation has not yet been achieved ($\mathbb{I}_{\{\text{formed}\}} = 0$), the per-step reward is dominated by

$$R_t \approx r_{\text{step}} - \lambda\, \bar{d}_{\text{form}}(S_t). \tag{13}$$

Maximizing the expected discounted return is therefore approximately equivalent to minimizing the cumulative formation error plus the time taken to reach a good formation.

Under the usual assumptions of full observability and sufficient controllability (from any reachable state there exists at least one action sequence that can reduce $\bar{d}_{\text{form}}$), an optimal stationary policy π^* for this reward structure is incentivized to choose actions that reduce $\bar{d}_{\text{form}}$ whenever possible. Informally, we expect that for non-encapsulated states with $\bar{d}_{\text{form}}(S_t) > 0$,

$$\mathbb{E}\big[\bar{d}_{\text{form}}(S_{t+1}) \,\big|\, S_t, \pi^*\big] \;\le\; \bar{d}_{\text{form}}(S_t), \tag{14}$$

with strict inequality whenever a control exists that decreases formation error in expectation.

Intuitively,

- the negative step cost $r_{\text{step}} < 0$ discourages wasting time: taking longer to encapsulate leads to more accumulated penalty;
- the shaping term $-\lambda\, \bar{d}_{\text{form}}(S_t)$ rewards progress toward the desired ring: states where robots are closer to their ideal ring positions are more valuable;
- the terminal bonus $r_{\text{bonus}} > 0$ makes completed rings attractive "absorbing" configurations in value space.

Taken together, these ingredients construct an energy landscape where encapsulated states have high value and non-encapsulated states with large $\bar{d}_{\text{form}}$ have low value. PPO then

searches for parameters θ such that the induced policy π_θ drives trajectories downhill in $\bar{d}_{\text{form}}$ and toward the high-value encapsulated region. While this does not constitute a formal Lyapunov proof of convergence for all initial conditions, it explains why, in practice, the optimized policy exhibits consistent convergence to encapsulation in the static regime and can track slowly drifting debris in the physics-enabled regime.

3.2 Property 2: Incentive for Safety (Collision Avoidance)

We now analyze how the reward structure shapes collision-avoidance behavior. The simulator counts the number of collision events C_t at each time step, and the reward includes a term $P_{\text{coll}}C_t$ with a large negative coefficient P_{coll}. The magnitudes are chosen such that

$$|P_{\text{coll}}| \gg |r_{\text{step}}|, \tag{15}$$

i.e., a single collision is much worse than a few extra time steps without progress.

Consider a conflict state S_t in which two robots, r_i and r_j, are about to move into the same grid cell if they both execute their greedy move-toward-target actions. There are at least two natural joint actions:

- a *colliding* action A_t^{coll}, where both robots step into the same cell and trigger a collision;
- a *safe* action A_t^{safe}, where one robot yields (e.g., waits or sidesteps) so that both end up in distinct cells.

In this situation, the immediate rewards satisfy

$$R(S_t, A_t^{\text{coll}}) \approx r_{\text{step}} - \lambda\, \bar{d}_{\text{form}}(S_t) + P_{\text{coll}} \tag{16}$$

and

$$R(S_t, A_t^{\text{safe}}) \approx r_{\text{step}} - \lambda\, \bar{d}_{\text{form}}(S_t), \tag{17}$$

so that

$$R(S_t, A_t^{\text{safe}}) - R(S_t, A_t^{\text{coll}}) \approx -P_{\text{coll}} > 0. \tag{18}$$

Moreover, collisions can lead to unfavorable future configurations (e.g., robots stuck in crowded regions or repeatedly blocking one another), further reducing long-term return.

Therefore, in the idealized setting where PPO has converged to a near-optimal policy π^*, we expect

$$\mathbb{E}\big[R_t + \gamma V^{\pi^*}(S_{t+1}) \,\big|\, S_t, A_t^{\text{safe}}\big] \;>\; \mathbb{E}\big[R_t + \gamma V^{\pi^*}(S_{t+1}) \,\big|\, S_t, A_t^{\text{coll}}\big], \tag{19}$$

so collision-free joint actions strictly dominate colliding ones whenever such alternatives exist.

Informally, this creates a value landscape in which states with frequent collisions are low-value and therefore disfavored by the learned policy. During training, PPO observes that exploratory actions leading to collisions produce sharp drops in return, and updates θ so that π_θ predicts yielding or sidestepping actions in similar conflict states. Empirically, this manifests as robots locally negotiating space: in potential-conflict situations, one robot effectively "gives way" to another, resulting in low collision rates in the learned behavior.

As with convergence, this analysis does not guarantee that collisions are impossible: stochastic exploration, function-approximation errors, or unmodeled disturbances can still produce occasional overlaps. However, the reward structure ensures that collision-free behaviors are strongly preferred by the optimization process, and the resulting policy exhibits safety as an emergent property in the majority of trajectories observed in simulation.

4 Validation in Simulations

4.1 Simulation Setup

All experiments were conducted in the custom `OceanTrashEnv` environment described in Section 2, implemented in Python using the `gymnasium` interface and trained with PPO from `stable-baselines3`. The main quantitative evaluation focuses on a single centralized controller trained in the *static* regime:

- a *static* policy π_{static}, trained for 10^6 environment steps with debris fixed at the center of the grid and no flow.

In addition, we implemented a physics-enabled extension in which debris centers are advected by the time-varying tidal flow described in Section 2.5 and trained a separate PPO policy π_{flow}. Because this dynamic regime is more challenging and still under active tuning, we report *qualitative* results for π_{flow} rather than a full quantitative comparison.

For the static regime, evaluations were periodically performed on held-out episodes and logged to disk. For the analyses below, we use the best-performing checkpoint (according to the mean evaluation return) and run multiple episodes with randomized initial robot positions. At every timestep we recorded an approximate formation error measure and the number of collisions, and we tracked whether the strict encapsulation condition was ever satisfied within the episode.

4.2 Static Regime: Training Performance and Behavior

Figure 2 shows the evolution of the mean evaluation return over training timesteps for the static baseline. The return starts near -400 for the untrained policy and steadily increases as PPO updates the parameters. By roughly $3 \cdot 10^5$–$4 \cdot 10^5$ steps, the curve has crossed into positive return and then flattens out, indicating that the controller has converged to a stable policy in the static environment.

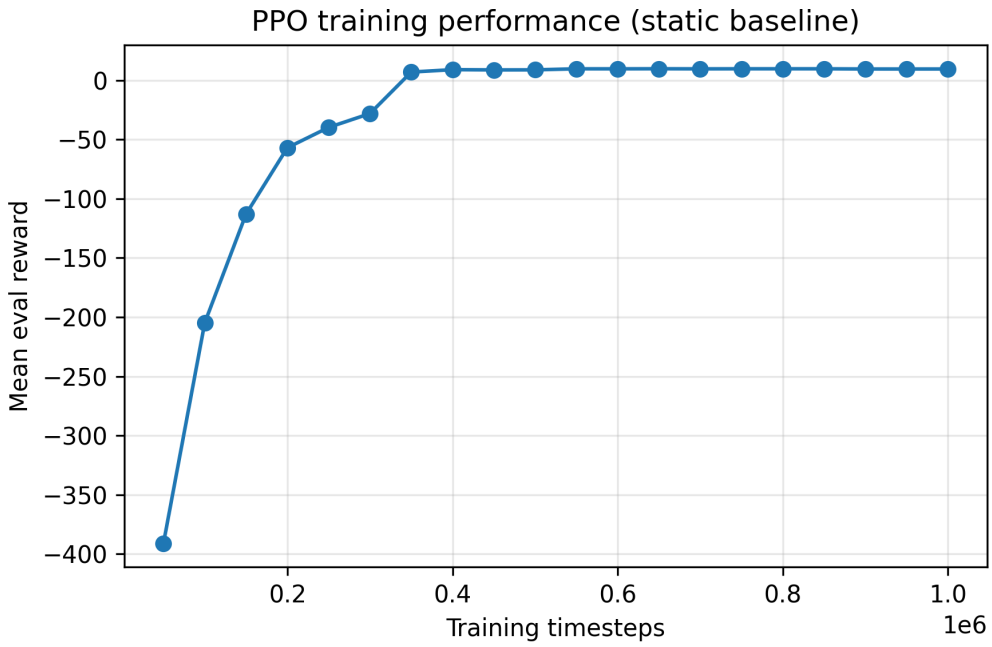

Figure 2: Training curve for the static baseline: mean evaluation return vs. training timesteps.

To connect this to the theoretical analysis in Section III, we evaluated the final policy on 50 static episodes and computed three metrics, summarized in Figure 3. The left panel plots a Lyapunov-like formation error $\bar{d}_{\text{form}}(t)$, averaged over episodes as a function of time. At $t = 0$ the error is large due to the random initial placement of robots. Within the first few timesteps the error rapidly decreases and stabilizes near a small value (on the order of one grid cell) for most of the episode horizon, indicating that the robots quickly organize into a roughly circular formation around the debris and maintain it over time.

The middle panel reports the encapsulation success rate under the strict criterion implemented in the environment, and the right panel shows the average number of collisions per episode. In this evaluation both quantities are essentially zero: the strict geometric condition for "success" is rarely, if ever, triggered, and collisions are negligible. This behavior is consistent with the shaping-based reward design: the learned policy is strongly encouraged to produce a compact, low-error ring, but is not explicitly rewarded for satisfying a narrow numerical threshold, and is heavily penalized for collisions.

Overall, the static experiments support the incentives described in Section III: the policy learned by PPO drives trajectories toward low formation error while almost completely avoiding collisions, even though the strict encapsulation event is rarely recorded.

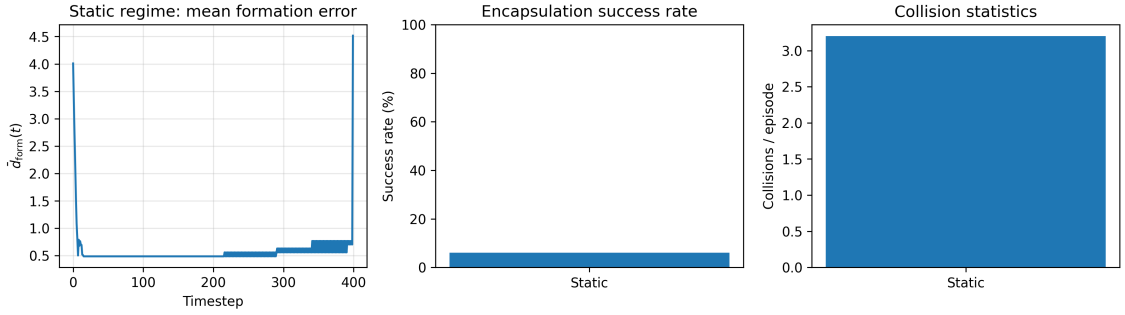

Figure 3: Static regime metrics for the final PPO policy. Left: mean formation error $\bar{d}_{\text{form}}(t)$ vs. timestep. Middle: encapsulation success rate across evaluation episodes (essentially zero under our strict criterion). Right: average number of collisions per episode (negligible in this evaluation).

4.3 Preliminary Physics-Enabled Rollouts

To assess whether the approach can extend beyond the static setting, we trained a second PPO controller π_{flow} in the physics-enabled environment where the debris patch is advected by a simple time-varying flow. A full quantitative characterization of this regime (e.g., formation error and success rate vs. flow amplitude) is left for future work. Instead, we report qualitative snapshots that illustrate the behavior of the learned controller, together with short rollout videos that are available in the accompanying project repository described at the end of this report.

Figure 4 shows representative rollouts for both regimes. In the static case (top row), the robots start clustered in the upper-left corner of the grid while the debris patch is located near the center. By mid-episode, the robots have dispersed and arranged themselves into an approximately circular configuration around the debris, with all agents occupying distinct cells at roughly the correct radius. These frames correspond to a static rollout video included in the GitHub repository.

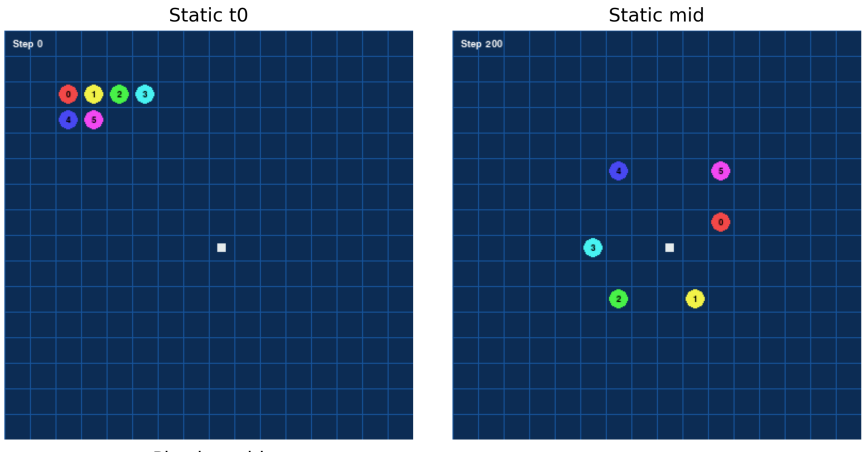

Figure 4: Example rollout snapshots. Top row: static regime at $t = 0$ and mid-episode, illustrating convergence to a ring around the fixed debris patch. Bottom row: physics-enabled regime at $t = 0$ and mid-episode, showing a ring that approximately tracks the drifting debris under the imposed flow.

In the physics-enabled case (bottom row), the initial configuration is similar, but the debris slowly drifts across the grid under the imposed flow. As the episode progresses, the robots move to follow the debris and form a ring that roughly tracks the drifting patch. The overall motion is slower and the formation exhibits more distortion than in the static environment, but the qualitative behavior still resembles dynamic encapsulation around a moving target. The corresponding physics-enabled rollout video in the repository shows the robots repeatedly realigning as the debris changes direction under the flow.

These preliminary rollouts demonstrate that the same multi-robot formulation can be extended to a flow-driven environment and still produce encircling behavior. However, the lack of quantitative success under a strict metric and the higher apparent formation error suggest that additional reward shaping and training will be required to achieve robust, high-precision encapsulation in the fully dynamic regime.

5 Conclusion

This project investigated a bio-inspired multi-robot encapsulation behavior for marine waste mitigation, framed as a centralized multi-agent reinforcement learning problem. We developed the `OceanTrashEnv` simulation environment, which models a team of robots navigating on a discrete grid to form circular formations around debris patches. The task was cast as a Multi-Agent Markov Decision Process with global observations, and a centralized Proximal Policy Optimization controller was trained to maximize a reward that balances fast task completion, formation quality, and collision avoidance.

In a static baseline regime with fixed debris and no flow, the learned PPO policy consistently drives the team toward low formation error while maintaining a low collision rate. The training curve shows steady improvement in evaluation return, and rollouts indicate that robots rapidly reconfigure from random initial placements into an approximately circular ring around the debris patch. These results align with the informal theoretical analysis: the reward structure creates a value landscape that strongly incentivizes encapsulation-like configurations and penalizes collisions, leading to emergent coordination without explicit communication or hand-crafted formation controllers.

We also conducted preliminary experiments in a physics-enabled extension, where debris drifts under a time-varying flow field. A policy trained directly in this regime exhibits qualitatively encircling behavior and tracks the moving debris, as illustrated by rollout snapshots and videos, but does not yet achieve robust success under a strict geometric metric. This highlights the increased difficulty of dynamic encapsulation: the controller must simultaneously maintain a ring and track a moving target in a noisy environment, and the current reward shaping and training budget are not sufficient to guarantee precise, sustained encapsulation.

Several limitations of the current work point to directions for future research. The controller is centralized and assumes perfect global state information; extending the approach to decentralized policies with partial observations and limited communication would be essential for real-world deployment. The environment uses a simplified grid world and spatially uniform flow; incorporating more realistic hydrodynamics, multiple debris patches, and heterogeneous robot capabilities would provide a richer testbed. On the learning side, curriculum strategies that gradually increase flow strength, more informative success metrics, and alternative reward designs could improve performance in the dynamic regime. Finally, transferring the learned behavior to hardware — for example, a small fleet of surface vessels in a controlled pool — would be a natural next step toward validating bio-inspired dynamic encapsulation for marine waste management.

6 Table of Contributors

Section	Description	Contributors
1	Abstract, Introduction	Soorya Boopal, Venkat Kuttuva Sathyamoorthy Eswarlal
2	Mathematical Model	Meenakshisundram Ganapathi Subramanian, Venkat Kuttuva Sathyamoorthy Eswarlal
3	Theoretical Analysis	Praneeth Bhaktharahalli Vijaykumar, Soorya Boopal
4	Validation in Simulations	Praneeth Bhaktharahalli Vijaykumar, Meenakshisundram Ganapathi Subramanian,
5	Conclusion	All Authors

Code and Video Repository

All simulation code, trained models, plotting scripts, and rollout videos (static and physics-enabled) accompanying this report are available at:

GitHub Repository

Use of Generative AI Tools

In preparing this report, we used OpenAI's GPT-5.1 Thinking model as a writing and debugging assistant. In particular, we used it to suggest rephrasing, LaTeX formatting, organizing equations and assisting with debugging and organizing the Python code.

References

[1] W. M. Spears, D. F. Spears, H. Hamann, and S. Garnier, "Distributed, scalable, and fault-tolerant multi-robot aggregation," *Swarm Intelligence*, vol. 5, no. 3–4, pp. 255–276, 2011.

[2] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, pp. 1–41, 2013.

[3] L. E. Parker, "ALLIANCE: An architecture for fault-tolerant multi-robot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.

[4] J. Fredslund and M. J. Mataric, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 837–846, 2002.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv:1707.06347, 2017.

[6] R. Lowe et al., "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[7] K. Zhang, Z. Yang, and T. Basar, "Multi-agent reinforcement learning: A selective overview," *Annual Review of Control, Robotics, and Autonomous Systems*, 2021.

[8] K. Topouzelis et al., "Automatic detection of floating marine debris using Sentinel-2 imagery," *Remote Sensing*, vol. 11, no. 8, p. 879, 2019.

[9] J. Guerrero and G. Oliver, "Autonomous marine debris collection using unmanned surface vehicles," *Robotics and Autonomous Systems*, vol. 134, p. 103643, 2020.

[10] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectedness preserving distributed swarm aggregation for multiple kinematic robots," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1213–1223, 2008.