Carpenter Ant-Inspired *Decentralized Collective Intelligence* for Swarm Robotics

BACKGROUND

MOTIVATIONS

Problem

Modern AI systems scale intelligence through centralized architectures by increasing model size on specialized hardware [1–4]. In centralized multi-robot planning systems, computational complexity grows rapidly with swarm size, making centralized coordination increasingly **fragile and unscalable** [6].

Inspiration

Individual carpenter ants are simple, yet collectively they exhibit efficient, complex behaviour. Interestingly, this is achieved **without central control** or command hierarchy [7].

Solution

A **decentralized multi-agent system** coordinated through stigmergy mimicking carpenter ants archives coordinated scalable behaviour without central control.

Goal

Evaluate whether **collective intelligence can emerge and scale** through local interactions among simple agents, providing an alternative to centralized AI architectures.

Research Question

Does stigmergic communication improve the scalability of decentralized swarm systems as the number of agents increases?

SWARM INTELLIGENCE IN NATURE

Carpenter ants (*Camponotus spp.*) exhibit extraordinary **collective intelligence** despite severe individual limitations:

- Individual ants operate with restricted local sensing and minimal memory.
- Yet, as a collective, ants solve complex tasks such as shortest path discovery, resource allocation, and adaptive re-routing [7].
- These capabilities emerge through **stigmergy,** a mechanism of indirect coordination mediated by the environment [9].
- Ants deposit **pheromones** proportional to path utility, and subsequent ants bias their motion along the pheromone gradients. This creates a **distributed positive feedback loop**; frequently traversed paths are reinforced while unused paths decay [9].

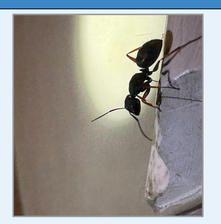


Figure 1. *Camponotus modoc, a carpenter ant species common in British Columbia.*

To understand the natural behavior of Carpenter Ants, **direct observations** were conducted in controlled settings:

- Ants were captured and placed in a bounded environment with introduced food sources.
- A consistent exploration-exploitation transition was observed: Initial random motion followed by rapid convergence onto stable routes.
- These routes exhibited reinforcement and persistence as ants followed the trail; a hallmark of stigmergic coordination.

These observations inspired the design of my system. Implementing “digital stigmergy”, the system enables agents to coordinate without direct communication.

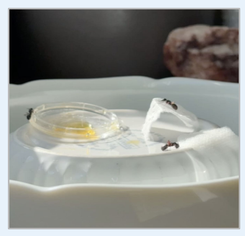


Figure 2. *Conducting observations on ants.*

SYSTEM

SIMULATION ENVIRONMENT

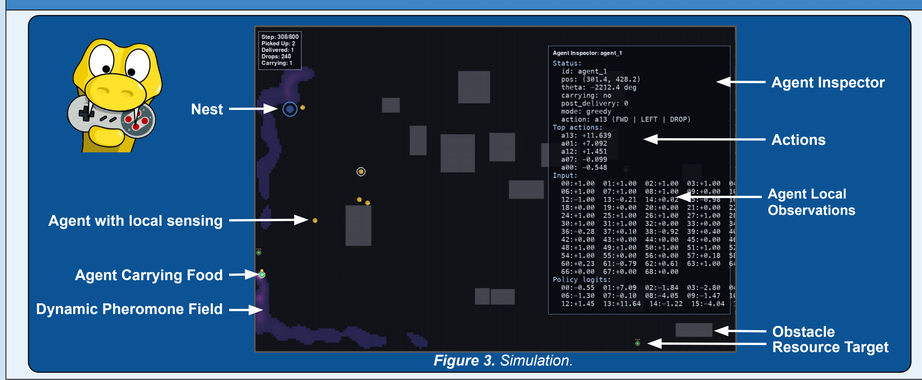


Figure 3. *Simulation.*

A 2D continuous simulation environment was developed using **PyGame** to conduct research.

- Agents were created and trained in the simulation.
- The simulation was used to conduct experiments validating the learned model.
- The simulation models agent motion, collisions, and spatial interactions.

A **digital pheromone field** was created.

- The field is updated continuously based on agent actions.
- The pheromone field decays over time, enabling temporal encoding of path information.

AGENT MODEL

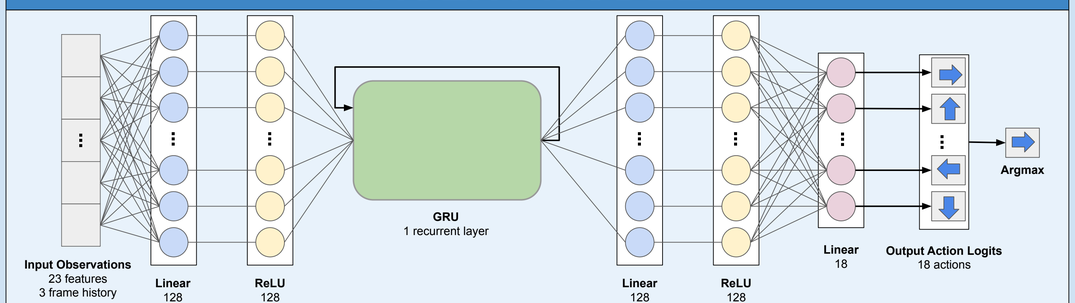


Figure 4. *Model architecture for decision-making, trained through reinforcement learning. At each timestep, agents observe their local surroundings and independently choose actions for movement and pheromone deposition. Through training, the agents learned cooperative behaviors such as exploration, food retrieval, and coordinated trail-following without centralized control or direct communication.*

TRAINING PIPELINE & LEARNING FRAMEWORK

Curriculum Guided Reinforcement Learning (CRL) Via Recurrent Multi-Agent Proximal Policy Optimization (RMAPPO) with Gated Recurrent Unit (GRU)

Multi-Agent Environment State
At each timestep, the environment stores agent positions, obstacles, food, and pheromones.

Agents

Local Observation Construction
Each agent receives its own local observation.

Observation Preprocessing and Normalization
Raw environment data is converted into a normalized observation vector.

Temporal Processing
Three consecutive 23-dimensional frames are stacked to form a 69-dimensional observation, providing short-term temporal context.

Per-Agent Policy Input
Each agent receives its local observation and feeds it into an identical policy, maintaining its own recurrent hidden state.

Actor Network
Observations are processed through an **MLP**, **GRU**, and **policy head** to produce an action distribution.

Environment Update
All agents act simultaneously, and the environment updates accordingly.

Critic

Trajectory Rollout Collection
On-policy trajectories are collected over multiple timesteps, including observations, actions, rewards, and done signals.

Centralized Critic Input
A centralized recurrent critic uses a richer state representation to evaluate the joint system.

Value Estimation
The critic processes input through an **MLP**, **GRU**, and **value head** to produce a scalar estimate of expected future reward.

Generalized Advantage Estimation
Trajectories and rewards are combined with critic estimates to compute advantages and returns.

Clipped Policy Update
Proximal Policy Optimization (PPO) updates the policy while constraining policy changes for stability.

Curriculum Learning Stages
Training is conducted across 17 environments organized into 3 phases of increasing difficulty. By first learning foundational skills in simpler settings, the agent can efficiently solve a complex target task that would be difficult to learn from scratch [20].

REINFORCEMENT LEARNING

Reinforcement Learning (RL) is a machine learning method focused on training an “agent” to make optimal decisions in a dynamic environment [14].

The goal of RL is to:
Learn a policy that maximizes total future reward over time.

Reward Shaping
Rewards are defined to guide learning. Intermediate rewards are also used to accelerate training.

Key Rewards
Pickup = +6.0
Delivery = +30.0
Collision = -1.5
Undelivered Resource = -10.0

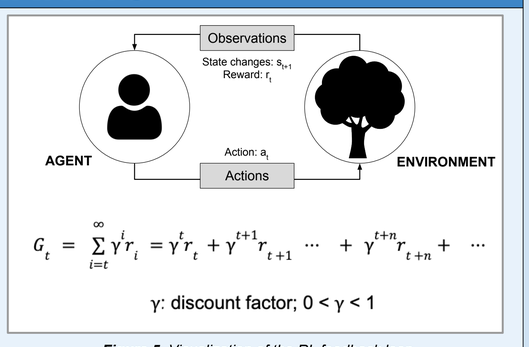


Figure 5. *Visualization of the RL feedback loop.*

An agent learns to make sequential decisions through interaction with an environment.

- The objective is not to maximize immediate reward, but to maximize expected cumulative future reward over time. This is formalized as $\mathbf{G_t}$.
- To make decisions, the agent must estimate how good a state is in terms of future outcomes. This is captured by the **value function $\mathbf{V^{\pi}(s)}$**.
- The **Bellman Equation** defines the value function recursively as the sum of immediate reward and discounted future value.

This is critical in this research because rewards are delayed. Exploration actions may yield no immediate reward, but are necessary for later successful food collection and delivery.

Value Function

$$V^{\pi}(s) = \mathbb{E}[G_t \mid s_t = s]$$

Bellman Equation

$$V(s) = \mathbb{E}[r + \gamma V(s')]$$

MULTI-AGENT PROXIMAL POLICY OPTIMIZATION

Policy Learning

A **policy distribution** is learned directly.

- Policy learning optimizes a policy so that it maximizes expected return.
- An **advantage function** enables learning, measuring whether an action is better or worse than expected. This reinforces beneficial actions and suppress poor ones.

Proximal Policy Optimization (PPO)

The policy is optimized using PPO.

- PPO updates the policy while constraining how much it can change in a single step.
- This is achieved through a clipped objective, ensuring stable learning [17].

Multi-Agent Proximal Policy Optimization (MAPPO)

RL is extended to a **multi-agent setting** using MAPPO (Multi-Agent PPO).

- Each agent follows an identical policy but acts based only on its **local observation** [15].
- This makes the system decentralized at execution [15].

Centralized Training with Decentralized Execution (CTDE)

Training multiple agents independently is unstable.

- An individual's learning is disrupted by the simultaneously changing behaviour of others [15].
- CTDE resolves this by providing agents global information through a centralized critic during training. Agents still act solely on local observations during execution [15, 16].

Policy

$$\pi(a \mid o)$$

Advantage Function

$$A(s,a) = Q(s,a) - V(s)$$

Given…

$$r_t = \frac{\pi_{\text{new}}}{\pi_{\text{old}}}$$

PPO Clipped Objective Function

$$L^{\text{CLIP}} = \mathbb{E}\left[\min\left(r_t A_t,\ \text{clip}(r_t, 1-\epsilon, 1+\epsilon)\, A_t\right)\right]$$

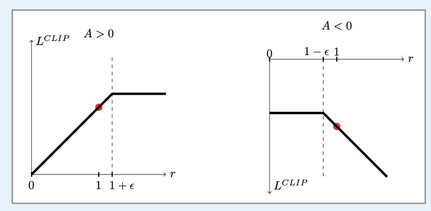


Figure 6. *Plots showing one term of the surrogate function L^{CLIP} [17].*

CURRICULUM LEARNING

Phase 1
Single agent, targeting:
- Picking up resources
- Delivering resources
- Managing obstacles

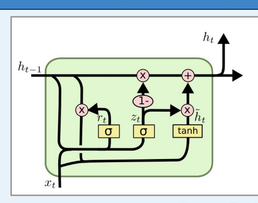
Phase 2
Small swarm, targeting:
- Picking up and delivering resources together
- Avoid interfering with other agents
- Form rudimentary pheromone trails

Phase 3
Full swarm, targeting:
- Scaling coordination across many agents
- Handling multiple resources and obstacles
- Pheromone trails for route formation
- Stable performance in the final complex environment

GATED RECURRENT UNITS

The model incorporates a type of **Recurrent Neural Network (RNN)** called **Gated Recurrent Units (GRUs)**.

- Enables each agent to maintain a **memory** of past observations.
- It works by introducing a dedicated cell state that persists over time and interacts with a set of gates to control information flow [18, 19].



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

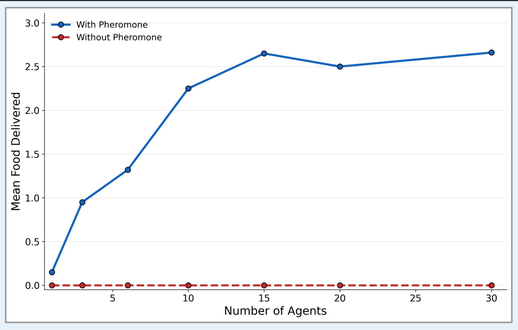
Figure 7. *GRU cell [19].*

RESULTS & VALIDATION

Hypothesis

A curriculum-trained stigmergic swarm will outperform non-stigmergic and non-learning controls on foraging performance, and this advantage will increase with swarm size.

STIGMERGY



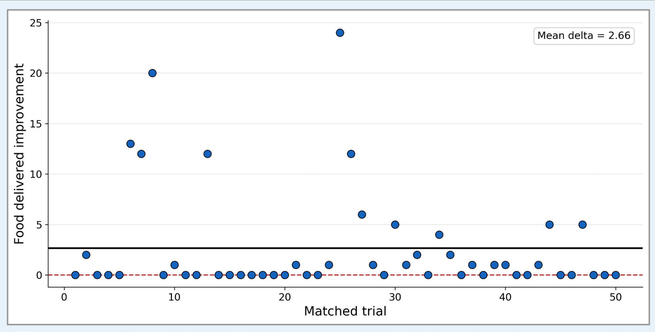
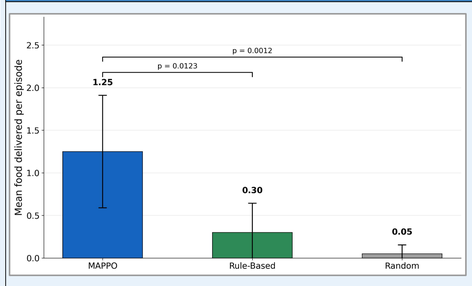


Figure 9. *Two-way ANOVA revealed a highly significant effect of pheromone condition on food delivery performance* ***($p < 10^{-8}$)****. Although the interaction effect was not statistically significant, strong positive scaling trends were observed under stigmergic communication, while non-pheromone swarms remained near zero performance across all swarm sizes. Spearman correlation analysis on all raw episode data showed a significant positive relationship between swarm size and food delivery* ***($\rho = 0.2045$, $p = 0.00368$)****.*

Figure 10. *Trial-by-trial improvement in food delivery from pheromone-enabled coordination. A paired Wilcoxon signed-rank test on matched 30-agent evaluations showed that pheromone-enabled swarms significantly outperformed non-pheromone swarms* ***($n = 50$ pairs, mean $\Delta = 2.66$, $p = 2.34 \times 10^{-6}$)****. Positive performance improvements occurred in 23 trials, while no negative performance differences were observed.*

MAPPO LEARNING



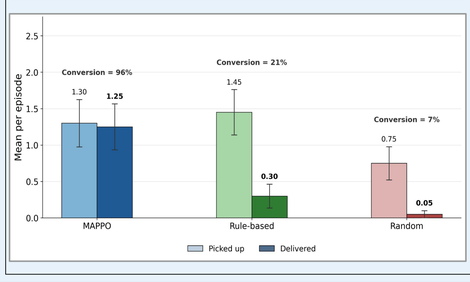
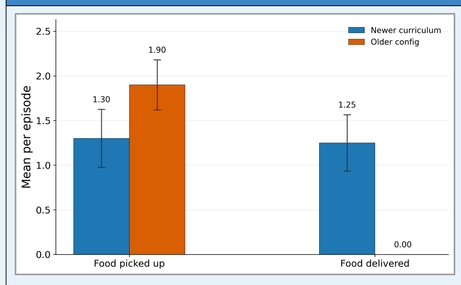


Figure 11. *Comparison of mean food delivery performance between MAPPO, rule-based, and random policies. A one-way ANOVA revealed a significant overall difference among policies* ***($F(2,57) = 9.345$, $p < 0.001$, $\eta^2 = 0.247$)****. Post-hoc Welch's t-tests showed that MAPPO significantly outperformed the rule-based policy* ***($p = 0.0123$)*** *and the random baseline* ***($p = 0.0012$)****.*

Figure 12. *Comparison of pickup and successful delivery rates across policies. MAPPO achieved the highest delivery conversion efficiency, indicating that learned swarm coordination was substantially more effective at converting discoveries into completed deliveries.*

CURRICULUM LEARNING



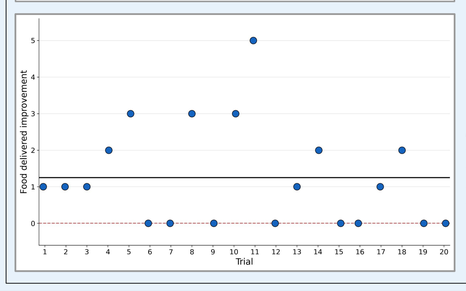
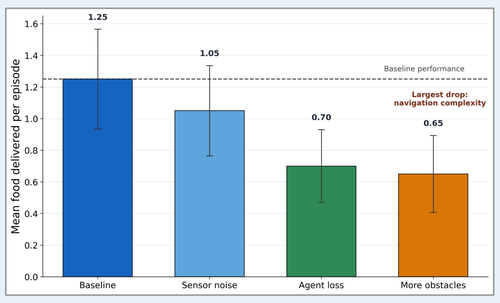
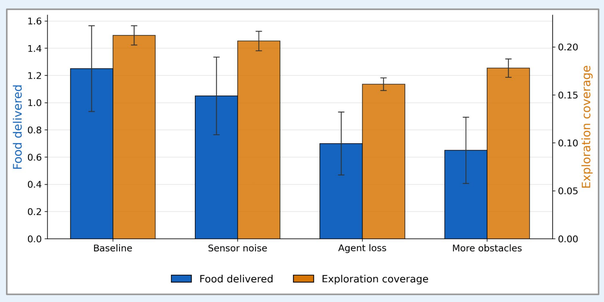


Figure 13. *Comparison between the new curriculum-trained configuration and the older training configuration. The older model could pick up food but failed to complete the full foraging loop.*

Figure 14. *Trial-by-trial delivery improvement of the new curriculum over the older configuration. A paired Wilcoxon signed-rank test showed that the newer curriculum significantly outperformed the older configuration* ***($n = 20$, mean $\Delta = 1.25$, $p = 0.00198$)****.*

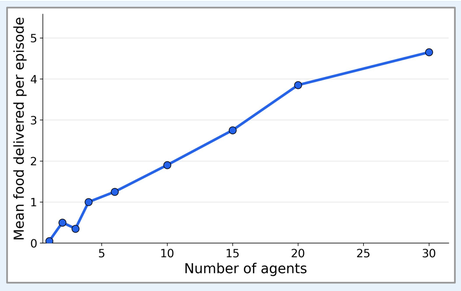
ROBUSTNESS

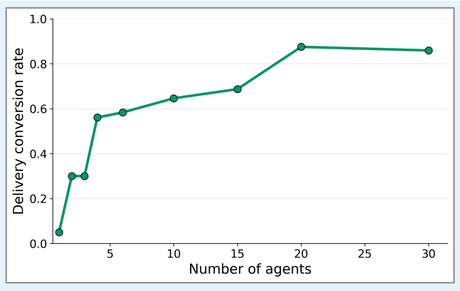




Figures 14-15. *Comparison of food delivery performance and exploration coverage under robustness stress conditions, including sensor noise, agent loss, and increased obstacle density. Food delivery performance decreased under more difficult conditions but did not show a statistically significant overall difference across robustness conditions* ***(one-way ANOVA: $F(3,76) = 1.122$, $p = 0.345$, $\eta^2 = 0.042$)****. In contrast, exploration coverage showed a significant overall difference across conditions* ***(one-way ANOVA: $F(3,76) = 6.719$, $p < 0.001$, $\eta^2 = 0.210$)****. Post-hoc Welch's t-tests revealed significant reductions in exploration coverage under agent loss* ***($p < 0.001$, $d = 1.332$)*** *and increased obstacle density* ***($p = 0.019$, $d = 0.773$)****, while sensor noise produced no significant effect. These results suggest that decentralized task completion remained partially resilient under environmental stress, whereas efficient spatial exploration was more sensitive to swarm disruption.*

SCALABILITY





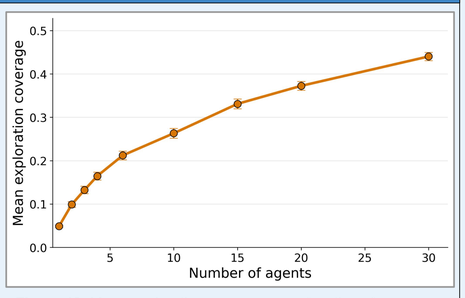


Figure 16. *Mean food delivered per episode as a function of swarm size. Food delivery performance increased with swarm size, suggesting improved decentralized collective foraging as additional agents contributed to exploration and retrieval. A significant positive monotonic relationship was observed between swarm size and food delivery performance* ***(Spearman's $\rho = 0.611$, $p < 0.001$)****.*

Figure 17. *Delivery conversion rate as a function of swarm size. Delivery conversion efficiency increased with swarm size and began to plateau at larger swarm sizes, suggesting the emergence of increasingly efficient collective coordination. A significant positive monotonic relationship was observed between swarm size and delivery conversion efficiency* ***(Spearman's $\rho = 0.496$, $p < 0.001$)****.*

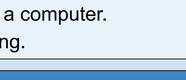
Figure 18. *Mean exploration coverage as a function of swarm size. Exploration coverage increased strongly with swarm size, indicating improved collective spatial exploration as the number of agents increased. An extremely strong positive monotonic relationship was observed between swarm size and exploration coverage* ***(Spearman's $\rho = 0.953$, $p < 0.001$)****.*

SIM-TO-REAL DEPLOYMENT

TO DEMONSTRATE REAL-WORLD VIABILITY, MODELS WERE DEPLOYED ONTO PHYSICAL ROBOTS.

ELECTRONIC ARCHITECTURE

Raspberry Pi 5
Executes the learned policy and **high-level** decision-making.



Camera
Provides visual data.

ESP32
Microcontroller which manages sensors, motor control, and **real-time hardware interaction**.

Differential Drive System
Two N20 (200 rpm) motors control the **locomotion** of the robot.

DRV8833 Motor Driver

N20 Motors W/ Encoders

2D LiDAR System
Front-facing 180° distance sensing, implemented using a VL53L0X time-of-flight sensor mounted on an SG90 servo for scanning.

VL53L0X ToF Distance Sensor

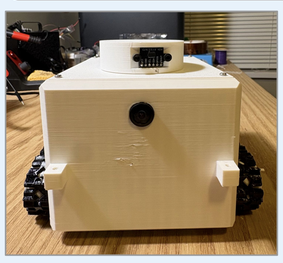
SG90 Servo Motor

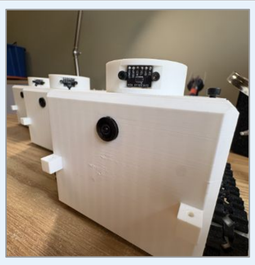
Encoders
Two encoders on the motors provide real-time feedback on speed, position, and angle.

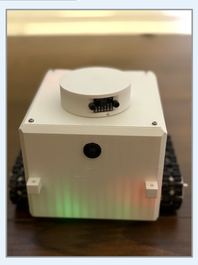
Power
The robot runs on 7.4v **standard RC car batteries**. Motors run directly on 7.4V DC, while a step down buck converter lowers the voltage to 5.2V DC for the compute and sensors.

7.4V LiPo RC Car Battery

Step Down Buck Converter (7.4V -> 5.2V)







Figures 19-21. *Photographs of the physical robots.*

SOFTWARE

A custom software layer was deployed onto a Raspberry Pi 5 on the robot, handling high-level control:

- **Control library** abstracts the robot's actions, allowing for convenient retrieval of sensor information, movement control, and pheromone information.
- **Python runtime** feeds inputs through the learned policy and continuously issues real-time commands.
- **Bluetooth library** connects the robot to a computer.
- **Camera library** enables visual processing.

FIRMWARE

A developed collection of **C++ modules** was deployed onto an ESP32, handling low-level control such as:

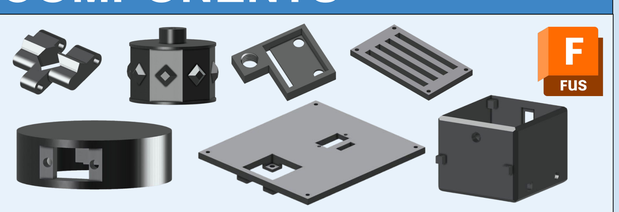
- **Calibration**
- **Gear motor actuation**
- **Executing action commands received**
- **Servo motor actuation**
- **Sensor data acquisition** (ToF sensor)

An ESP32 and Raspberry Pi communicate via **JSON protocol over UART**, connecting high-level decision-making to low-level control.

STRUCTURAL COMPONENTS

Custom robot components were created using computer-aided design (CAD) in **Autodesk Fusion.**

- Design focused on compactness, stability, and modular integration of all subsystems.
- Fused Deposition Modeling (**3D printing**) was used to manufacture the components in TPU and PLA.



DIGITAL ENVIRONMENTAL MEMORY

To implement stigmergic coordination in the physical system, a virtual pheromone field is maintained by Mission Control.

Virtual pheromone:

- Provides a digital environmental memory for stigmergy
- Stored global pheromone field maintaining deposition, evaporation, and dispersion

Communicates with robots via TCP (internet) or Bluetooth

Enables the sharing of:

- Position and heading from the robot
- Pheromones from Mission Control.
- **Agents perceive only local pheromone signals rather than the global state.**

Mission control, importantly, also provides a interactive visual dashboard for action control and demonstrations.

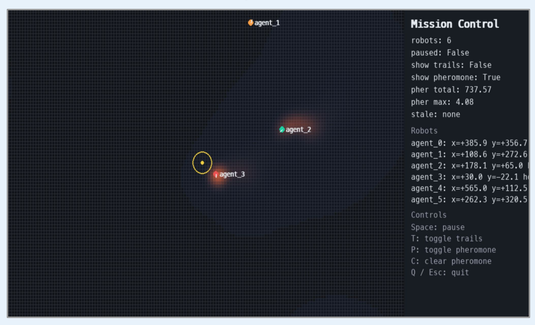


Figure 22. *Mission Control.*

IMPACT

SIGNIFICANCE

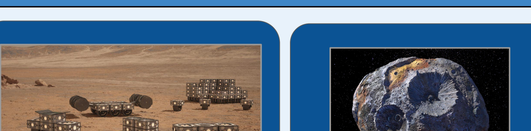
- Demonstrates **emergent collective intelligence** from decentralized local interactions.
- Introduces an **alternative to centralized AI architectures.**
- Demonstrates a method for designing **scalable multi-agent systems.**
- Shows robustness in **challenging environments.**
- Establishes a **framework** for decentralized robotic systems.

The research is completely accessible. *All experiments, simulators, software, and firmware are available online.* The research is made open source through GitHub.

Scan this QR Code to visit the GitHub Repository!

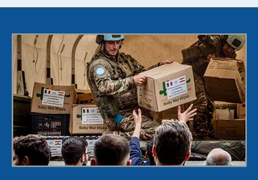
NOVELTY

- This research is the **first of its kind** to incorporate **emergent learned collective intelligence through stigmergy**. Most previous research has focused on rule-based mathematical models.
- Unlike much previous research, **no central overhead camera is needed** to track robots – each robot is self-sufficient navigationally and computationally – making the system **truly decentralized** and **far more scalable**.



Figures 23-25. *(From left to right) Kilobots, Colias Robots, and E-Puck robots. Most previous research explores rule-based models and requires information from a centralized source [21-23].*

APPLICATIONS



Aid Delivery
Dynamically discover and reinforce efficient routes in chaotic environments to deliver humanitarian aid.

Environmental Monitoring
Persistent large-scale sensing through low-cost agents, enabling vast coverage of ecosystems.

Extraplanetary Exploration
Replace single-point-of-failure rovers with scalable swarms that explore in parallel and coordinate without communication delays.

Asteroid Mining
Fully autonomous resource extraction, where robots form trails to resource locations.

The model developed is effective in high risk and uncertain environments.

CONCLUSIONS

This project demonstrates that decentralized collective intelligence can emerge through multi-agent reinforcement learning and stigmergic environmental communication. Inspired by carpenter ant foraging behavior, agents learned to coordinate exploration, resource retrieval, and cooperative task completion **without centralized control** or explicit inter-agent communication.

Experimental results showed that pheromone-based environmental memory improved coordination **efficiency, scalability**, and collective task **performance**, while maintaining **robustness** under sensor noise, agent loss, and obstacle-dense conditions. The project also established a sim-to-real framework for transferring learned decentralized policies from simulation to physical robotic systems.

Overall, the results suggest that **complex collective behavior** can emerge from local interactions among simple decentralized agents operating through shared **environmental information**.

FUTURE WORK

INVESTIGATING TRAINING BENEFIT

When agents are trained with pheromone, disabling pheromone at evaluation time **does not significantly reduce performance**. This may suggest that pheromones provide a training benefit.

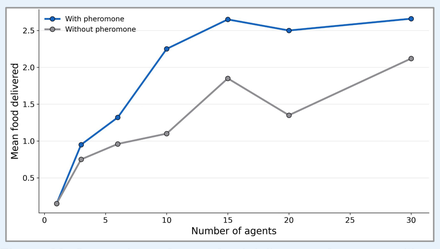


Figure 23. *Comparison of mean food delivery performance with and without pheromone-based stigmergic communication.*

THREE DIMENSIONAL PHEROMONES

Further work in this research will involve extending the swarm system from **2D to fully 3D environments**.

- While this research demonstrates emergent collective intelligence on a flat plane, many real-world applications occur in complex 3D spaces with height, depth, and multi-level terrain.
- Agents would need to learn a number of extra behaviors moving from a 2D to 3D environment.
- Notably, the pheromone system would evolve from a 2D grid into a dynamic volumetric field, allowing agents to communicate through **3D environmental memory**.