Experimental Validation of mEDEA on Pogobot Swarms

Ekaterina Antipova, Luiz Felipe Nery, Thomas Marchand, Enzo Pinho Fernandes

January 2026

Contents

1 Introduction **2**
1.1 Project context and objectives . . . 2
1.2 Brief overview of mEDEA . . . 2

2 System Overview **3**
2.1 Pogobot Robots . . . 3
2.2 Pogosim: Pogobot Robot Simulator . . . 4

3 Methodology **5**
3.1 mEDEA implementation details . . . 5
3.2 Genome representation and mutation . . . 6

4 Experiments and Results **7**
4.1 Overview . . . 7
4.2 Circle environment - baseline . . . 7
4.3 Circle environment with gradient light . . . 10
4.4 Circle environment with solid light . . . 13
4.5 Circle environment with orbiting gradient light . . . 15
4.6 Circular environments with obstacles . . . 17
4.7 Behavior under constraints . . . 18

5 Conclusion **20**
5.1 Interpretation of results . . . 20
5.2 Future work . . . 20

6 Appendix **21**

1 Introduction

1.1 Project context and objectives

Swarm robotics investigates how large numbers of simple robots can collectively solve tasks without centralized control. This project is situated in that context, using the Pogobot project, to study decentralized learning in a realistic swarm setting.

The main objective of our project is to implement the mEDEA algorithm, described in the reference paper [1], and evaluate its behavior on Pogobots. We aim to reproduce the algorithm's core dynamics, run controlled experiments across different environments and seeds, and extract some metrics to assess the swarm's adaptation. Our implementation works in both simulation and on real robots, but the experiments will be performed only in simulation.

Additionally, this work will focus on testing the algorithm in different environments and assessing how it behaves in such scenarios. For instance, an interesting scenario to explore is exposing the robots to a light source. We suppose this light can serve as a reference point for the Pogobots to meet and share their genomes.

1.2 Brief overview of mEDEA

Sometimes, a population of autonomous agents can face dynamic, unpredictable environments and/or unknown, especially to the human operator. This kind of problem makes it hard (or impossible) to explicitly make a fitness function. That's why we need a distributed online optimization algorithm targeting agent self-adaptation in the long term. It's important for the population to be able to evolve and optimize its behavior under an implicit environmental pressure.

mEDEA *(Minimal Environment-driven Distributed Evolutionary Adaptation)* takes into account selection pressure from the environment.

In mEDEA, each robot carries a genome that encodes its behavior. During a generation, robots broadcast their genome to nearby peers while interacting with the environment. At the end of the generation, each robot selects a random genome from its local list of received genomes, mutates it, and continues with the new genome. If no genome has been received, the robot becomes inactive until it later receives one. This simple rule creates an implicit selection process! Genomes that lead to better survival and encounter rates tend to spread, while poorly adapted agents disappear from the active population.

The algorithm is fully decentralized, without an explicit fitness function or global ranking. Instead, selection emerges from local interactions and environmental constraints. This makes mEDEA suitable for large robot swarms operating in unpredictable environments, where adaptation must happen online and continuously.

[1] N. Bredeche and J.-M. Montanier, "Environment-driven Embodied Evolution in a Population of Autonomous Agents," in `Parallel Problem Solving from Nature`, Krakow, Poland, Sep. 2010, pp. 290–299. `https://inria.hal.science/inria-00506771`

2 System Overview

2.1 Pogobot Robots

Pogobots [2] are small mobile robots specifically designed for the study of collective behaviors and social learning in robotic swarms. With a diameter of 6 cm and a modular architecture, each unit is composed of two main modules : a head and a belly.

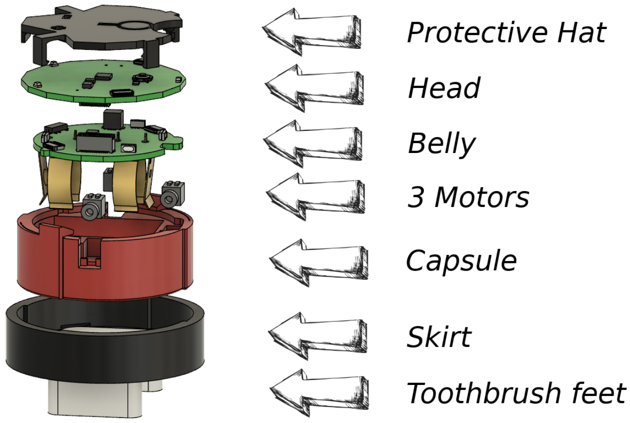

Figure 1: Simplified architecture of a Pogobot

The head integrates an iCE40UP5K FPGA running a soft-core processor, which ensures real-time management of sensors and communication. An inertial measurement unit (IMU) provides information about the robot's orientation and experienced accelerations, while four high-speed infrared transceivers enable inter-robot communication. In addition, the head includes three photoreceptors (rear, front-left, and front-right) to measure ambient light levels. Finally, a front-facing LED provides a visual indication of the robot's internal state.

The belly, attached beneath the head, houses the actuation and power components. It contains several side LEDs, two motors controlling the power delivered to the left, middle, and right wheels (or toothbrush feet) respectively, as well as the battery.

It should be noted that the Pogobot project is open hardware [3]. The 3D models and KiCad design files are publicly available on GitHub and can be freely used, modified, contributed to, or forked.

The Pogobot software is developed using an open-source SDK [4] written in C. The SDK is available online and is accompanied by API documentation that enables users to learn how to control the various functionalities of the Pogobots, along with several example programs.

[2]Pogobot Website: `https://pogobot.github.io/`
[3]Pogobot project repository: `https://github.com/nekonaute/pogobot`
[4]Pogobot SDK repository: `https://github.com/nekonaute/pogobot-sdk`

2.2 Pogosim: Pogobot Robot Simulator

Pogosim [5] is a simulator for Pogobot robots that is currently under development by Léo Cazenille. Its goal is to reproduce the Pogobot API used on the real robots, allowing the same code to run both in simulation and in real-world experiments.

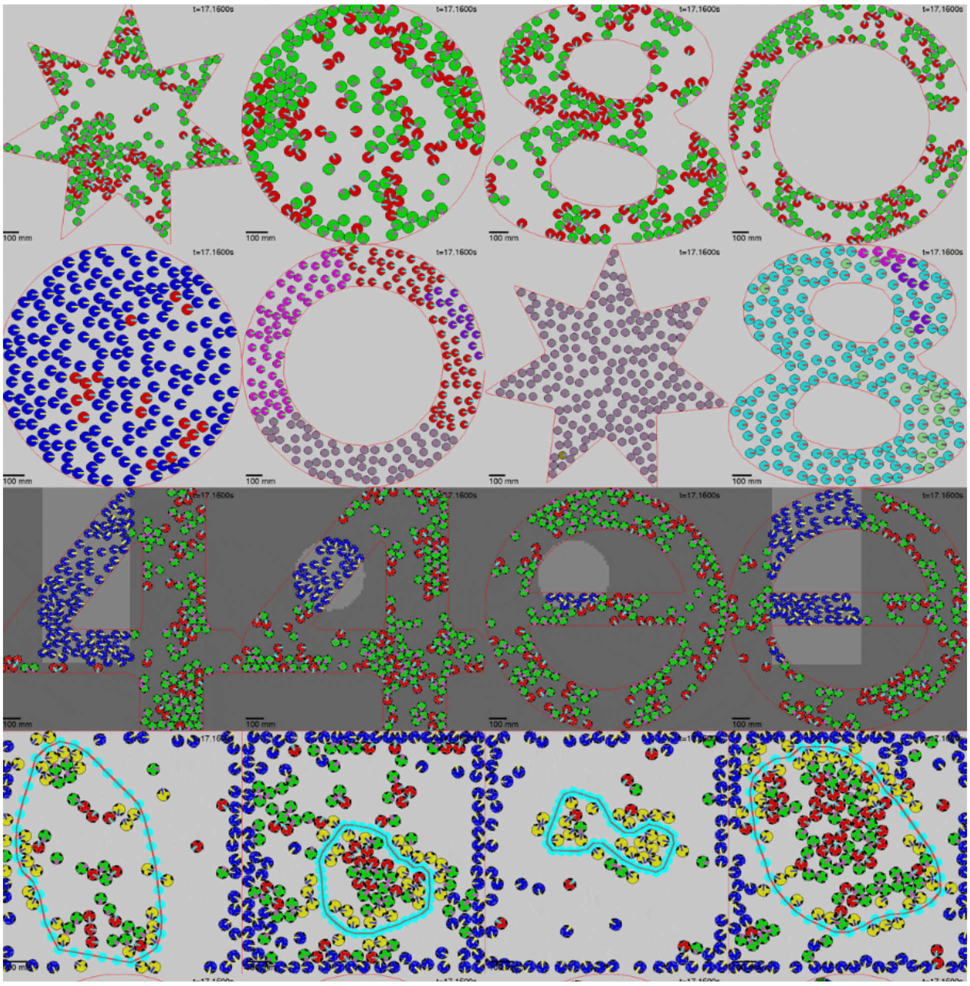

Figure 2: Examples of Pogosim simulations

The simulator allows us to create a wide variety of interesting arenas, both in terms of shape and content, including wall, obstacles, membranes, light sources, light ray casting... In addition, several properties can be modified, such as the physical parameters of the Pogobots, their communication range, the intensity and radius of light sources... Of course, many types of information can be gathered at each step, such as the robots' positions, angles, or even custom metrics. As a result, Pogosim provides a cost-effective and flexible entry point for testing the algorithm under a wide range of conditions.

For more information, feel free to visit the Pogosim website. [6]

[5]Pogosim repository: `https://github.com/Adacoma/pogosim`

[6]Pogosim website `https://adacoma.github.io/pogosim/`

3 Methodology

3.1 mEDEA implementation details

Our implementation follows the canonical mEDEA cycle on Pogobots with fully decentralized selection and reproduction. Indeed, each robot maintains an active genome controlling its behavior, a local list of genomes received from neighbors, and a generation timer. At startup, each robot initializes its active genome with random weights and enters the `active state`.

At each time step, the robot reads its sensors (three photosensors, 6-axis IMU, and battery level), normalizes them, and feeds them to a small MLP controller. The network outputs two commands, which are converted to speeds and directions, and then applied to the left and right motors. Robots broadcast their current genome at regular intervals (here, 10 steps) using omnidirectional IR messages.

A generation lasts a fixed number of steps (400 in our setup). At the end of each generation, the robot clears its current genome and applies the mEDEA selection rule.

- If at least one genome has been received during the generation, one genome is selected uniformly at random from the local list, mutated, and becomes the new active genome.
- If no genome has been received, the robot becomes inactive and doesn't move until it receives a genome in a future generation.

After selection, the genome list is cleared, and the next generation starts. This method implements implicit selection pressure. Indeed, genomes that promote survival, motio,n and encounters are more likely to be transmitted multiple times in a single generation, and so, persist in the population.

The local genome list is capped at 50 to respect the memory constraints of the physical Pogobots. Mutation is Gaussian with a fixed standard deviation, here $\sigma = 0.02$. The controller runs at 60 Hz. Only during a simulation, we record per-generation logs for later analysis (position, angle, genomes weights...).

3.2 Genome representation and mutation

Because the Pogobot has different characteristics from the robot described in the paper, we adapted the neural network.

Each genome is a vector of real-valued weights defining a two-layer neural controller. The input layer consists of 10 sensor values: three photosensors, a 3-axis accelerometer, a 3-axis gyroscope, and a normalized battery level, and we added a bias term. These feed 5 hidden neurons with sigmoid activation. The hidden layer (and a bias) connects to two output neurons, which are mapped to left and right motor commands, controlling their speed and direction.

In total, the genome contains 67 weights :

$$((\text{Input\_size} + 1) \times \text{Hidden\_Neurons} + (\text{Hidden\_neurons} + 1) \times \text{Output\_size} = 67.$$

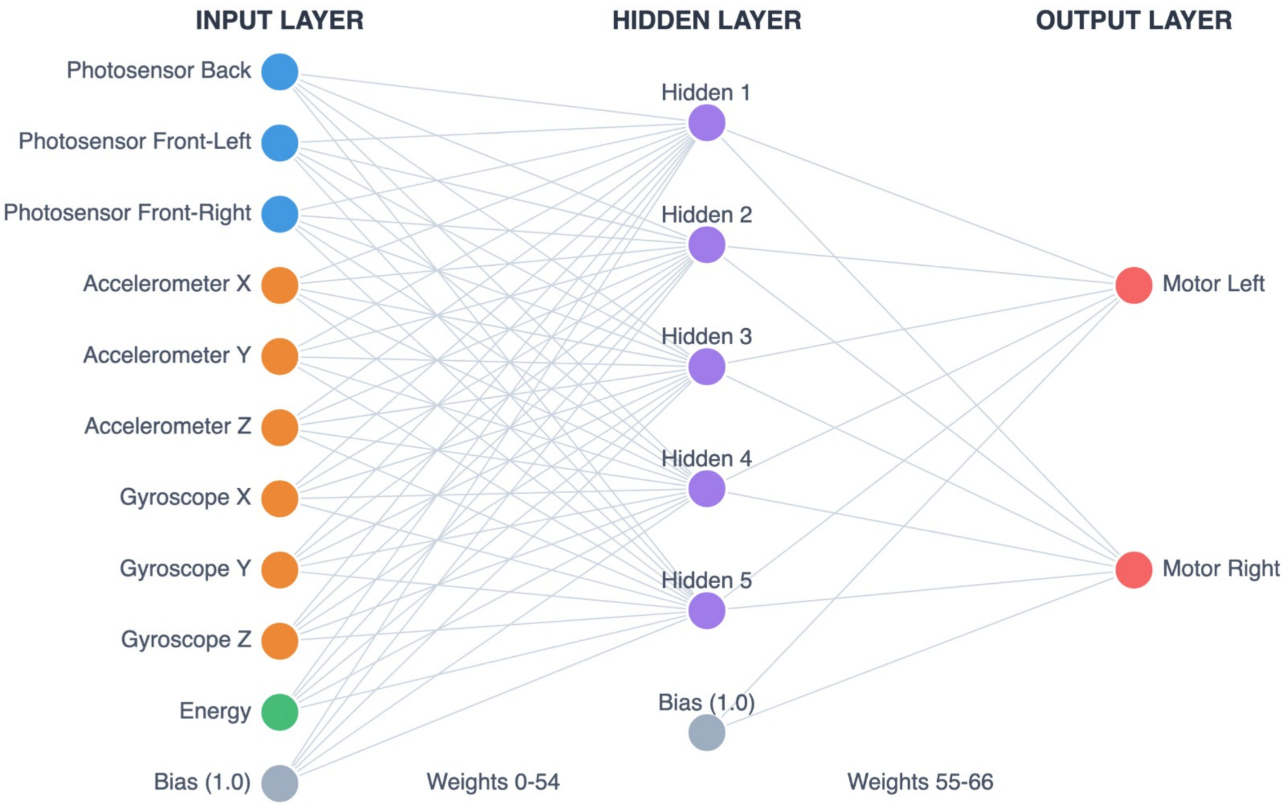

Figure 3: Neural controller structure used by each Pogobot.

4 Experiments and Results

4.1 Overview

Multiple experiments were conducted in order to assess mEDEA behavior and test it under different setups. All of them involved creating different environments composed of lights, solid objects, or complex, dynamic, moving objects. All environments were built upon a circular arena available at the Pogosim repository [5].

Initially, to establish a baseline for Pogobots' behavior with mEDEA, an experiment was conducted in a simple environment. The environment is the `disk` (circle) with no objects or lights, just the 30 robots. This served as a baseline for understanding the algorithm's behavior when running with Pogosim. Although this experiment was simple, it was essential to establish the role of certain environmental parameters. Due to the large amount of such parameters (robot size, robot max velocity, arena size, etc), we decided to fix them for our experiments, and we explore, depending on the experiment, whether it's necessary to tune the number of robots and their communication range. Next, we will present each experiment and its respective environment.

4.2 Circle environment - baseline

In this baseline experiment, we observed that depending on initialization, the number of robots, and their communication range, we could end up in three main states: (i) robots turning around at the same place in groups. (ii) groups of robots close to the walls and (iii) extinction. Those first two states are illustrated in Figure 4, and their occurrence is because they maximize the possibility of passing their genomes. And the last one occurred when the environment's parameters or initialization were unfavorable for the swarm.

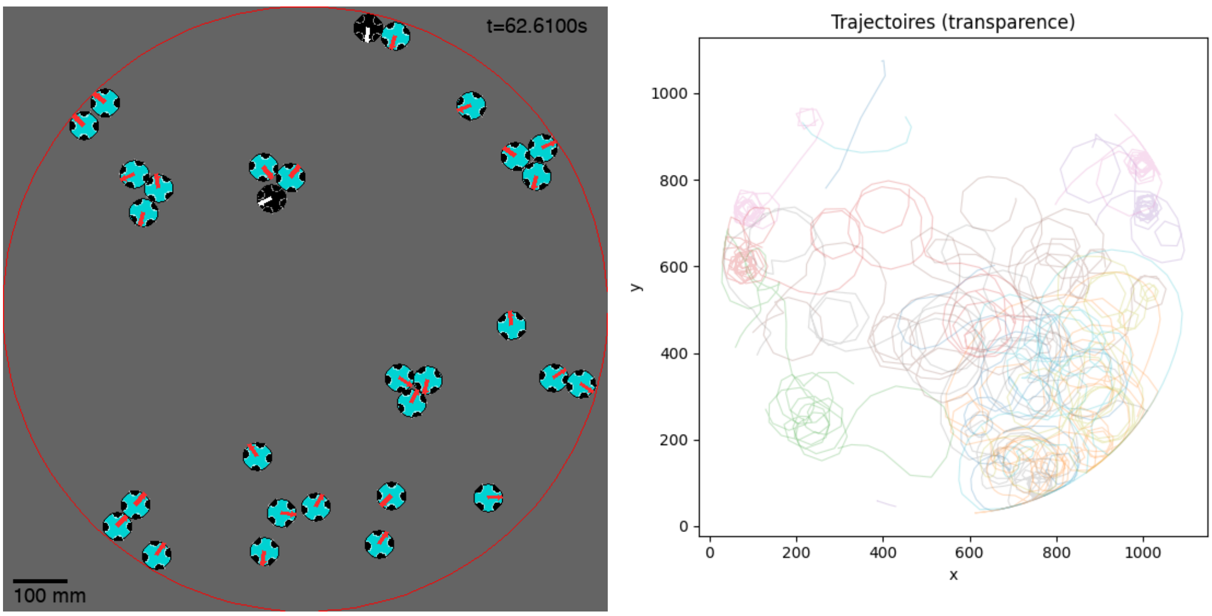

Figure 4: Left: Simple circular arena environment. In colors, active pogobots, in black and white, inactive pogobots. Right: Pogobots trajectories over 100 seconds.

Additionally, we observed some interesting group behaviors. Sometimes the robots establish formations with other robots. coupling in pairs or trios for many interactions. Figure 5 exhibits those behaviors as well as an example of robots being attracted by the walls.

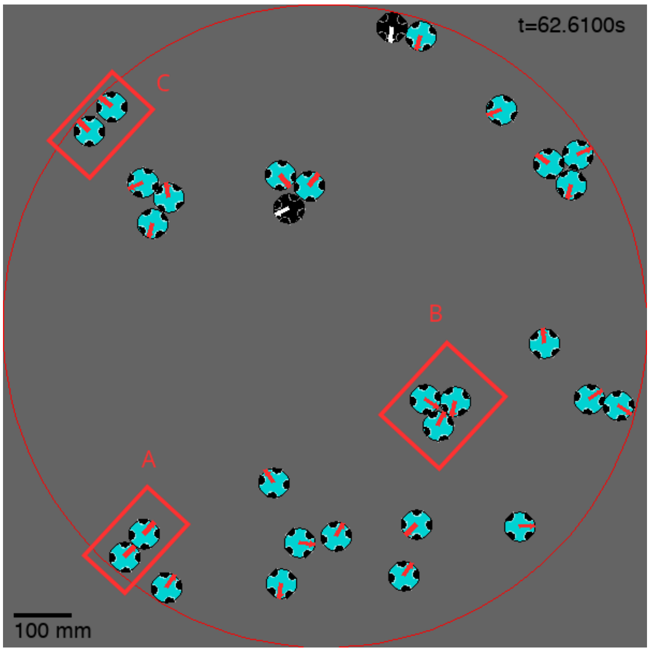

Figure 5: A: couple formation. B: Trio formation. C: Robots using the wall as a reference to meet.

In Figure 9 (right-hand side), we observe the effects of the previously mentioned behaviors. First, we observe many closely spaced peaks, indicating spots where the robots orbited in groups to increase the likelihood of spreading their genomes. There are also two rings representing a region near the walls that the robots are attracted to. The first one, the outer ring at the limit of the heatmap, contains the largest number of samples and is consistent across all boards of the heatmap, indicating that the pogobots rely heavily on the wall as a reference point to meet and pass their genomes, as illustrated by formation C in Figure 5. However, upon closer analysis, we observe an inner circle generated by the pogobots in the A or B formation, as illustrated in Figure 5, showing that they don't just hug the edges, but can also cluster around them.

In Figure 6, we observe a typical execution of this environment with 30 robots. In the left image, we can see that in the first half of generations, the distance of the active robots from the center is scattered overall, but then in the second half, they are all clustered away from the center, showing the behaviour where they use the wall as a reference to meet and exchange genomes. We can see from the right image that most robots that are not at the edge of the environment are inactive, indicating they were unable to exchange genomes away from the edges.

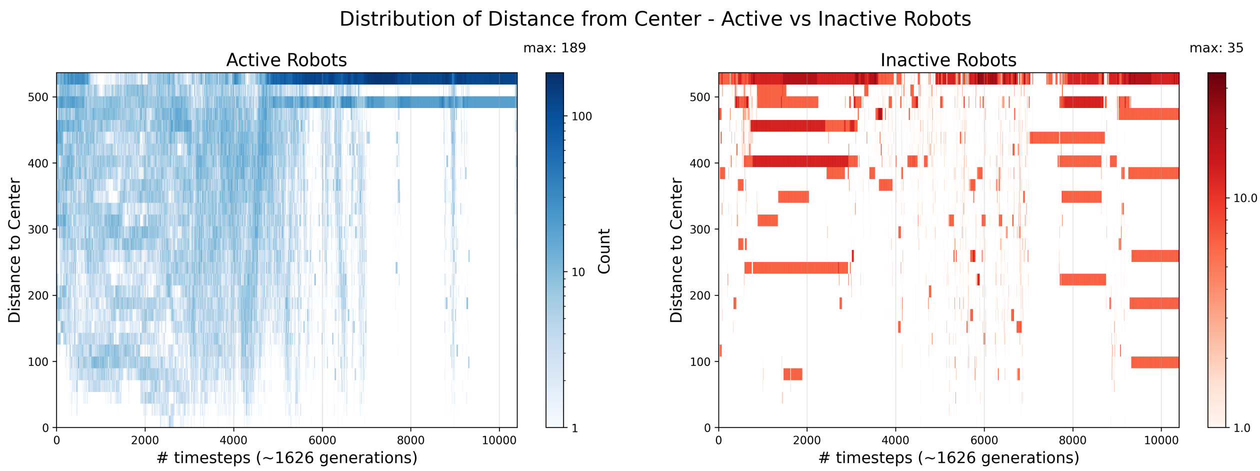

Figure 6: Distribution of the Distance from the Environment Center for Active (left) and Inactive (right) robots over time.

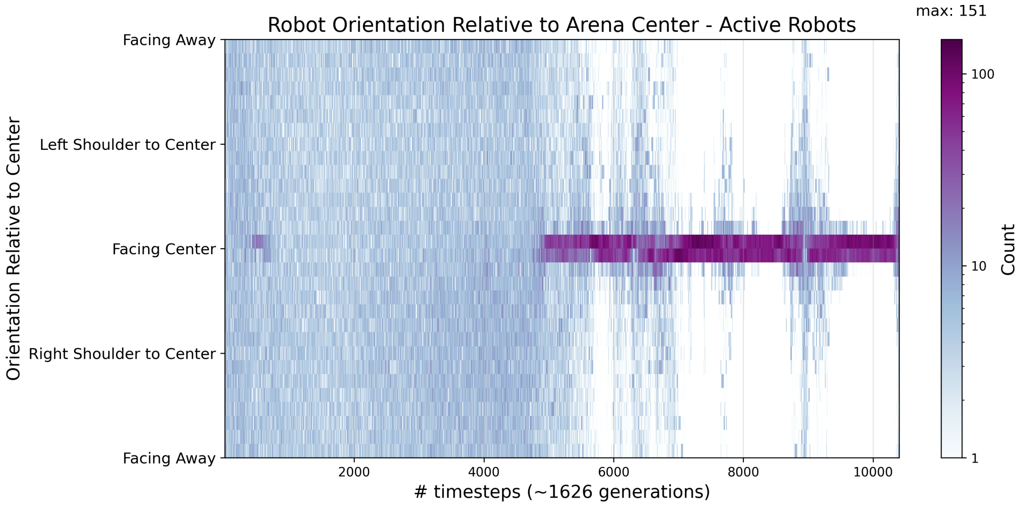

Figure 7: Distribution of the Orientation relative to the Environment Center for all robots over time.

In Figure 7, we see how initially their orientation seems to be all random, until they started meeting at the edges at the halfway point in time, where all the robots would be facing the center.

4.3 Circle environment with gradient light

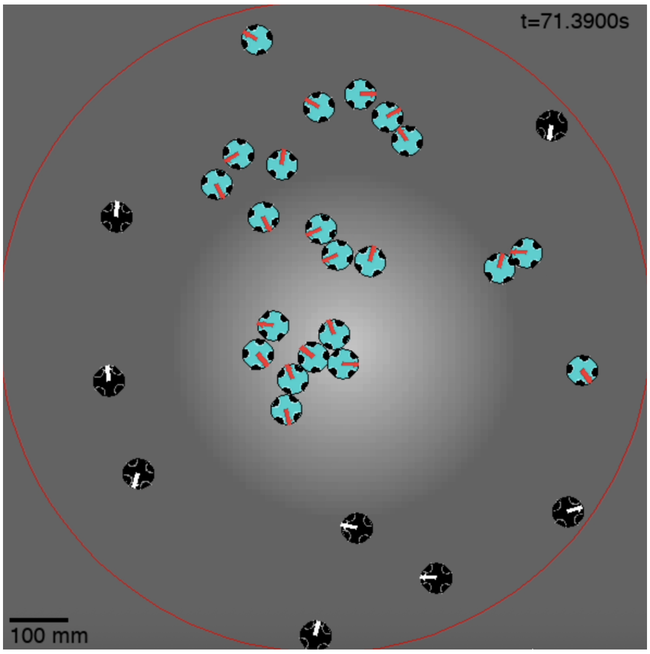

Figure 8: Circular arena with static gradient light source.

This next experiment was done to investigate the effect of a light source in the arena. For this, a gradient light source was positioned at (600, 600) in the circular arena as shown in Figure 8. Our hypothesis is that the light would create another reference point for the pogobots to meet and pass their genomes. Figure 9 confirms this hypothesis. Compared with the baseline, we observe a large concentration of robots at the center of the gradient light, confirming that, over iterations, the Pogobots were attracted to the center of the light and used it as a reference point to meet other robots. We can still observe inner and outer rings, indicating that robots still use the walls as a reference, as in the baseline. However, comparing the two graphs, we observe that the inner ring is less important, which suggests that the pogobots rely less on duo and trio formations for matching in this setting.

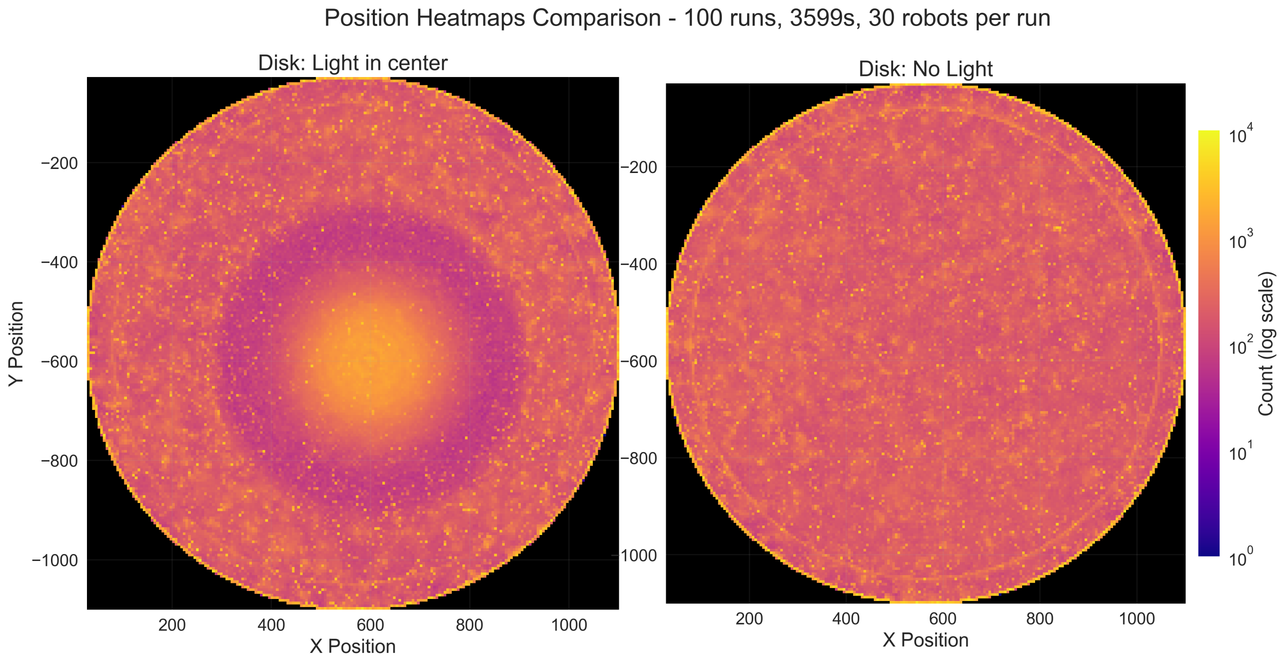

Figure 9: Left side: Circle arena with circle gradient light positioned at (600, 600). Right side: Simple circle arena.

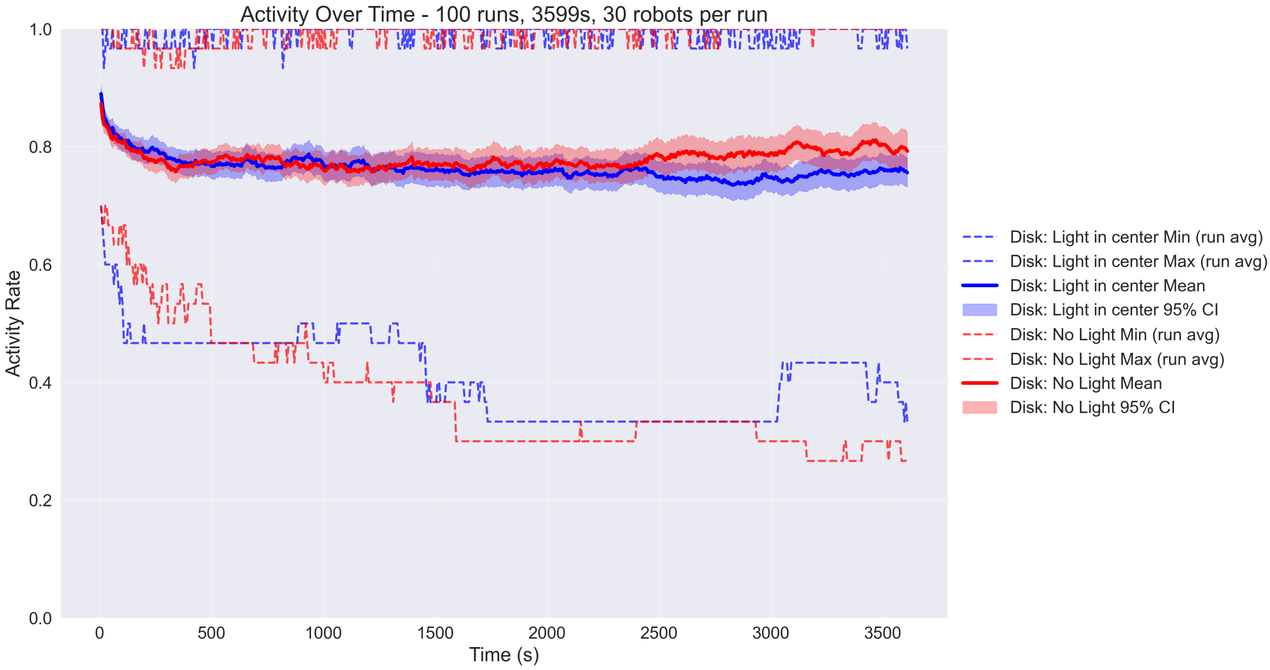

Figure 10: The activity rate of robots over 100 runs over 3600 simulation seconds, comparing the circle arena with a gradient light (Blue) and with no light (Red).

Here in Figure 10 we plot the average activity of robots in the two environments. We observe that the environment with no light in the center has a slightly higher active number of robots, even though its minimum is slightly lower. Both environments seem to have active robots at around 30% in the worst-case scenario. This makes it hard to determine which environment is more survivable for the robots, but they are good at staying active as a group in both.

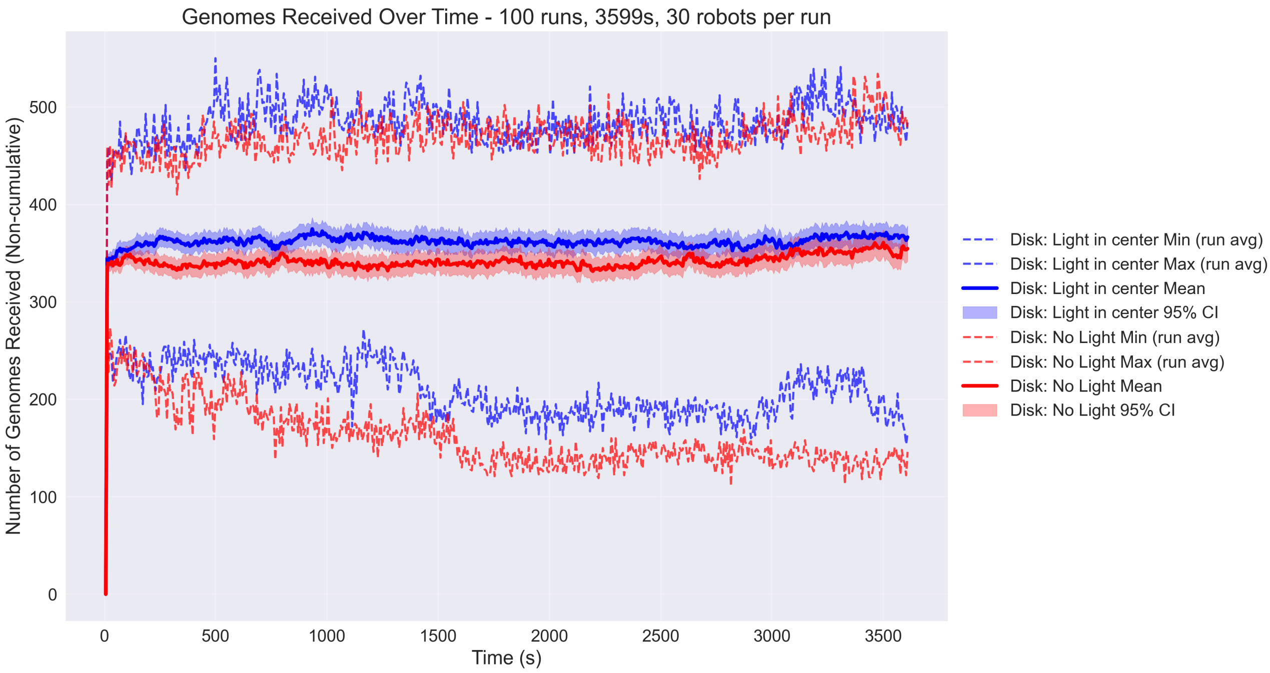

Figure 11: The number of genomes received on average over 100 runs over 3600 simulation seconds, comparing the circle arena with a gradient light (Blue) and with no light (Red).

In Figure 11, we compare the number of genomes received in the two environments. We observe that the environment with soft light receives a slightly higher average number of genomes.

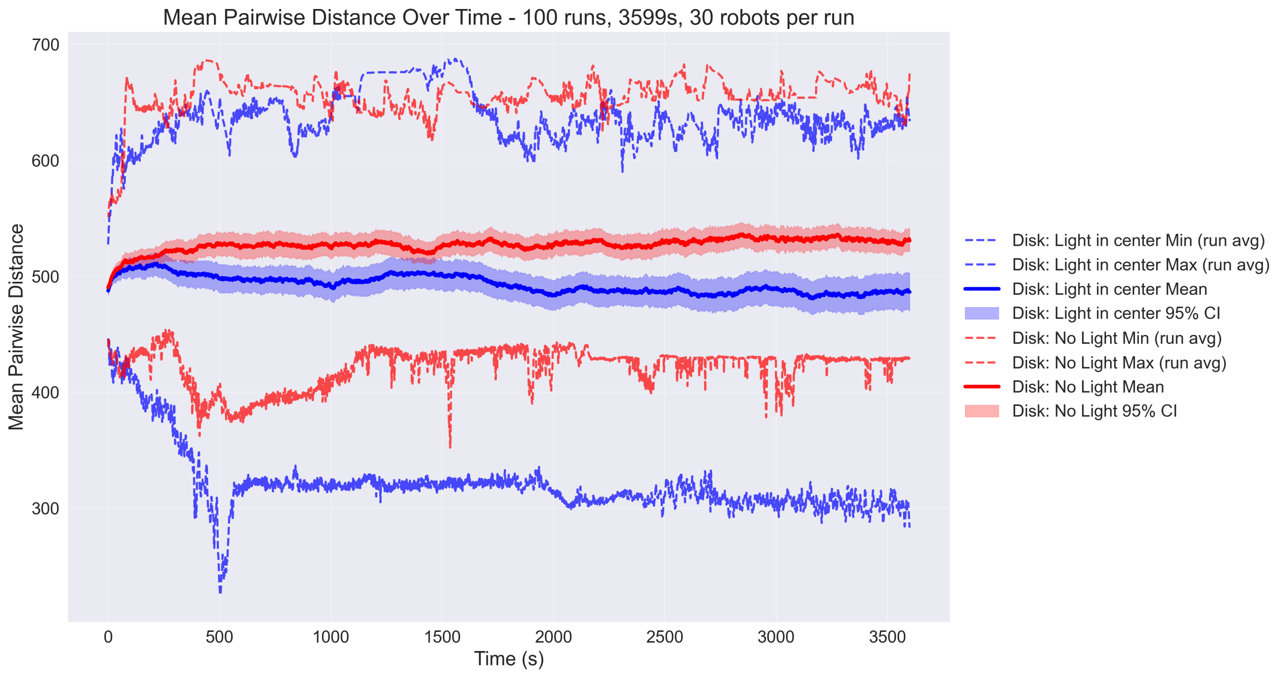

Figure 12: The pairwise distance of robots on average, comparing the circle arena with a gradient light (Blue) and with no light (Red).

In Figure 12, we observe that the average distance between two robots is lower in the environment with the soft light, showing that they cluster together more on average. We also see that the run with the lowest average in the soft light environment had a mean pairwise distance much lower than that in the no light base environment.

Moreover, Figure 9 shows a big concentration of samples in the center of the light and almost no sample points in the colder regions of the light. This experiment raised the question of whether the light gradient plays such a great role in the robot's behavior.

4.4 Circle environment with solid light

In order to answer this question, we defined a new experiment based on the environment shown in Figure 13. Instead of using the gradient light, a solid uniform light was placed in the same position.

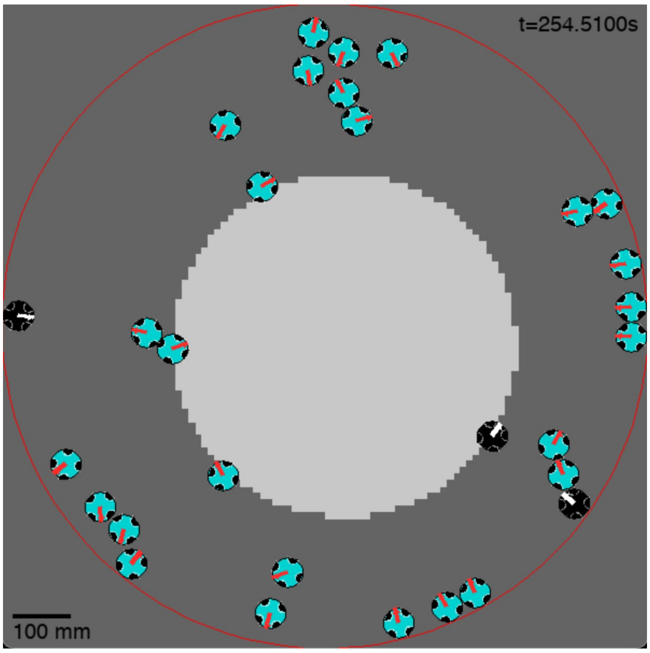

Figure 13: Solid uniform light placed at position (600,600).

Figure 14 compares both settings: A Soft light (gradient) and a Hard light (no gradient). We observe that the uniform light does not produce any gathering effects on the pogobots and the region defined by this light is not more popular among the pogobots than other regions, which suggests that the gradient of the light plays a big role in the pogobots, as it works as a reference direction to the center of the light, which is a common meeting point for the pogobots.

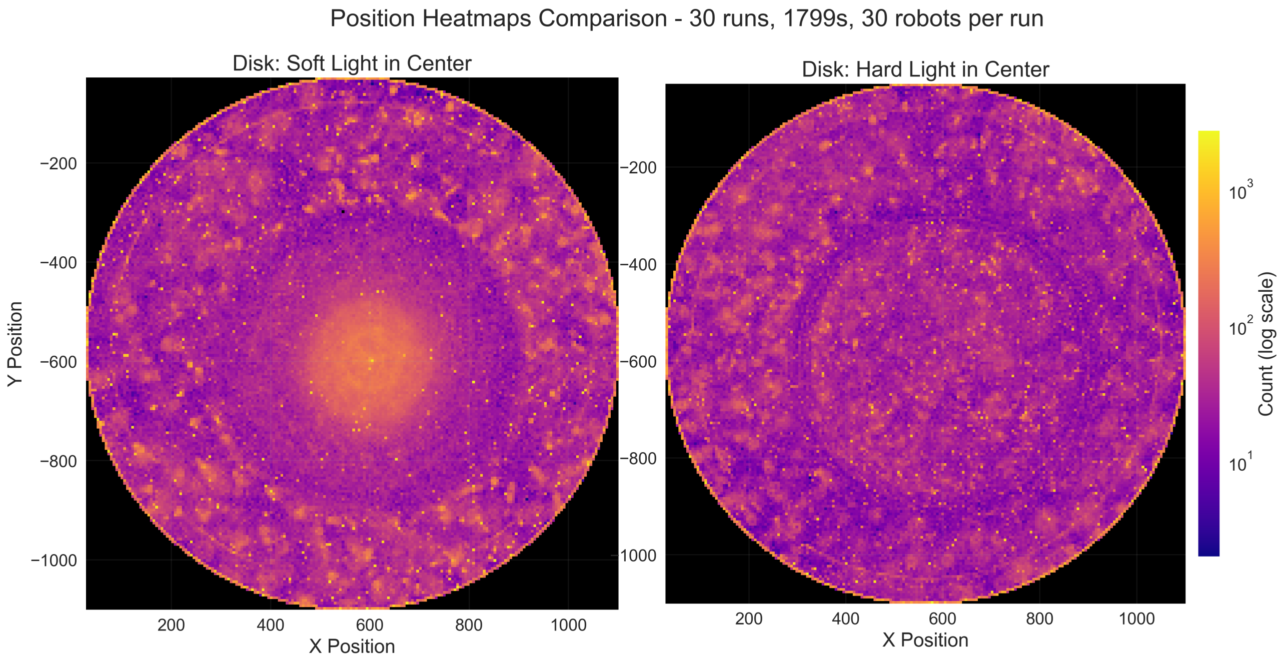

Figure 14: Left side: Circle arena with circle gradient light positioned at (600, 600). Right side: Circle arena with no gradient light positioned at (600, 600).

We now compare the base environment with no light to the environment with hard light to see whether the hard light still affects the pogobots.

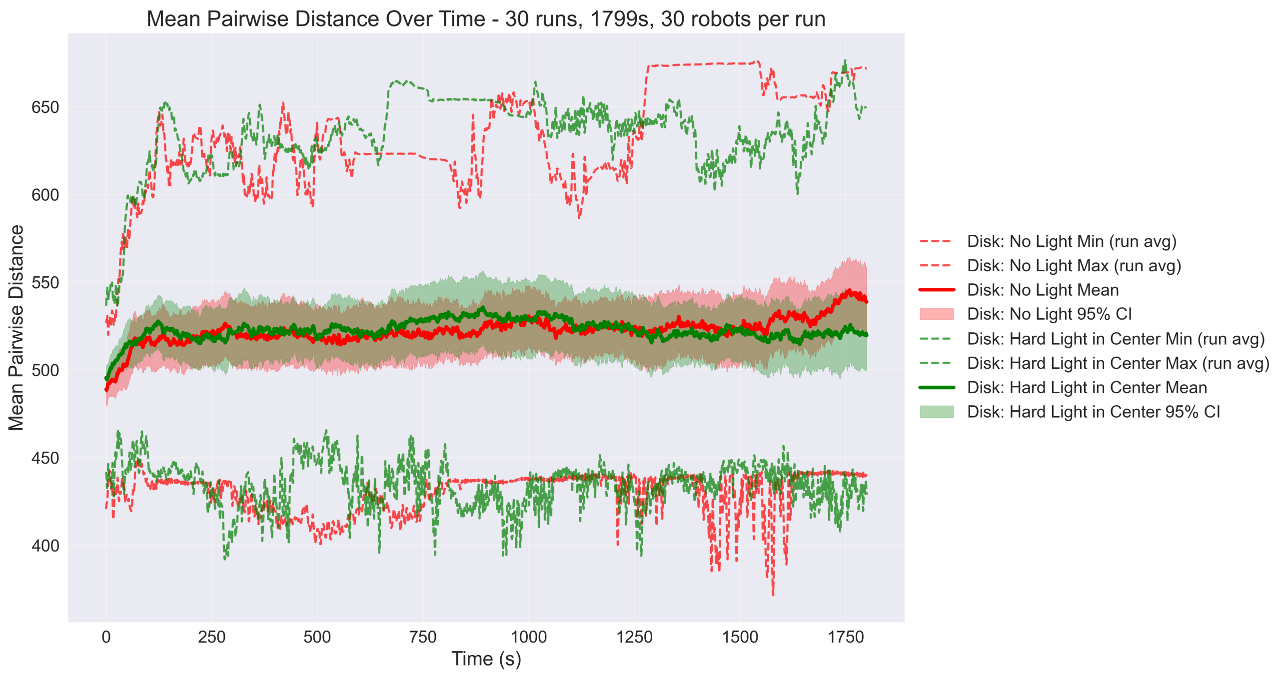

Figure 15: The pairwise distance of robots on average, comparing the circle arena with no light (Red) and with a hard light (Green).

In Figure 15, we observe that there is no statistically significant difference between the two environments for the first half-hour that we executed it for, showing that the hard light does not seem to cluster the robots together.

(a) Activity over Time

(b) Genomes received over time

Figure 16: Activity and Genomes received on average, comparing the circle arena with no light (Red) and with a hard light (Green).

We can see in Figure 16a and 16b that the other metrics do not show a clear difference between the two environments, confirming that the hard light does not seem to affect the robots in any significant way for these timesteps.

4.5 Circle environment with orbiting gradient light

A limitation of the gradient light environment is that the light is confined to a certain region. When Pogobots are trapped outside this region, they may be deactivated, and it can take a long time to reactivate. We then propose an experiment in which light moves in an orbital pattern around the circular environment, so that all positions are reached at a given time. In this experiment, we planned to test whether the robots could follow the light even when it was moving. To do so, given the limited number of objects available in the Pogosim simulation, it was necessary to modify the simulation code to add a new type of moving light. In this case, a gradient light that moves in an orbital pattern around a point with customized velocity. This was done in a fork of the simulation [7].

Figure 17 shows how this light works.

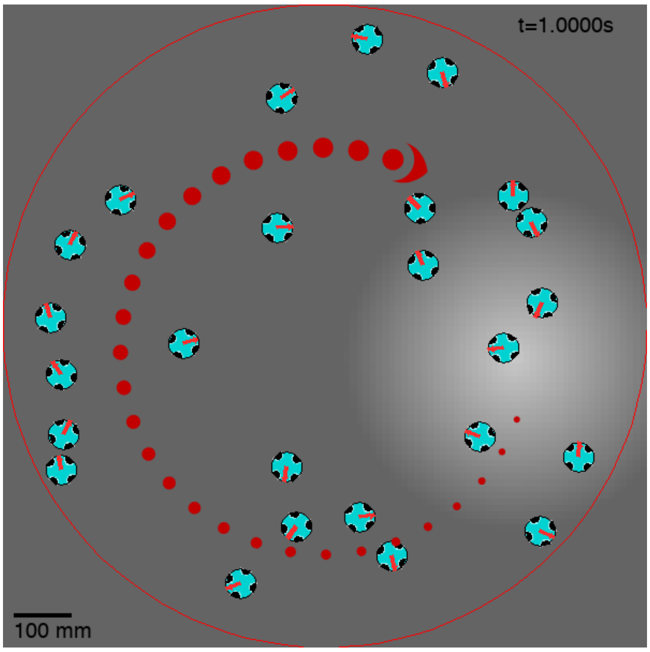

Figure 17: Orbitating light path.

[7]Pogosim fork repository: `https://github.com/luiznery/pogosim`

Figure 18 displays an interesting finding of this work. Even though we can still observe other formations, we clearly see that in the orbiting gradient light environment, the robots learn to follow the light. On the left-hand side of Figure 4, we observe the trajectories of the Pogobots for the last 100 seconds of the simulation. Even though we can still observe some star-shaped patterns formed by robots rotating in groups, we also see a clear donut-shaped pattern representing the other pogobots following the light. The heatmap on the right-hand side of this figure supports this hypothesis, clearly showing a donut-shaped concentration of samples along the orbital path.

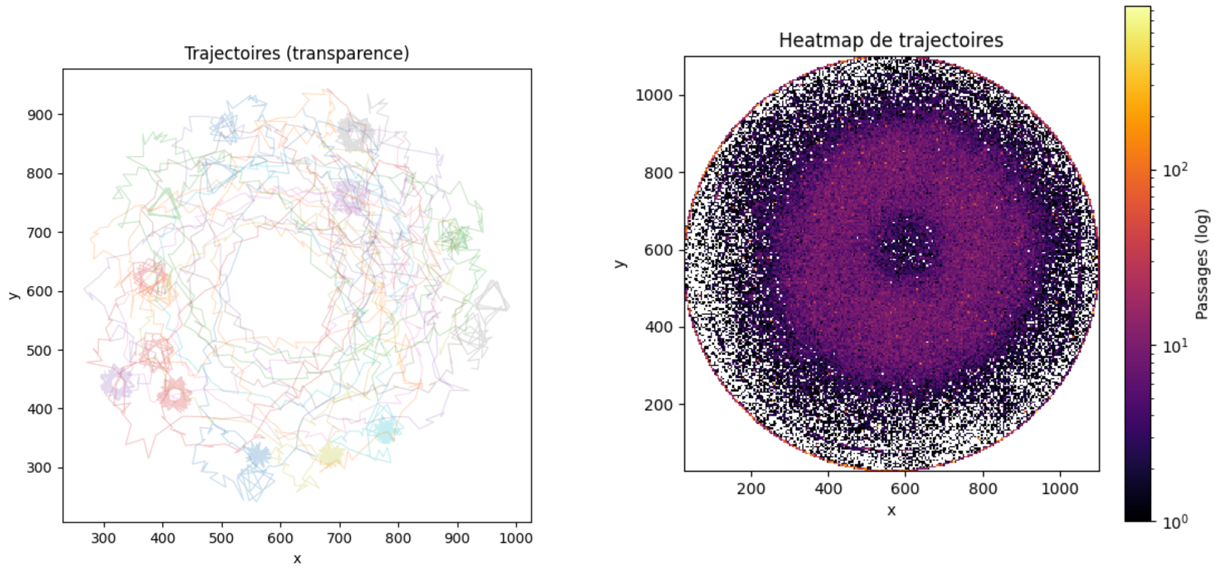

Figure 18: Left side: Pogobots' trajectories for the orbital gradient light environment. Right side: Heatmap of pogobots' positions over the simulation time for the orbital gradient light environment.

4.6 Circular environments with obstacles

We further explore the Pogobot behavior by further complicating the arenas - adding obstacles and additional light sources to track the evolution and differences in Pogobot activity (Figures 19 and 20).

We explore:

A) Circular arena with static gradient light.

B) Circular arena with a static central object and static gradient light.

C) Circular arena with a static central object and a rotating light beam.

D) Circular arena with several smaller fixed objects and static gradient light in one of the objects (imitating a maze-like structure).

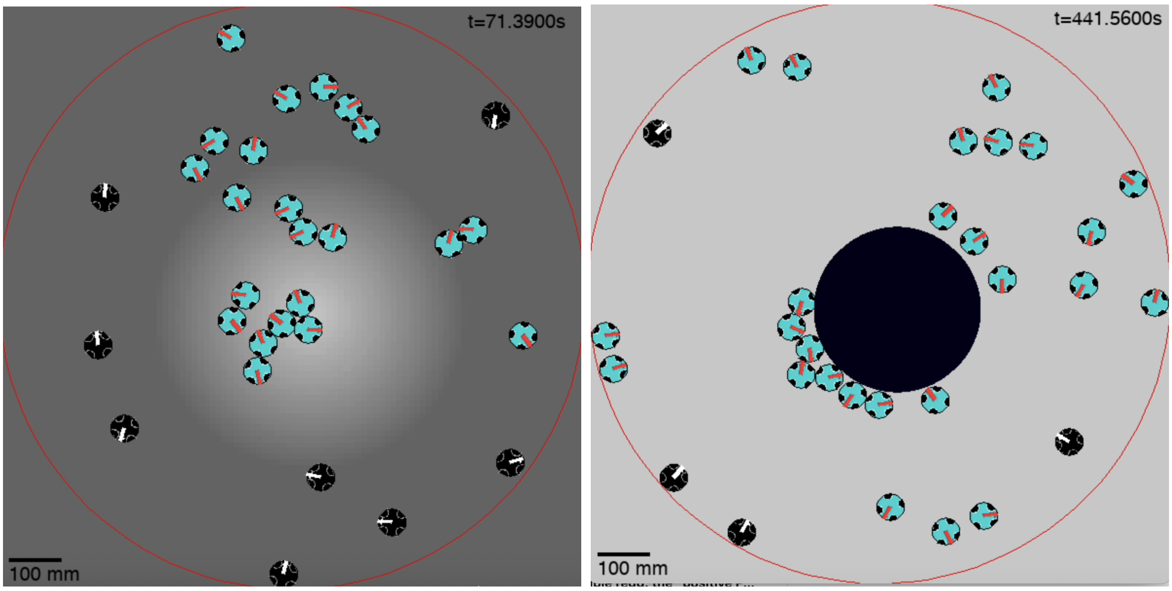

Figure 19: Arenas A and B

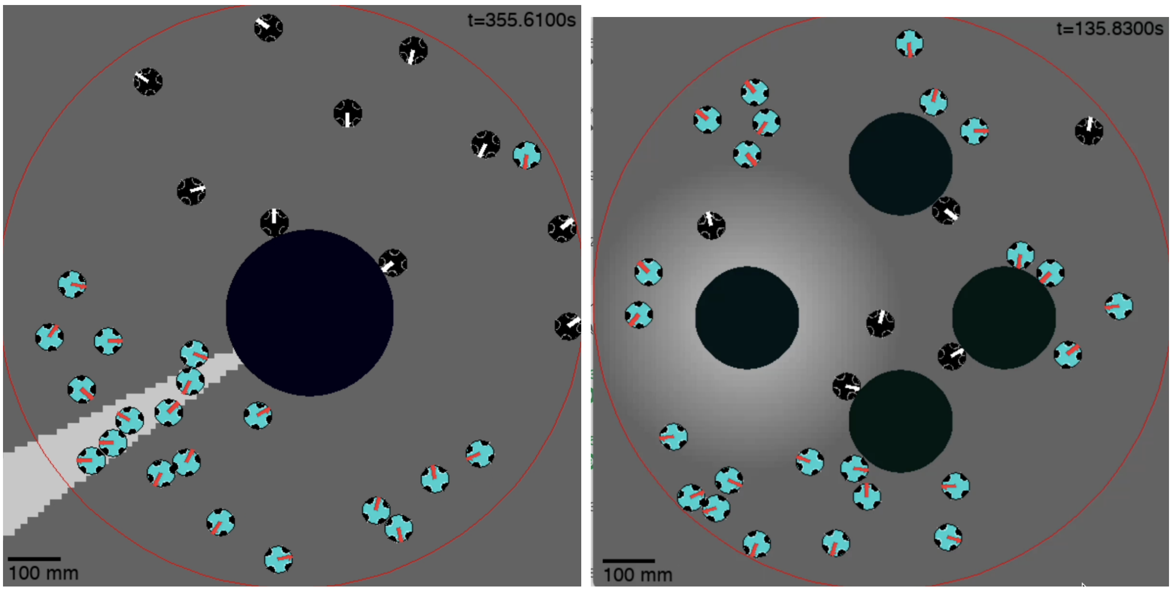

Figure 20: Arenas C and D

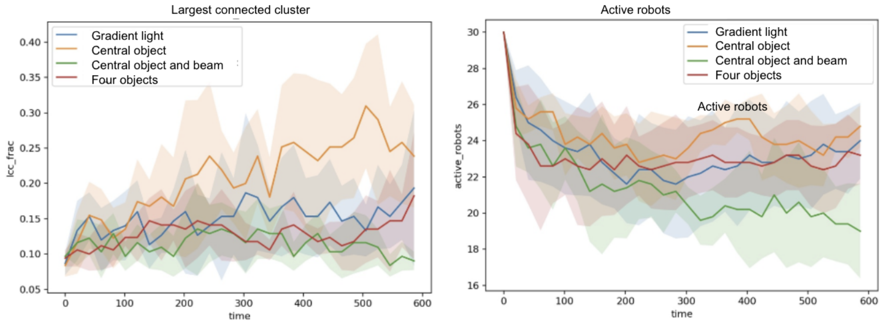

Figure 21: Average largest connected cluster per arena (left), share of active robots per arena (right)

4.7 Behavior under constraints

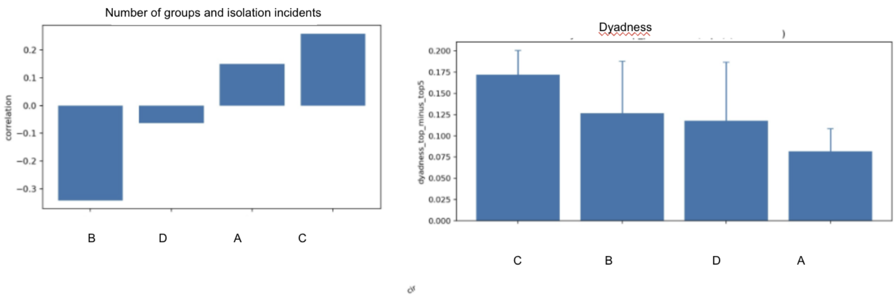

Figure 22: Correlation between the number of groups and isolation per second (left), Dyadness (right)

Throughout, we track the agent's location and patterns of interaction with other Pogobots in the arenas A to D. The defined Pogobot behavior would not allow for sophisticated cooperation (which implies the existence of an incentive to withhold cooperation). Here, robots effectively cooperate when the genome successfully moves, keeping the agents active.

In general, it appears that the introduction of a constraint affects robot activity negatively, as the original empty circular arena keeps the largest number of robots active.

While the count of active robots in the environments A-D drops from the start of a run, we note that the measurement is the highest for arena A and the lowest for arena D (Figure 21). Furthermore, a similar pattern is noted in the measurement of the largest connected cluster present. Arena A also promotes the largest share of clustered robots as well as the highest genome receiving rates, with D being on the lower end of both metrics. We also note that in the presence of static objects, robots tend to cluster around the object rather than anywhere else in the space, including the walls. This in itself produces stable cluster formation.

Moreover, we observe that the rotating light beam disrupts clustering in the Pogobots. As they follow the beam for several steps after it passes by, they appear incapable of forming coherent groups, but produce higher directional alignment than in any other arena. Gradient static light does not appear to have a significant influence on the robots' interaction with the obstacles in this setting.

As Figure 22 demonstrates, the relationship between death by social isolation (lack of genome reception) and the number of groups varies across the arenas. Intuitively, arenas C and D produce the most distraction and limitation - in them, robots cannot opt for the most optimal strategy of forming a coherent, stable cluster, thus they form smaller groups (dyads, trios) and avoid isolation. With less reason to separate, such as in arenas A and B, remaining split is suboptimal, as most robots naturally remain in each other's vicinity.

A similar idea can be suggested based on the right-hand graph in Figure 22: arena D produces the largest disruption, forcing robots to pair up throughout. Notably, the presence of static objects creates a medium level of dyaness, and the arena with the central light does not commonly produce such a behavior.

Finally, we suggest that forming a coherent group remains the best available strategy to achieve the highest levels of survival and genome spread. However, in the absence of such an option, robots form smaller groups, which, however, restricts genome flow and restricts the swarm's activity. Moreover, with the best strategy blocked, Pogobots reform into a less optimal strategy that stabilizes the swarm on a lower level of activity.

5 Conclusion

In this work, we observed the behavior of Pogobots with the mEDEA model in the Pogosim simulation. Our objective was to test mEDEA behavior under different environments. For that, we created several environment setups based on lighting, static objects, and dynamic objects. We observed that under those conditions, the Pogobots converge to group around a common point of interest, such as the walls of the environment, forming large groups or even following light gradients. Surprisingly, even a five-neuron neural network could describe more complex behaviors, such as following a light that exhibits a regular motion pattern. Another important observation is that the swarm system is highly sensitive to environmental parameters, which may need to be fine-tuned for slightly different environments.

5.1 Interpretation of results

We suggest that Pogobots' optimal strategy is to form a large, stable group. This provides for the highest share of active robots and genome reception. However, the introduction of obstacles and distractions complicates that option, forcing the robots to form smaller groups, which naturally limits their level of activity and genome spread.

5.2 Future work

A natural continuation of this project would be to test our implementation on real robots, using a variety of environments and experimental conditions. In particular, it would be interesting to introduce abrupt environmental changes during execution to observe how the robots adapt and whether the population can survive.

Furthermore, greater attention could be paid to robots' behavior when their best strategy (forming a large cluster) is unavailable. More attention could be paid to the persistence of smaller groups (dyads or trios) and whether certain genomes are beneficial in adverse rather than favorable scenarios. We also raise the question of whether sticking to one's chosen group or actively exploring is more beneficial in the face of adversity.

In addition, an important question concerns the impact of uncertainty in the robot's physical properties on performance. For example, Pogobots equipped with toothbrush feet exhibit motion that is highly sensitive to surface conditions, dust, and, in particular, the quality of the bristles. Identical motor power and direction settings may result in significantly different movements across two robots. Would the swarm naturally adapt to such variability? Do agents need to share similar physical priorities to ensure their survival?

Furthermore, it would be interesting to explore the robots' behaviours in the proposed environments across different NN sizes. And also to perform an explicability analysis to check which inputs and weights are important to make each decision, for example, to follow the orbiting light.

6 Appendix

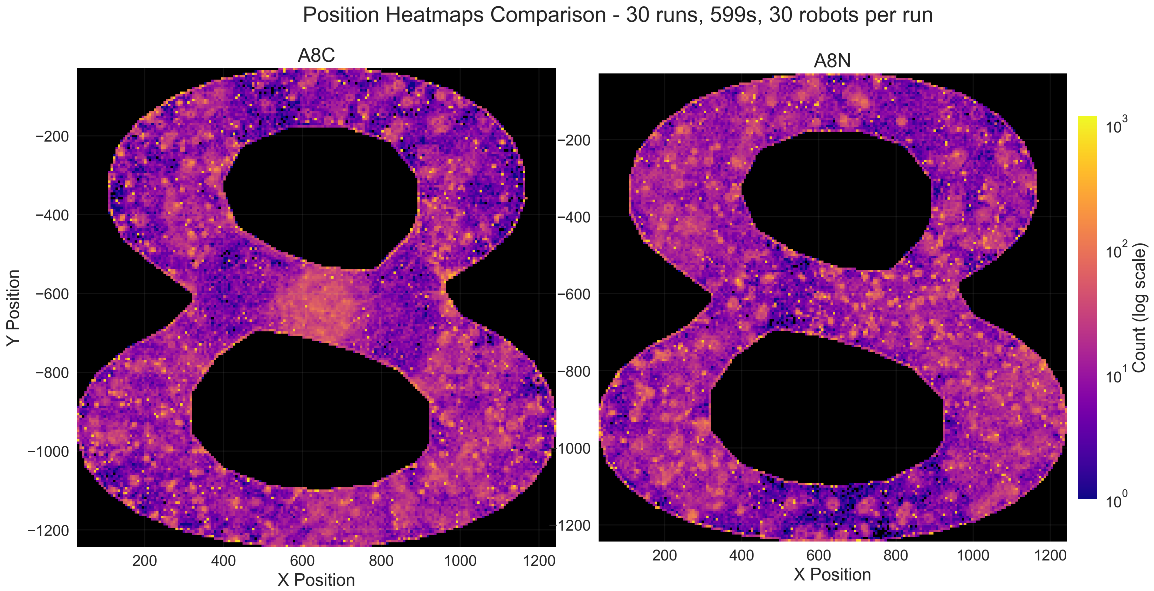

Figure 23: Left side: Digit 8 arena with circle gradient light, Right side: Empty Digit 8 arena. We can see the robots clustering towards the center.

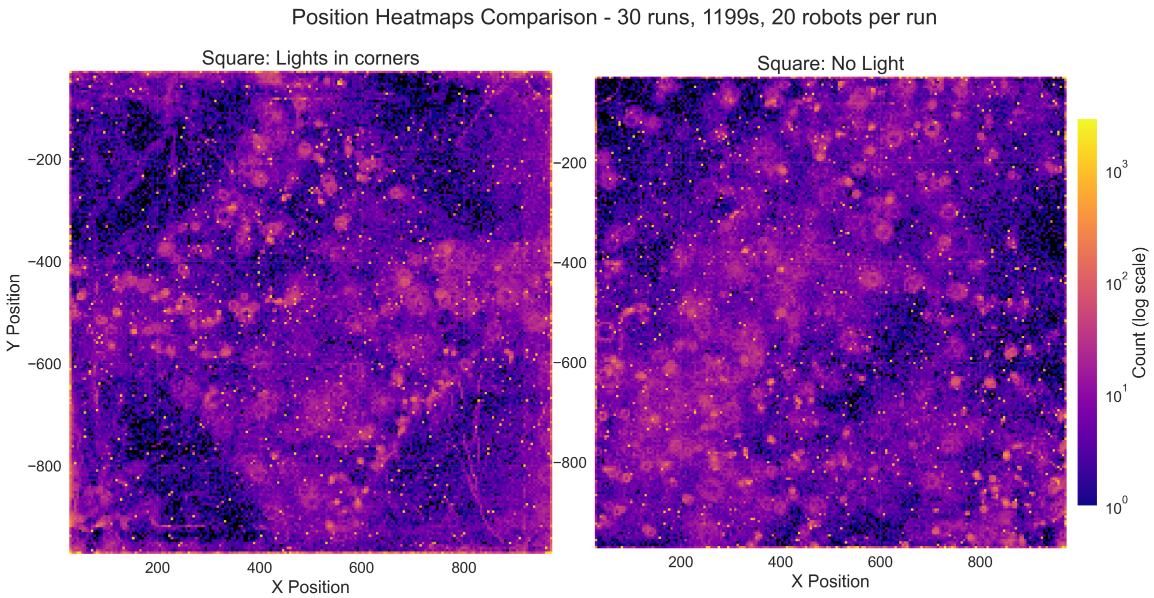

Figure 24: Left side: Square arena with 4 circle gradient lights in the corners, Right side: Empty Square arena. With so many lights, they tend not to cluster around the lights, but away from the lights near the center.