Using Reinforcement Learning to Support a Falling Pillar with Swarm Intelligence

CMSC421
Group 14
Michael Yang, Alex Wang, Will Chambers, Tae Jung, Radvilas Mikonis, Jnanadeep Dandu

INTRODUCTION

Swarm construction is primarily studied in the context of cooperative group robotics; researchers have spent decades trying various methods of coercing multiple agents to achieve tasks together that would be impossible individually. Early work in this field employed organizational structure based learning in which each agent optimized a local self evaluation function based on global parameters[4]. Other early models encouraged cooperation with biologically inspired cooperation methods, such as pheromone trails with particle swarm optimization to enable load balancing[3]. Recently, multi-agent Reinforcement Learning (RL) has proven itself a viable replacement, as the required cooperative properties enforced by earlier methodologies come about as emergent behaviors in properly constructed models [1].

Advancements in theory and training techniques [7] have made multi-agent reinforcement more and more appealing, and the integration of RL and industry standard physics simulation tools is publicly available for free through Unity in the Unity Machine Learning Agents Toolkit (ML-Agents). Unity ML-Agents [2] is a package for the Unity Game Engine, which exposes the high-quality physics, and scene description tools inside Unity to the rich ML ecosystem of Python.

Due to its well documented github repository, many tutorials, ease of use, and advanced physics engine, ML-agents is popular for modelling many types of real-world based reinforcement learning [2]. Unity offers a detailed user interface that allows users to easily and quickly create and model unique environments, and training can be watched live via the Unity engine. Using this engine, swarm construction can be modeled with a large array of Unity objects each undergoing reinforcement learning.

Swarm construction is a large field of research, and its uses can be found in many areas of study. The implementation of having simple robots construct surprisingly complex shapes or systems is difficult, but these systems can frequently be found in nature. Many early swarm construction models were based after insects, such as ants or wasps. Each worker performs simple tasks, but together the results can be extremely complex systems of habitation or paths. These ideas, when used on a human scale, can increase the possible types of automated constructs, the efficiency of construction, and even potentially the safety of human operators [9].

To construct any structure, a swarm must at least be able to stand up structural members. Understanding how groups of independent agents could work together to construct or reinforce simple structural supports could reduce the need or enhance the ability of structural engineers working in the field designing at a higher level. In the best case scenario, a swarm construction could be discovered that outperforms traditional methods.

In order to investigate whether a group of simple agents could work together to support an unstable pillar, a training program was designed and modeled within Unity. A wide array of variables were examined to gather a breadth of understanding about possible variables that could impact the ability of the swarm to support the pillar. Six different models were constructed with different success criteria, and any emergent cooperation was recorded.

METHODS

The goal of each agent is to keep a central pillar upright. In order to force episodes to end, the pillar was given an initial force in an arbitrary direction. This initial force was either maintained in the same direction each episode, or given a random direction each episode.

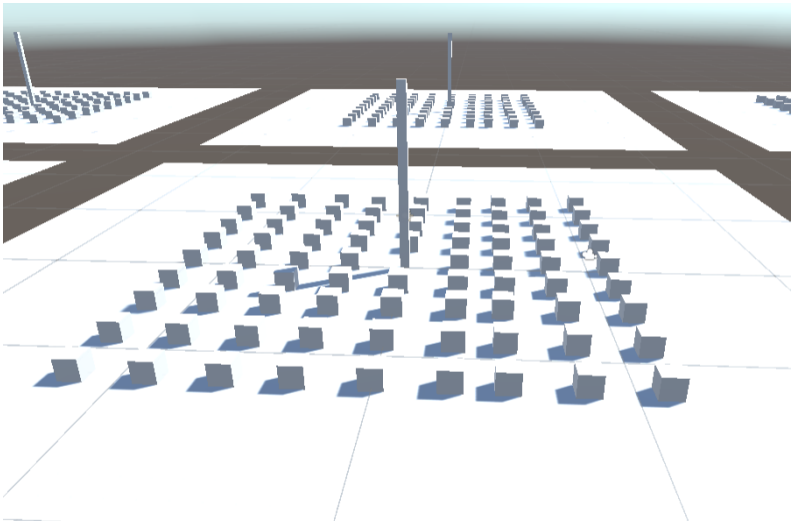


Figure 1: *Example experimental setup*

Observation spaces are denoted complex or simple based on size. Complex observation spaces include information which scales with the number of agents, or are otherwise very large. Simple observation spaces are those which depend on only a few inputs.

Available unity action types for individual agents can be discrete or continuous. Discrete actions allow for simpler implementations and fewer decisions for a model to train on, whereas continuous actions allow for the agents to move in more complex ways, but increase the learning difficulty.

The shape of the agent affects the nature of the physics simulation, we used either cubes or spheres. Additionally, the number of agents used varied.

We used two different training models, PPO, the default learning algorithm at OpenAI, and MA-POCA. Both models are developed by unity ml-agents based on extensive prior research in single, and multi-agent reinforcement learning. PPO internally builds a neural-net, which is used to assign actions to input sets. MA_POCA is specialized for group learning; the model evaluates agents from a centralized critic, thus equating individual and group rewards, based on [8].

Table 1: Experimental Hyperparameters

Exp #	Observation Space (Complex/Simple)	Action type (continuous vs discrete, grounded vs jumping)	Physical Parameters (# of balls / squares)	Model Type (PPO/MA-poca)
1	Complex	Discrete, Grounded	3 cubes	MA-poca
2	Simple	Continuous, Jumping	80 cubes	PPO
3	Complex	Discrete, Grounded	5 cubes	PPO
4	Complex	Continuous, Grounded	7 spheres	PPO
5	Complex	Discrete, Jumping	6 spheres	PPO
6	Simple	Continuous, Grounded	10 spheres	PPO

Table 2: Experiment Rewards and Basic Description

Exp #	Reward Scheme	Description
1	Linear in pillar up time, large penalty for pillar down, agent out of bounds.	Observe effect of inter-agent position information on emergence of cooperation.
2	Linear with respect to relative distance to pillar and relative height, time	Large multi-agent test with extremely simple agents
3	Linear in collisions with pillar and pillar uptime, penalty for agents out of bounds or not colliding with pillar or pillar down	Examined the consistency of result with randomly spawned agents without a group reward system
4	Linear, proximity to pillar, pillar angle, small penalties for failure and leaving area	Different sized agents with limited resources that leverage complex observations to find a solution to the goal
5	Linear reward, penalty when agents collide, reward when agent collides with pillar, pillar height change.	Broke down the goal into multiple sub goals
6	Linear in pillar up time and proximity to pillar. Penalty for pillar down, agent out of bounds	Observed simple reward structure for individual sphere agents

RESULTS

Experiment 1:

We found that introducing partner information does not improve pillar support performance, indeed, the learning stagnates at roughly half the max group reward for the same system without partner observations. Additionally, cooperative behavior is observed without partner information.

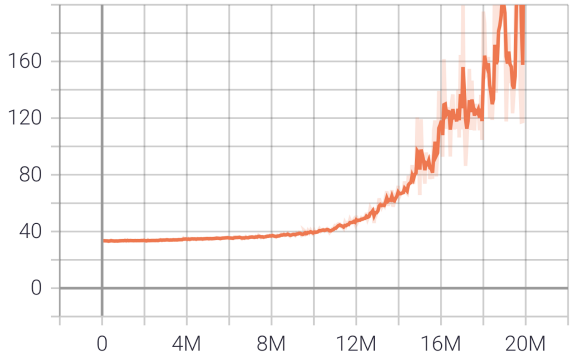


Fig 1.1. *No partner positions*

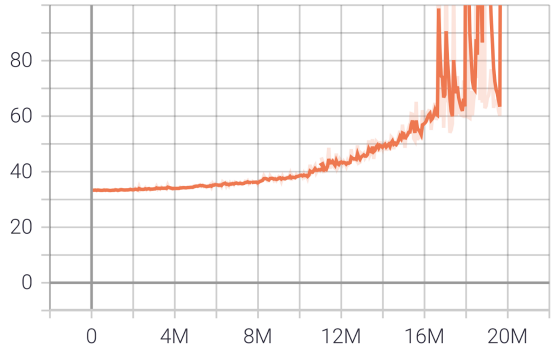


Fig 1.2. *With partner Positions*

Experiment 2:

For this system, the ideal number of training steps was approximately 4 million, as can be seen at the peaks in both Fig 2.1 and Fig 2.2. Beyond 4 million, the system started to overtrain, and slowly saw a decrease in episode length and cumulative reward. In the optimal case at 4.1M training steps, agents would first swarm the pillar. When the pillar eventually fell down, the agents would fan out to potentially prevent the pillar from falling all the way down. After the agents have fanned out and the pillar has reached a low y position, the agents quickly begin to jump to increase the score, but because of the instability of the agents the pillar often quickly falls through the gaps between the agents and the episode ends.

Fig 2.1. *Massive swarm cumulative reward*

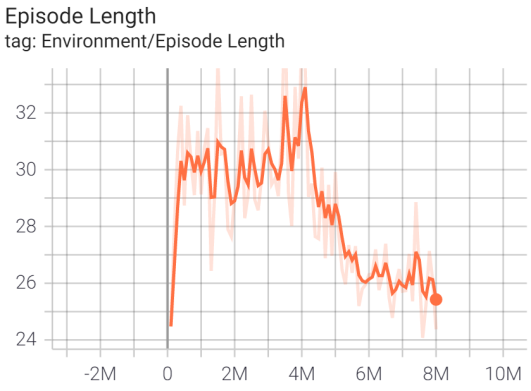


Fig 2.2. *Massive swarm episode length*

Experiment 3:

Approximately 4 million steps were taken as shown in Fig. 3.1 and Fig. 3.2. These two figures are almost identical in chart since the agents were being rewarded for the pillar not falling over the limit and each episode being terminated when the pillar has fallen over. Despite the fact that these two figures show an overall uptrend, inconsistencies were observed during the experimentation. Since agents were starting from random locations and that the agents were not

set up as a multi-agent group, the agents were successful at holding the pillar up only when they were spawned from every side of the pillar shown in Fig. 3.3. When all the agents started from one side of the pilar, it was pushed to one side of the map shown in Fig. 3.4. The spikes in Fig. 3.1 and Fig. 3.2 displays the outcome during a perfect starting point of each agent relative to the pillar.

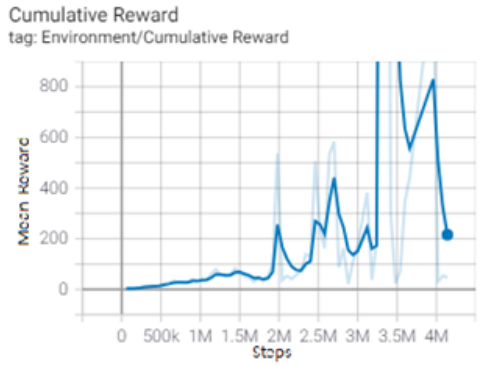


Fig. 3.1 *Mean rewards for steps taken*

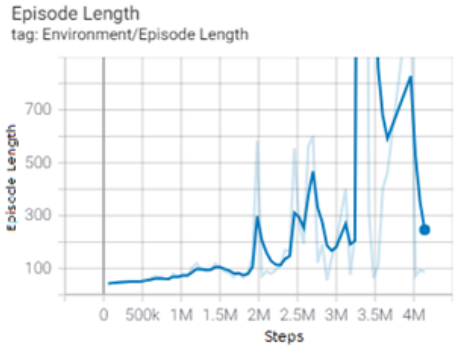


Fig. 3.2 *Length of episodes for steps taken*

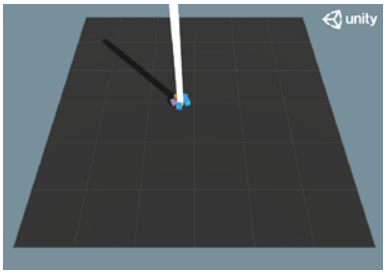


Fig. 3.3 *Agents from all sides*

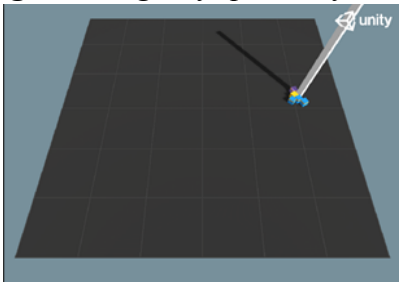


Fig.3.4 *Agents on one side*

Experiment 4:

This experiment used 2 small spheres, 2 medium spheres, and 3 large spheres that forced a different approach to be taken by the agents to accomplish the goal. It was trained for 2 million steps. The behavior that emerged was to support the pillar along with the other agents once it had fallen to a side as seen in Fig. 4.3 with a "carrying" behavior. The cumulative reward saw a linear increase, although saw dips during interruptions in training. Similar results were seen in episode length, with some outliers that aligned with the interruptions in Fig. 4.1 and 4.2. After resuming the training, the trend continued in a linear manner.

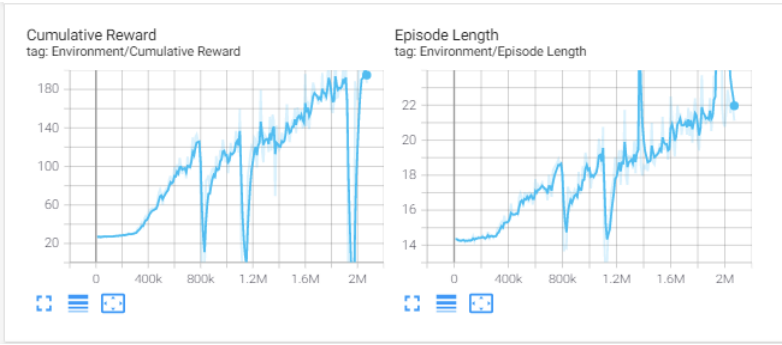


Fig. 4.1 *Interruptions in training* **Fig. 4.2** *Episode Length*

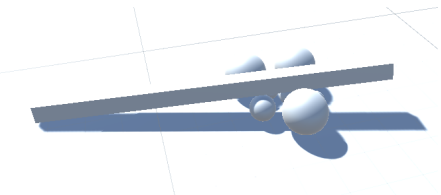
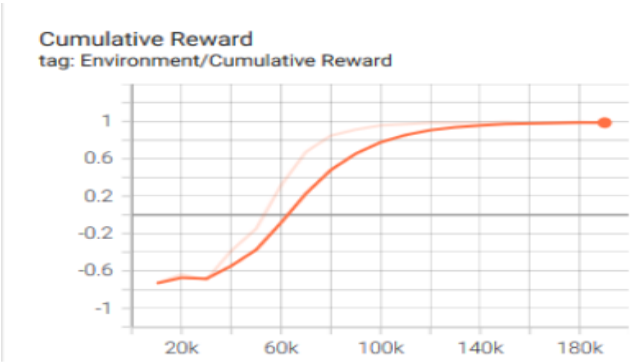


Fig. 4.3 *"Carrying" Behavior*

Experiment 5:

The goal of holding up a pillar was broken up into two separate sub goals. The first goal was to teach the agents how to find the pillar. The agent's average reward starts to converge to 1 at around the 140kth step from Fig. 5.1 and the episode's length starts to drop to 0 from Fig. 5.2, meaning that the agent is almost immediately able to find the pillar at the start of every episode.



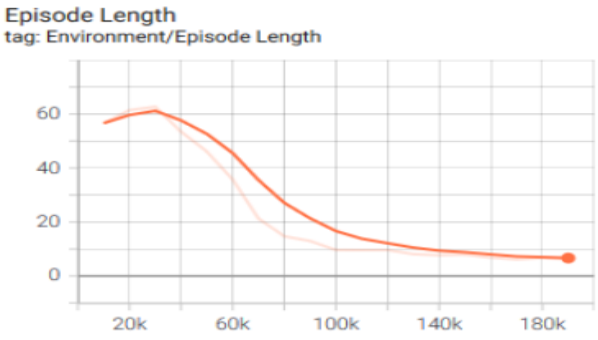


Fig. 5.1 *Mean rewards for finding pillar* **Fig. 5.2** *length of episodes for finding pillar*

Once this goal was achieved, the next goal was to teach them how to hold up the pillar. For this goal, all the agents were given the brain model learned from the first goal of finding the pillar. The average reward and episode length are increasing together over time at a slow rate from Fig 5.3 and Fig 5.4, but there are also random up and down jumps. This is because when the agent runs under the pillar and jumps, the pillar is still falling at an angle so the agent ends up missing the pillar. Once the agent jumps and misses, it is most likely that the pillar will fall and end the episode.

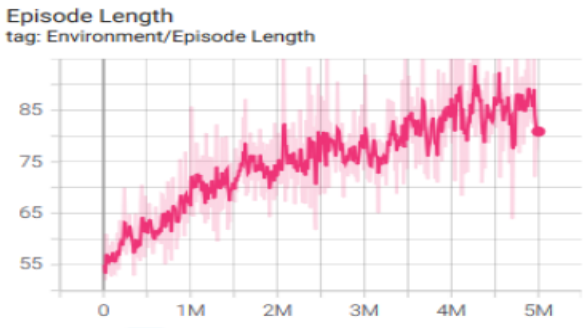


Fig 5.3 *Cumulative reward for steps taken* **Fig 5.4** *Length of episodes for steps taken*

Experiment 6:

There are 10 agents acting individually with a simple reward structure. Approximately three million steps were taken as shown in Fig 6.1 and 6.2. The reward structure was based largely on proximity of agents to the pillar as well as incrementing the reward by a small amount for each action. The graphs for cumulative reward and episode length are generally increasing together and share observable similarities in structure due to the agents getting rewarded for the pillar not falling and the episode ending when the pillar falls. The average reward and episode length was reliably increasing over time at a greater rate than most other experiments. However, around 1.97M steps, there is a clear indication of overtraining resulting in a significant decrease in episode length that did not recover.

Fig 6.1. *Cumulative reward for steps taken*

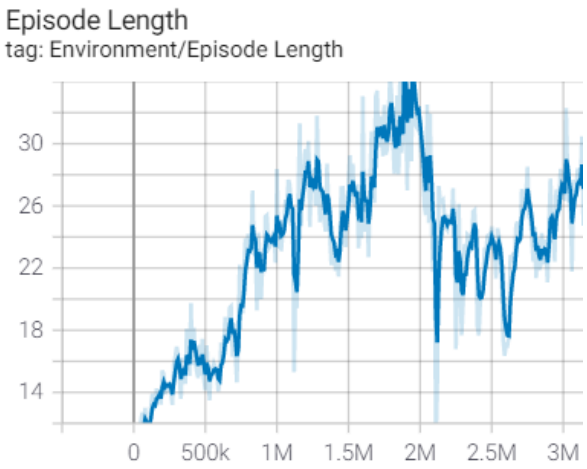


Fig 6.2. *Length of episodes for steps taken*

DISCUSSION

By fanning out and experimenting in several different directions many potentially important model parameters in achieving emergent cooperation were found and the cooperation was applied to pillar support. However, due to our broad approach, many of our findings require more work to be definitive.

In general, it was found that simpler rewards performed better. Increasing the complexity of the rewards appears to increase the difficulty of the agents to learn how to optimize more rewards methods and results in agent confusion over a long period of training.

Giving agents the ability to jump generally seems to have a positive impact on support performance. With both simpler and more complex models, enabling jumping gave agents a critical advantage over grounded agents, as having an extra dimension of movement enabled much more complex action types. Grounded agents rely on early contact with the pillar to have enough leverage to push the pillar to a vertical position, but enabling jumping allows for much more flexible behavior.

Observation space size has unclear impact. In experiment 1, the partner position information added so much complexity that cooperative performance decreased, while in experiment 4 a large observation has seemingly no negative effect on the training. This could simply be a difference between a multi-agent model, and a single agent model applied to several agents, but more work is needed.

The rewards were also found to be more useful when applied in a positive reinforcement format to drive the agents to accomplish the goal. Focusing on adding large penalties encouraged the agents to avoid the penalty rather than accomplish the task, which resulted in behavior such as avoiding the ledge but not supporting the pillar.

Breaking the goal into simpler goals seems to have helped the agents learn more efficiently. In Experiment 5, after the agents learned the location of the pillar, they immediately knew where to go to try to hold the pillar up, allowing them to support it sooner.

Additionally, adding more reward methods can result in inappropriately large ratios of magnitude for different rewards. Adding the height of the pillar could be a reasonable metric, but if the metric of change in pillar height is used for its sign, the height of the pillar can significantly outweigh the other metrics. These behaviors also differed based on the experimental constraints of each method. This is seen in the carrying behavior example as opposed to the jumping

behavior. Sometimes, the scope of the experiment was not sufficient to reach the desired outcome, therefore reevaluation of experiment parameters was often necessary.

In an optimally trained solution we expected to see the agents stand the pillar up, and then cease movement, as this would indefinitely maximize the rewards for the entire group of agents. However, in all of our experiments, regardless of whether the rewards were group based or individualized, this never occurred. Thus, we conclude that the local minimum of all agents struggling over the pillar for an extended length of time was not overcome by any parameter permutation that we tried. This would be the ultimate goal of further research on pillar support.

Were we to continue research, further improvements could be made. Standardizing more between experiments would increase comparability between each experiment. Explicitly exploring the effect of observation space size or model type would be good directions to explore individually. Another option for exploration could be the significance of agents' shape, as it was experimented with but not fully analyzed. A potential related experiment would be to try to use a cell based building system. Currently, the number of observations and variables necessary for each agent to have full knowledge of the environment is extremely large.

BIBLIOGRAPHY

[1] Sheikh, Hassam Ullah, and Ladislau Bölöni. "Multi-agent reinforcement learning for problems with combined individual and team reward." *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020.

[2] Nandy, Abhishek, and Manisha Biswas. "Unity ml-agents." *Neural Networks in Unity*. Apress, Berkeley, CA, 2018. 27-67.

[3] Meng, Yan, and Jing Gan. "A distributed swarm intelligence based algorithm for a cooperative multi-robot construction task." *2008 IEEE Swarm Intelligence Symposium*. IEEE, 2008.

[4] Takadama, K., et al. "Learning model for adaptive behaviors as an organized group of swarm robots." *Artificial Life and Robotics* 2.3 (1998): 123-128.

[5] Matta, M., et al. "Q-RTS: a real-time swarm intelligence based on multi-agent Q-learning." *Electronics Letters* 55.10 (2019): 589-591.

[6] Stewart, Robert L., and R. Andrew Russell. "A distributed feedback mechanism to regulate wall construction by a robotic swarm." *Adaptive Behavior* 14.1 (2006): 21-51.

[7] Zhang, Kaiqing, Zhuoran Yang, and Tamer Başar. "Multi-agent reinforcement learning: A selective overview of theories and algorithms." *arXiv preprint arXiv:1911.10635* (2019).

[8] Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." *arXiv preprint arXiv:1706.02275* (2017).

[9] Alexander Grushin, James A. Reggia, "*Automated design of distributed control rules for the self-assembly of prespecified artificial structures*", Robotics and Autonomous Systems, Volume 56, Issue 4, 2008, Pages 334-359, ISSN 0921-8890, https://doi.org/10.1016/j.robot.2007.08.006.

CONTRIBUTION STATEMENT

Alex Wang - Introduction, results #2, discussion, methodology #2, distribution of model - 20%

Will Chambers - Introduction, original concept for pillar support, results, methods #1, discussion - 16%

Radvilas - Methodology/results #4, editing of the final report, portion of the discussion section, collaboration with the group throughout. - 16%

Michael Yang - Method/result 5, discussion- 16%

Tae Jung - Methods/results 3, discussion - 16%

Jnanadeep Dandu - Methods/results 6, discussion- 16%