# Advancing Soft Robots with Evolutionary and Reinforcement Learning

Lead Researcher: Aba Onumah    Faculty Advisor: Dr. Yu Xiang    Anish Reddy Alle    Arjun Krisnan    Jessica Myoungbin Choi    Rohit Penna

## Introduction and Purpose

Soft robotics is an emerging field that utilizes flexible materials such as rubber or EcoFlex to create robots capable of adapting to a wide variety of tasks. These robots are especially useful where traditional rigid robots may struggle, in dynamic environments like cave systems, disaster sites, agricultural fields, and usage in deep-sea exploration.

However, designing and building soft robots from scratch to autonomously interact with these environments is a highly resource-intensive process, often requiring significant time, effort, and coordination to test and refine their performance. As scientists seek ways to streamline this process, we present a unique approach by integrating evolutionary algorithms (EAs) with reinforcement learning (RLs). This combination enables more efficient design and adaptation of soft robots, reducing both time and resource expenditures while enhancing their autonomy and robustness in unpredictable environments.

## Components

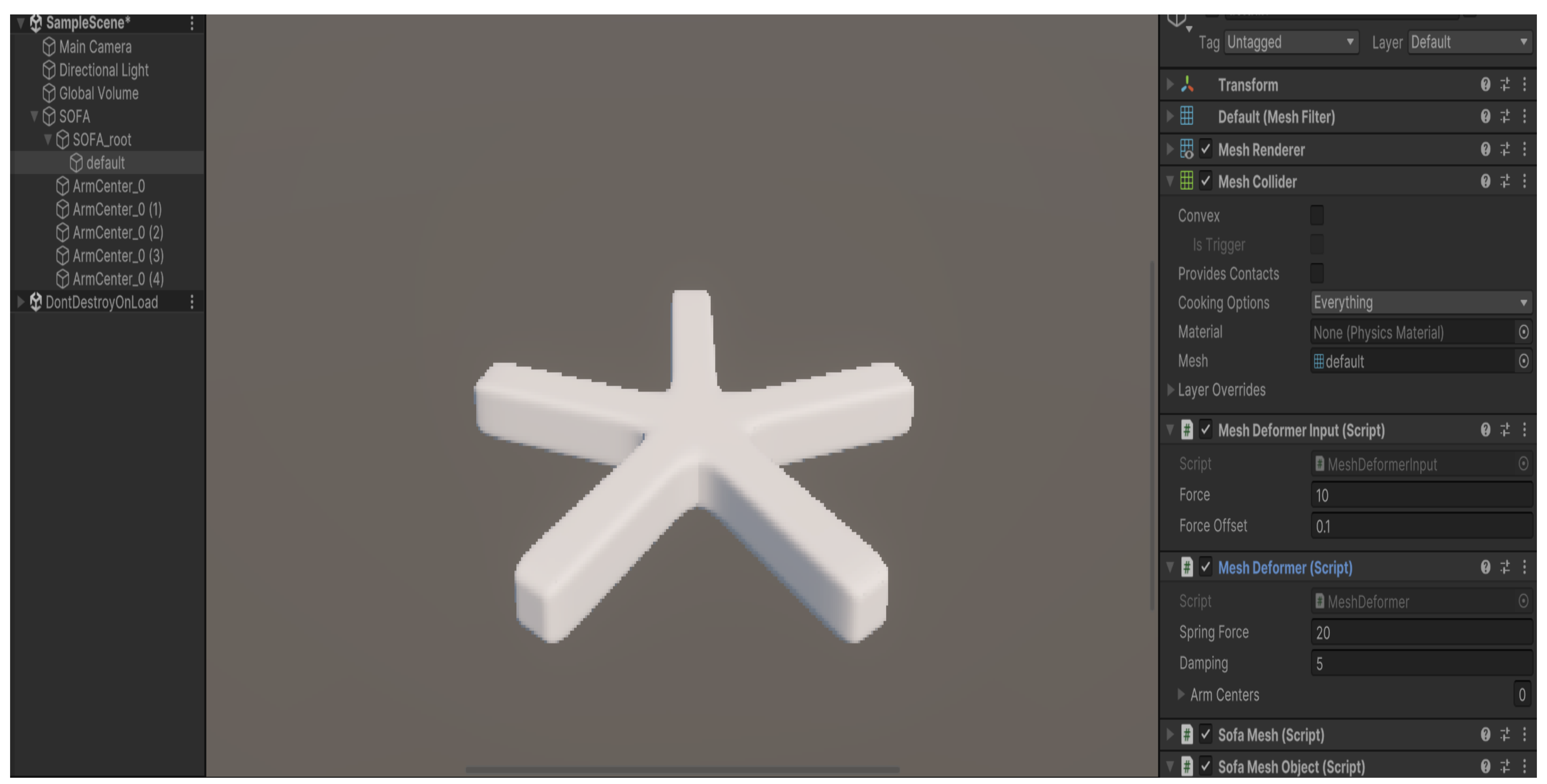
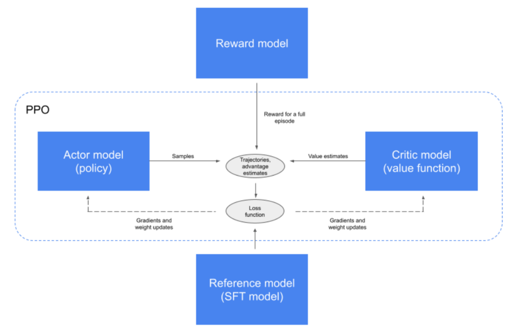


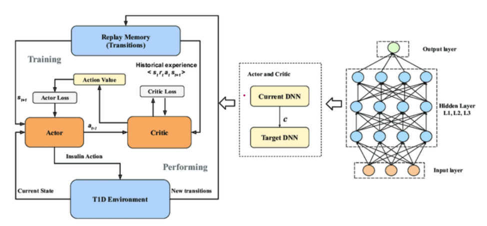
Figure 1. Unity and Sofa combined

For the development of our soft robotic model, we utilized two primary tools: Unity and the SOFA Framework. Unity provided the platform for simulating and visualizing the soft robot's environment, leveraging its powerful physics engine and machine learning capabilities. The SOFA Framework was employed to simulate soft body dynamics, enabling the modeling of deformable structures with high realism.

We integrated these two tools by using Unity's ML-Agents for environment simulation and SOFA's components for the soft body physics. While we successfully managed to integrate deformations into the model, the robot's movement was not fully realized at the time of this poster's creation. Further research and development will focus on refining the movement dynamics and enhancing the overall interaction between the components.

## Models





**Proximal Policy Optimization (PPO)**: A *reinforcement learning* (RL) algorithm that utilizes an **Actor-Critic Network** with shared layers for *feature extraction*, where the **Actor** determines the probabilities of taking certain actions from the given state, and the **Critic** assesses an action by evaluating the current state. The model then manages **policy updates** to constrain them within a threshold, ensuring that policy changes aren't too drastic.

**Deep Deterministic Policy Gradient (DDPG)**: Another *reinforcement learning algorithm* (RLA) that is designed to handle *continuous spaces* by once again utilizing an **actor-critic model** where the **actor** learns a *deterministic policy* by mapping states directly to specific actions and a **critic** that estimates the value of these actions through *Q-learning*. It leverages a **replay buffer** to sample past experiences and employs *Ornstein-Uhlenbeck processes* to add noise to the actor's actions, encouraging *exploration*.

## Results


Figure 2

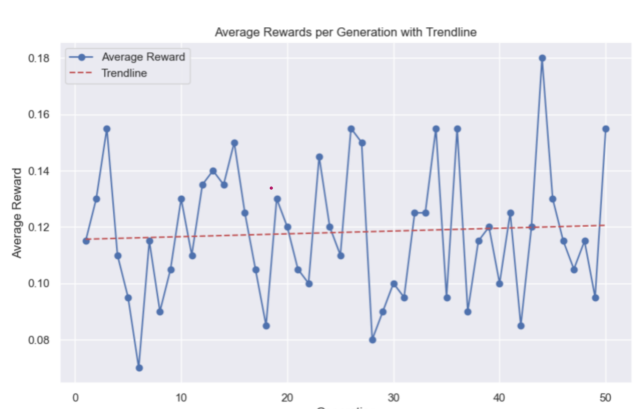



Figure 3

**Deep Deterministic Policy Gradient**: Figure 1 shows the changing episodic reward of a DDPG model during training. We used a replay buffer of state-action pairs to avoid the need for infinite integration.

We used Unity's ML-Agents Python library to simulate gymnasium environments, allowing us to visualize the training results and evolution of each model.

**Proximal Policy Optimization**: For the proximal policy optimization technique, rather than evolve through hyperparameters we opted to evolve by crossing the weights of the neural nets of the parents randomly to create the next generation. However,this optimization technique is still a work in progress. The model has been producing results in simple template gym environments, but fails when given a more complex environment as provided by Unity.

## Analysis

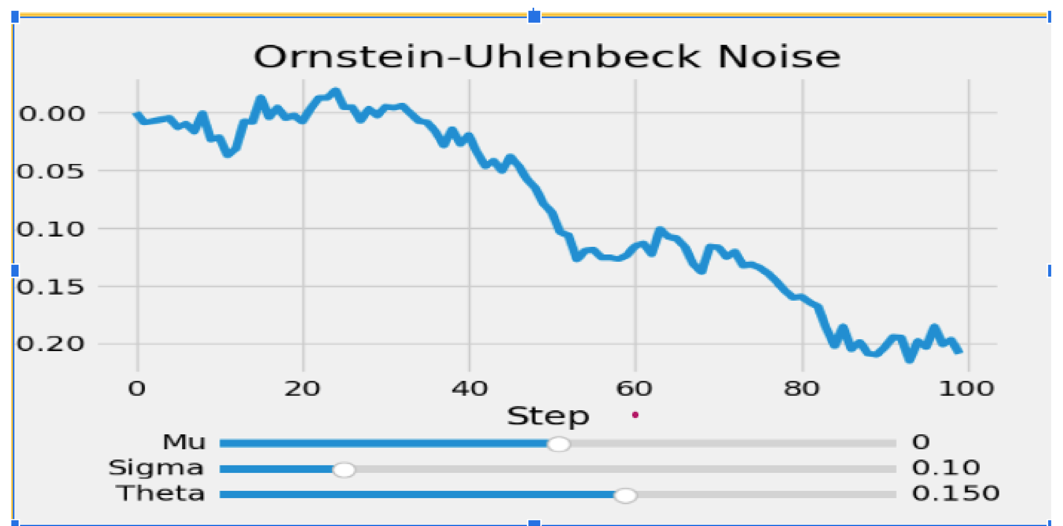



Figure 4

By evolving populations of agents across multiple generations, the framework allows for EA to complement the RL aspect by culling half the population, resulting in higher learning efficiency and policy performance. Incorporating other EA techniques such as elitism could even further improve the training performance.

The results indicate that while DDPG's EA-based approach effectively converged on suitable hyperparameters, PPO's weight-crossing strategy, although beneficial in simpler settings, struggled in complex environments.

## Conclusion

Although the results available at the time of this poster are limited, the preliminary findings indicate a promising trend. Specifically, the tuning of hyperparameters has led to an exponential increase in reaching optimal rewards for both reinforcement learning models (RL) and evolutionary algorithms (EA). These early results suggest that with further research, the potential of EAs and RLs in soft robotics could be substantial. We are optimistic that continued exploration will highlight their effectiveness and the critical role they could play in enhancing the performance of soft robots in complex, dynamic environments.

## References

- M. A. Graule, T. P. McCarthy, C. B. Teeple, J. Werfel and R. J. Wood, "SoMoGym: A Toolkit for Developing and Evaluating Controllers and Reinforcement Learning Algorithms for Soft Robots," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 4071-4078, April 2022
- Bai, Hui, et al. "Evolutionary Reinforcement Learning: A Survey." Intelligent Computing, vol. 2, 2023, pp. 1-10, doi:10.34133/icomputing.0025. arxiv.org. https://arxiv.org/pdf/2303.04150
- Mertan, Alican, and Nick Cheney. "Modular Controllers Facilitate the Co-Optimization of Morphology and Control in Soft Robots." Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23, 2023, pp. 174-183, doi:10.1145/3583131.3590416. arxiv. https://arxiv.org/pdf/2306.09358v1