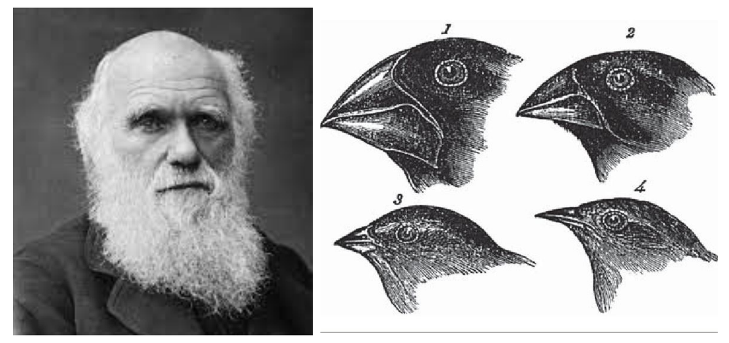
On the Origins of Robot Morphologies

Mike Zhang, Mikhail Terekhov

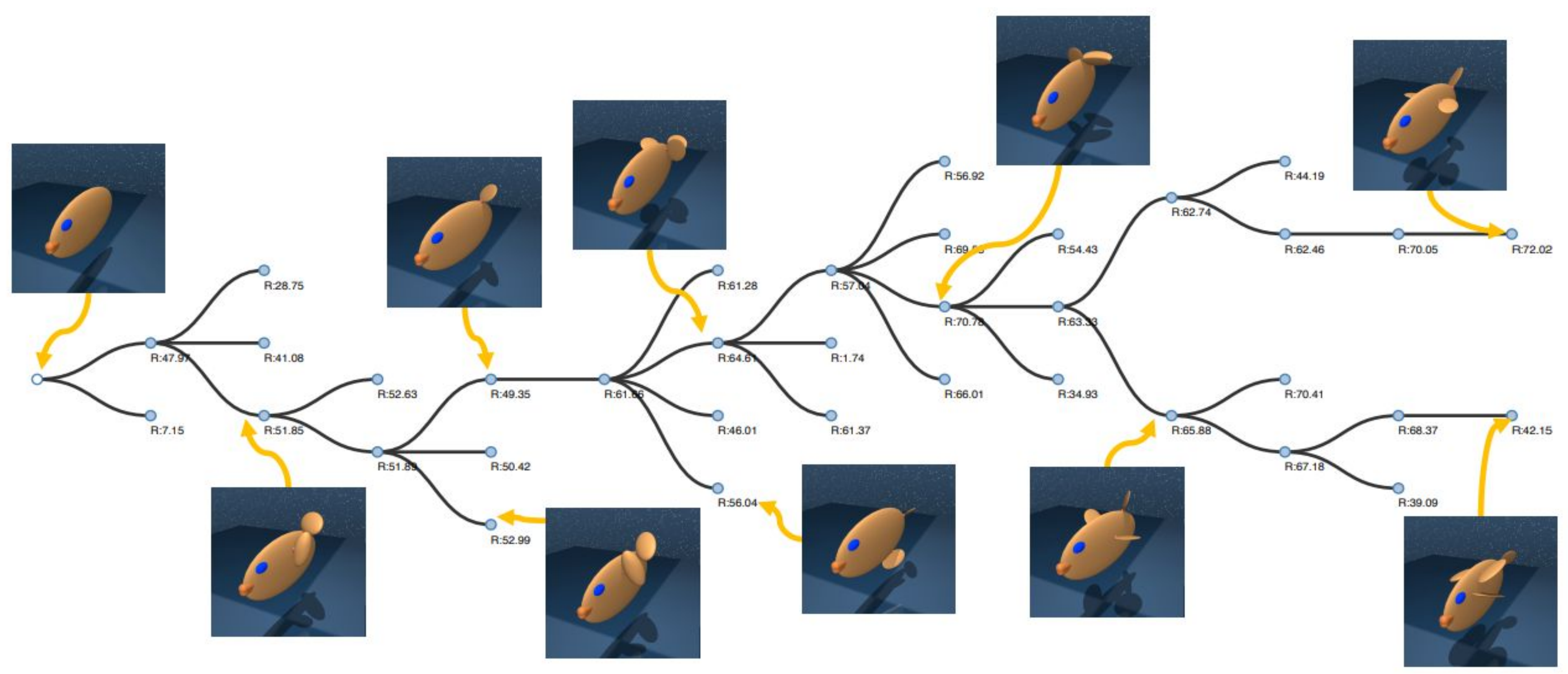


MOTIVATION: ROBOT MORPHOLOGY OPTIMIZATION

A robot is composed of both its physical design (morphology) and a controller. The morphology is generally first designed by humans and a controller is designed later.

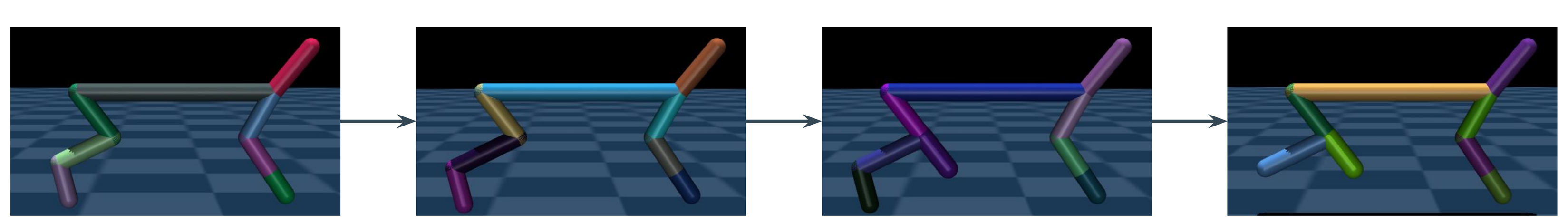
Ideally, a joint design and optimization of the morphology and controller would result in better performance. This idea leads to the field of *evolutionary robotics*. The field is inspired by its namesake process whereby embodied organisms change both in physical morphology and neurological control over generations in response to their environment. Evolutionary Search (ES) is commonly employed as an outer optimization searching over morphologies by applying random mutations, while an inner optimization computes a controller for each morphology.

Existing work in ES for morphology optimization have used data-driven methods to impart an "intuition" to the search process [3]. *We investigate if this can be taken further by having the evolutionary search process be entirely data-driven, using deep reinforcement learning.*



Example of robot morphology optimization using evolutionary search. Figure credit [1]

OUR CONTRIBUTION: *EVOLUTION ITSELF* AS AN MDP



Fitness f_1 Fitness f_2 Fitness f_3 Fitness f_4

In contrast to the use of RL only for finding a control policy for a given morphology, we frame a sequential application of mutations (evolution) as a Markov Decision Process:

- State Space: morphology and (optionally) its motion policy
- Action Space: a pre-selected set of mutations
- Transition Rules: mutation applied to the morphology, (optional) re-trained policy
- Reward: Change in the morphology's fitness $r_i = f_i - f_{i-1}$

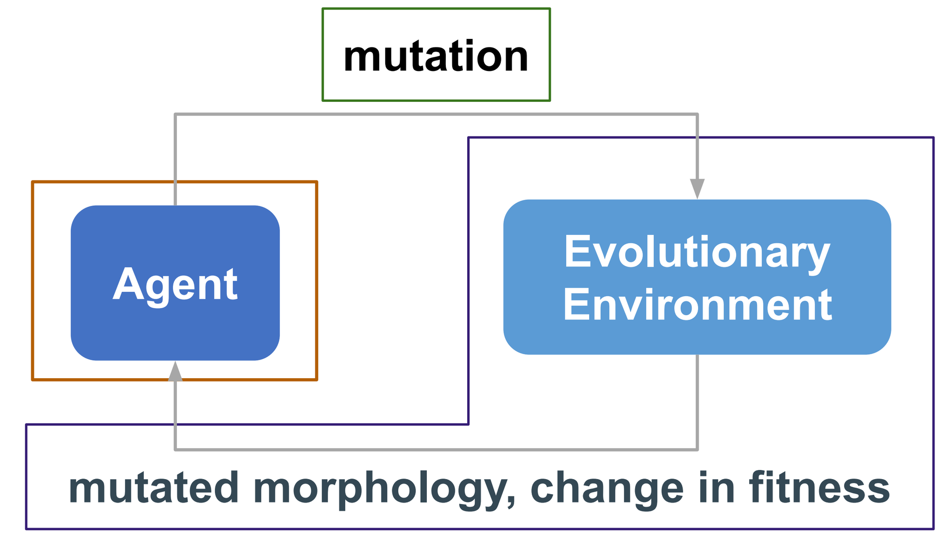
Expected benefits

- More efficient application of mutations compared to random search
- Consideration of the long-term effect of a mutation (i.e. a mutation now may allow a very useful mutation in the future)

METHODOLOGY

Action/Mutation Space: Define a mutation space amenable to neural network regression

Evolutionary RL Agent: Implementing algorithms to aid mutation selection with RL



Evolutionary Gym: Implement training environment modelled on OpenAI Gym interface

ENGINEERING CHALLENGES

Our novel approach presents unique challenges and a lack of established results in literature

Efficient Morphology Fitness Evaluation

Computing the True Reward is Intractable. The true fitness of a morphology in a task environment can only be determined by computing the optimal controller. This is intractable and we must resort to heuristic methods and estimate a lower bound on the true fitness.

We implement two methods to approximately evaluate the fitness:

1. Based on *Neural Graph Evolution* [1], which computes a controller using RL. The use of GNNs allows for parent-to-child control policy inheritance.
2. Based on *Task-Agnostic Morphology Evolution* [2], which formulates an alternative task-agnostic notion of fitness based on morphology's *predictability* and *capacity to reach distant states.*

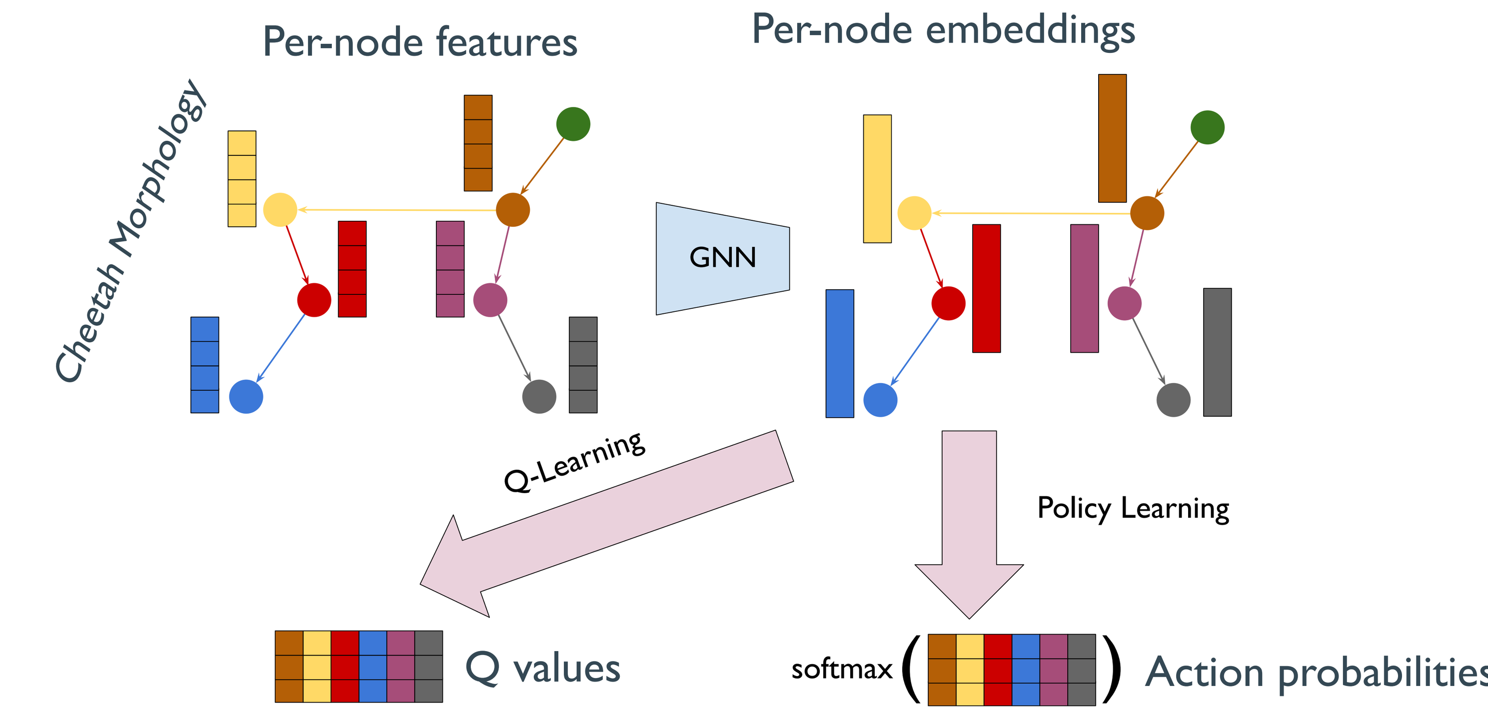
Stochasticity of the Reward. Both methods of approximating the fitness are stochastic: the same morphology may achieve different fitness under the same evaluation method. This presents another significant challenge for implementing RL: the randomness of state transition is embodied in the randomness of the reward.

Action Space Design

Variability of the Action Space. A common assumption in Deep RL is that the action space remains constant throughout the trajectory. In this work, possible actions/mutations that can be applied to a morphology depend on the number of nodes/limbs of the morphology. Over an evolutionary rollout, nodes can be added or removed.

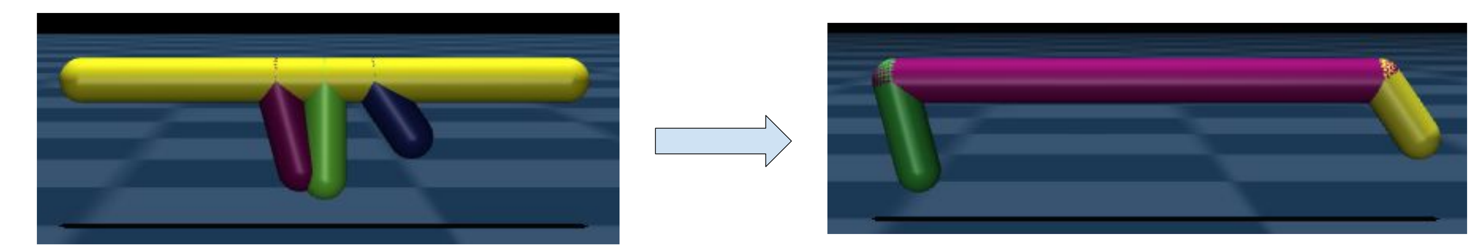
Action Variance. The mutations must be specified with enough precision to reduce the variance associated with each action. However, too precise actions may result in too many actions and difficulty in exploration during training.

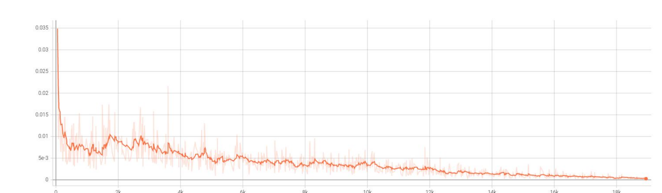
NEURAL NETWORK POLICY ARCHITECTURES



CURRENT RESULTS

- Implemented individual mutations based on TAME codebase
- Completed the evolutionary environment with TAME- and NGE-based fitnes
- Discovered TAME model unsuitablity
- Implemented the first version of Deep Q-Learning





Q-Function Estimator Loss

Average Episode Reward

Experiment design: *bad walker.* Evolutionary agent needs to remove the obstructing limb and move functional limbs to the ends of the robot. So far the fitness of a morphology is *pre-defined*, not estimated with TAME or NGE.

NEXT STEPS AND FUTURE WORK

Alternative Methods for Fitness Evaluation

Investigate methods for approximating the optimal controller for a morphology without a time consuming inner optimization. For example, a model predictive control approach used in the morphology optimization of Zhao et.al [3].

Asynchronous Reinforcement Learning

Due to the long time it takes to take a single step of the evolutionary training environment we want to improve training time by implementing parallel RL algorithms such as Asynchronous Q-Learning or A3C.

Batch Reinforcement Learning

Another approach to scaling up RL training would to massively parallelize the roll-out data collection and train an agent using offline/batch RL methods.

REFERENCES

[1] Tingwu Wang, Yuhao Zhou, Sanja Fidler, and Jimmy Ba. Neural graph evolution: Towards efficient automatic robot design. arXiv preprint arXiv:1906.05370, 2019.

[2] Hejna III, Donald J., Pieter Abbeel, and Lerrel Pinto. Task-Agnostic Morphology Evolution. arXiv preprint arXiv:2102.13100, 2021.

[3] Allan Zhao, Jie Xu, Mina Konakovic-Lukovíc, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Robogrammar: graph grammar for terrain-optimized robot design. ACM Transactions on Graphics (TOG), 39(6):1–16, 2020.