Learning a locomotion controller for a quadrupedal robot with Deep Reinforcement Learning

MaRBle 2018-2019

Mathieu Coenegracht, Charlotte Dalenbrook, Kaspar Kallast, Samuel Kopp

Department of Data Science and Knowledge Engineering, 08.07.2019

Maastricht University

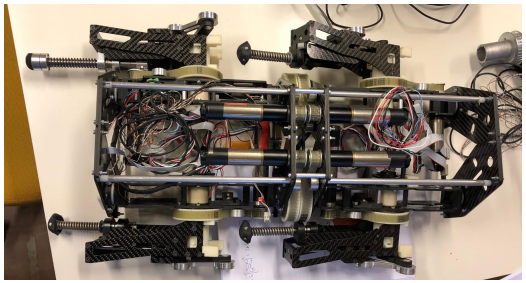
Outline

1. Project Story
2. Learning a Locomotion Gait
3. Learning on Rough Terrain
4. Robot Morphology for Rough Terrain
5. Project Conclusion & Credits
6. References

1. Project story: Goals and challenges

Original Goals:

- Rough terrain controller by guiding learning progress
- Find optimal morphology for the SwarmLab-Robot



Pivot

3 sections with different focus:

- Learning
- Terrain Generation
- Morphology

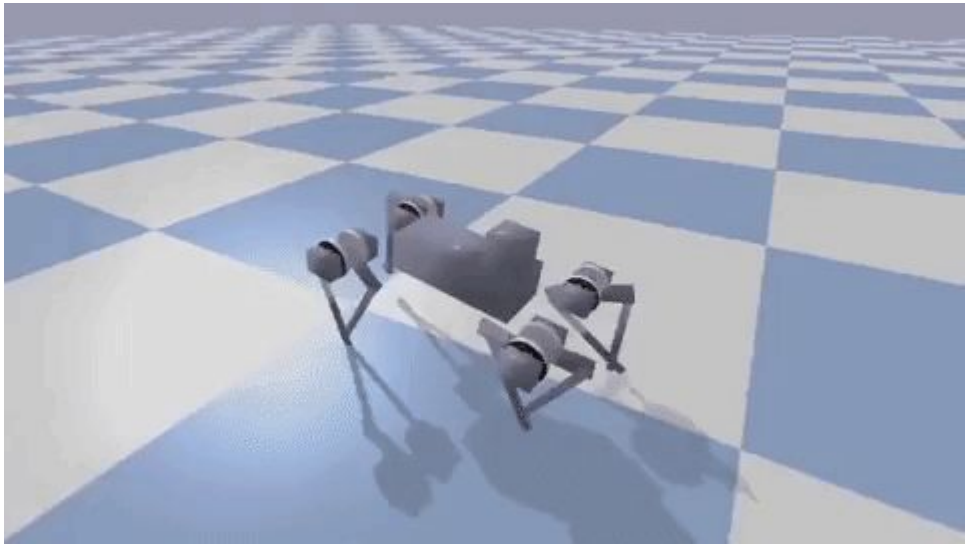
Hurdles:

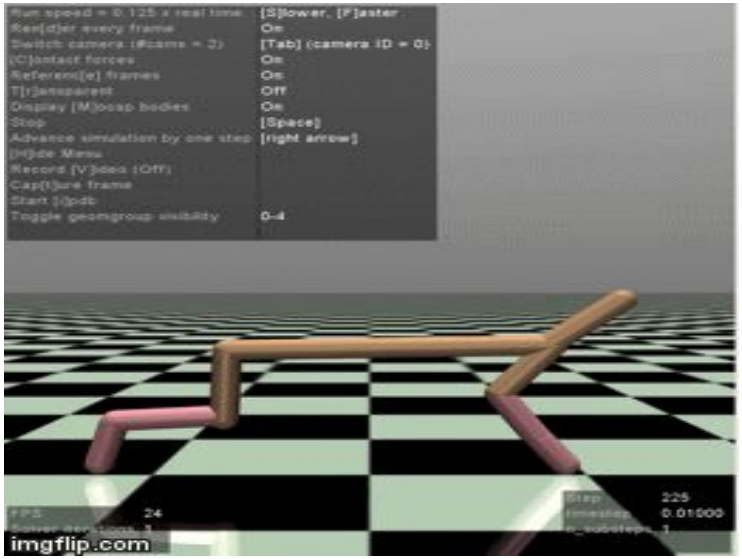
- Time & Experience
- License for Simulation Tool
- Testing Speed
- Motor Controllers

Learning a Locomotion Gait

1. Why use reinforcement learning?
2. Research question
3. Methodology
4. Experiments
5. Results & Discussion

Why Use Reinforcement Learning?





Research Question

How do different choices for the observation and action space affect the learning process?

RL-Background

- Markov Decision process (S, A, P, R)
  - S: State Space
  - A: Action Space
  - P(s'|s,a): S x A → Pr(S)
  - R(s, a, s'): S x A x S → $\mathbb{R}$
- Policy π(s, a)
- Expected return J(π)
- Goal: find a policy π that maximizes J(π)

Proximal Policy Optimization(PPO)

- On-policy
- Trust-region
- Tends to be monotonically improving
- Multiple Actors in Parallel

Schulman, John, et al. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347* (2017).

Proximal Policy Optimization(PPO)

Algorithm 1 PPO-Clip

1: Input: initial policy parameters θ_0, initial value function parameters ϕ_0
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
4: Compute rewards-to-go $\hat{R}_t$.
5: Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.
6: Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \quad g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

typically via stochastic gradient ascent with Adam.
7: Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

typically via some gradient descent algorithm.
8: **end for**

Actions & Motor Control

A: Direct Joint Angles

- PID-controller
- significantly better learning, gait quality and robustness than torques or velocity[1]

B: Feedback Oscillation with 2 DoF

, f is frequency

$$\phi(t) = f2\pi t$$

$$X_{front,j} = a_{1,j} sin(\phi(t)) + a_{2,j}$$

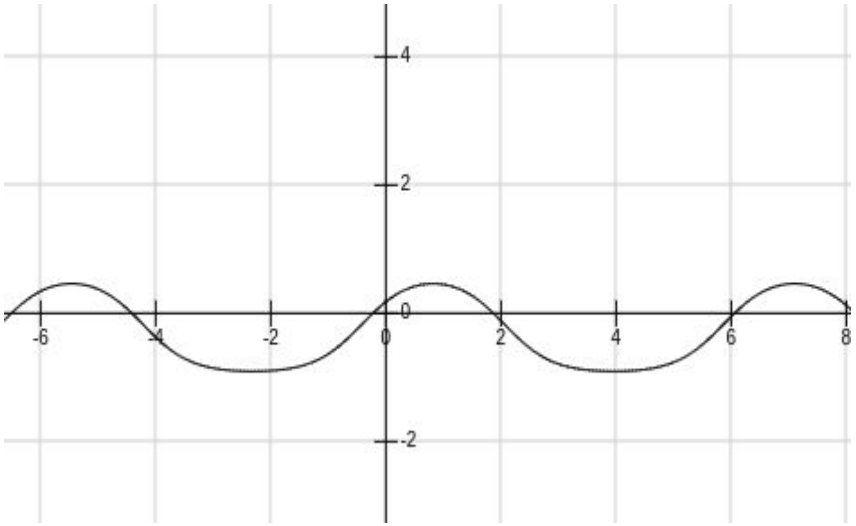
$$X_{back,j} = a_{3,j} sin(\phi(t) + a_{4,j}) + a_{5,j}$$

→ Galloping Motion

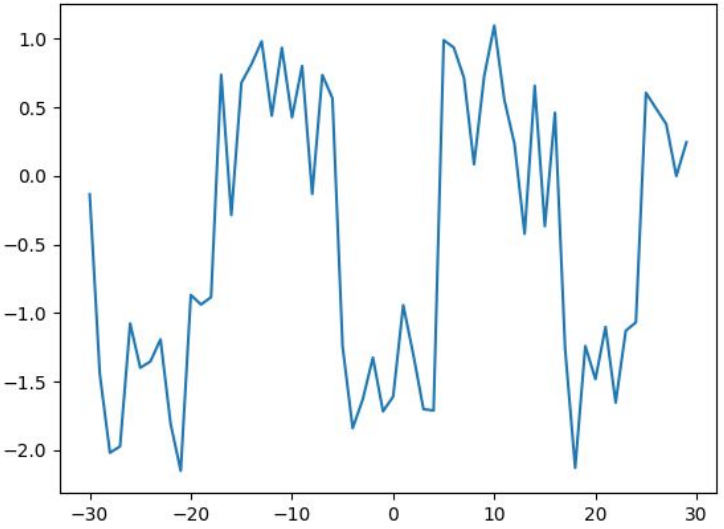
Feedback Oscillation with 6 DoF

→ Combination of A for Shoulders & B for Contraction

[1] Bin et al, *Learning locomotion skills using deeprl: Does the choice of action space matter? (2017)*



Oscillation with constants



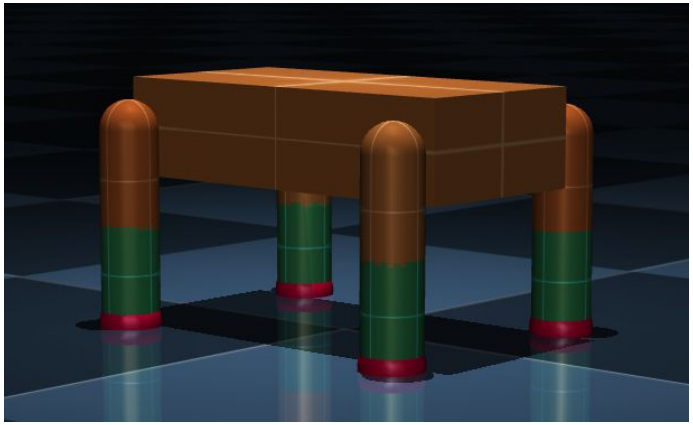
Feedback Oscillation

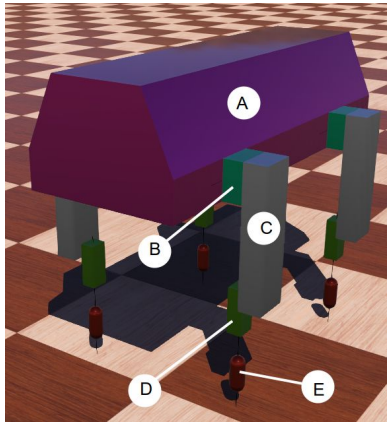
Observation Space

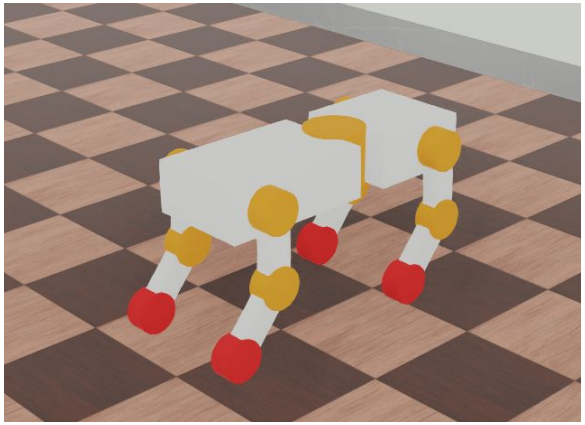
- Environment partially observable
- Increase in dimensionality → increase in time[2]
- Comparison of 5 observations spaces:
  - Joint positions
  - Joint positions and velocities
  - External forces
  - Inertia and velocities of com
  - External forces + Inertia and velocities

Simulation & Robot Models

- MuJoCo SLIP Quadruped
- Webots Swarmlab Quadruped
- Webots EPFL Ghostdog





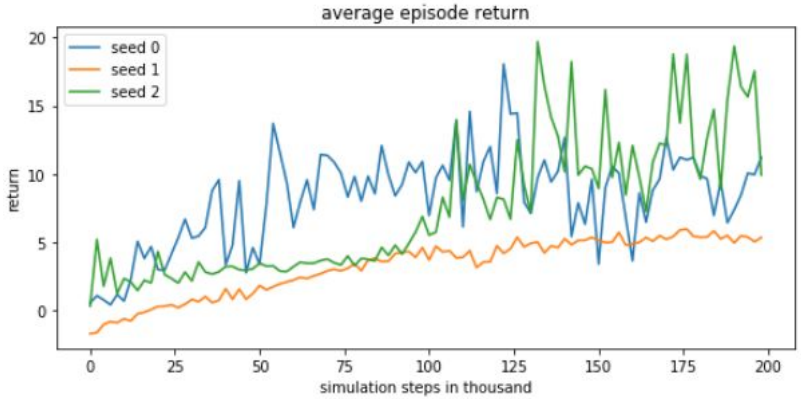


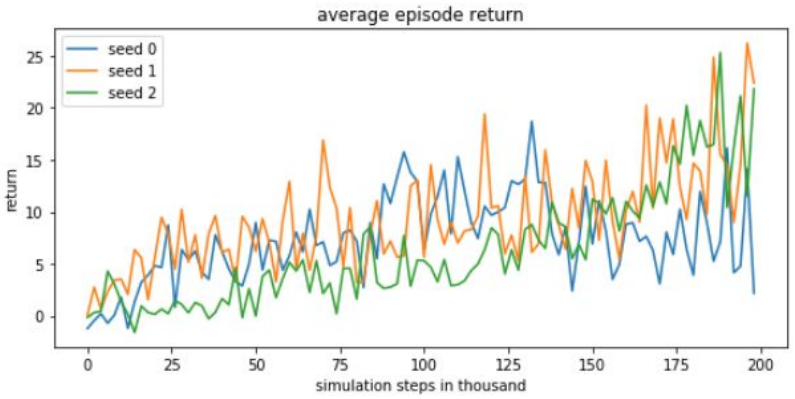
Experiments

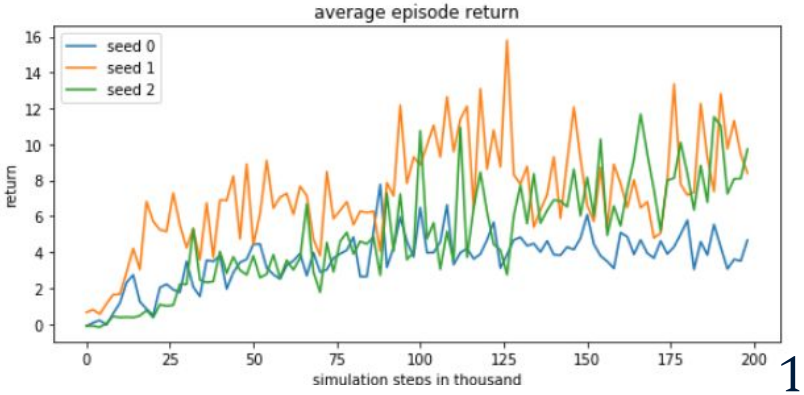
- Action space
  - Ghostdog, Swarmlab Quadruped
  - 3 seeds, 50/100 epochs, 1000/2000 steps per epoch
- Observation space:
  - MuJoCo SLIP Quadruped
  - 10 seeds, 500 epochs, 4000 environment steps
- Reward: r = d + s - e

Results: Action Space

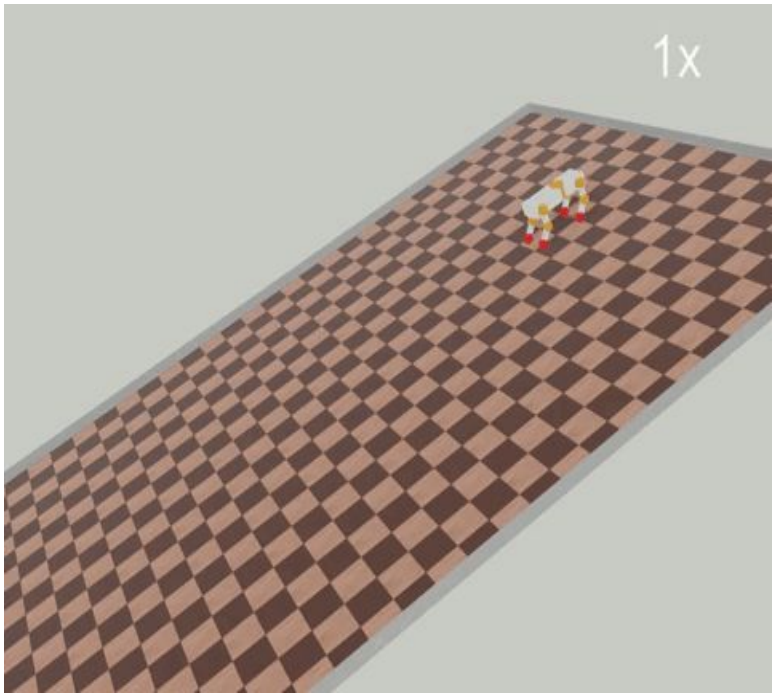
- Swarmlab quadruped
  - Agent unable to learn a gait
  - Model might be unable to balance
  - Possible that actions rarely lead to balanced step
- EPFL Ghostdog
  - Feedback-oscillation with 2 DoF produced consistently stable gait
  - Feedback-oscillation and position control 6 DoF unable to learn stable gait
  - PPO seems to have trouble learning stable gaits when dimensionality is higher



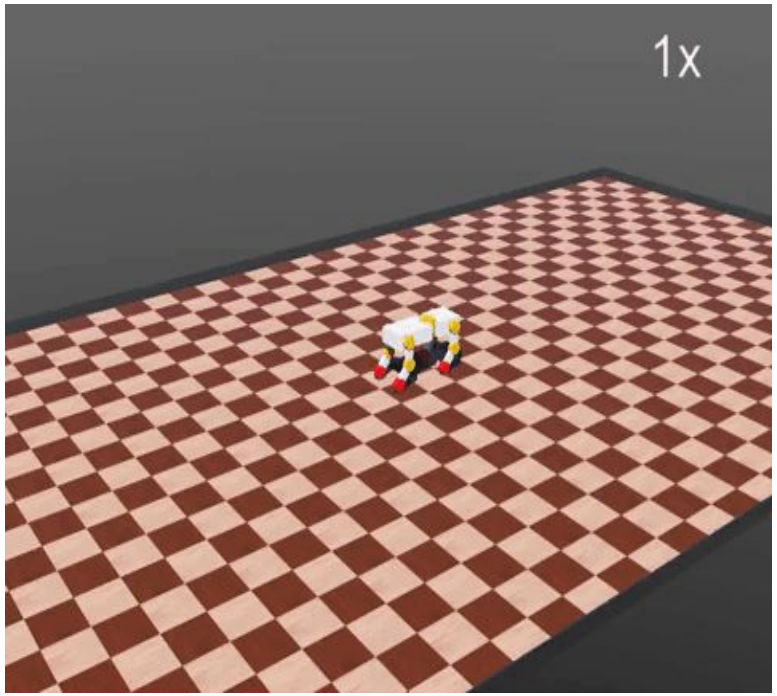




Trained policies on Ghostdog



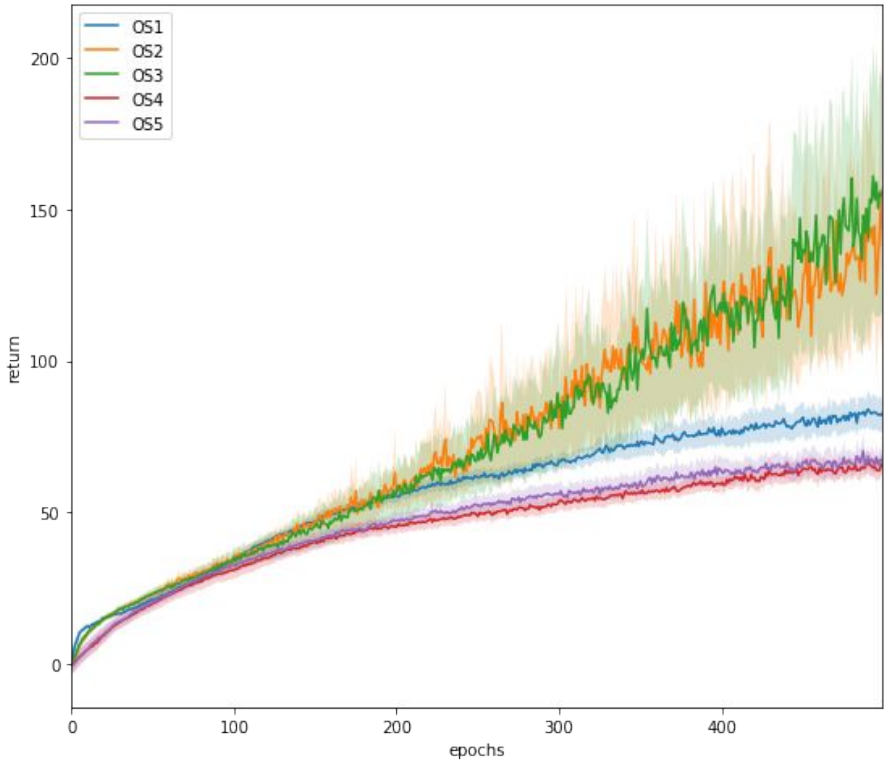
12 DoF direct position control



2 DoF feedback oscillation

Results: Observation space

- Agent unable to successfully infer velocities using positions.
- Similar performance between using joint velocities and external forces
  - Despite OS3's larger size
- Performance drop when including COM inertia and velocities
- Similar performance when additionally including external forces



Learning on Rough Terrain

- Apply reinforcement learning to learn locomotion on various kinds of courses
- Webots
- Terrain Generator

Making the Terrain Generator

- Controller/Supervisor scripts
- VRML97 world description
- Webots excellent UI

Terrain Generator Overview

- Main platform
- Elevation grid
- Obstacles
- Contact Properties
- Randomization of terrain properties

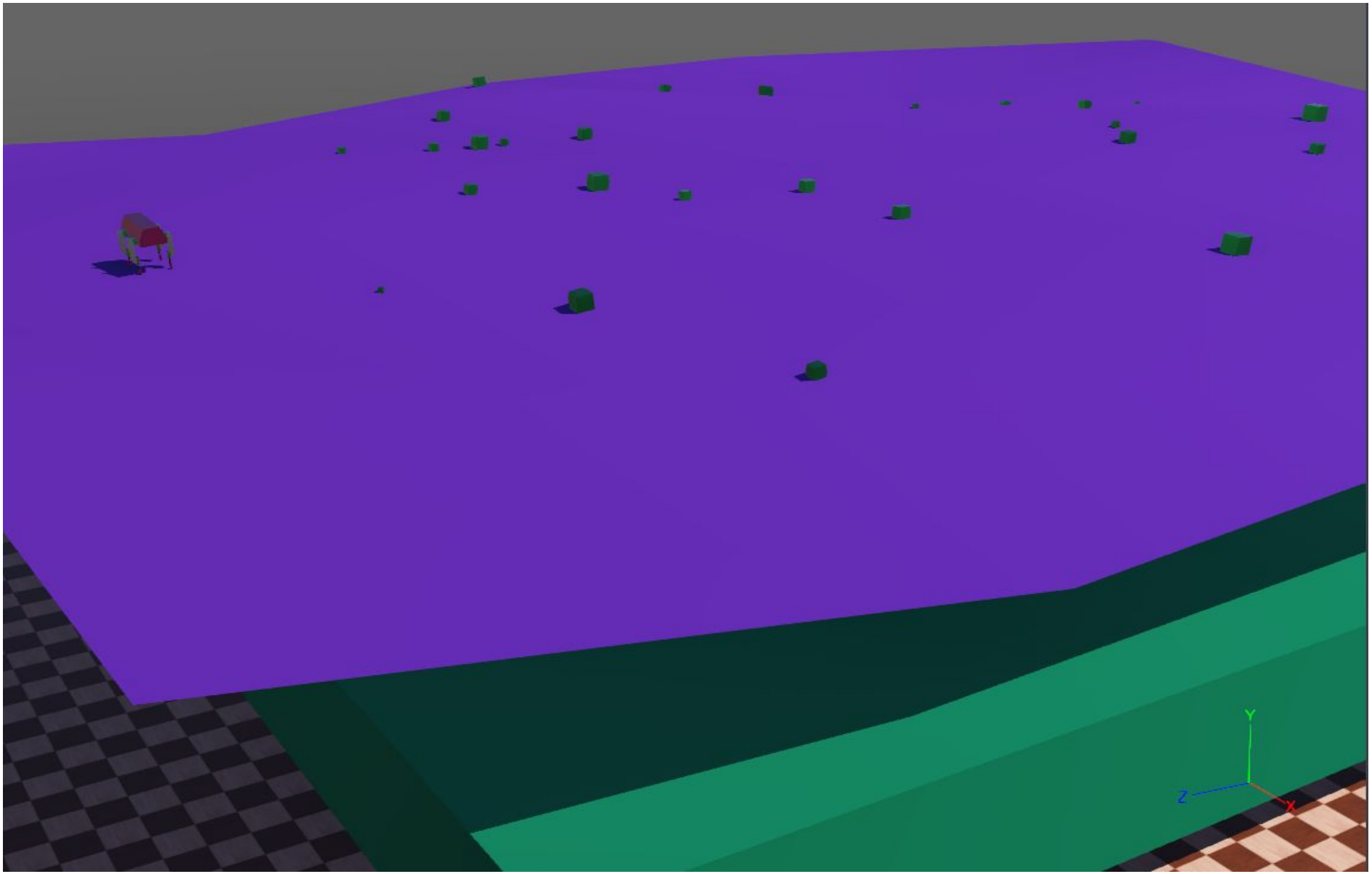
Elevation Grid

- Robot dropped on elevation grid
- Can instantly fall after drop due to uneven terrain
- To counter this: stabilizing force and reset physics upon contact with elevation grid
- Elev. grid is fit to main platform's size

Encountered Bugs

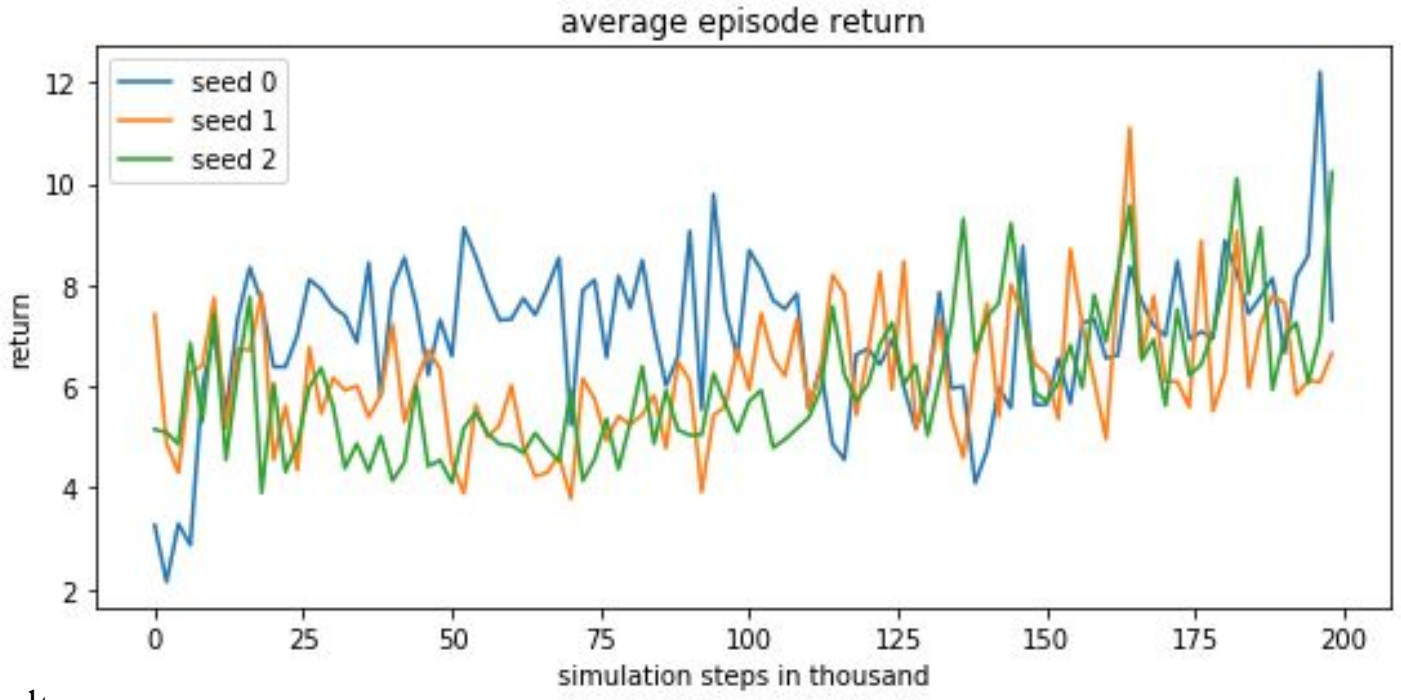
- Bounding object of solid not updating correctly
- Change in centre of mass when spawning new robot model

Example Terrain



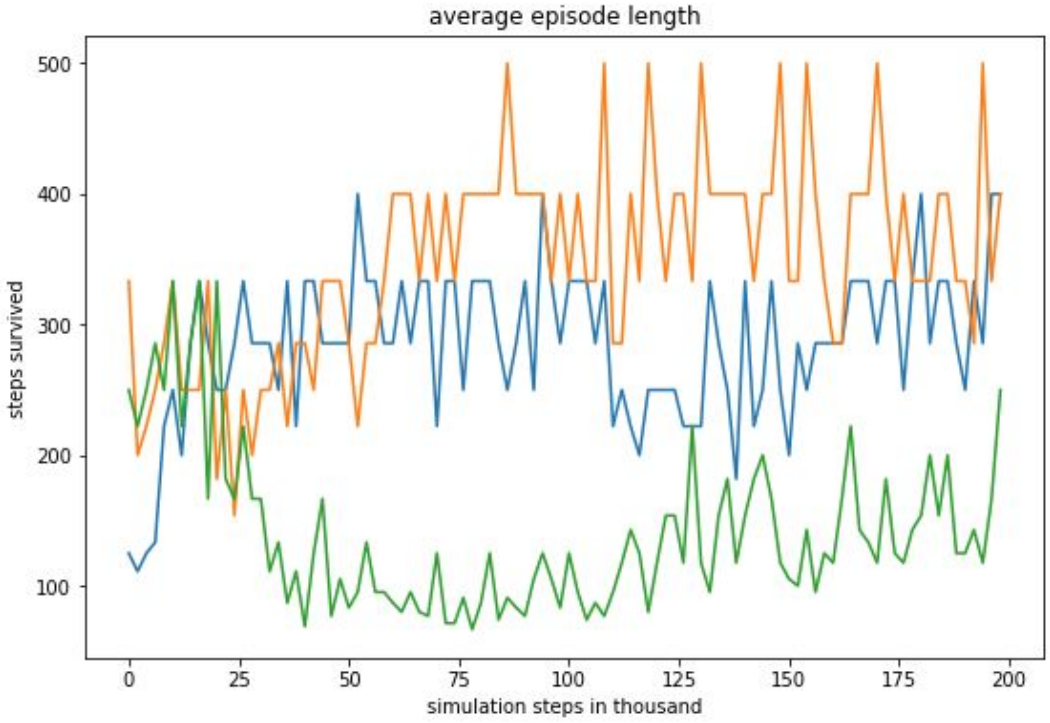
Experiments

- GhostDog model was used
- Easy elevation grid
- Hard elevation grid
- 3 * 100 epochs each terrain (2000 steps/epoch)
- Terrain generator and learning procedure not directly connected (but ideally should’ve been)



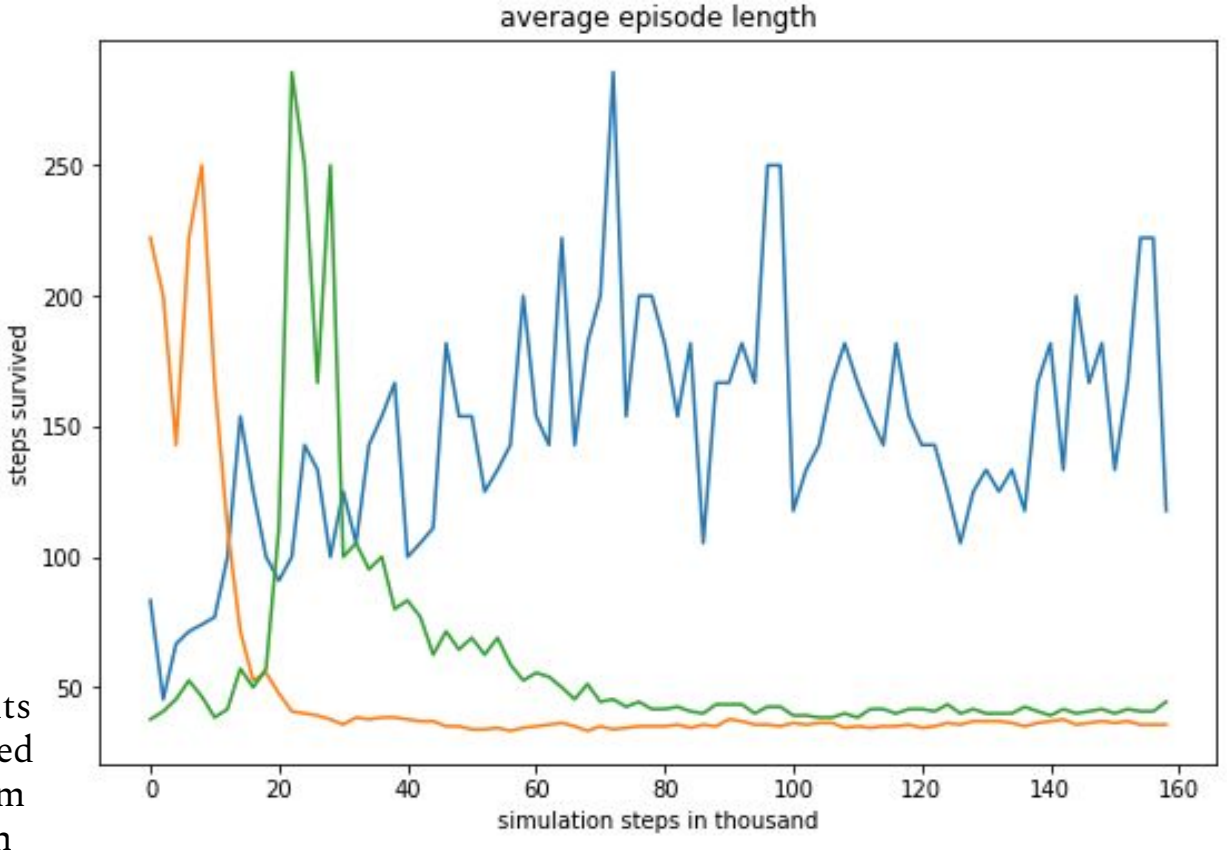
Easy Terrain Results

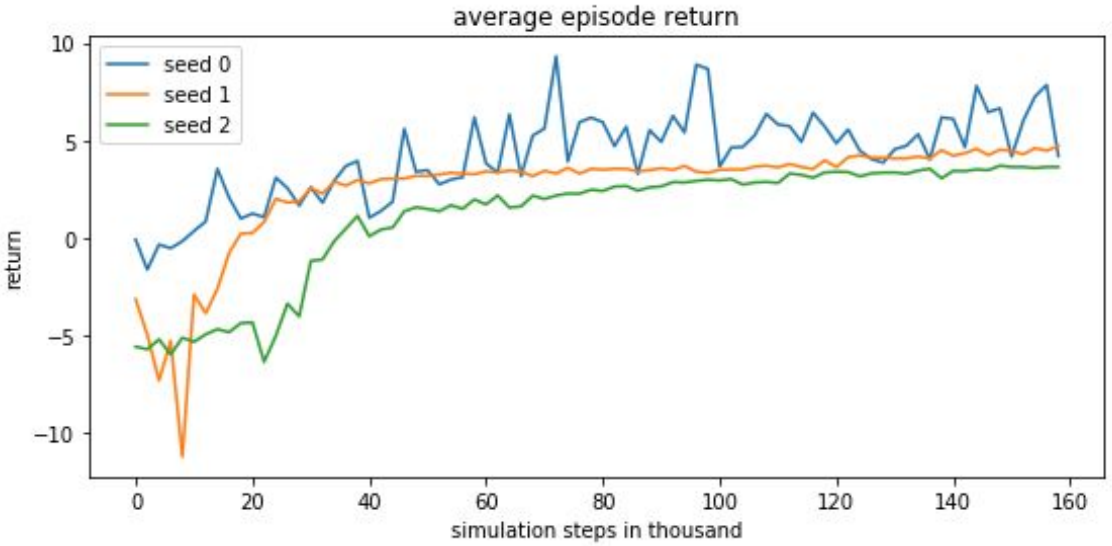
- All three trials learnt a slow gait
- Dragging rear legs with two front legs



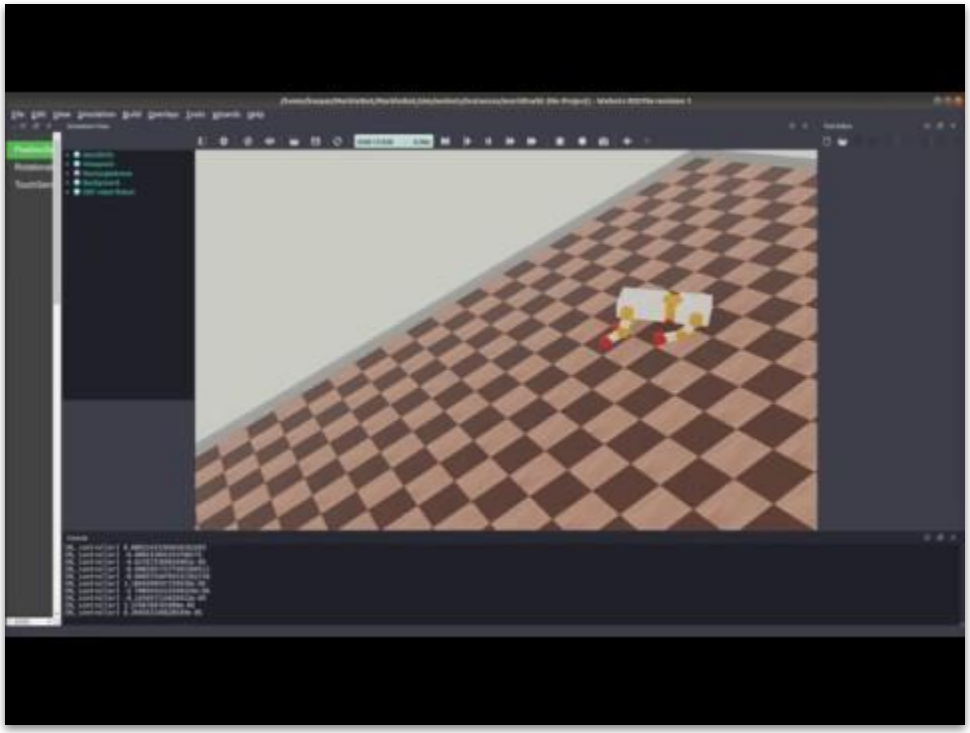
Hard Terrain Results

- 2/3 trials failed to learn a form of locomotion
- Stopped after 80 epochs on 2 failed trials

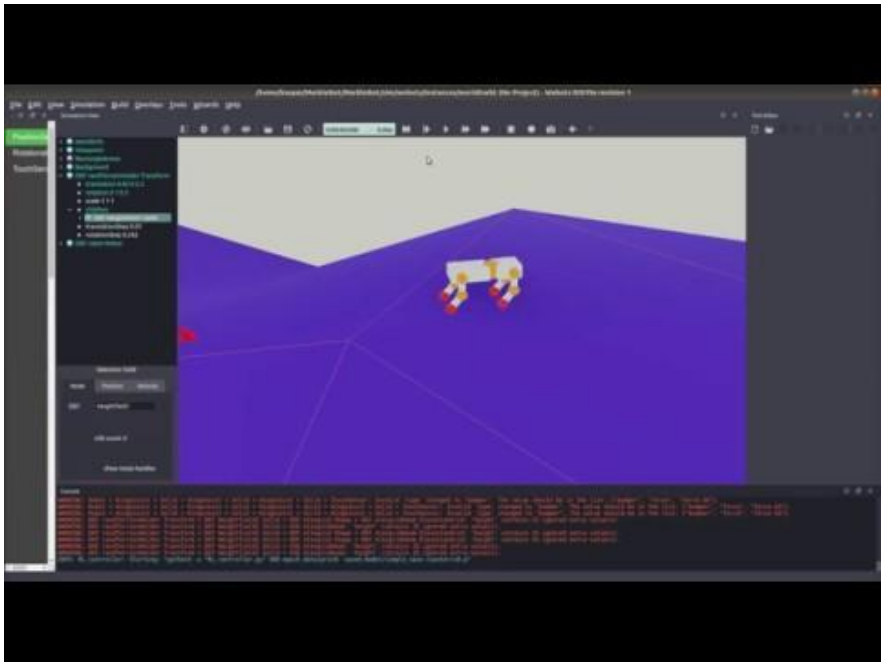




Trial samples



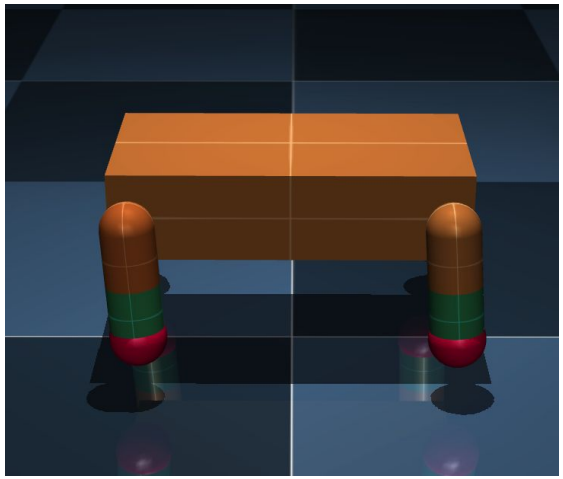
Gait learnt on easy elev

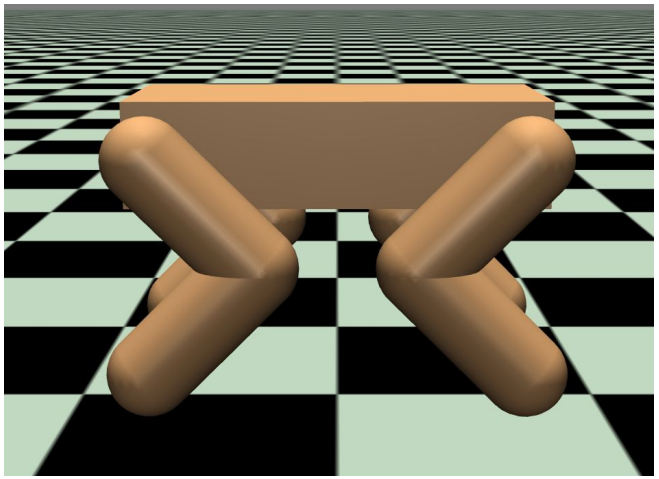


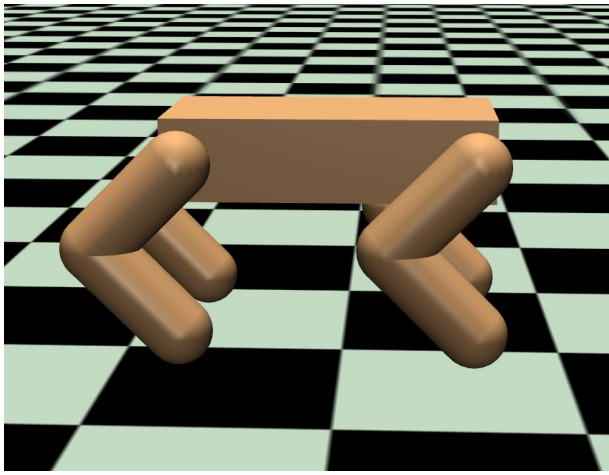
Hard elevation grid trial

Robot Morphology

- 3 morphologies (shown below)
- Rough and Flat Terrain
- Compare performance
- Goal: suggestion of morphology according to terrain





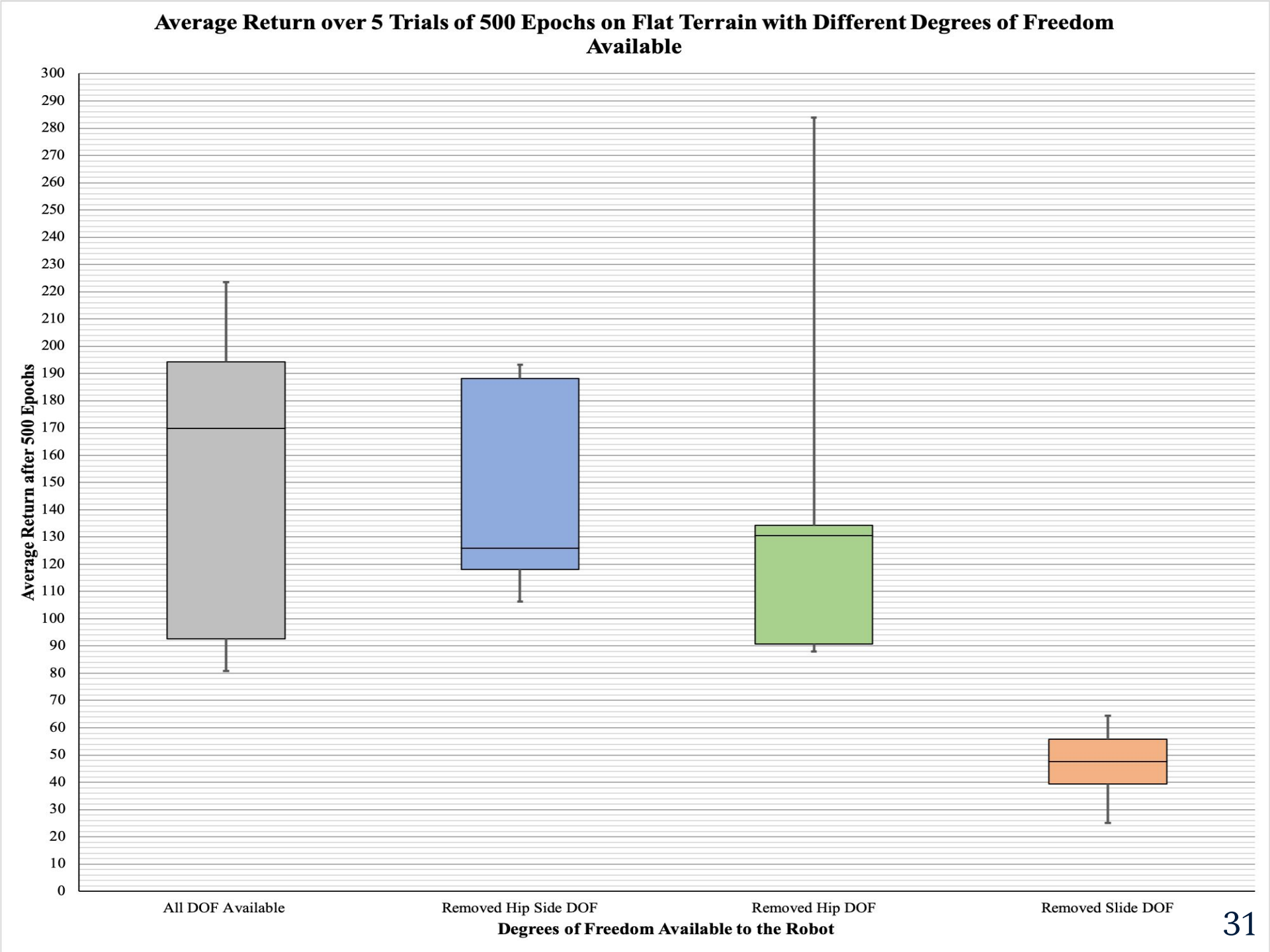


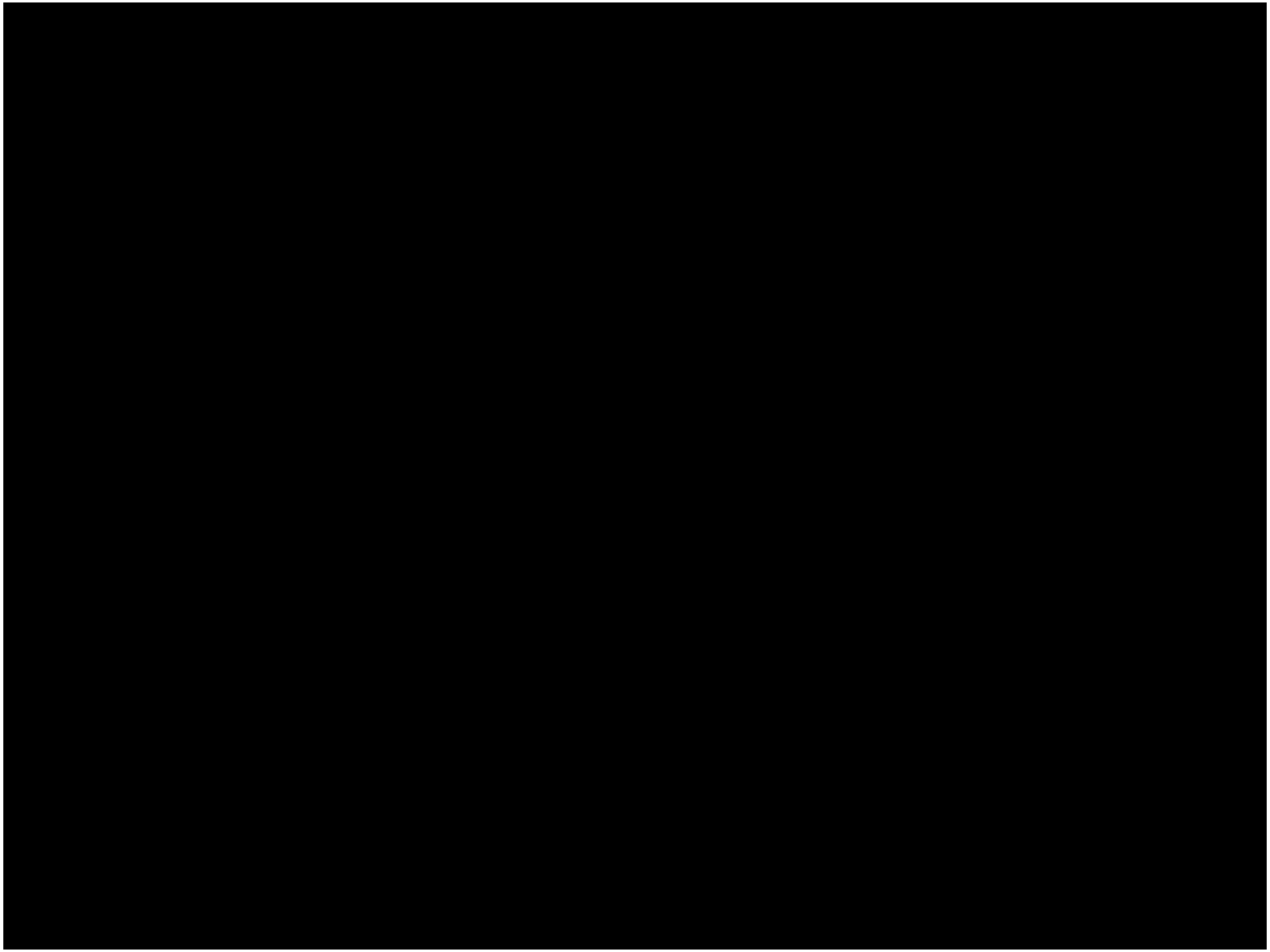
Research Question

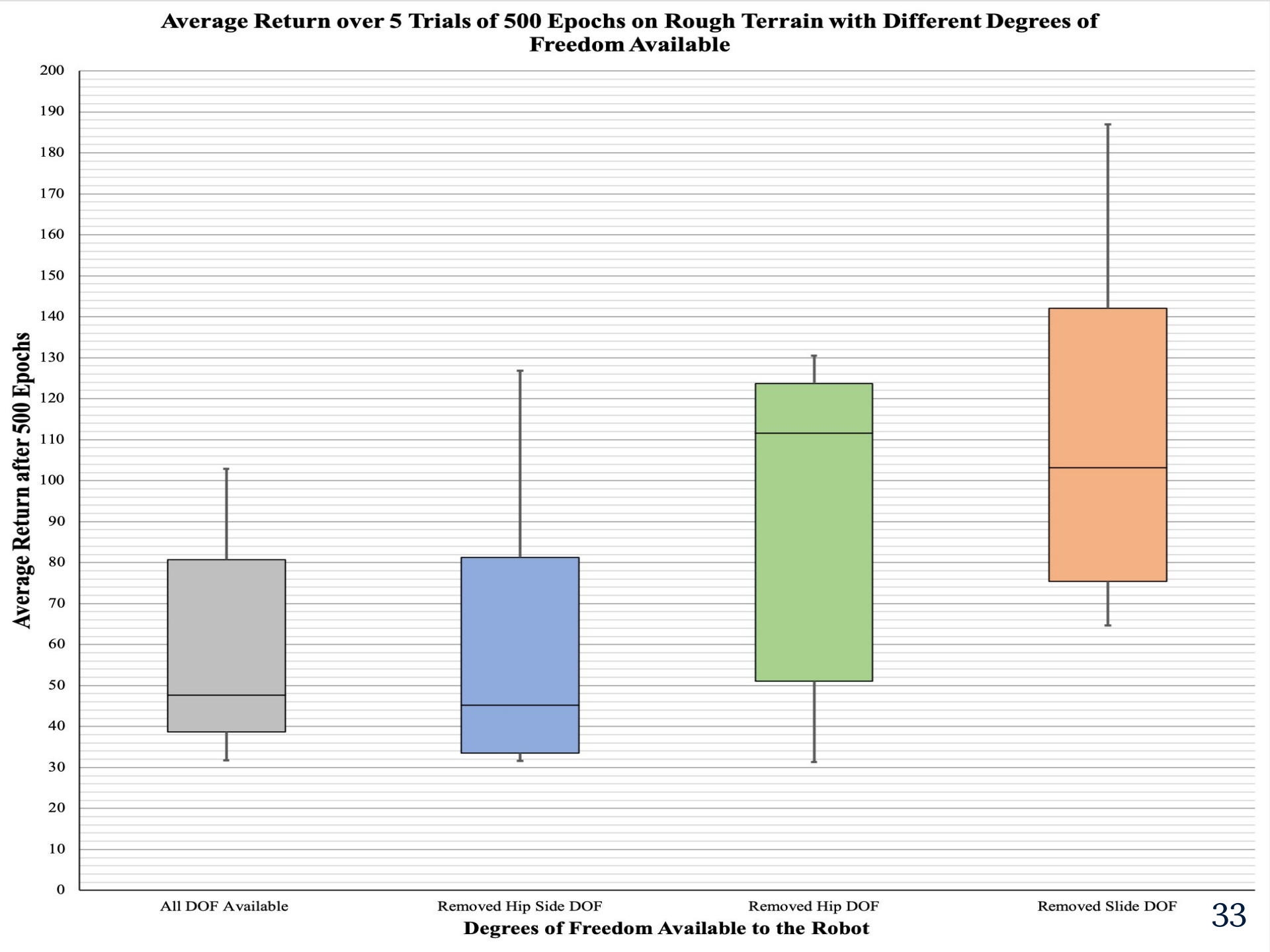
To what extent do hardware alterations such as the full length of the legs, the leg type, and the number of degrees of freedom available change the performance of a quadrupedal robot over both rough and flat terrain?

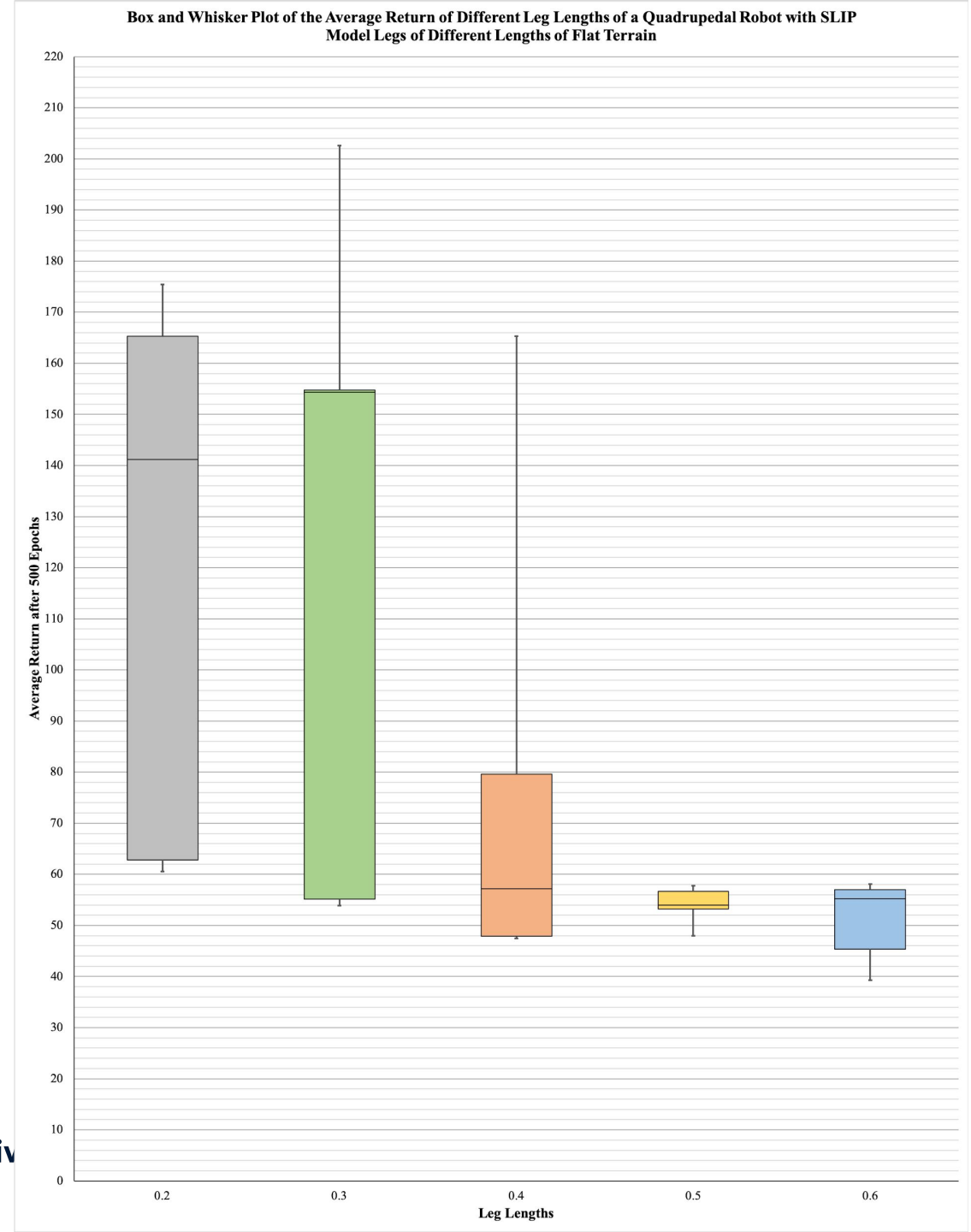
Experiments

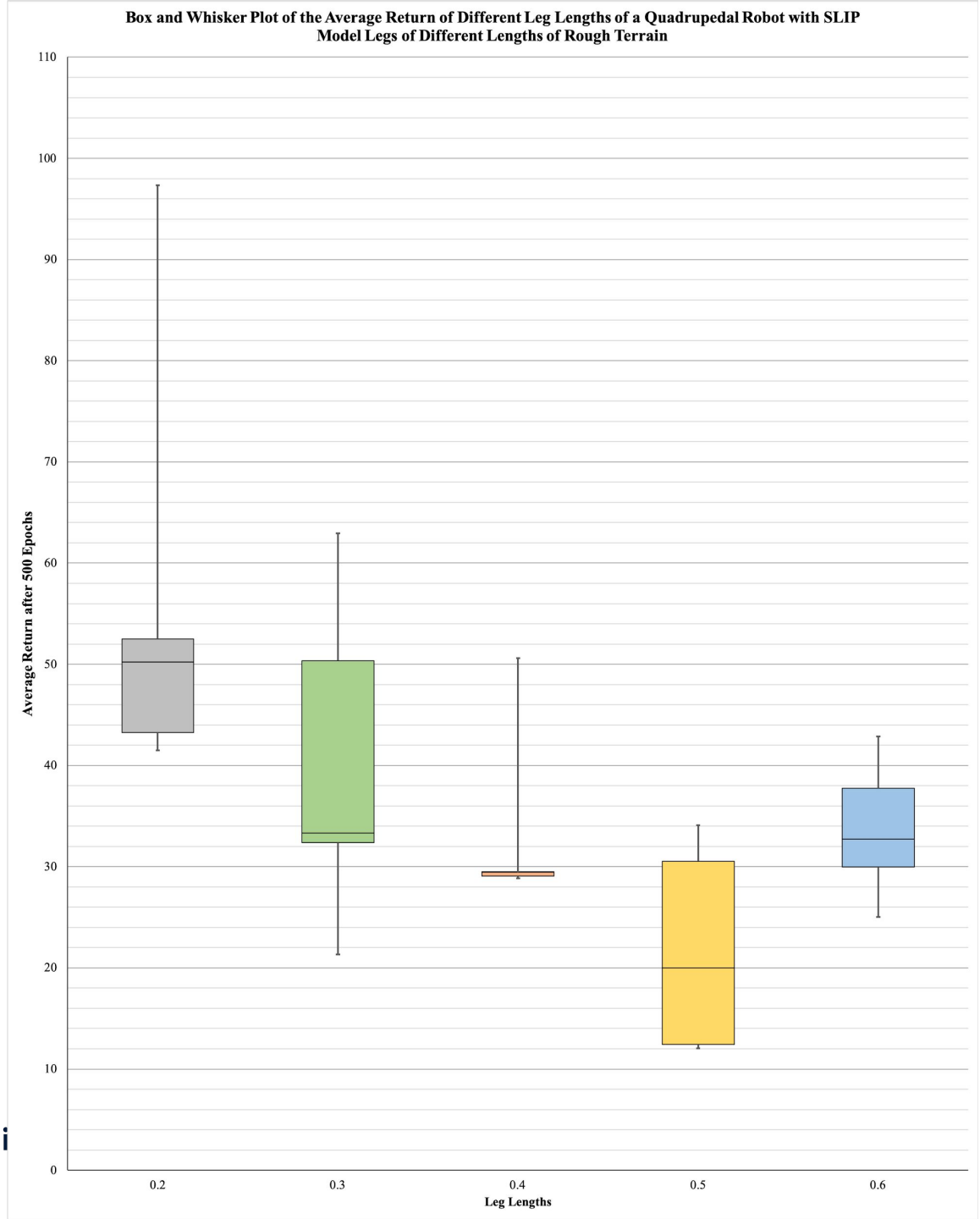
- How does...
  - Leg length of a SLIP model robot
  - Number of degrees of freedom on a SLIP model robot
  - Leg design on a quadrupedal
- Affect performance
- Each experiment:
  - 500 epochs
  - 5 trials

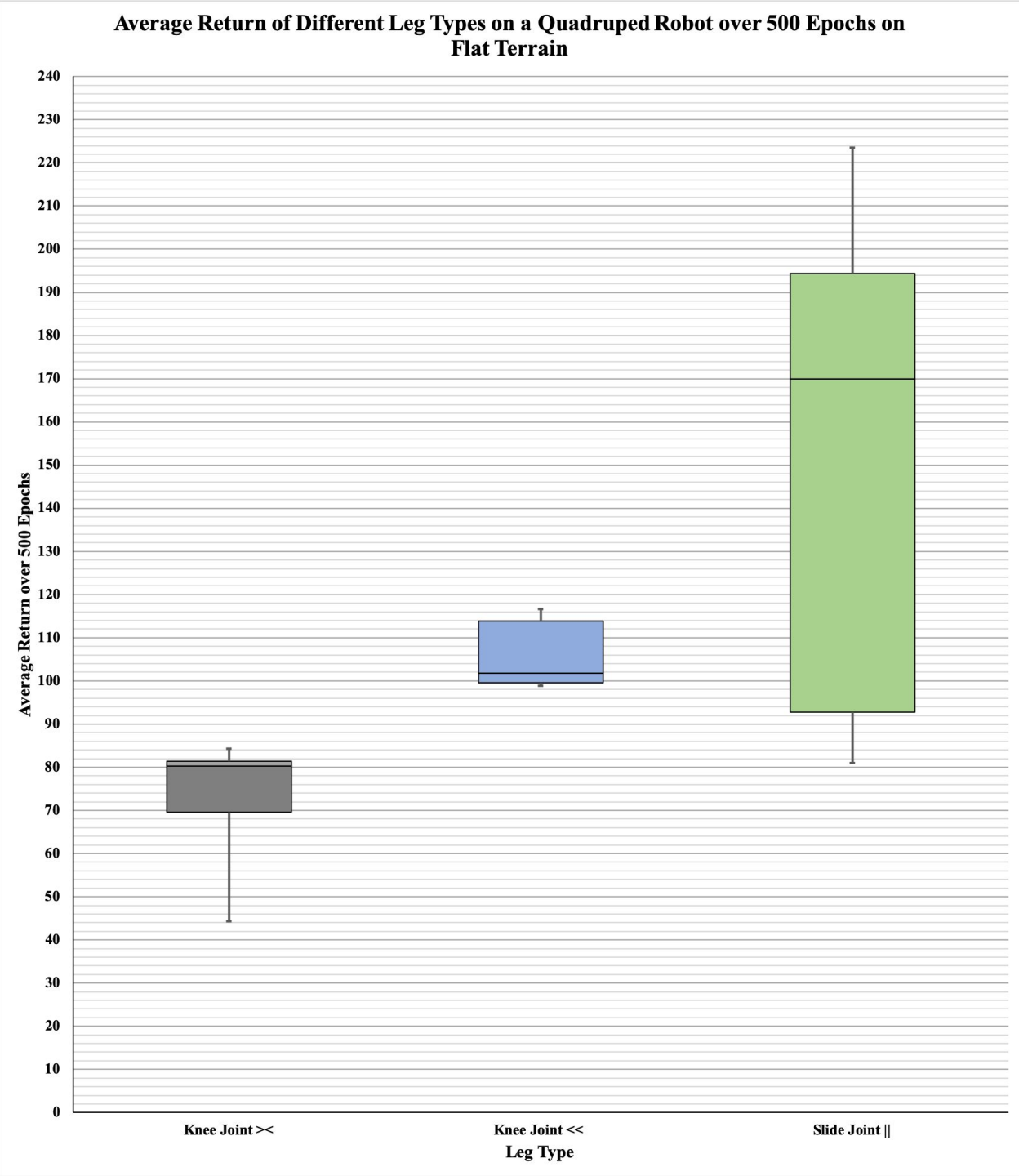

Average Return over 5 Trials of 500 Epochs on Flat Terrain with Different Degrees of Freedom Available
Average Return after 500 Epochs
300
290
280
270
260
250
240
230
220
210
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
All DOF Available
Removed Hip Side DOF
Removed Hip DOF
Removed Slide DOF
Degrees of Freedom Available to the Robot

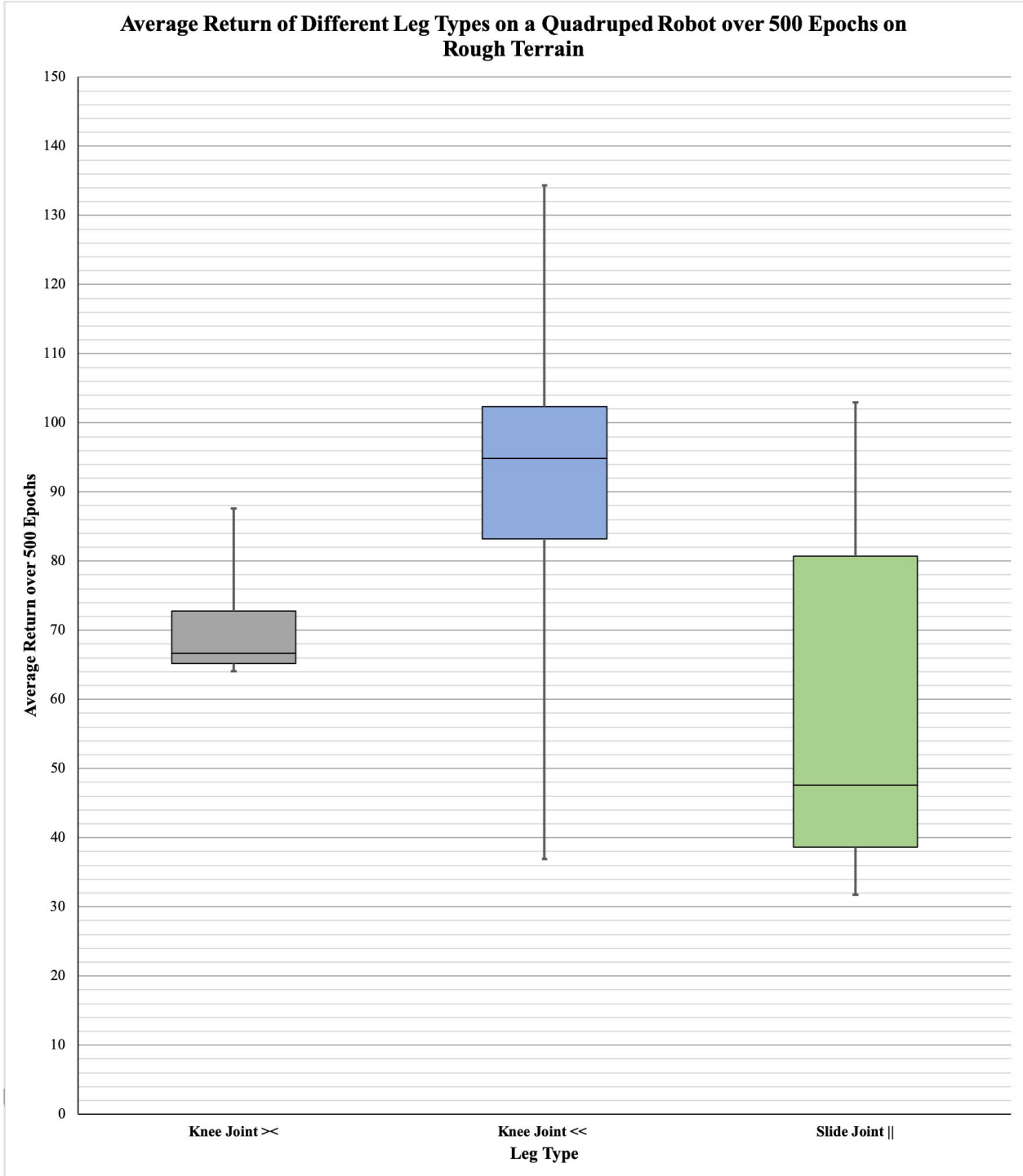


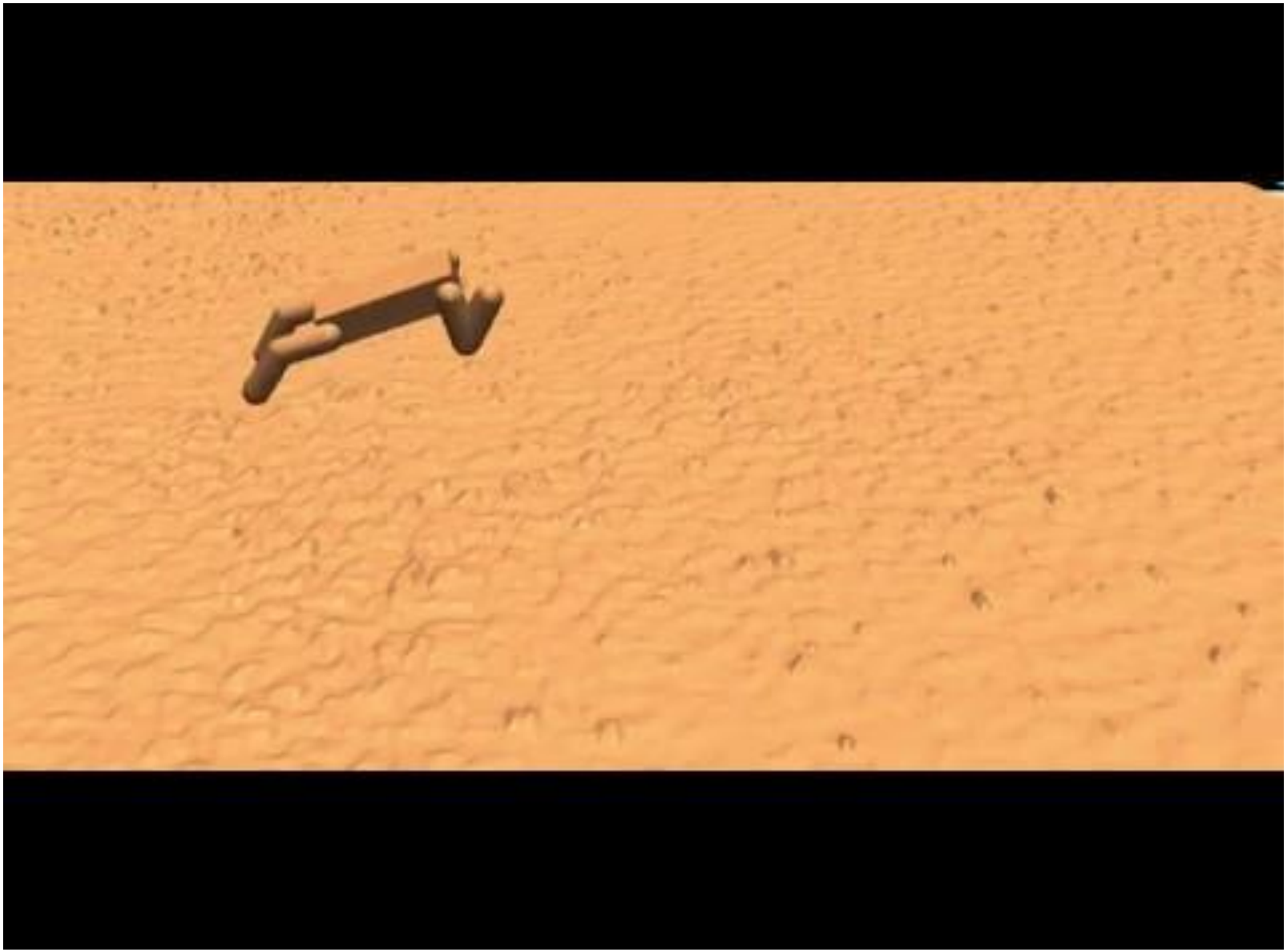

Average Return over 5 Trials of 500 Epochs on Rough Terrain with Different Degrees of Freedom Available
Average Return after 500 Epochs
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
All DOF Available
Removed Hip Side DOF
Removed Hip DOF
Removed Slide DOF
Degrees of Freedom Available to the Robot


Box and Whisker Plot of the Average Return of Different Leg Lengths of a Quadrupedal Robot with SLIP Model Legs of Different Lengths of Flat Terrain
Average Return after 500 Epochs
220
210
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
0.2
0.3
0.4
0.5
0.6
Leg Lengths


Box and Whisker Plot of the Average Return of Different Leg Lengths of a Quadrupedal Robot with SLIP Model Legs of Different Lengths of Rough Terrain
Average Return after 500 Epochs
110
100
90
80
70
60
50
40
30
20
10
0
0.2
0.3
0.4
0.5
0.6
Leg Lengths


Average Return of Different Leg Types on a Quadruped Robot over 500 Epochs on Flat Terrain
Average Return over 500 Epochs
240
230
220
210
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
Knee Joint ><
Knee Joint <<
Slide Joint ||
Leg Type


Average Return of Different Leg Types on a Quadruped Robot over 500 Epochs on Rough Terrain
Average Return over 500 Epochs
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
Knee Joint ><
Knee Joint <<
Slide Joint ||
Leg Type



Morphology Conclusions

- Rough Terrain:
  - **Leg design:** knee joints facing in same direction
  - **Leg length:** 0.2; ratio leg length to torso length: 1:2
  - **Degrees of freedom:** hip degree removed
- Flat Terrain:
  - **Leg design:** SLIP model straight legs
  - **Leg length:** 0.25; ratio leg length to torso length 5:8
  - **Degrees of freedom:** all

Conclusion

Our project:

- A good algorithm isn't enough
- One algorithm performs very differently in distinct situations
- Many changes need to be made

References

[[1]Peng, Xue Bin, and Michiel van de Panne. "Learning locomotion skills using deeprl: Does the choice of action space matter?." *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2017.
[2] WHITEHEAD, Steven D. A Complexity Analysis of Cooperative Mechanisms in Reinforcement Learning. In: *AAAI*. 1991. p. 607-613.
APA