New virtual rodent environments and dimensionality reduction: Improved computational tractability for autonomous navigation

Preston Fu[1, 2] James Queeney[2] Ioannis Ch. Paschalidis[2]

[1]Saratoga High School, 20300 Herriman Ave, Saratoga, CA 95070
[2]Division of Systems Engineering, Boston University, Boston, MA 02215

Motivation and Contributions

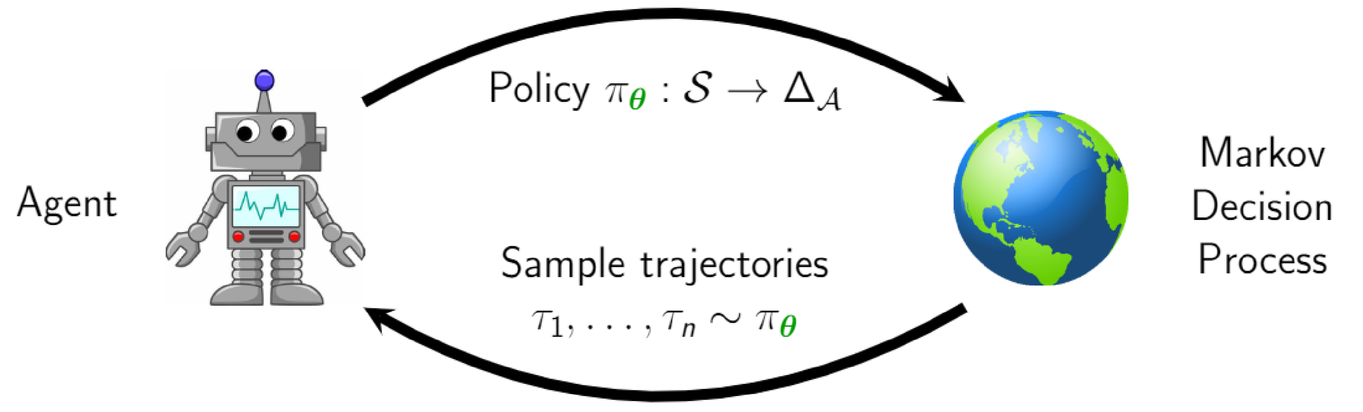
Motivation: The capabilities envisioned for next-generation autonomous vehicles — learning on-the-fly and adapting to novel environments — are already exhibited by biological organisms such as rodents. However, training the existing virtual rodent [1] on complex tasks remains fairly computationally intensive.

Contributions: We develop simplifications of the existing rodent model and its tasks to increase the efficiency of its training.

- We construct simplified environments in which a virtual rodent is rewarded for accomplishing the goals of locomotion or collecting food items which appear on regular time intervals.
- We further analyze task simulation data and identify the subset of virtual actuators that are most crucial to accomplishing a task.

Reinforcement Learning (RL) Background

Reinforcement Learning: Agent interacts with an environment using a policy π_θ parameterized by $\theta \in \mathbb{R}^d$, and receives sample trajectories $\tau_1, \dots, \tau_n \sim \pi_\theta$.



Goal: Find policy π_θ maximizing expected total discounted reward:

$$\mathop{\mathbb{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right].$$

Policy Gradient:

$$\max_\theta \; \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] \qquad \hat{g} = \hat{\mathbb{E}}_t\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$$

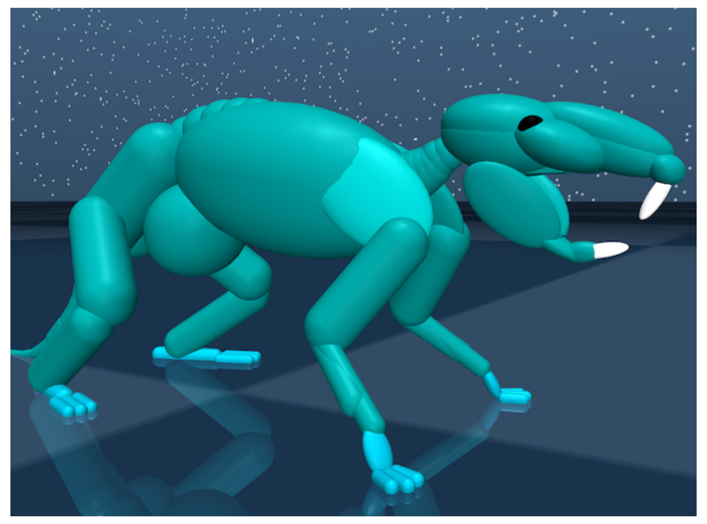
Proximal Policy Optimization (PPO) Algorithm: [2] Avoid excessive policy updates:

$$\text{clip} \quad \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \quad \text{to } [1-\varepsilon, 1+\varepsilon]$$

Deepmind's Virtual Rodent [3]

- Based on laboratory measurements
- 38 controllable degrees of freedom
- 158 observations: proprioceptive information (advanced kinematics, joint angles, forces) and raw egocentric RGB-camera input

Goal: Maintain the rodent model, but lower those numbers by presetting values.



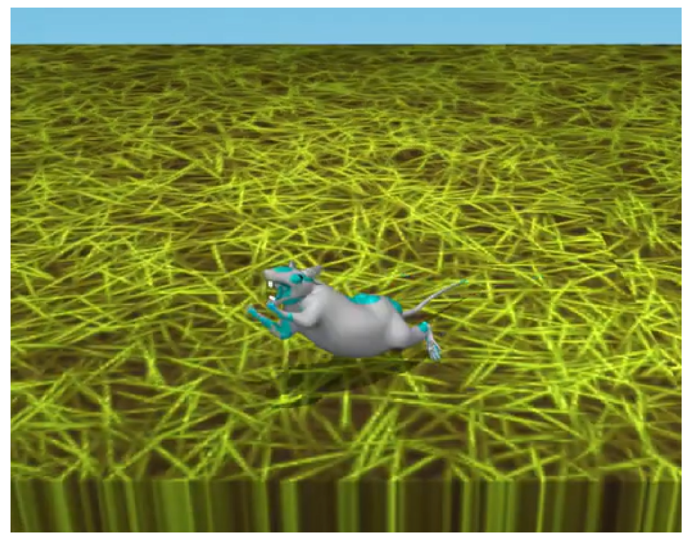
Virtual rodent with collision geometry

New virtual rodent environments

Existing environments: Jumping along a platform with gaps; searching for randomized targets in randomized maze.

Locomotion

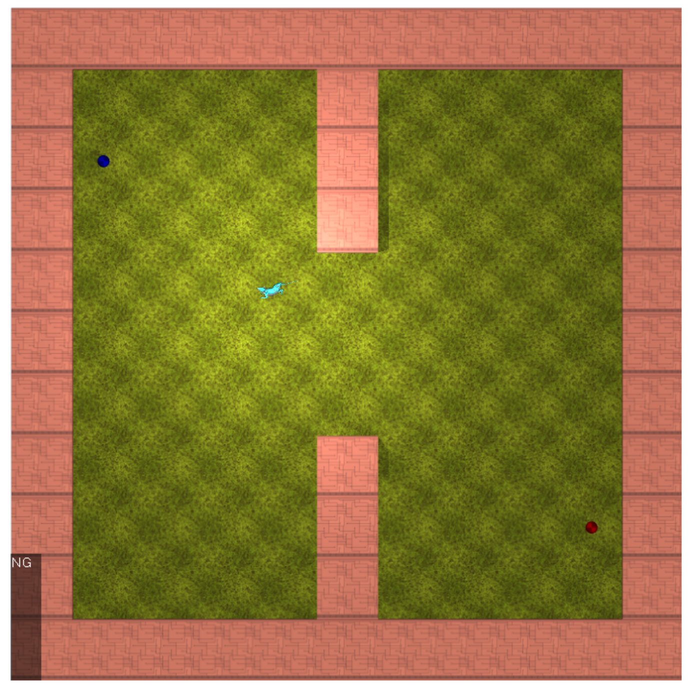
- Rodent rewarded as it approaches a given velocity
- As velocity is varied, rodent is trained to perform particular movements (i.e., running vs. walking)
- Does not use egocentric camera input
- **Method:** Adapt existing rodent environment so that task does not require jumping over platform gaps



https://youtu.be/sYFOoeicrnQ

Timed food collection

- Two-room layout fixed, specified by user-entered parameters
- Rodent rewarded for collecting food items
- Food items spawned and despawned at specific locations and times
- **Method:** Modified targets to enable manual activation; additional function to update the target status at fixed times



Dimensionality reduction for action space

Goal: Reduce the number of degrees of freedom of the rodent while maintaining a reasonable ability to accomplish tasks (i.e. locomotion and collecting food items).

Method:

Partial training

Run aggressive ($\varepsilon = 0.3$), short ($T = 50$, size $= 10^5$) simulation to train the rodent partially

- Final raw reward ≈ 0.51

Number	Action	Reward
0	lumbar_extend	0.2659841
1	lumbar_bend	0.01232401
2	lumbar_twist	0.037079815
3	cervical_extend	0.030297989
4	cervical_bend	0.031912517
5	cervical_twist	0.08995138
6	caudal_extend	0.03631608

Inspired by Principal Component Analysis (PCA), determine the actions that have the largest variance across 100 simulations

- Red = least variance, green = greatest

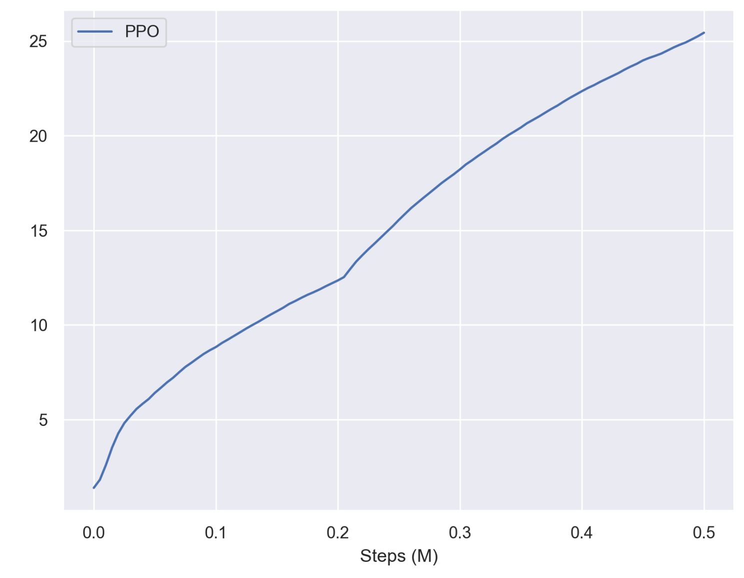
1: lumbar bend
3: cervical extend
4: cervical bend
⋮

Remove actions with least variance by replacing with their empirical averages. Repeat, using lowest-variance action at each step, while the reward remains above given threshold.

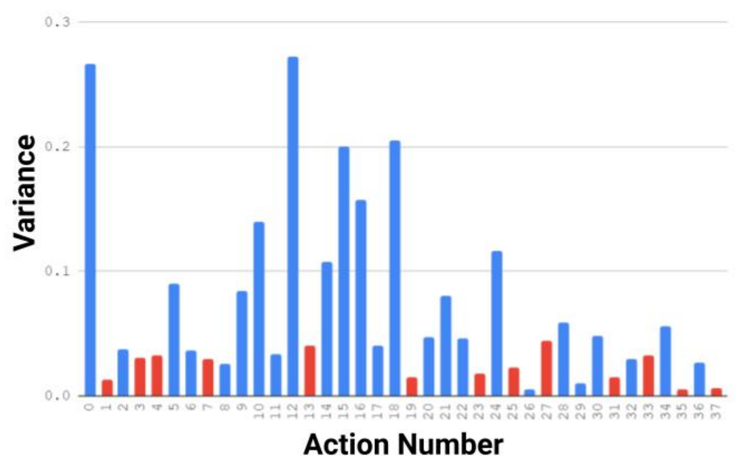
Results

Setup: We use the procedure for dimensionality reduction for the action space. In particular, we do not retrain the rodent using a modified policy; we only perform a post-evaluation on top of the learned policy.

- A reduction to **24** degrees of freedom (14 removed) only decreased the reward slightly to **0.45** (vs. original 0.51).



Tests are based on the policy learned at the end of this locomotion learning progress curve, PPO ($T = 50$, size $= 5 \cdot 10^5$).



Red = removed actions, generally had lowest variance
Blue = remaining, removing had significant adverse effect on performance

Discussion and Future Work

Takeaway: Our simplifications marginally reduced performance while significantly decreasing the space size (hence runtime). Our work allows for faster prototyping of novel RL algorithms and efficient testing of neuroscience hypotheses.

Immediate goals: Formal testing of simplifications:

- Training the rodent on the locomotion task using our modified policy, excluding the aforementioned 14 degrees of freedom.
- Use $T = 10^3$ and $\varepsilon = 0.2$ while maintaining size — make learning more accurate.
- Apply these same principles to observation space and timed food collection task.

Long-term goals: We seek to utilize methods in feature learning (beyond PCA, etc.) to find a more efficient and optimal method of simplification.

References

[1] Josh Merel, Diego Aldarondo, Jesse Marshall, Yuval Tassa, Greg Wayne, and Bence Ölveczky. Deep neuroethology of a virtual rodent. arXiv preprint, 2019. arXiv:1911.09451.

[2] John Schulman, Prafulla Dhariwal Filip Wolski, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint, 2017. arXiv:1707.06347.

[3] Yuval Tassa, Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, and Nicolas Heess. dm control: Software and tasks for continuous control, 2020.

Acknowledgements

I would like to thank my mentors Jimmy Queeney and Professor Yannis Paschalidis for their continued support throughout this process. I would also like to thank BU RISE for providing this exciting research opportunity to me.