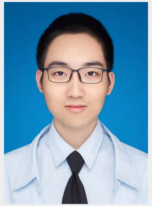
NEURAL INFORMATION PROCESSING SYSTEMS

NeurIPS’22 Competition Track Winner Presentation

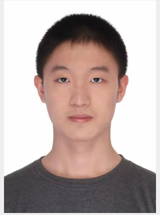
MyoChallenge: Die Rotation

Speaker: Yiran Geng, Boshi An

Yiran Geng*



Boshi An*

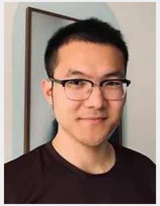


Yifan Zhong*

Jiaming Ji

Yuanpei Chen

Hao Dong



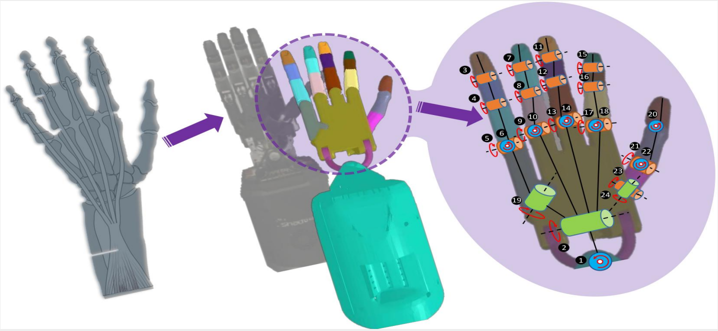
Yaodong Yang

Contents

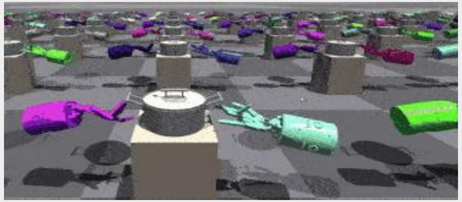
Contents

Our previous attempts on Dexterous Hands

NEURAL INFORMATION PROCESSING SYSTEMS

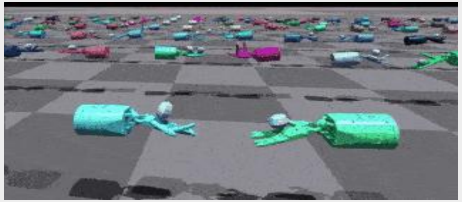


Shadow Hands Degree-Of-Freedom (DOF) [2]

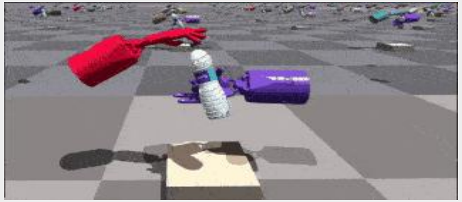


Lift Pot [2]

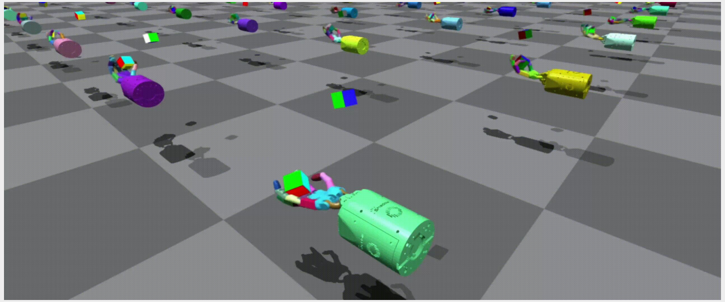
Open Door [2]



Hand Over [2]



Open Bottle Cap [2]



Die Rotation [1]

Swing Cup [2]

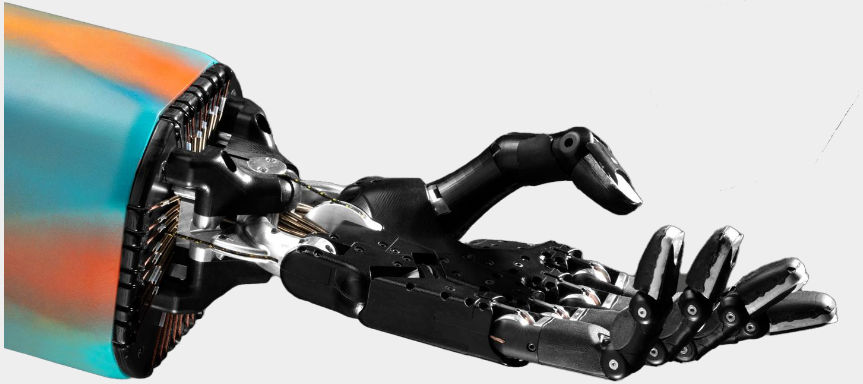
Open Scissors [2]

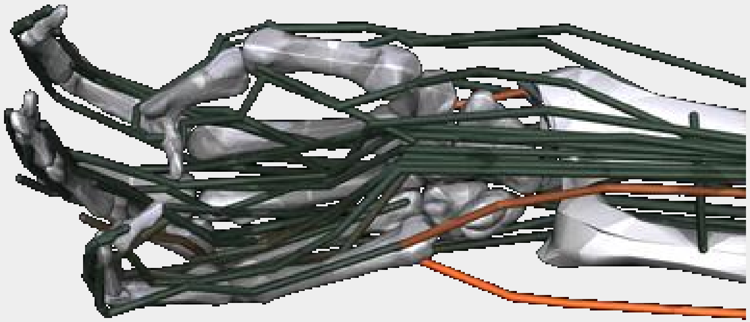
[1] Makoviychuk, Viktor, et al. "Isaac gym: High performance gpu-based physics simulation for robot learning." *arXiv preprint arXiv:2108.10470* (2021).
[2] Chen, Yuanpei, et al. "Towards human-level bimanual dexterous manipulation with reinforcement learning." *arXiv preprint arXiv:2206.08686* (2022).

Difficulties with MyoHand

NEURAL INFORMATION PROCESSING SYSTEMS

We attribute the difficulties to the difference in drive mode.

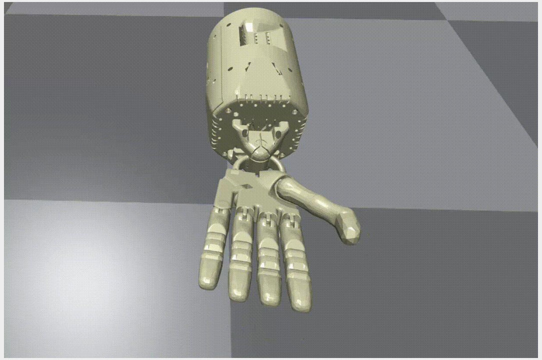


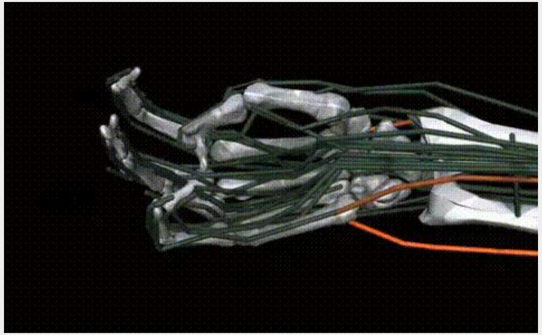


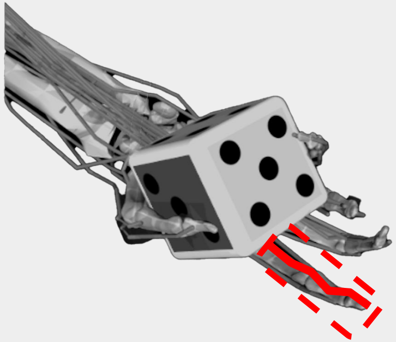
Images are from https://openai.com/blog/learning-dexterity/ and https://sites.google.com/view/myosuite

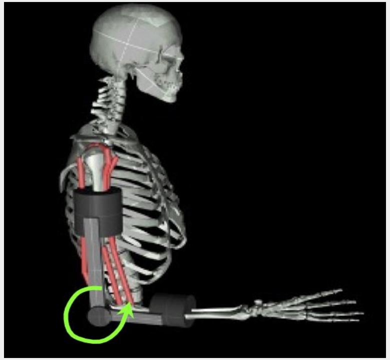
Difficulties with MyoHand

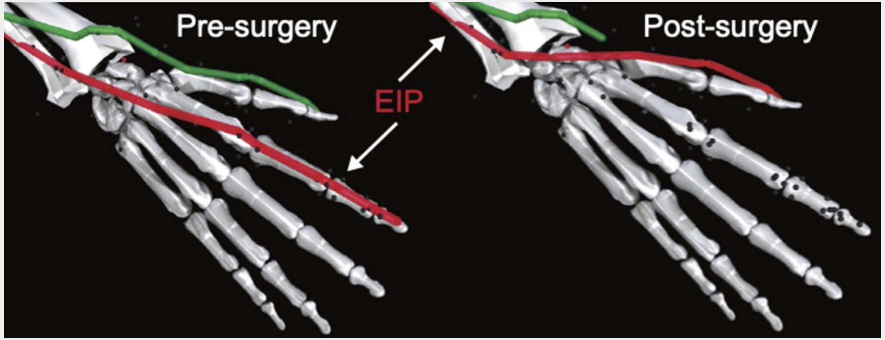
NEURAL INFORMATION PROCESSING SYSTEMS


Move a joint


Apply a force to a muscle


Muscle fatigue


Exoskeleton assistance



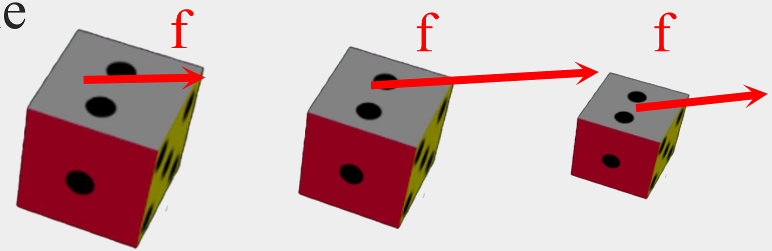
Tendon transfer

Difficulties with MyoHand

- Random Initialization
- Random Goal

Task	Goal	Environment initialization	Evaluation
Die *Phase1*	$Goal_{pos} \sim (-.010, .010)_{xyz}$ $Goal_{rot} \sim (-1.57, 1.57)_{xyz}$	$init_{hand} : palm\ up$ $init_{die} : over\ palm$	$score = (\sum_{t=T-5}^{T} success[t]) > 0$
Die *Phase2*	$Goal_{pos} \sim (-.020, .020)_{xyz}$ $Goal_{rot} \sim (-3.14, 3.14)_{xyz}$	$init_{hand} : palm\ up + noise$ $init_{die} : over\ palm + noise$	$effort = \sum_{t=0}^{T} act_{mag}[t]/T$

- Radom physical properties of the die
  - Random die size
  - Random die friction



Contents

- Introduction
- Methods
  - Reward Shaping
  - Curriculum Learning
  - Multi-target Training
- Limitation
- Future works

Our Method

NEURAL INFORMATION PROCESSING SYSTEMS

Reinforcement Learning (NPG/PPO)

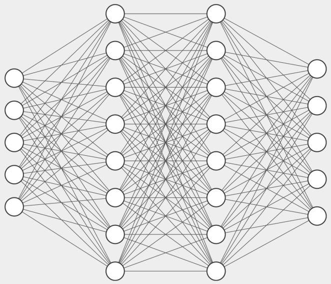
Reward Shaping

Curriculum Learning

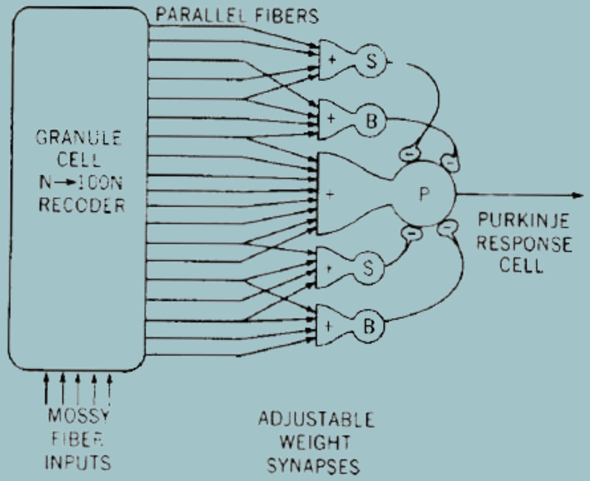
Multi-target Training

Reinforcement Learning Framework

Simplest models, but excellent performance.
Policy network is a MLP with hidden size 64.
Trained with natural policy gradient,
on a 32-core machine.



James Albus' Model



Human Cerebellum consists of a structure similar to MLP.

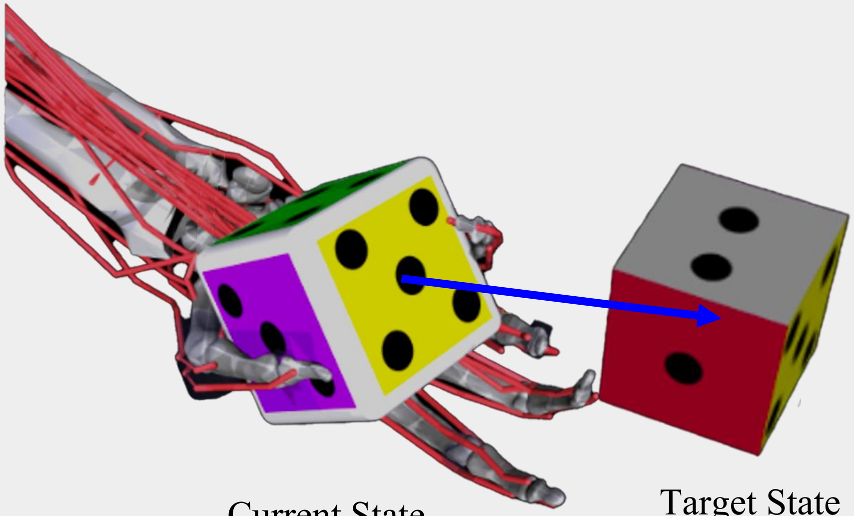
Image of James Albus' Model is from https://en.wikipedia.org/wiki/James_S._Albus

Reward Shaping

The most powerful tool for us to improve performance is reward shaping

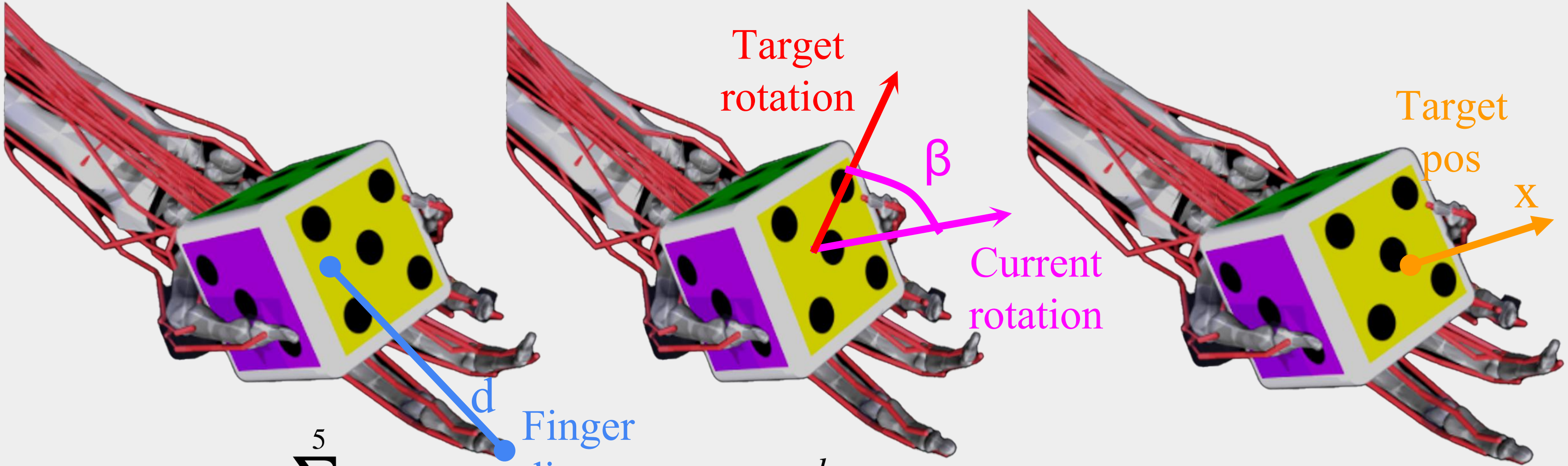
Criterias should be met:

1. Distance within a range
2. Rotational error within a range
3. ≥5 successes in a trial



Reward Shaping

The most powerful tool for us to improve performance is reward shaping



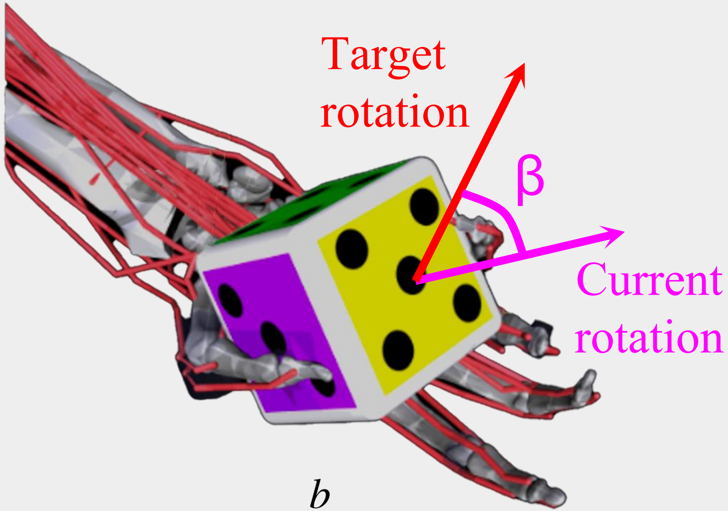
$$r_d = -\sum_{i=1}^{5} d_i$$

$$r_r = \frac{b}{\boldsymbol{\beta}^2 + b}$$

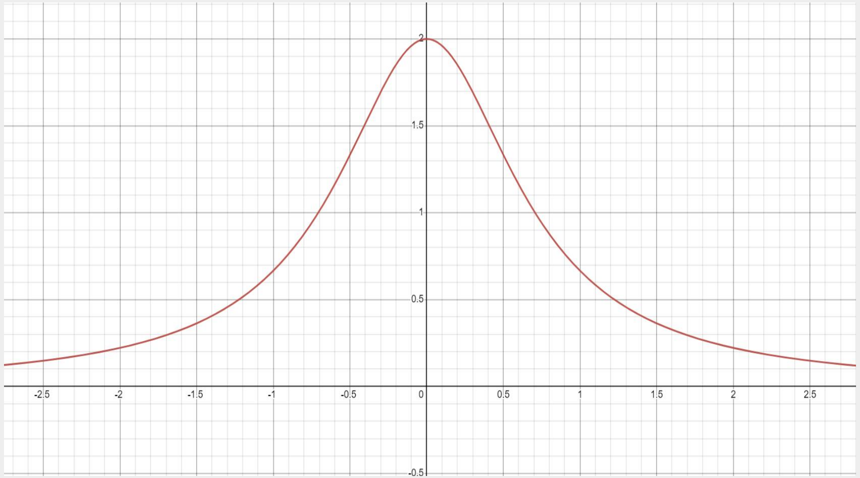
$$r_x = -x$$

Reward Shaping

The most difficult criteria to met is the rotational error limit.



$$r_r = \frac{b}{\boldsymbol{\beta}^2 + b}$$



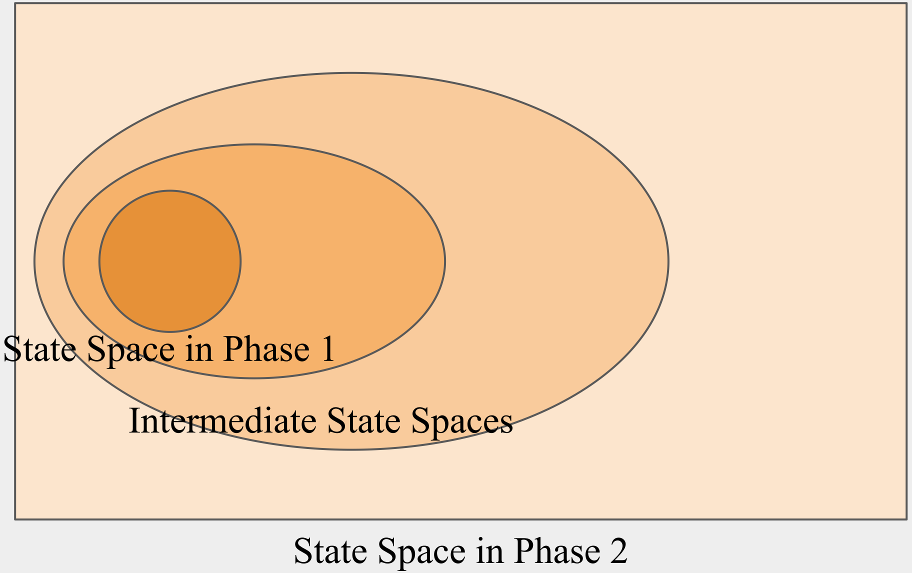
Curriculum Learning

NEURAL INFORMATION PROCESSING SYSTEMS

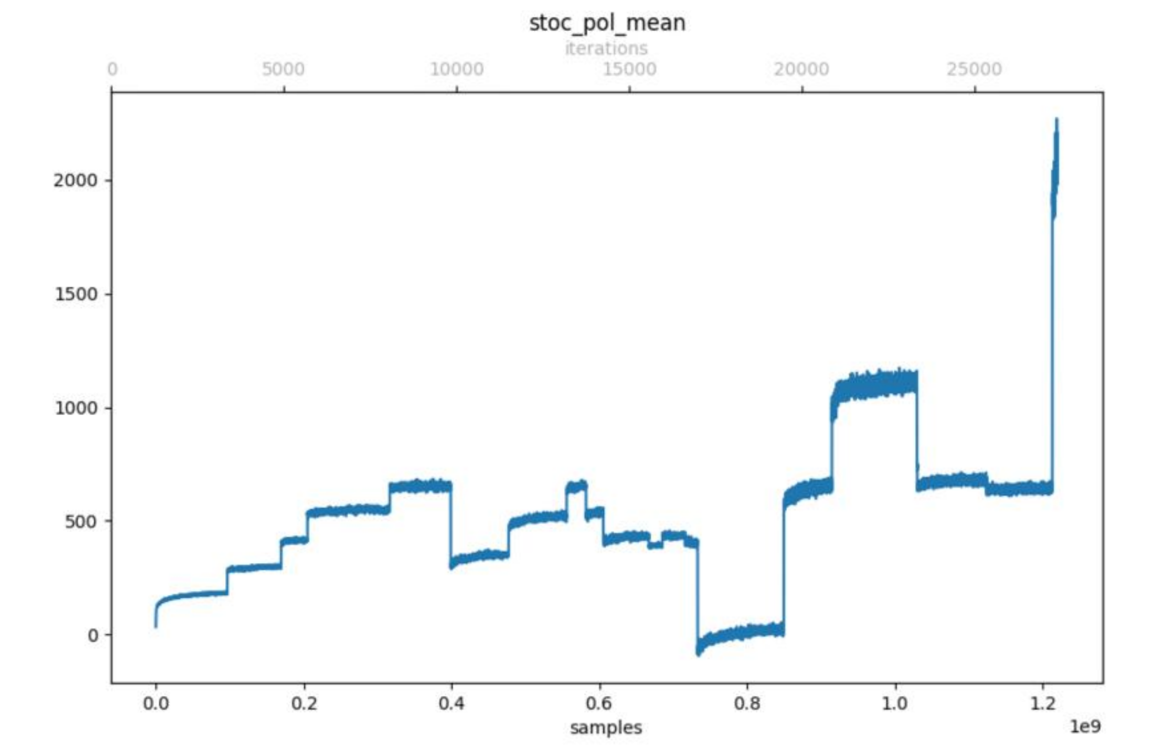
Our discoveries:

1. Generally speaking, the difficulty of a task is positively related to the size of state (or observation) space.
2. Constrain to less randomization and shorter episode length result in faster training but lower performance in original task.
3. When the learning rate of value function is much greater than policy network, adjusting reward function in the training process won’t cause the policy to fail.

Curriculum Learning



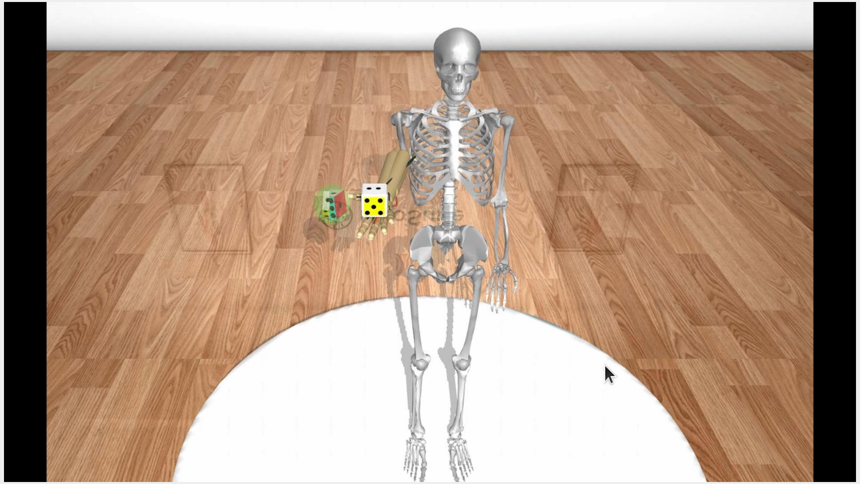
Curriculum Learning



Curriculum Learning

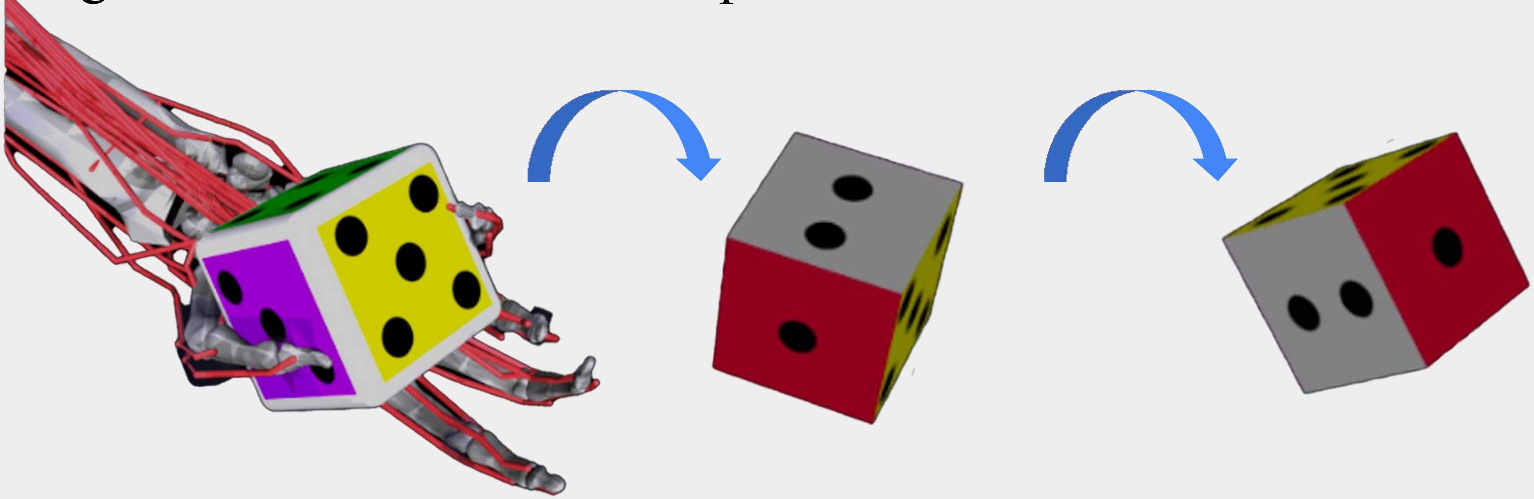
NEURAL INFORMATION PROCESSING SYSTEMS

It is still challenging for a single policy to handle rotations greater than 90 degrees, even with our curriculum learning technique. The reason is that reorientation within 90 degrees can be done with a single move, while a 180-degree rotation needs at least two moves



Multi-target Training

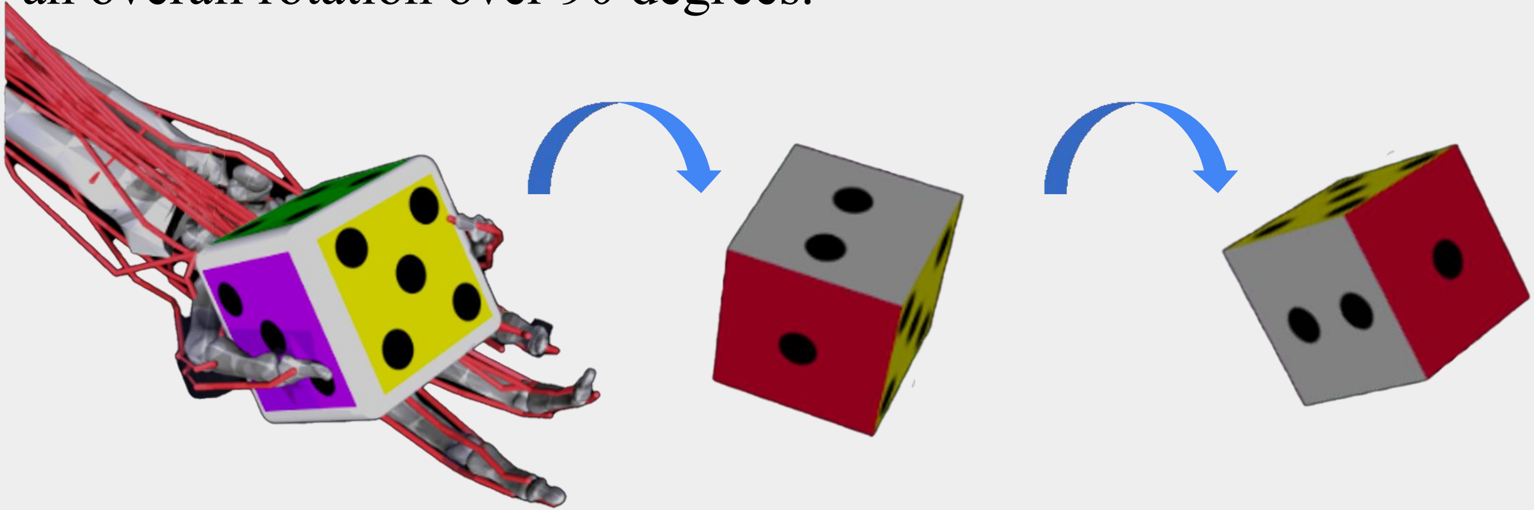
One way to encourage the agent to learn to manipulate the object with multiple moves is to adopt multi-target training: the target of reorientation will be updated after each success.



Original State Original Target Target after Success

Multi-target Training

Even if the agent can only handle reorientation within 90 degrees, by reaching consecutive targets, the agent can perform an overall rotation over 90 degrees.



Original State   Original Target   Target after Success

Multi-target Training

NEURAL INFORMATION PROCESSING SYSTEMS

However, our agent did not perform as expected during the multi-target training.

In phase 2, we have to focus on reaching a high success rate for rotations within 90 degrees (same as phase 1), and give up on large rotations.

Contents

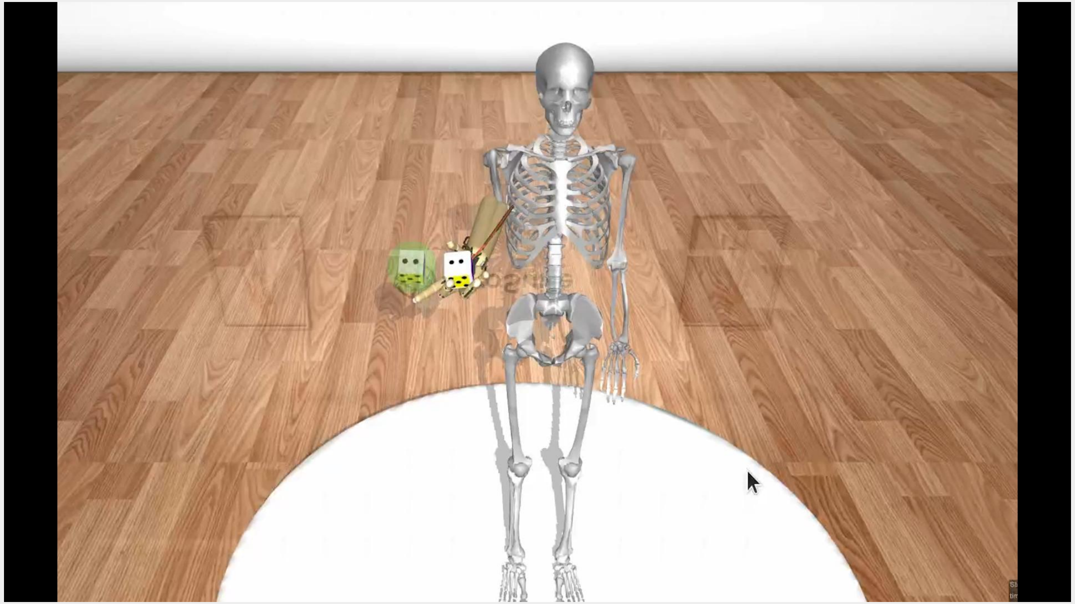
- Introduction
- Methods
  - Reward Shaping
  - Curriculum Learning
  - Multi-target Training
- Limitation
- Future works

NEURAL INFORMATION PROCESSING SYSTEMS

Limitations

NEURAL INFORMATION PROCESSING SYSTEMS

Our method failed to generate policies that is able to finish large rotations by multiple movements.

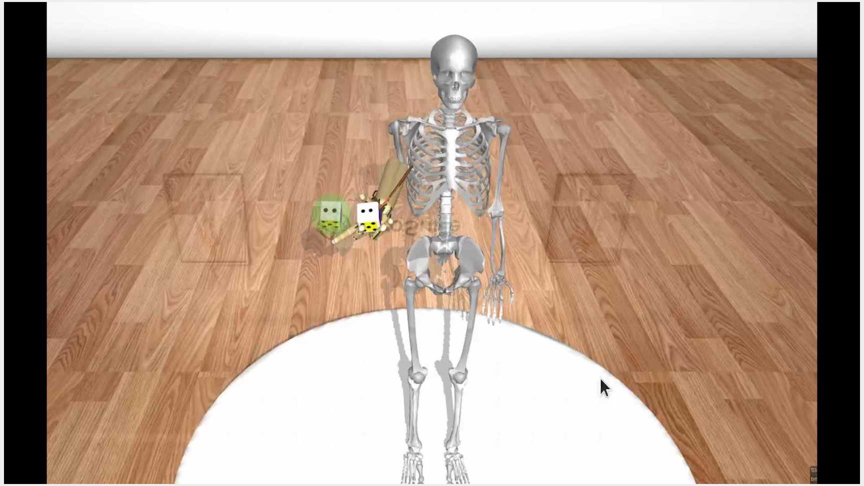


Contents

NEURAL INFORMATION PROCESSING SYSTEMS

Future Works

Our method failed to generate policies that is able to finish large rotations by multiple movements, which is a mismatched phenomenon comparing to animal behaviors.



Our policy only capable of proposing single movement in different conditions

Decerebrated cat exhibit multiple locomotion modes in different conditions

Future Works

NEURAL INFORMATION PROCESSING SYSTEMS

- Automatically merging different policy networks for different rotation ranges.
- Increasing the number of parallel environments.
- More improvements are yet to be studied!

Thank you!

Github Repo: https://github.com/PKU-MARL/MyoChallenge

Email: boshi_an@stu.pku.edu.cn and gyr@stu.pku.edu.cn

Websites: Yiran Geng: https://gengyiran.github.io/

Yaodong Yang: https://www.yangyaodong.com/