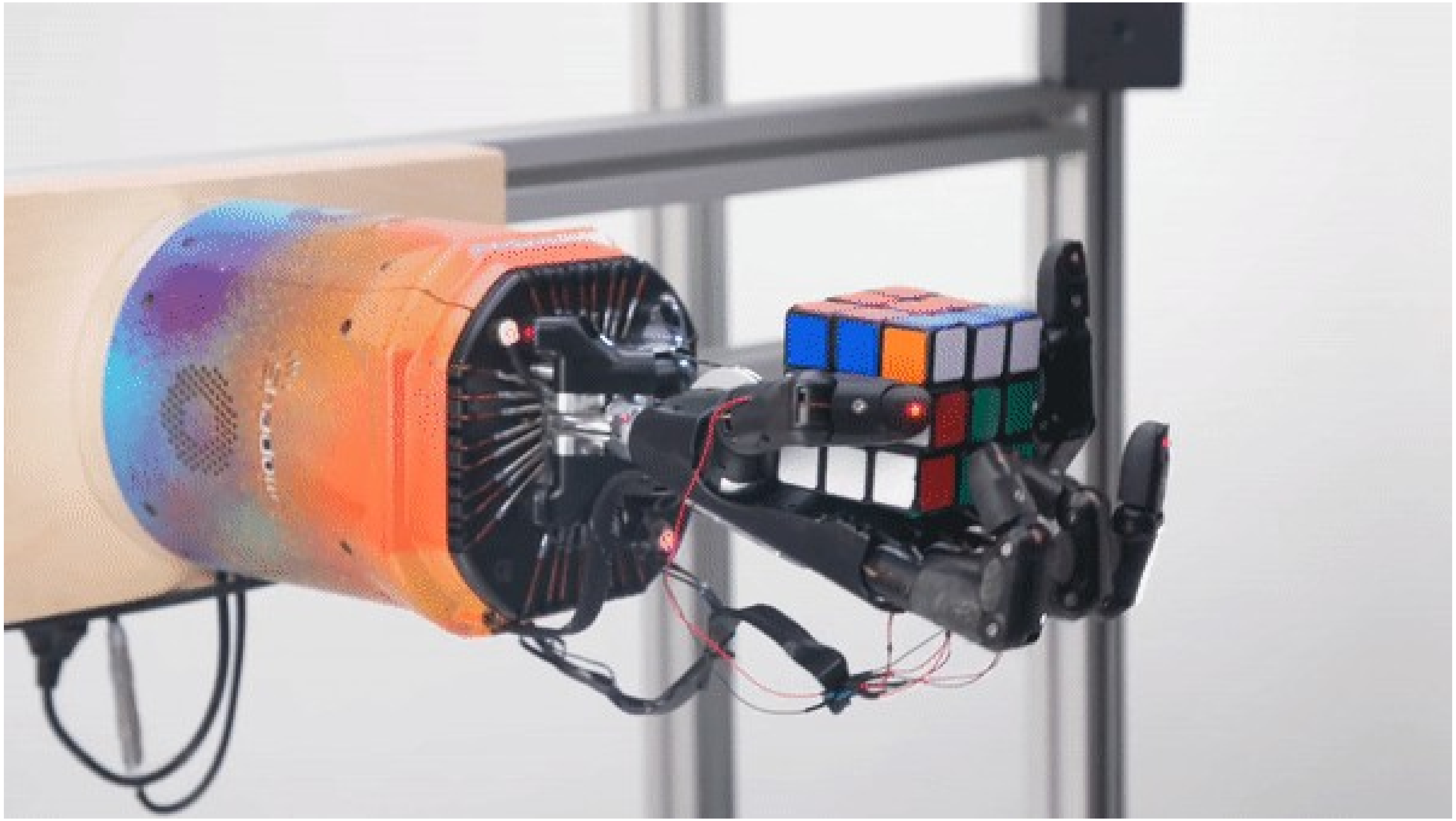
Welcome

https://www.youtube.com/watch?v=1EpJv34gQ88&t=183s



https://www.youtube.com/watch?v=kVmp0uGtShk&t=55s

Solving a Rubik‘s cube with a robotic hand (Learning dexterous manipulations)

Outline

- Why you should care
- How to train your robotic hand
- Learning dexterous manipulations

Outline

- Why you should care
- How to train your robotic hand
- Learning dexterous manipulations

Why you should care

- Human hands are awesome
- Custom robot for every task
- Learning to use a humanoid hand would give more freedom

Outline

- Why you should care
- How to train your robotic hand
- Learning dexterous manipulations

How to train your robotic hand

- Imitation Learning

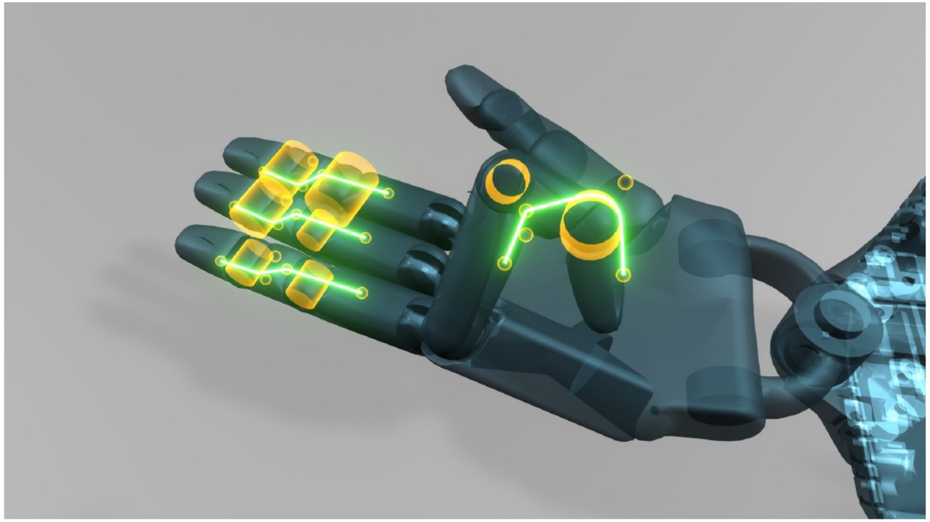
https://vcresearch.berkeley.edu/news/berkeley-startup-train-robots-puppets

- Simulation

Andrychowicz, Marcin, et al. "Learning dexterous in-hand manipulation." arXiv preprint arXiv:1808.00177 (2018)., Figure 3 left

Simulations

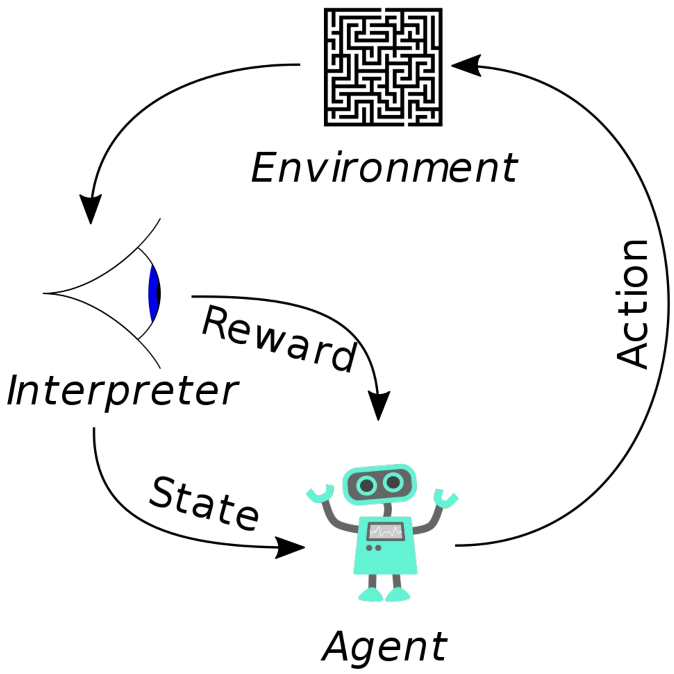
- Simulate everything
- Collect a lot of data for training
- Train policy in Sim



Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand.", Figure 7

Reinforcement learning

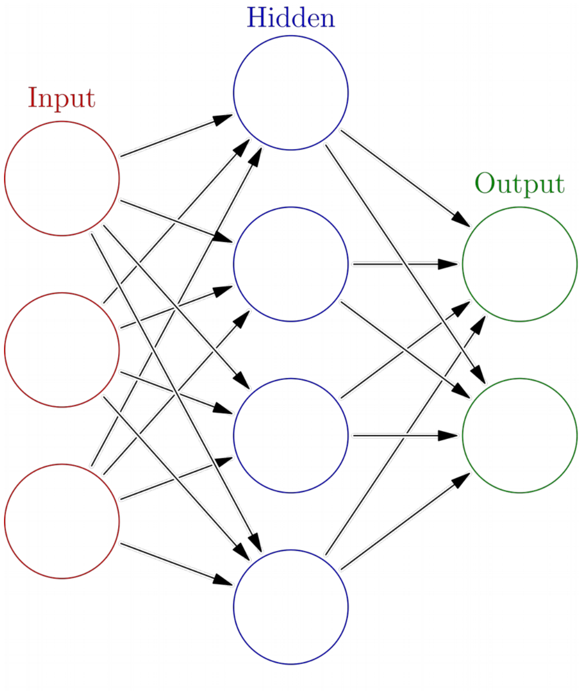
- Learning from mistakes
- Agenct, action, states and reward
- Goal is represented through a function



https://en.wikipedia.org/wiki/Reinforcement_learning#/media/File:Reinforcement_learning_diagram.svg

Deep Reinforcement learning

- Combine ANNs and RF
- Policy is learned by ANN
- Second ANN for state values



https://en.wikipedia.org/wiki/Artificial_neural_network

Memory

- Long-short-term-memory (LSTM)
- Well suited for clasification based on time series
  - Store important information
  - Can retrieve it ater arbitrary time

Outline

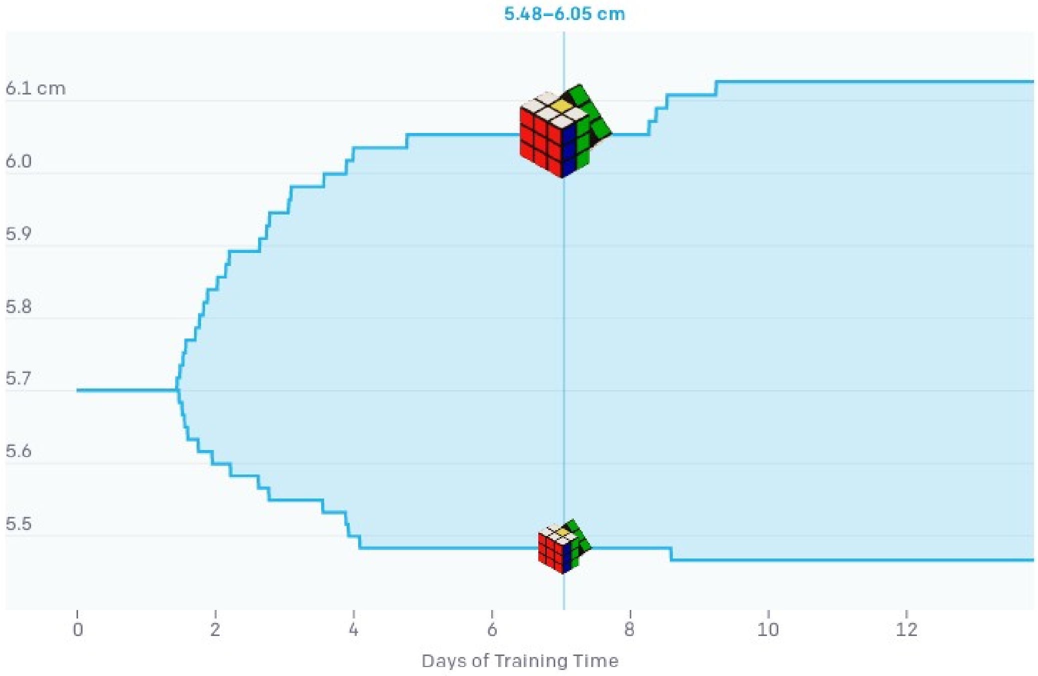
- Why you should care
- How to train your robotic hand
- Learning dexterous manipulations

Domain Randomizations (DR)

- Randomize physical properties of sim environments
- Hand-picked randomizations
  - Uniform distribution
- Problem:
  - What is important?
  - Not that robust

Automatic Domain Randomization (ADR)

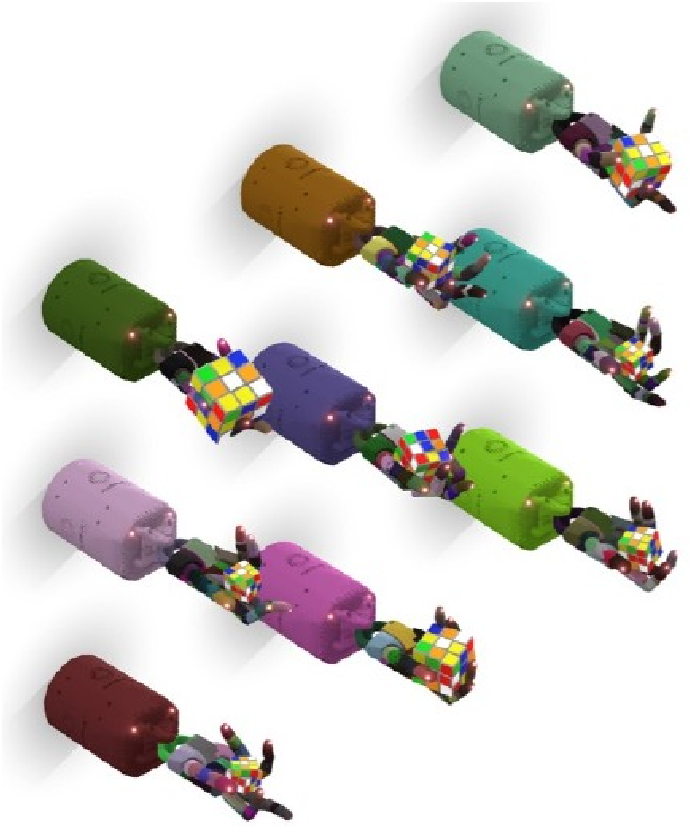
- Basic Idea:
  - Automatically change domain randomizations with progress



https://openai.com/blog/solving-rubiks-cube/

Automatic Domain Randomization (ADR)

- Changes can be made in:
  - Cube size
  - Friction of the hand
  - Gravity
  - Brightness
  - Action delay
  - Motor backlash

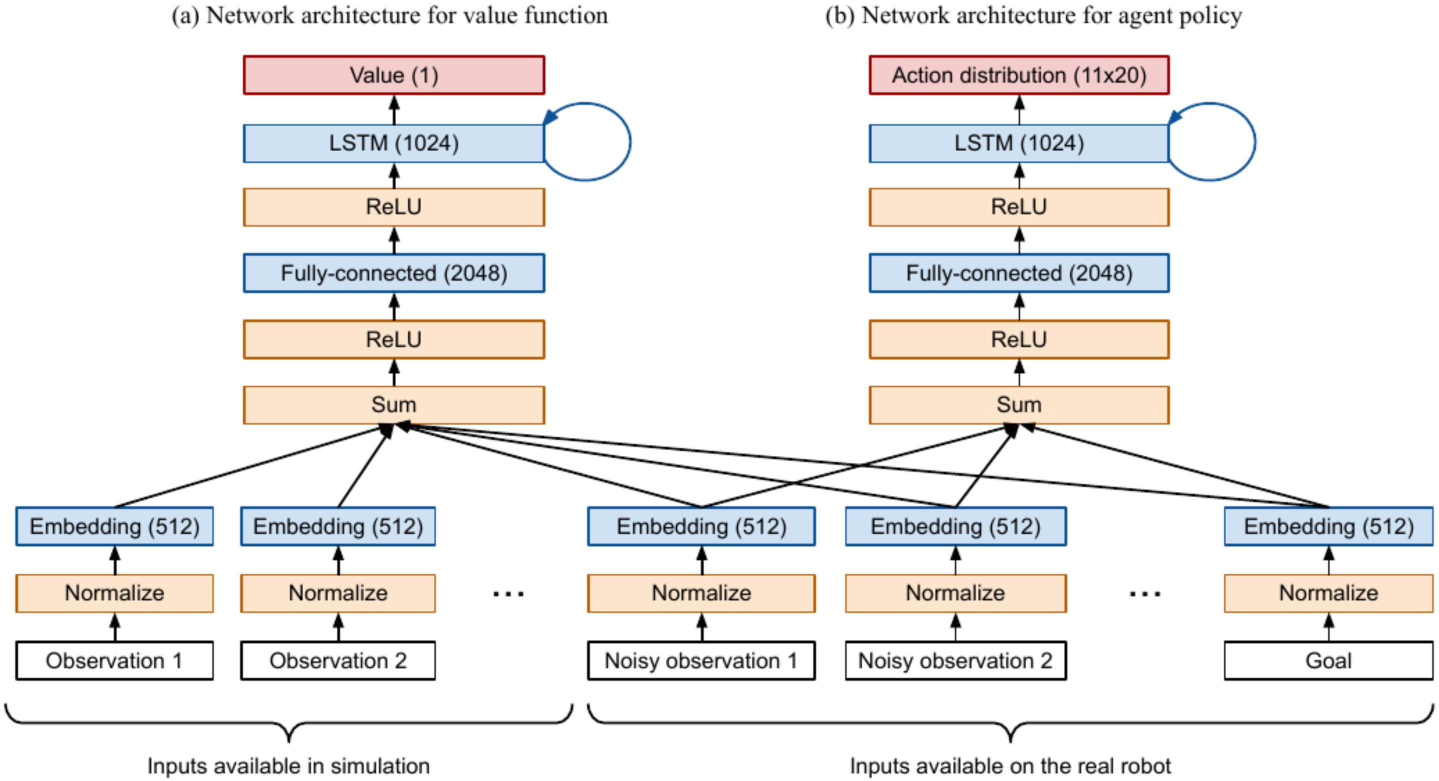


Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand.", Figure 2a

Learning dexterous manipulations

- Using ADR
- Train for several months (~13 Thausand years of sim)
- Two networks during training
  - One to predict value function
  - One for agent policy

Learning dexterous manipulations



Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand.", Figure 12

The robotic hand

Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand.", Figure 4a

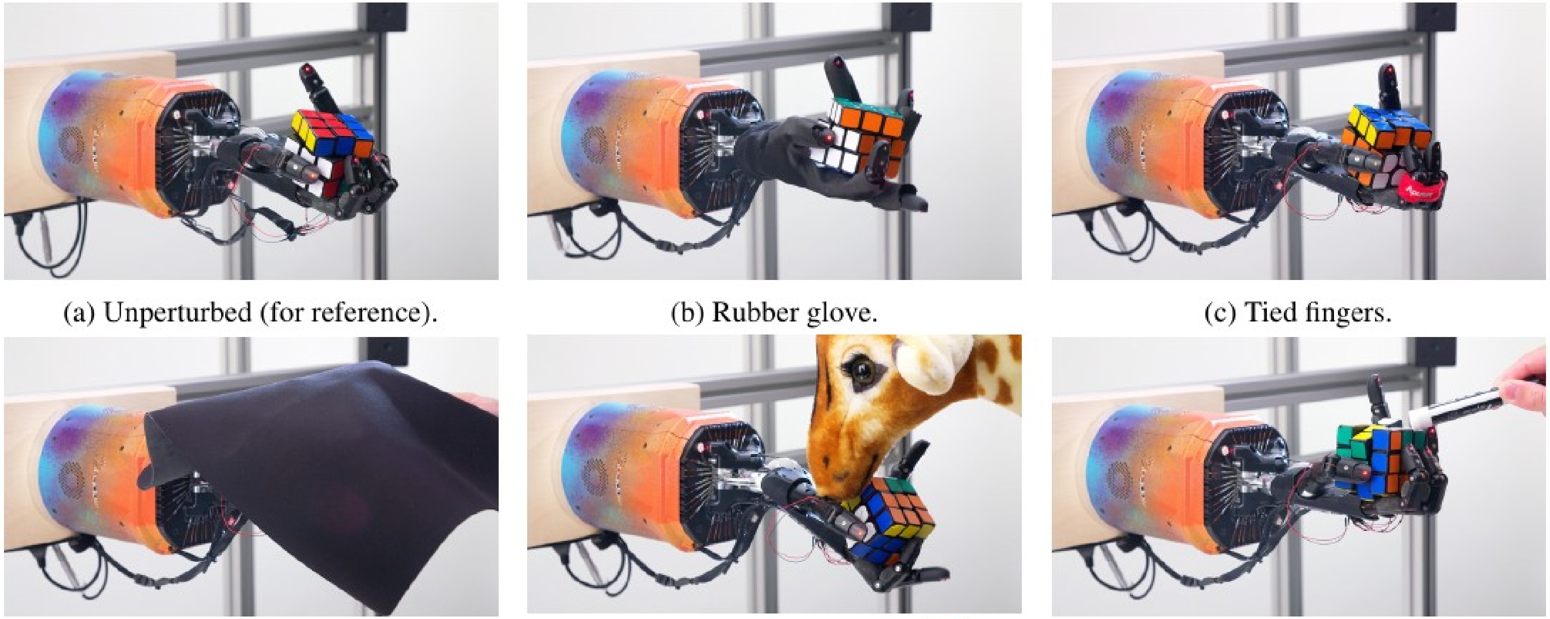
- The cage with 3 cameras from different angles
- Hand with tactile sensors
- Used CNN for vision

Comparisson

Policy	Training Time	ADR Entropy	Successes (Sim)		Successes (Real)	
			Mean	Median	Mean	Median
Baseline (data from [77])	—	—	43.4 ± 0.6	**50**	18.8 ± 5.4	13.0
Baseline (re-run of [77])	—	—	33.8 ± 0.9	**50**	4.0 ± 1.7	2.0
Manual DR	13.78 days	-0.348^{*} npd	42.5 ± 0.7	**50**	2.7 ± 1.1	1.0
ADR (Small)	0.64 days	-0.881 npd	21.0 ± 0.8	15	1.4 ± 0.9	0.5
ADR (Medium)	4.37 days	-0.135 npd	34.4 ± 0.9	**50**	3.2 ± 1.2	2.0
ADR (Large)	13.76 days	0.126 npd	40.5 ± 0.7	**50**	13.3 ± 3.6	11.5
ADR (XL)	—	0.305 npd	45.0 ± 0.6	**50**	16.0 ± 4.0	12.5
ADR (XXL)	—	**0.393 npd**	$\mathbf{46.7 \pm 0.5}$	**50**	$\mathbf{32.0 \pm 6.4}$	**42.0**

Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand.", Table 3

How robust is the outcome?



(a) Unperturbed (for reference).

(b) Rubber glove.

(c) Tied fingers.

(d) Blanket occlusion and perturbation.

(e) Plush giraffe perturbation.[17]

(f) Pen perturbation.

Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand.", Figure 17

Comparisson

Policy	Sensing		ADR Entropy	Successes (Real)		Success Rate	
	Pose	Face Angles		Mean	Median	Half	Full
Manual DR	Vision	Giiker	-0.569^{*} npd	1.8 ± 0.4	2.0	0 %	0 %
ADR	Vision	Giiker	-0.084 npd	3.8 ± 1.0	3.0	0 %	0 %
ADR (XL)	Vision	Giiker	0.467 npd	17.8 ± 4.2	12.5	30 %	10 %
ADR (XXL)	Vision	Giiker	**0.479 npd**	$\mathbf{26.8 \pm 4.9}$	**22.0**	**60 %**	**20 %**
ADR (XXL)	Vision	Vision	**0.479 npd**	12.8 ± 3.4	10.5	20 %	0 %

Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand.", Table 6

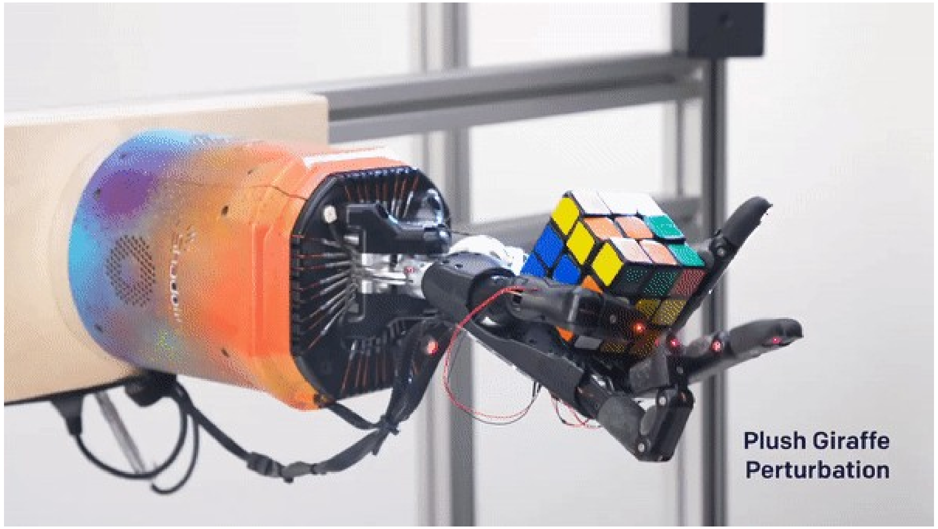
npd = nats per dimension, where nat is the natural unit of information

But ...

- Not a Rubik‘s Cube but Giiker‘s Cube
- Policy only solved 20% with a ‚fair scramble‘
- Other robotic hands can solve rubik‘s cube faster
- Solution steps were generated before

Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand.", Figure 13b

Thank you



https://www.youtube.com/watch?v=QyJGXc9WeNo

Questions?

Feedback

Source

- https://skymind.ai/wiki/deep-reinforcement-learning
- https://towardsdatascience.com/welcome-to-deep-reinforcement-learning-part-1-dqn-c3cab4d41b6b
- Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand."
- Andrychowicz, Marcin, et al. "Learning dexterous in-hand manipulation." arXiv preprint arXiv:1808.00177 (2018).
- https://openai.com/blog/solving-rubiks-cube/