

# Reinforcement Learning

## Computer Engineering Department

## Sharif University of Technology

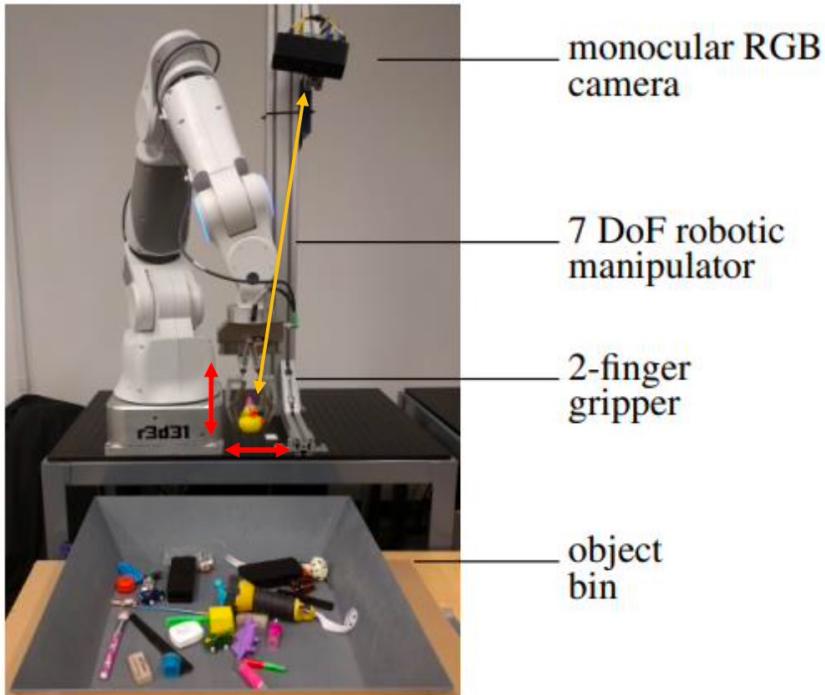
**Mohammad Hossein Rohban, Ph.D.**

**Hossein Hasani**

Spring 2023

Courtesy: Some slides are adopted from CS 285 Berkeley, and CS 234  
Stanford, and Pieter Abbeel's compact series on RL.

# Motivation



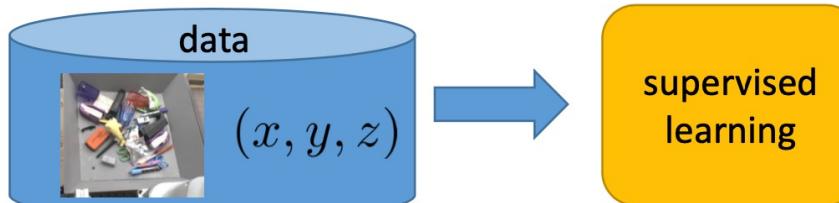
## Option 1:

Understand the problem, design a solution



## Option 2:

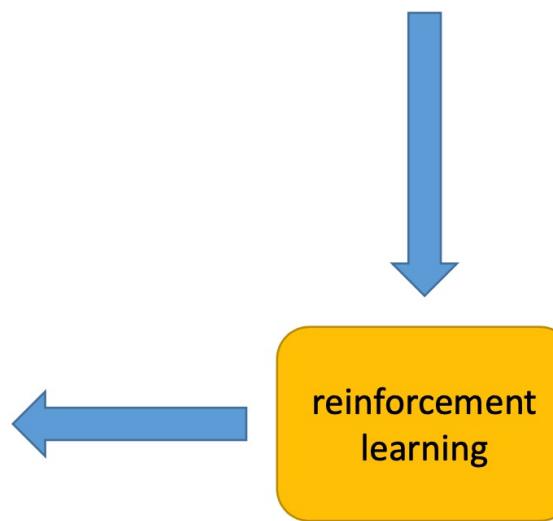
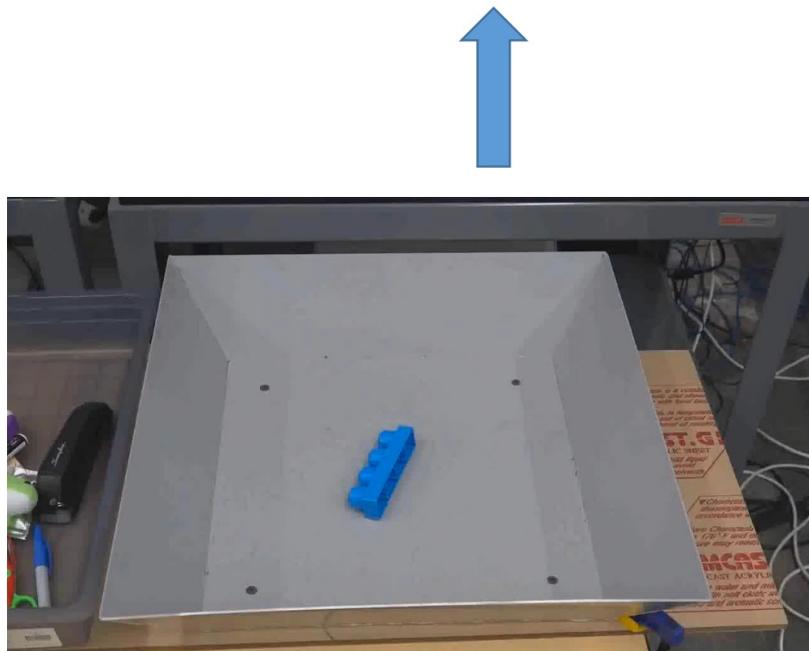
Set it up as a machine learning problem



# Motivation (cont.)



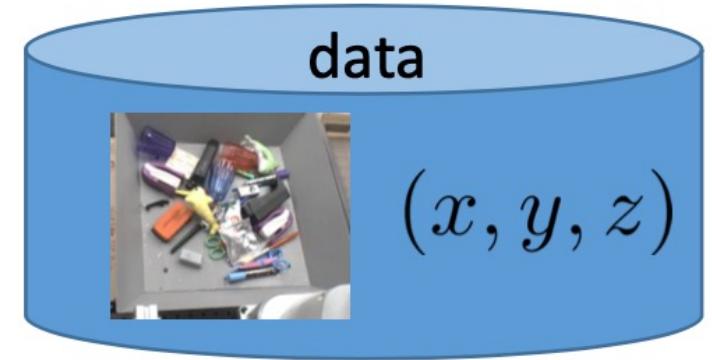
# Motivation (cont.)



# Motivation (cont.)

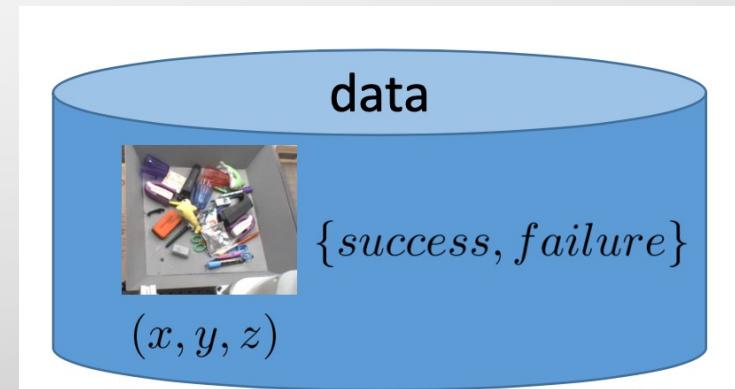
- Supervised learning:

- Ground truth is **known** in advance.
  - Training data are usually **static** and **iid**.

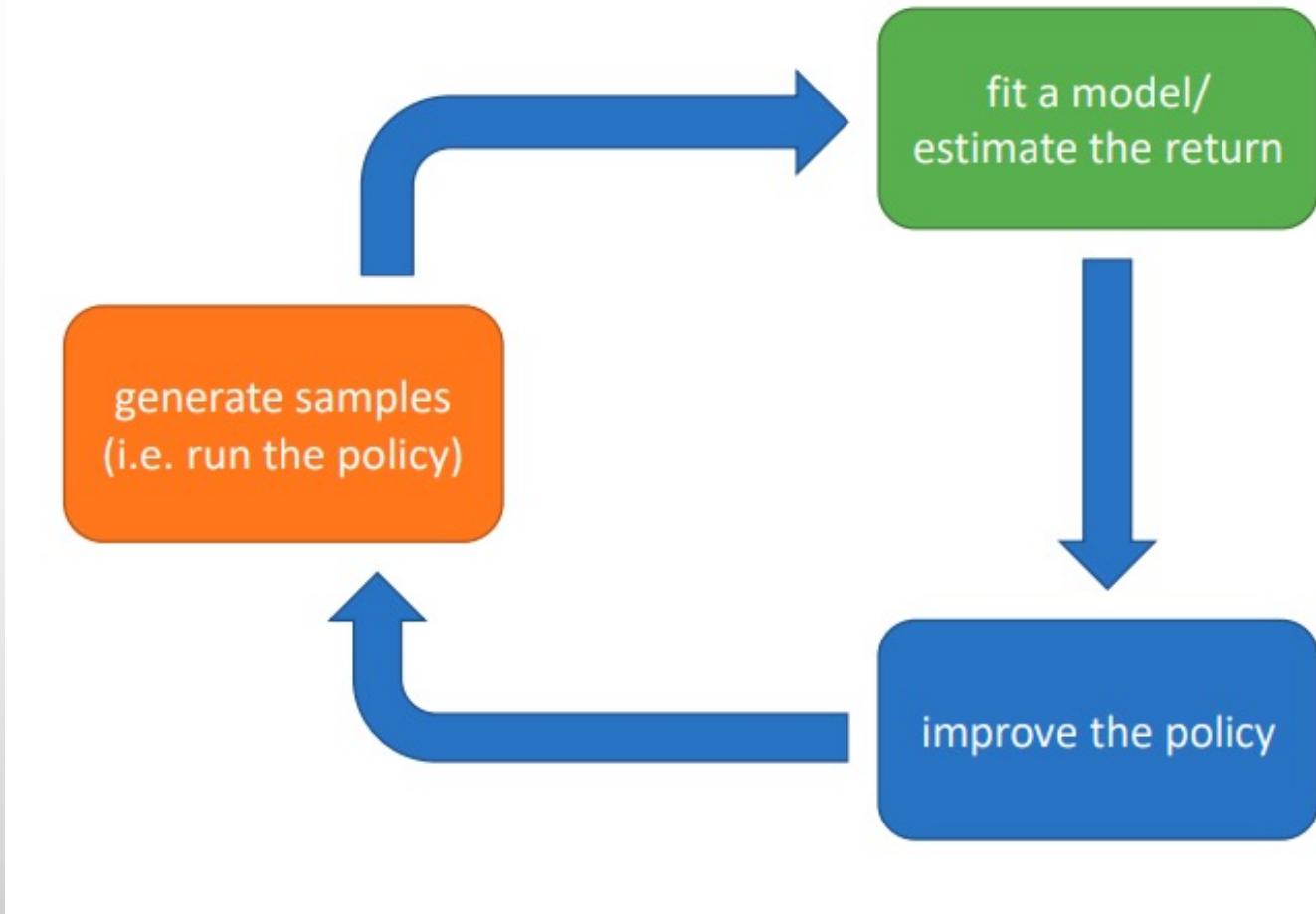


- Reinforcement learning:

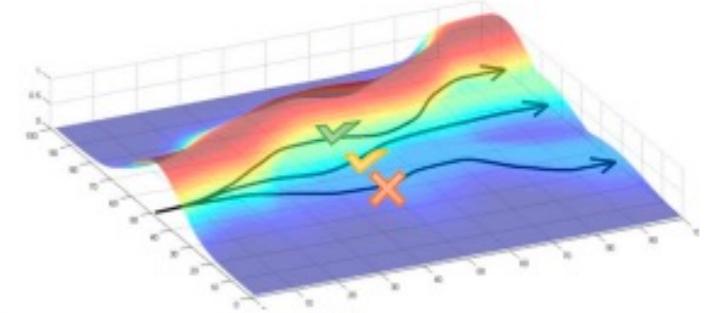
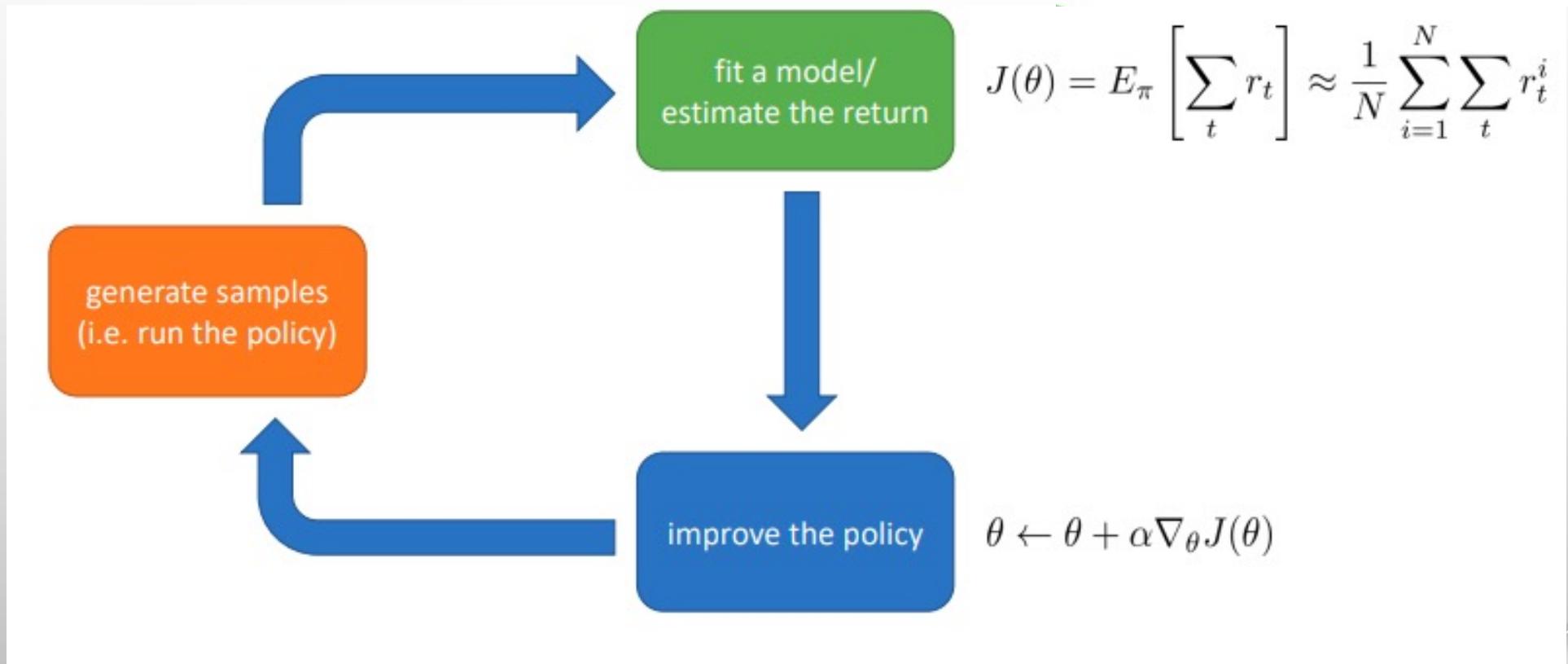
- The best action (**policy**) is usually **unknown a priori**.
  - **Sequence of actions** is needed.
  - A series of trial and error (**search**) is performed.
    - Usually **delayed reward** shows goodness of the trial.
  - Data is **dynamic (exploration)** and **non-iid**.



# The Anatomy of Reinforcement Learning



# A Simple Example



# Motivation (cont.)

	AI Planning	SL	UL	RL	IL
Optimization	X			X	X
Learns from experience		X	X	X	X
Generalization	X	X	X	X	X
Delayed Consequences	X			X	X
Exploration				X	

- SL = supervised learning; UL = unsupervised learning; RL = reinforcement learning; IL = imitation learning
- Imitation learning typically assumes input demonstrations of good policies
- IL reduces RL to SL. IL + RL is promising area

# Planning vs learning

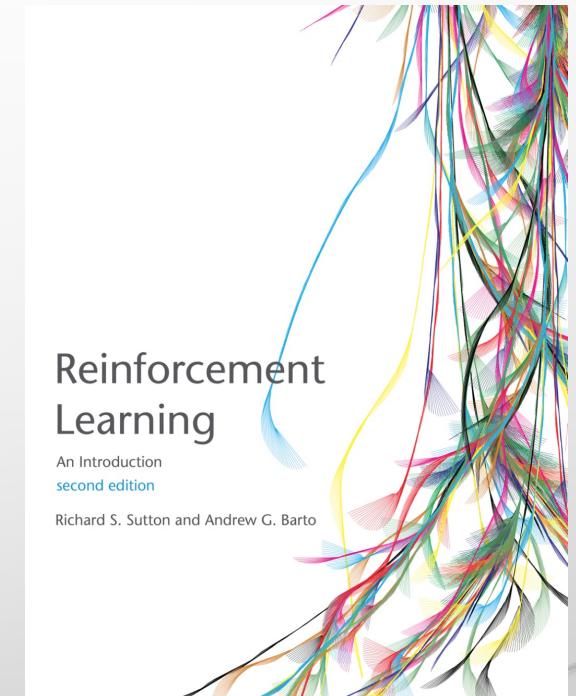
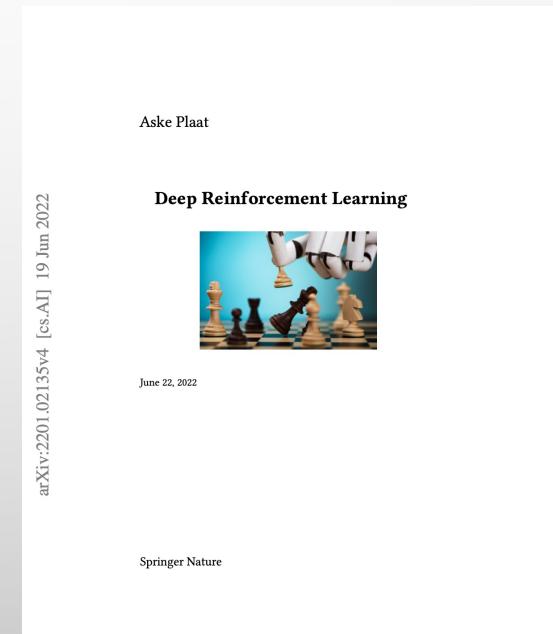
- Two fundamental problems in sequential decision making
  - Reinforcement learning:
    - The environment is initially unknown
    - The agent interacts with the environment
    - The agent improves its policy
  - Planning:
    - A model of the environment is known
    - The agent performs computations with its model (without any external interaction)
    - The agent improves its policy
    - A.K.A. Deliberation, reasoning, introspection, pondering, thought, search

# Prerequisites

- Stochastic Processes
  - Basic Probability and Statistics (Expectation, Conditional Prob. and Expectation, ...)
  - Estimation Theory (Max Likelihood Estimation, Bayesian Estimation, Monte Carlo Simulation)
  - Bayesian Networks and Markov Chains
  - Basic sampling methods
  - Variational Inference
- Basic Information Theory
  - Entropy, Mutual Information, KL Divergence
- Programming in Python
- Deep Learning (concepts and practice; PyTorch)

# References

- Reinforcement Learning: An Introduction by R. Sutton and A. Barto,  
2<sup>nd</sup> Edition, 2020.
- Deep Reinforcement Learning by A. Plaat, 2022.
- Original papers of some methods.



# Teaching Assistants

- Negin Hashemi (Head TA)
- Roozbeh Razavi
- Sepehr Ghobadi
- Hossein Khalili
- Mohammad Mozafari
- Parsa Haghghi
- Soroush Vafaei Tabar
- Mohammad Mousavi
- Ali Kahe
- Aida Afshar
- Amir Mesbah
- Kiana Asgari
- Seyed Abolfazl Rahimi
- Bardia Mohammadi

# Motivation (cont.) ChatGPT; Why RL?!

Step 1

Collect demonstration data  
and train a supervised policy.

A prompt is sampled from our prompt dataset.

Explain reinforcement learning to a 6 year old.

A labeler demonstrates the desired output behavior.

We give treats and punishments to teach...

This data is used to fine-tune GPT-3.5 with supervised learning.



Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

Explain reinforcement learning to a 6 year old.

- A In reinforcement learning, the agent is...
- B Explain rewards...
- C In machine learning...
- D We give treats and punishments to teach...

A labeler ranks the outputs from best to worst.

D > C > A > B

This data is used to train our reward model.

RM

D > C > A > B

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.

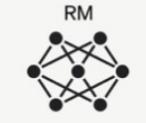
Write a story about otters.



The PPO model is initialized from the supervised policy.

The policy generates an output.

Once upon a time...

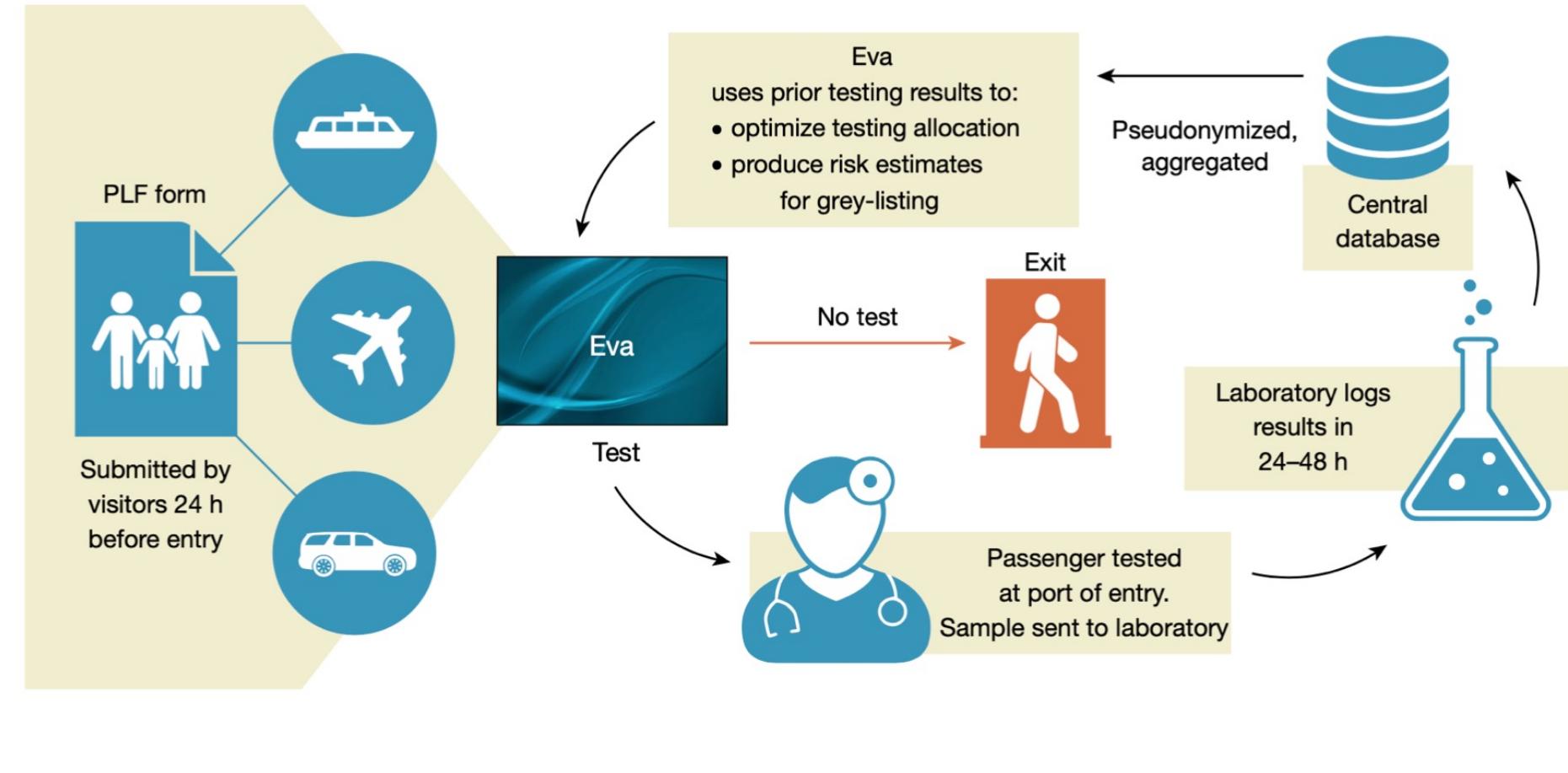


The reward model calculates a reward for the output.

$r_k$

The reward is used to update the policy using PPO.

# Motivation (cont.)



- <https://www.nature.com/articles/s41586-021-04014-z>

# History

2013

Atari (DQN)  
[Deepmind]



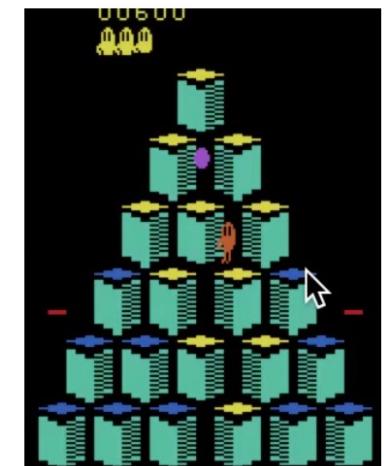
Pong



Enduro



Beamrider



Q\*bert

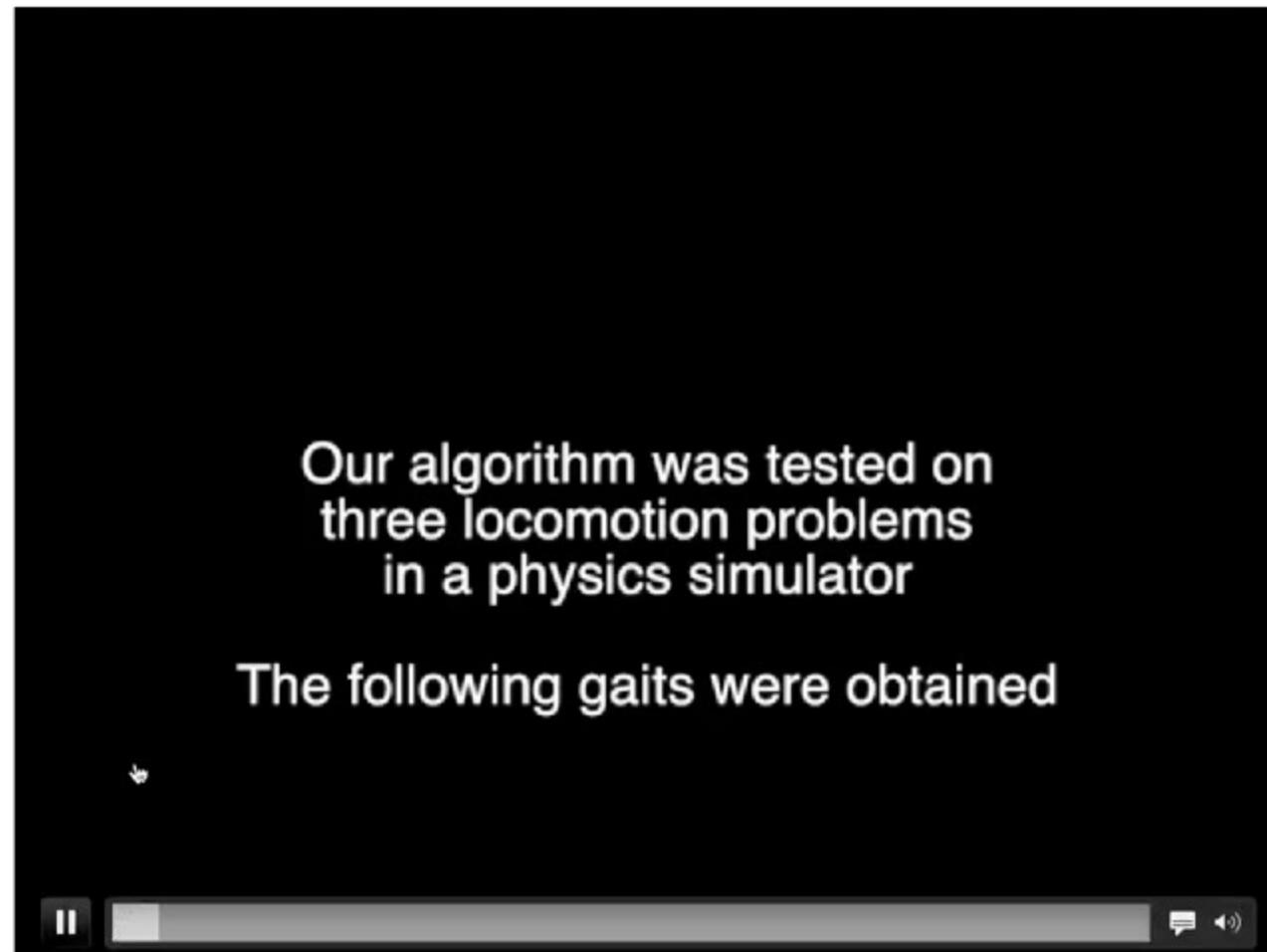
# A Few Deep RL Highlights

2013

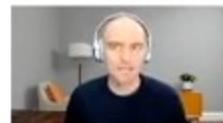
Atari (DQN)  
[Deepmind]

2014

2D locomotion (TRPO)  
[Berkeley]



Play 0:06 – 0:25



# History

2013	Atari (DQN) [Deepmind]
2014	2D locomotion (TRPO) [Berkeley]
2015	<b>AlphaGo</b> [Deepmind]



Tian et al, 2016; Maddison et al, 2014; Clark et al, 2015

# A Few Deep RL Highlights

2013

Atari (DQN)  
[Deepmind]

2014

2D locomotion (TRPO)  
[Berkeley]

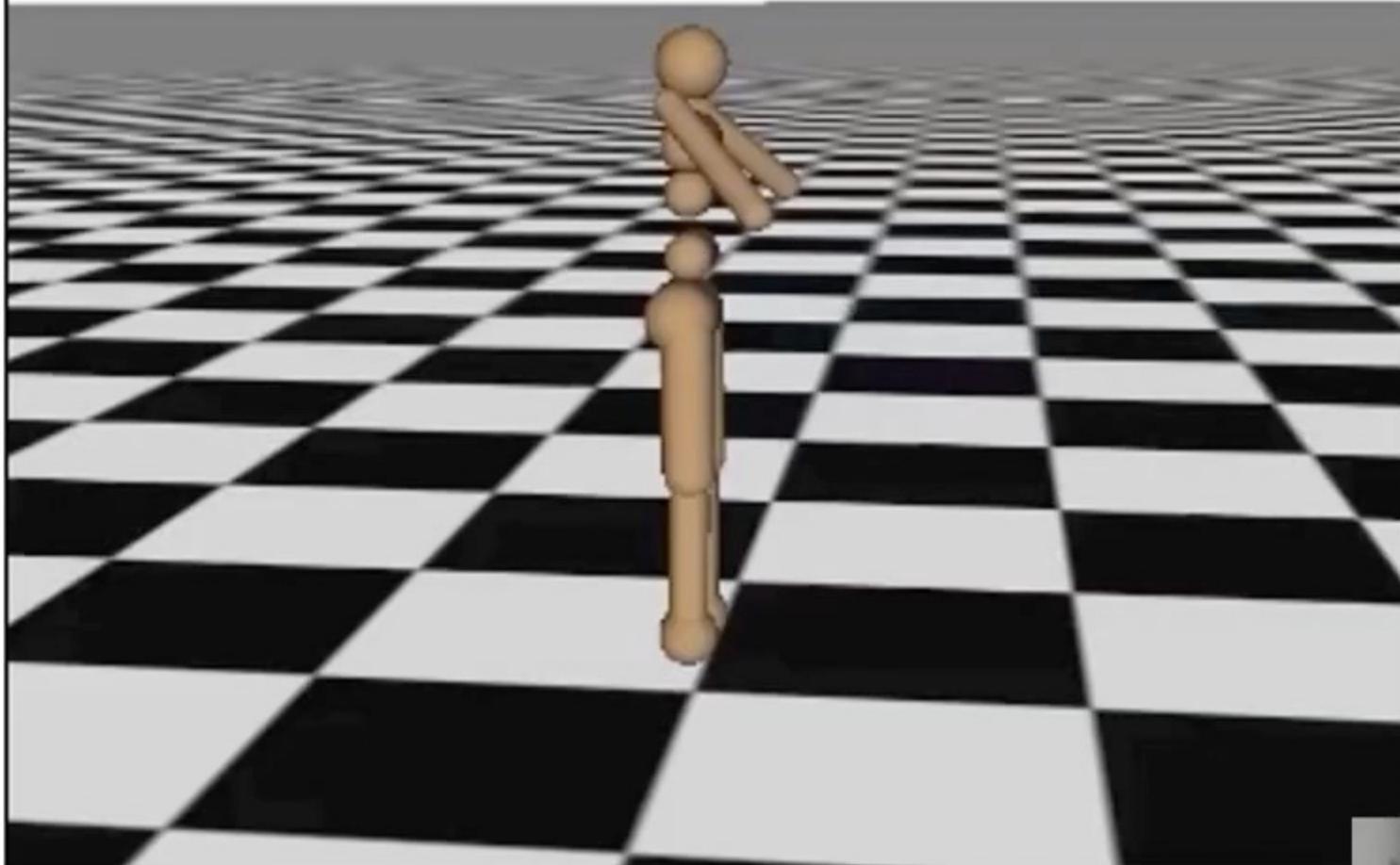
2015

AlphaGo  
[Deepmind]

2016

3D locomotion (TRPO+GAE)  
[Berkeley]

Iteration 0



[Schulman, Moritz, Levine, Jordan, Abbeel, ICLR 2016]



# A Few Deep RL Highlights

2013	Atari (DQN) [Deepmind]
2014	2D locomotion (TRPO) [Berkeley]
2015	AlphaGo [Deepmind]
2016	3D locomotion (TRPO+GAE) [Berkeley]
<b>2016</b>	<b>Real Robot Manipulation (GPS) [Berkeley]</b>

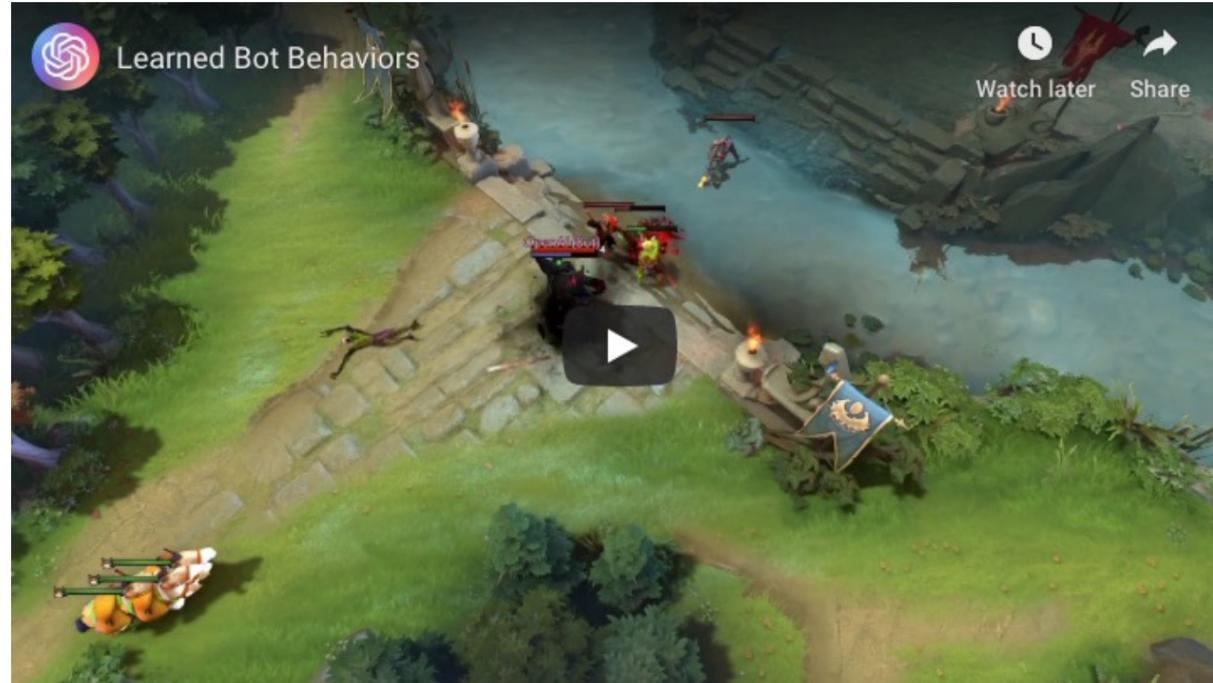


[Levine\*, Finn\*, Darrell, Abbeel, JMLR 2016]



# History

2013	Atari (DQN) [Deepmind]
2014	2D locomotion (TRPO) [Berkeley]
2015	AlphaGo [Deepmind]
2016	3D locomotion (TRPO+GAE) [Berkeley]
2016	Real Robot Manipulation (GPS) [Berkeley, Google]
2017	<b>Dota2</b> <b>(PPO)</b> [OpenAI]



OpenAI Dota Bot beat best humans 1:1 (Aug 2018)

# A Few Deep RL Highlights

2013

Atari (DQN)  
[Deepmind]

2014

2D locomotion (TRPO)  
[Berkeley]

2015

AlphaGo  
[Deepmind]

2016

3D locomotion (TRPO+GAE)  
[Berkeley]

2016

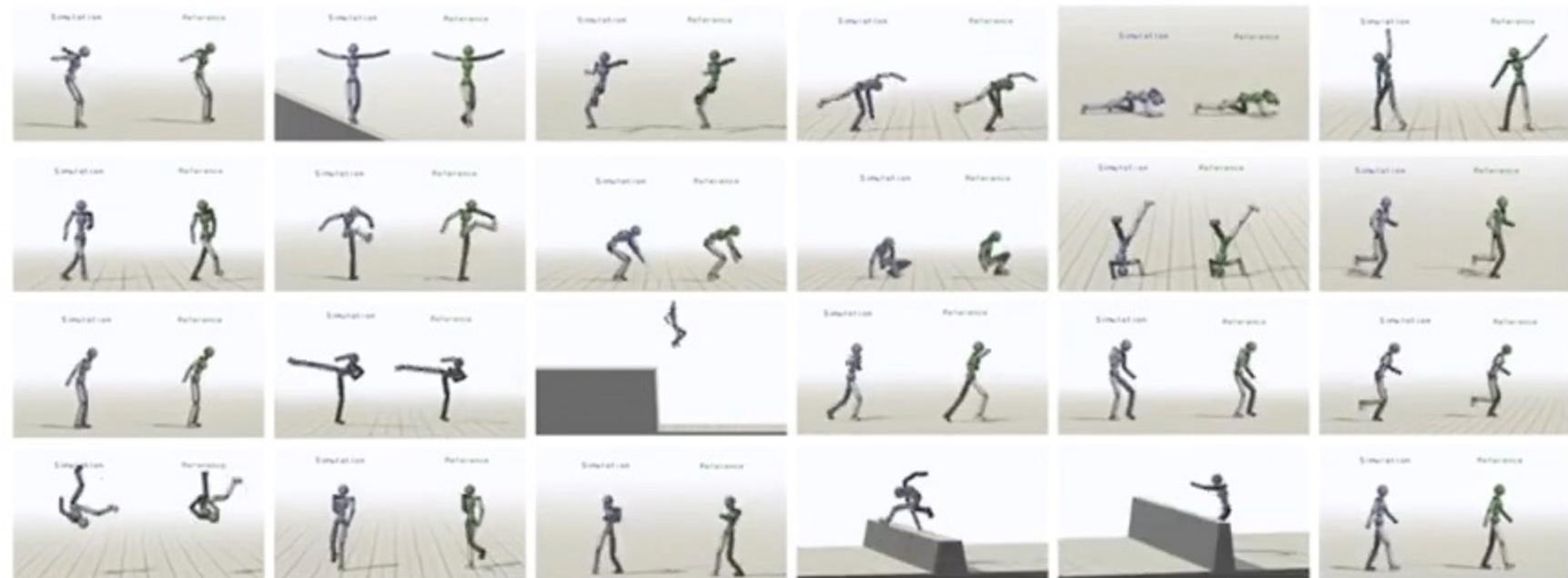
Real Robot Manipulation  
(GPS) [Berkeley, Google]

2017

Dota2  
(PPO) [OpenAI]

2018

DeepMimic  
[Berkeley]



[Peng, Abbeel, Levine, van de Panne, 2018]



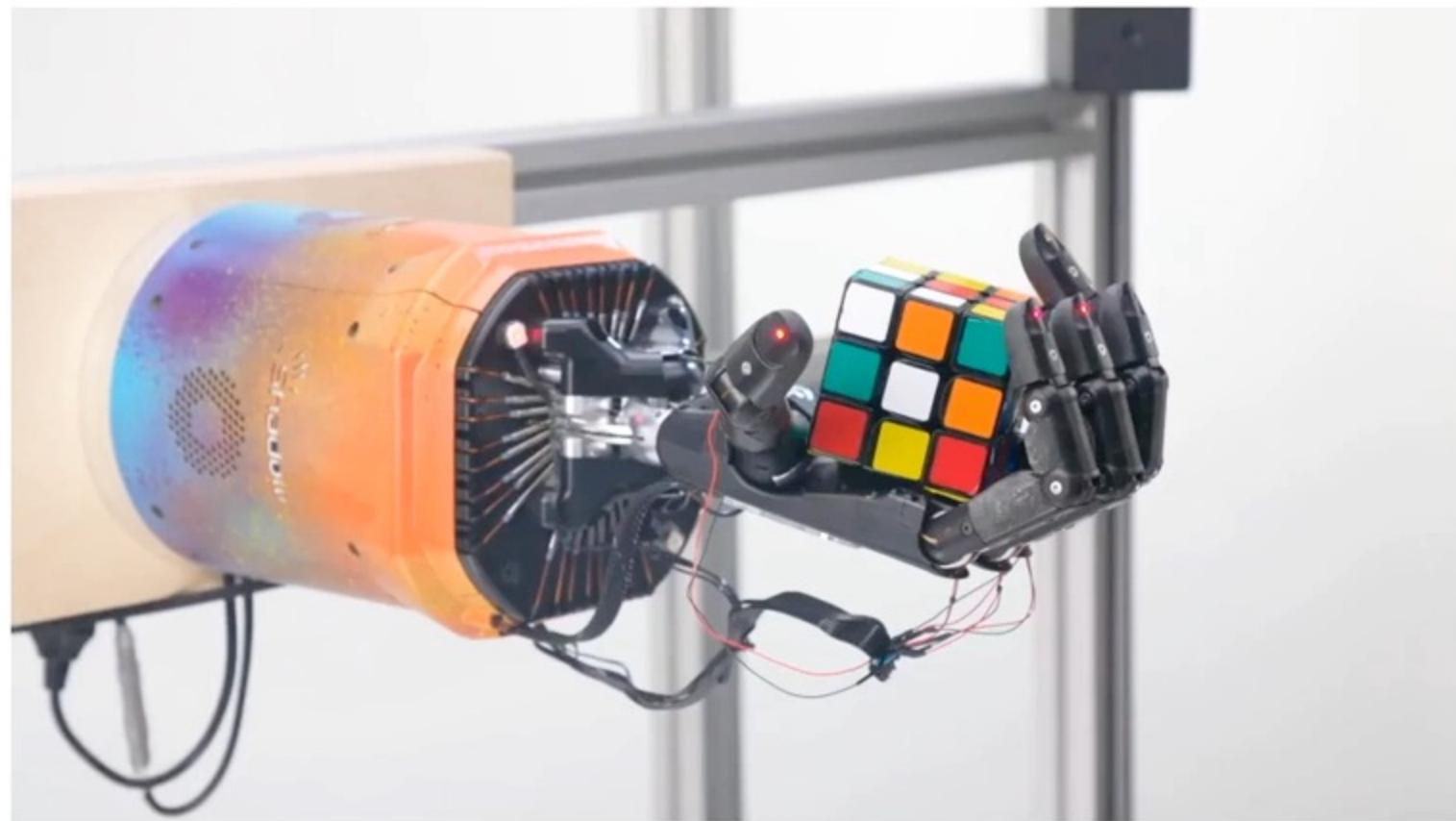
# History

2013	Atari (DQN) [Deepmind]
2014	2D locomotion (TRPO) [Berkeley]
2015	AlphaGo [Deepmind]
2016	3D locomotion (TRPO+GAE) [Berkeley]
2016	Real Robot Manipulation (GPS) [Berkeley, Google]
2017	Dota2 (PPO) [OpenAI]
2018	DeepMimic [Berkeley]
2019	<b>AlphaStar</b> [Deepmind]



# A Few Deep RL Highlights

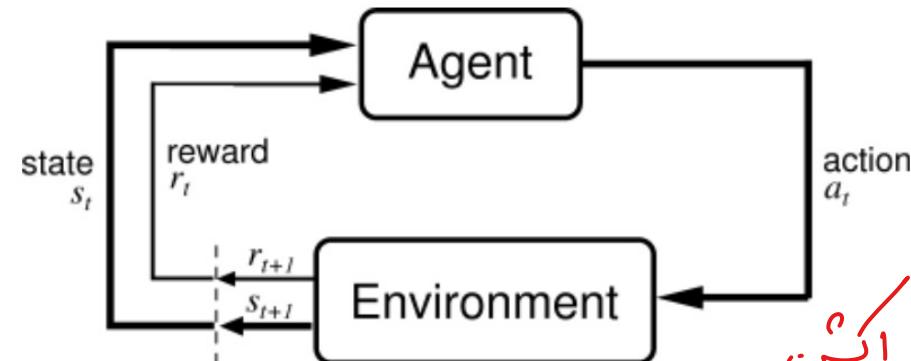
2013	Atari (DQN) [Deepmind]
2014	2D locomotion (TRPO) [Berkeley]
2015	AlphaGo [Deepmind]
2016	3D locomotion (TRPO+GAE) [Berkeley]
2016	Real Robot Manipulation (GPS) [Berkeley, Google]
2017	Dota2 (PPO) [OpenAI]
2018	DeepMimic [Berkeley]
2019	AlphaStar [Deepmind]
2019	Rubik's Cube (PPO+DR) [OpenAI]



# Let's Begin: Markov Decision Processes (MDPs)

An MDP is defined by:

- Set of states  $S$   $s_t \in S$
  - Set of actions  $A$   $a_t \in A$
  - Transition function  $P(s' | s, a)$
  - Reward function  $R(s, a, s')$
  - Start state  $s_0$
  - Discount factor  $\gamma$
  - Horizon  $H$



$$P(S_{t+1} = s' | S_t = s, A_t = a)$$

دُنْهَرِ اُنْ

داستنی مکمل → عافنه سخا نه طاریز

جواب، لغات

محل سازی که فن

مارکز بر مارکز

## The Goal

- The policy is  $\pi_\theta: S \rightarrow A$  for infinite horizon or  
 $\pi_\theta: S \times \{0, \dots, H\} \rightarrow A$  for finite horizon MDP.
- زمانی محدوده  
تکمیل کننده
- زمانی نامحدوده
- باشه
- کوچک ترین سریعترین
- ایجاد

MDP  $(S, A, T, R, \gamma, H)$ ,

goal: 
$$\max_{\pi} \mathbb{E} \left[ \sum_{t=0}^H \gamma^t R(S_t, A_t, S_{t+1}) | \pi \right]$$

Sometimes the policy could be stochastic:  $\pi : S \times A \rightarrow [0,1]$ , which is  
 $\pi(a|s) = \Pr(A_t = a | S_t = s)$ .

# Example: Grid World

An MDP is defined by:

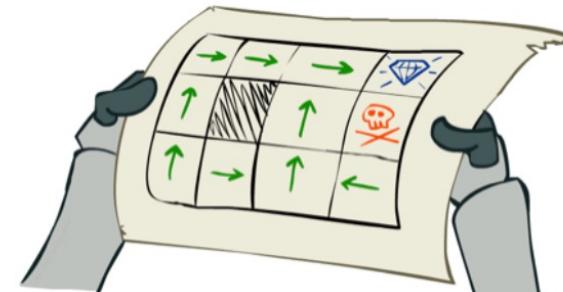
- Set of states  $S$
- Set of actions  $A$
- Transition function  $P(s' | s, a)$
- Reward function  $R(s, a, s')$
- Start state  $s_0$
- Discount factor  $\gamma$
- Horizon  $H$



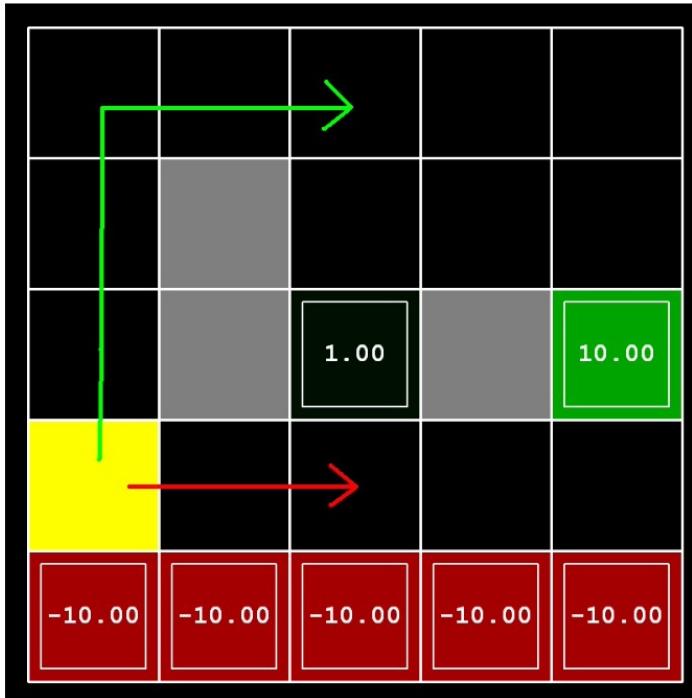
Goal:

$$\max_{\pi} \mathbb{E} \left[ \sum_{t=0}^H \gamma^t R(S_t, A_t, S_{t+1}) \middle| \pi \right]$$

$\pi^*$ :

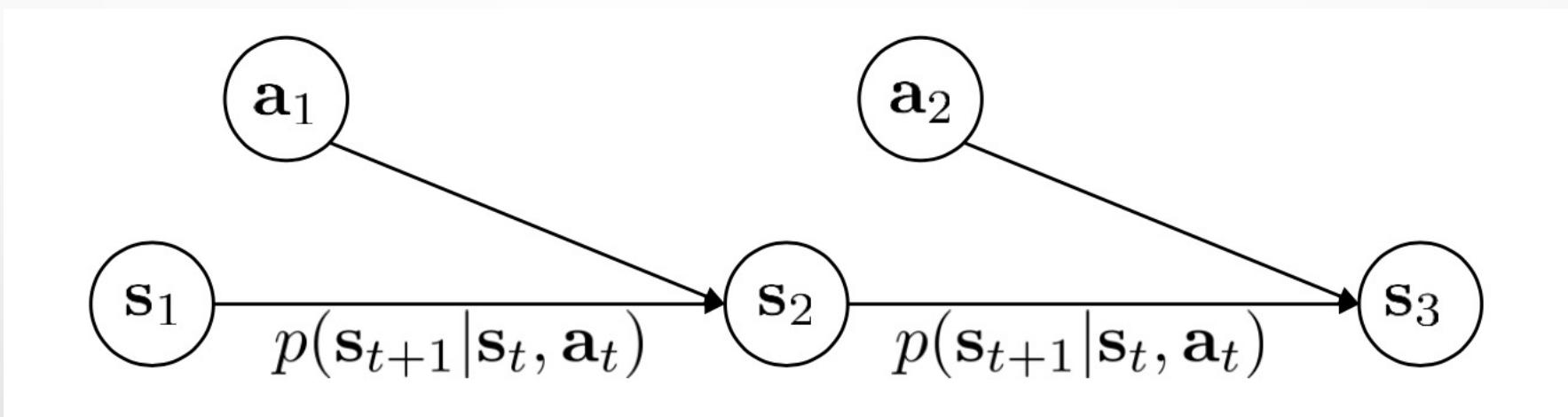


# Exercise



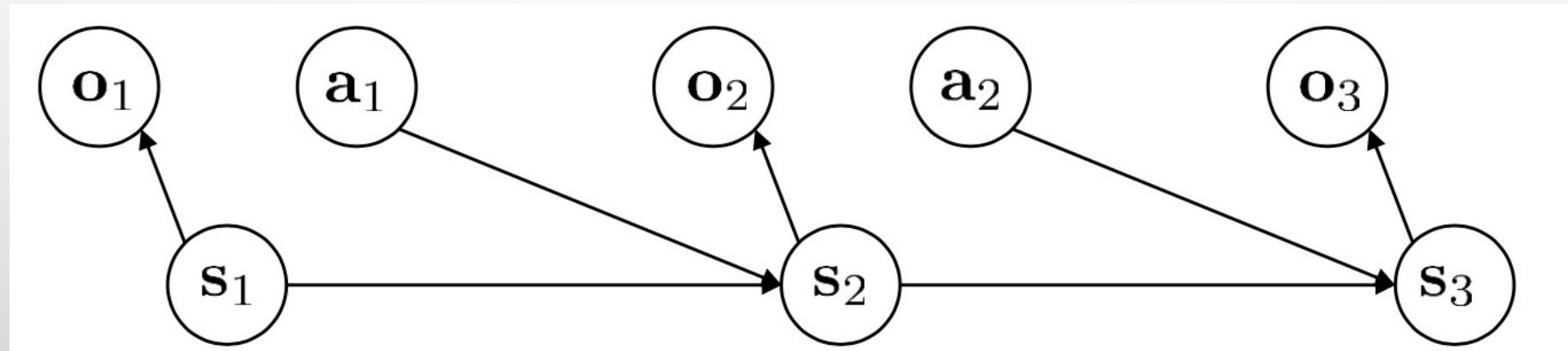
- (a) Prefer the close exit (+1), risking the cliff (-10) (1)  $\gamma = 0.1$ , noise = 0.5
  - (b) Prefer the close exit (+1), but avoiding the cliff (-10) (2)  $\gamma = 0.99$ , noise = 0
  - (c) Prefer the distant exit (+10), risking the cliff (-10) (3)  $\gamma = 0.99$ , noise = 0.5
  - (d) Prefer the distant exit (+10), avoiding the cliff (-10) (4)  $\gamma = 0.1$ , noise = 0

# Graphical Model of MDPs



# Partially Observable MDPs (POMDPs)

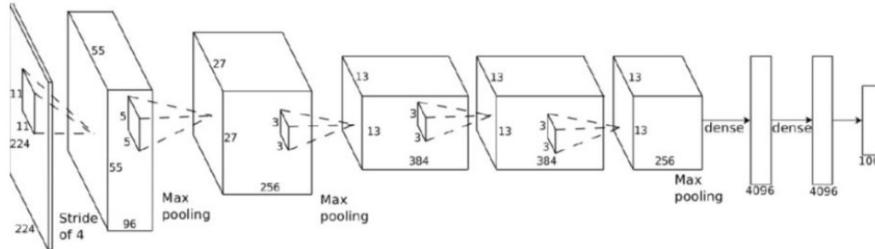
- Often times the state  $S_t$  is **hidden** from the agent, and only **noisy** or **incomplete** measurement of it is available  $O_t$ .



# Policy as a function of $S_t$ or $O_t$



$\mathbf{o}_t$



$\pi_{\theta}(\mathbf{a}_t | \mathbf{o}_t)$



$\mathbf{a}_t$

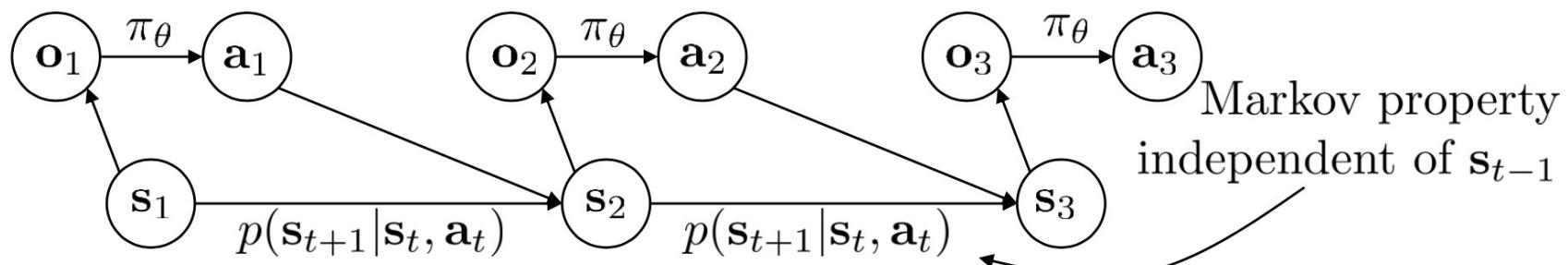
$s_t$  – state

$\mathbf{o}_t$  – observation

$\mathbf{a}_t$  – action

$\pi_{\theta}(\mathbf{a}_t | \mathbf{o}_t)$  – policy

$\pi_{\theta}(\mathbf{a}_t | s_t)$  – policy (fully observed)



# Optimal Value Function

MDP  $(S, A, T, R, \gamma, H)$ ,

goal:  $\max_{\pi} \mathbb{E} \left[ \sum_{t=0}^H \gamma^t R(S_t, A_t, S_{t+1}) | \pi \right]$

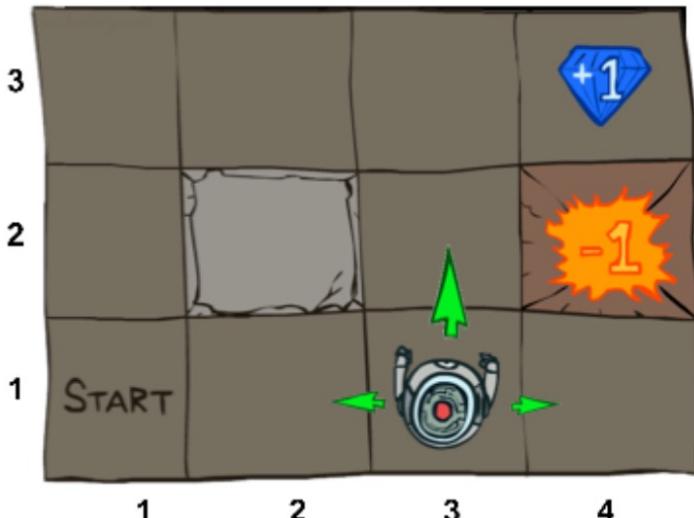
$$V^*(s) = \max_{\pi} \mathbb{E} \left[ \sum_{t=0}^H \gamma^t R(s_t, a_t, s_{t+1}) \mid \pi, s_0 = s \right]$$

= sum of discounted rewards when starting from state  $s$  and acting optimally

# Optimal Value Function

$$V^*(s) = \max_{\pi} \mathbb{E} \left[ \sum_{t=0}^H \gamma^t R(s_t, a_t, s_{t+1}) \mid \pi, s_0 = s \right]$$

= sum of discounted rewards when starting from state s and acting optimally



Let's assume:

actions deterministically successful, gamma = 1, H = 100

$$V^*(4,3) =$$

$$V^*(3,3) =$$

$$V^*(2,3) =$$

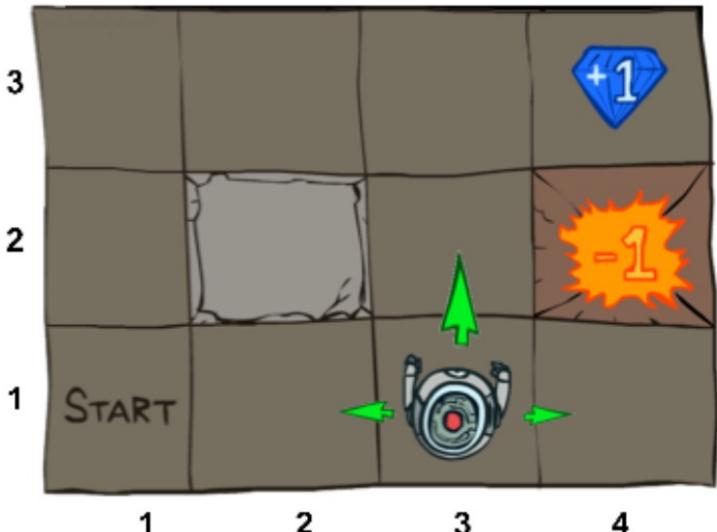
$$V^*(1,1) =$$

$$V^*(4,2) =$$

# Optimal Value Function

$$V^*(s) = \max_{\pi} \mathbb{E} \left[ \sum_{t=0}^H \gamma^t R(s_t, a_t, s_{t+1}) \mid \pi, s_0 = s \right]$$

= sum of discounted rewards when starting from state s and acting optimally



Let's assume:

actions deterministically successful, gamma = 0.9, H = 100

$$V^*(4,3) =$$

$$V^*(3,3) =$$

$$V^*(2,3) =$$

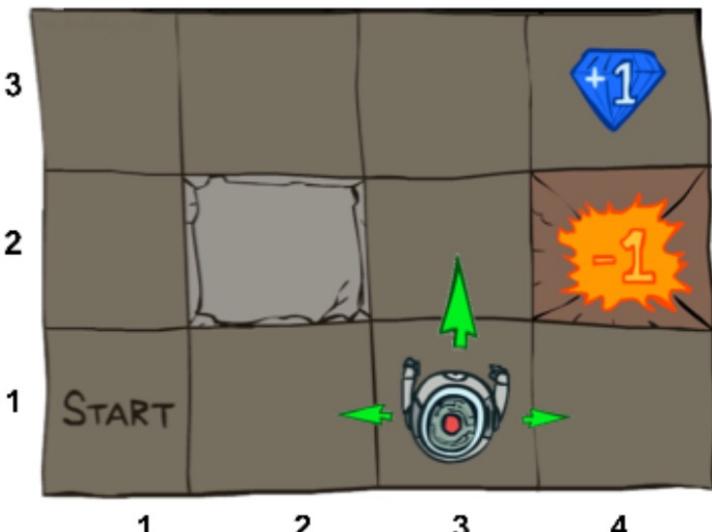
$$V^*(1,1) =$$

$$V^*(4,2) =$$

# Optimal Value Function

$$V^*(s) = \max_{\pi} \mathbb{E} \left[ \sum_{t=0}^H \gamma^t R(s_t, a_t, s_{t+1}) \mid \pi, s_0 = s \right]$$

= sum of discounted rewards when starting from state s and acting optimally



Let's assume:

actions successful w/probability 0.8, gamma = 0.9, H = 100

$$V^*(4,3) =$$

$$V^*(3,3) =$$

$$V^*(2,3) =$$

$$V^*(1,1) =$$

$$V^*(4,2) =$$