CS 4789/5789:

Introduction to Reinforcement Learning

Wen Sun

TAs:
Wen-Ding Li and Hadi Alzayer

Course website:
https://wensun.github.io/CS4789.html

(Lecture notes & reading materials)

Read the course website!

This course focuses on Reinforcement Learning

We care about:

(1) Algorithm design,
(2) Analysis of algorithm performance (e.g., convergence),
(3) How they work in practice

Four main themes we will cover in this course:

1. Markov Decision Process: Dynamic Programming & planning
2. Continuous Control
3. Learning in Markov Decision Process
4. Imitation Learning (i.e., learning from demonstrations)

Logistics

Four assignments (6 late days):
HW 0: 10%, HW 1-3: 20% each

Final exam:
30%

Attendance:
5% (bonus)

Tentative schedule for HWs are on course website

Final will be scheduled in the final week

Logistics

Four assignments (6 late days):
HW 0: 10%, HW 1-3: 20% each

Final exam:
30%

Attendance:
5% (bonus)

Tentative schedule for HWs are on course website

Final will be scheduled in the final week

Discussion on HW problems are **encouraged**;
But everyone needs to understand and write her/his **own solutions**;
Sharing answers inside/outside of the class is not allowed.
(see course website for more details)

Prerequisites

Strong grasp on Machine Learning (e.g., CS 4780)

Linear algebra & probability, programming in Python

Prerequisites

Strong grasp on Machine Learning (e.g., CS 4780)

Linear algebra & probability, programming in Python

Traditional Machine Learning such as supervised learning is a **small subset** of RL!

Reading Materials:
Reinforcement Learning: Theory & Algorithms

https://rltheorybook.github.io/

This is an extremely advanced RL book, so we will pick **specific subsections** for you to read

Please let us know if you find any typos or mistakes in the book

Questions?

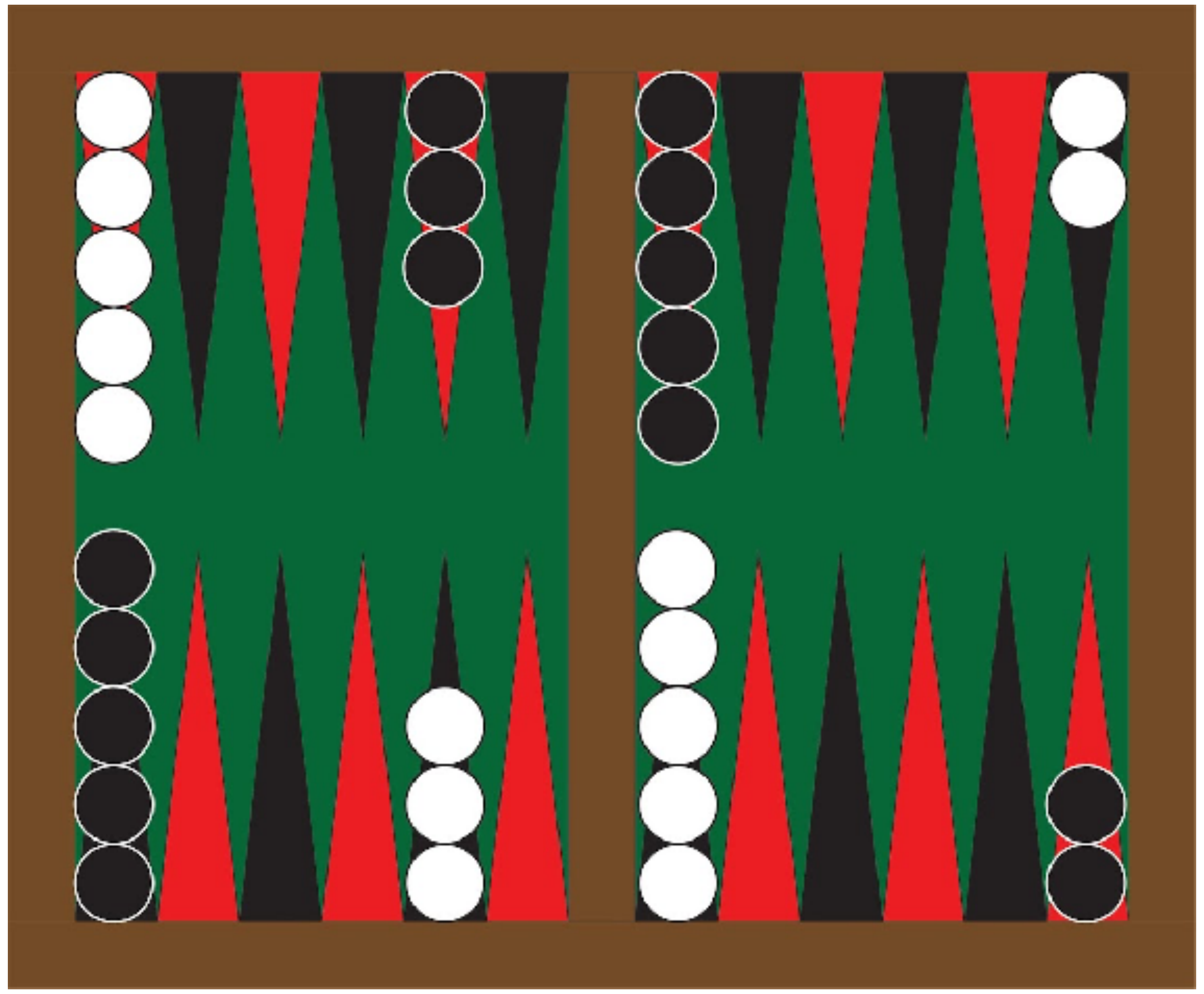
(Please read the course website after class)

Outlines:

1. Introduction: Applications of RL, RL versus Supervised Learning

2. Basics of Markov Decision Process (MDP): model, example, V & Q functions

Big Successful Stories of Reinforcement Learning



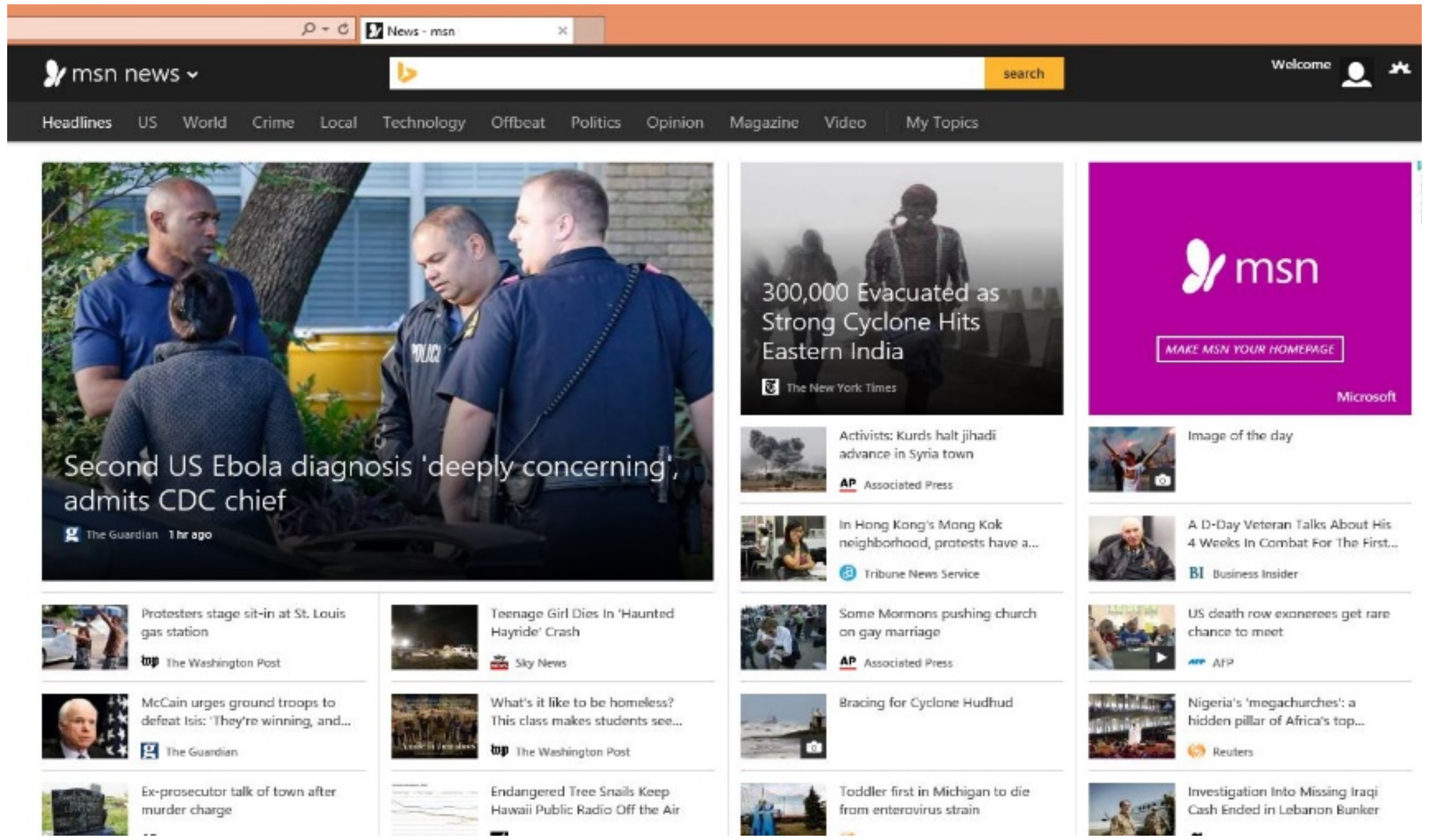
TD GAMMON [Tesauro 95]

[AlphaZero, Silver et.al, 17]

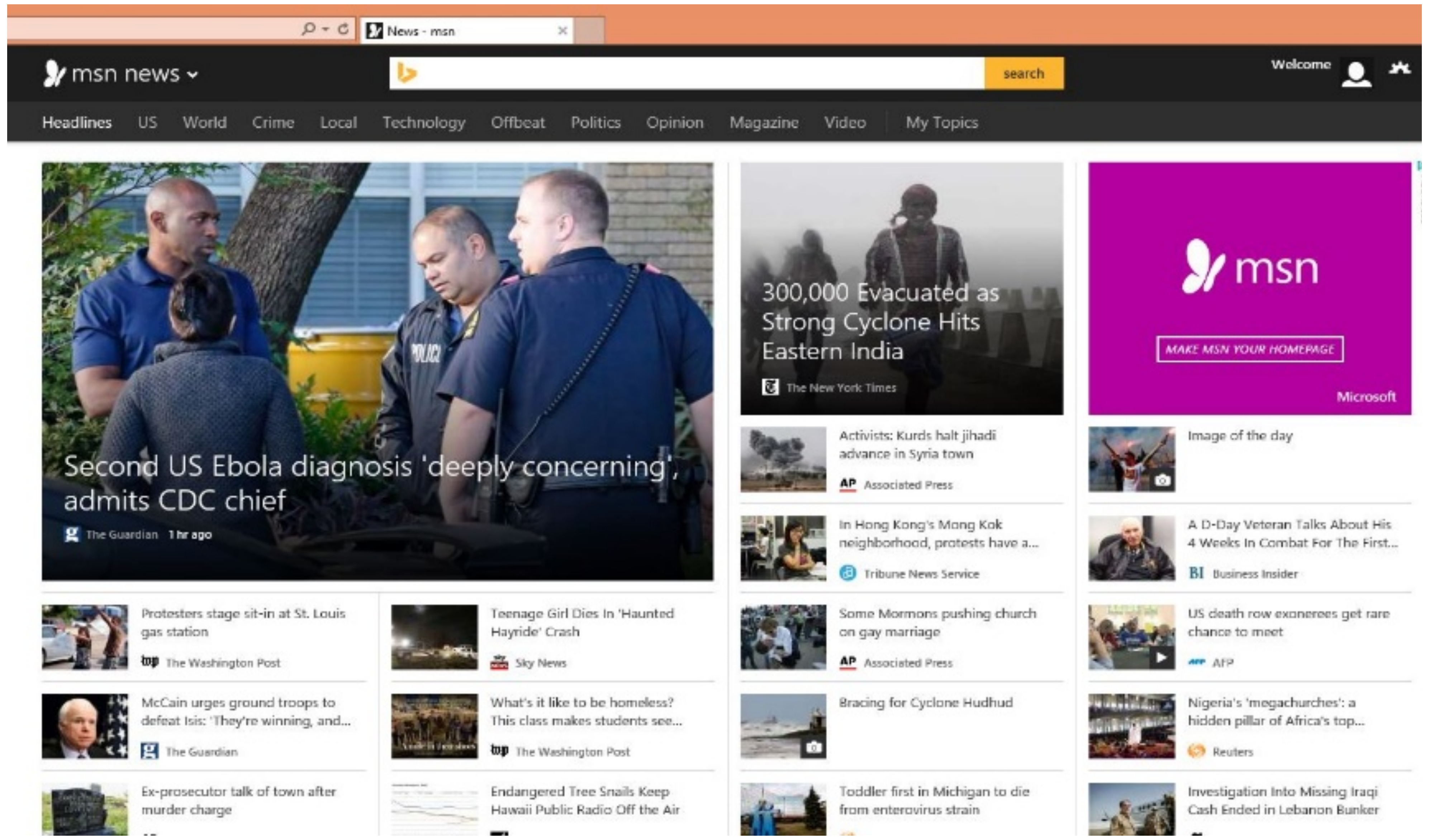
[OpenAI Five, 18]

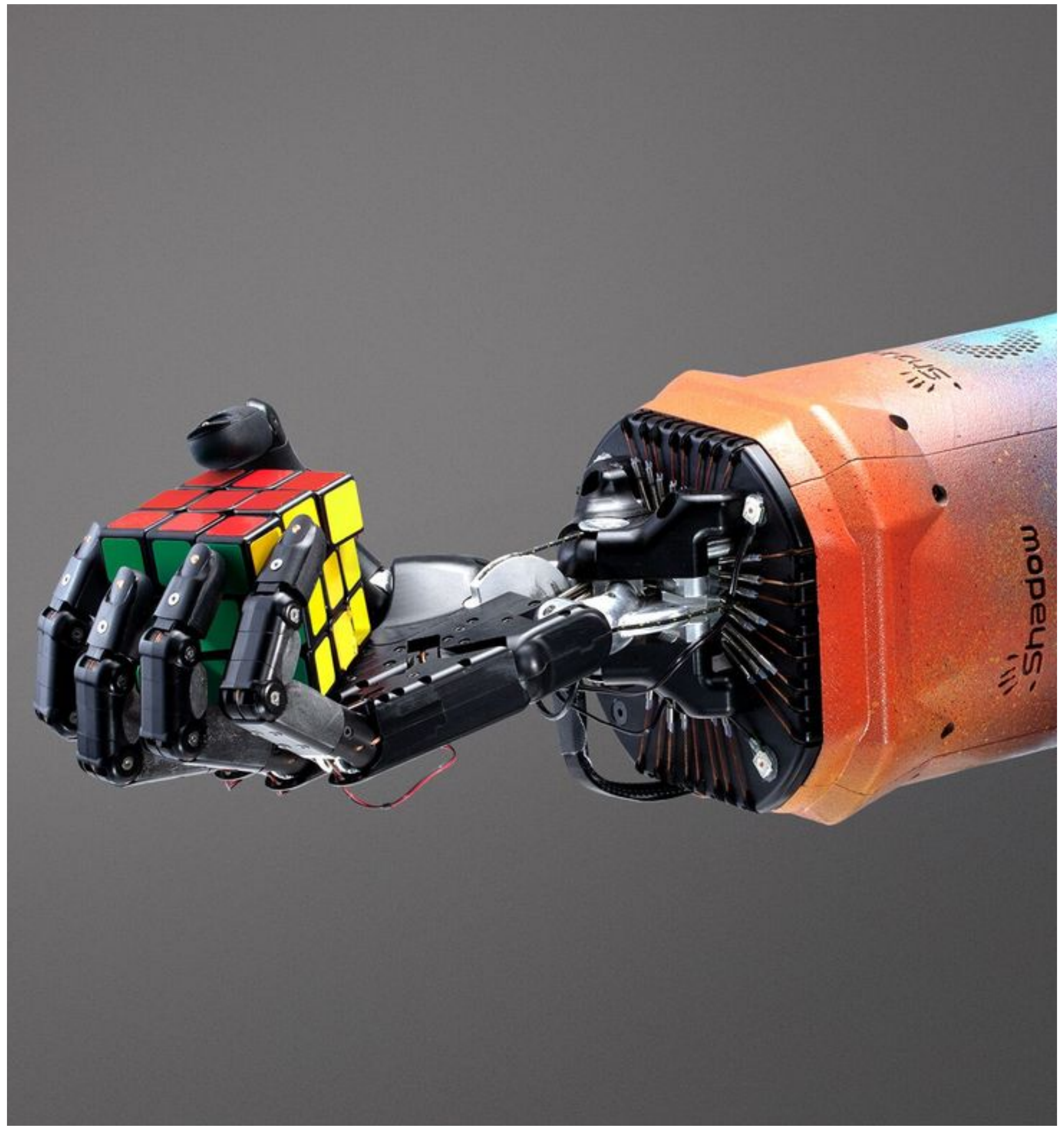
Reinforcement Learning in Real World:

Reinforcement Learning in Real World:



Reinforcement Learning in Real World:





To better understand RL,
let's recap Machine Learning 101

Recap: Supervised Learning

Recap: Supervised Learning

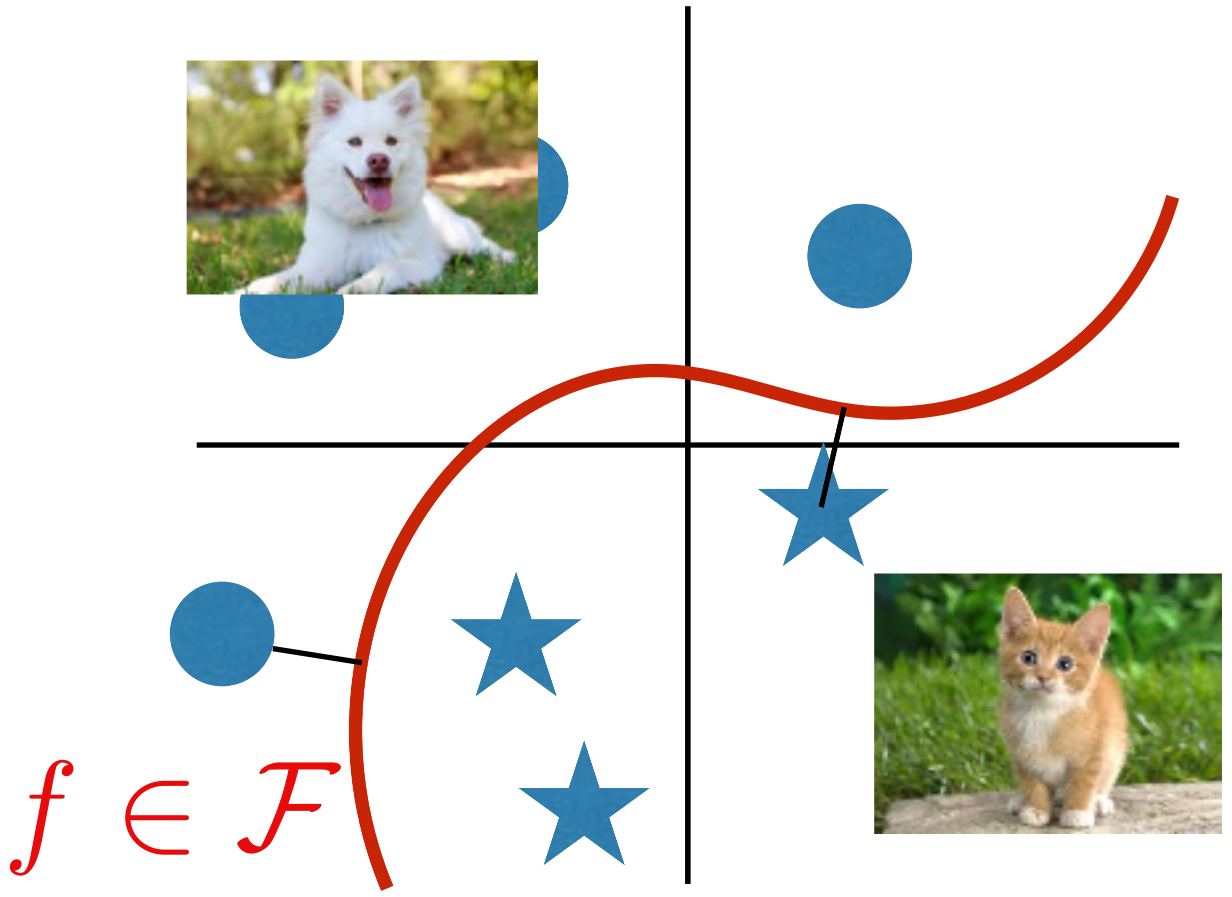
Given i.i.d examples at training:

( ,cat) ( ,cat) ( ,dog)

Recap: Supervised Learning

Given i.i.d examples at training:

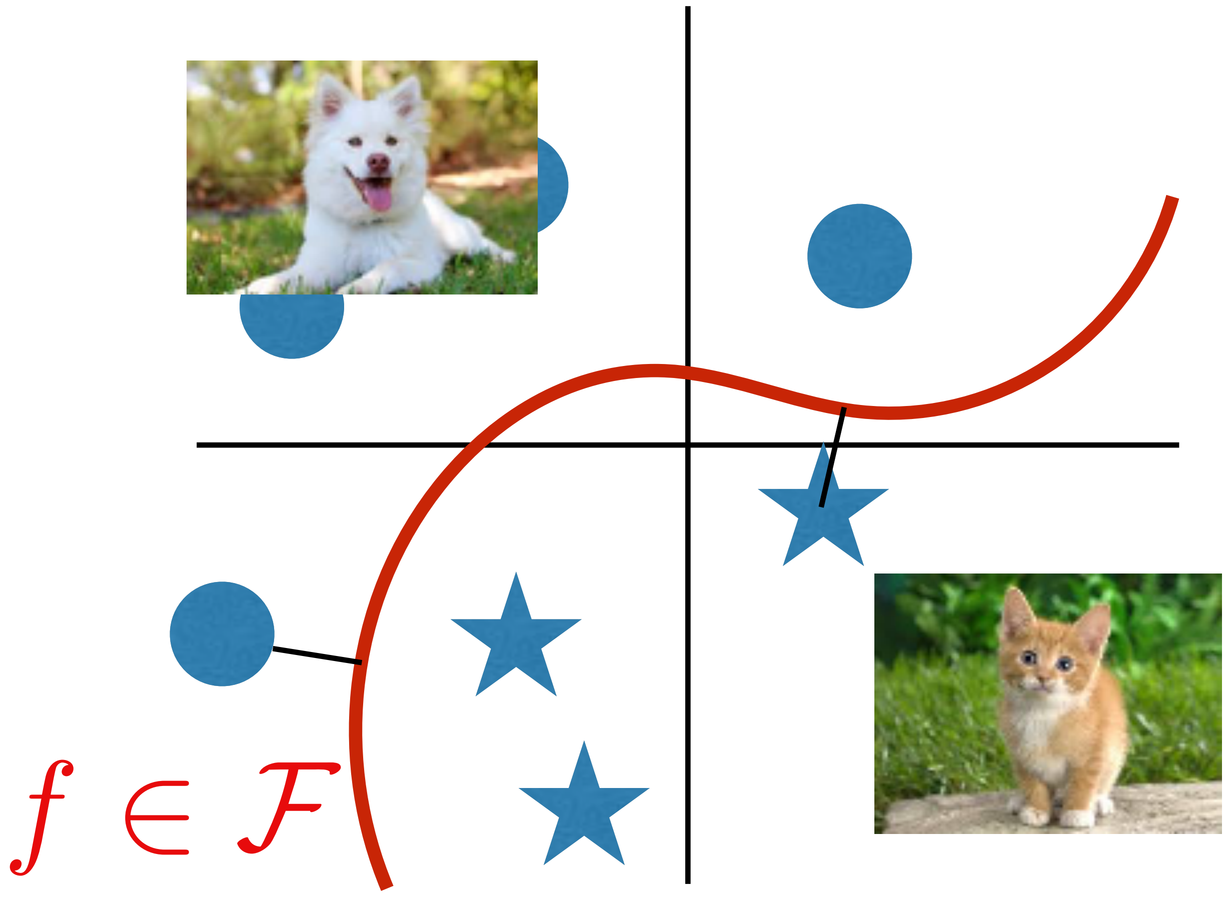
( ,cat) ( ,cat) ( ,dog)



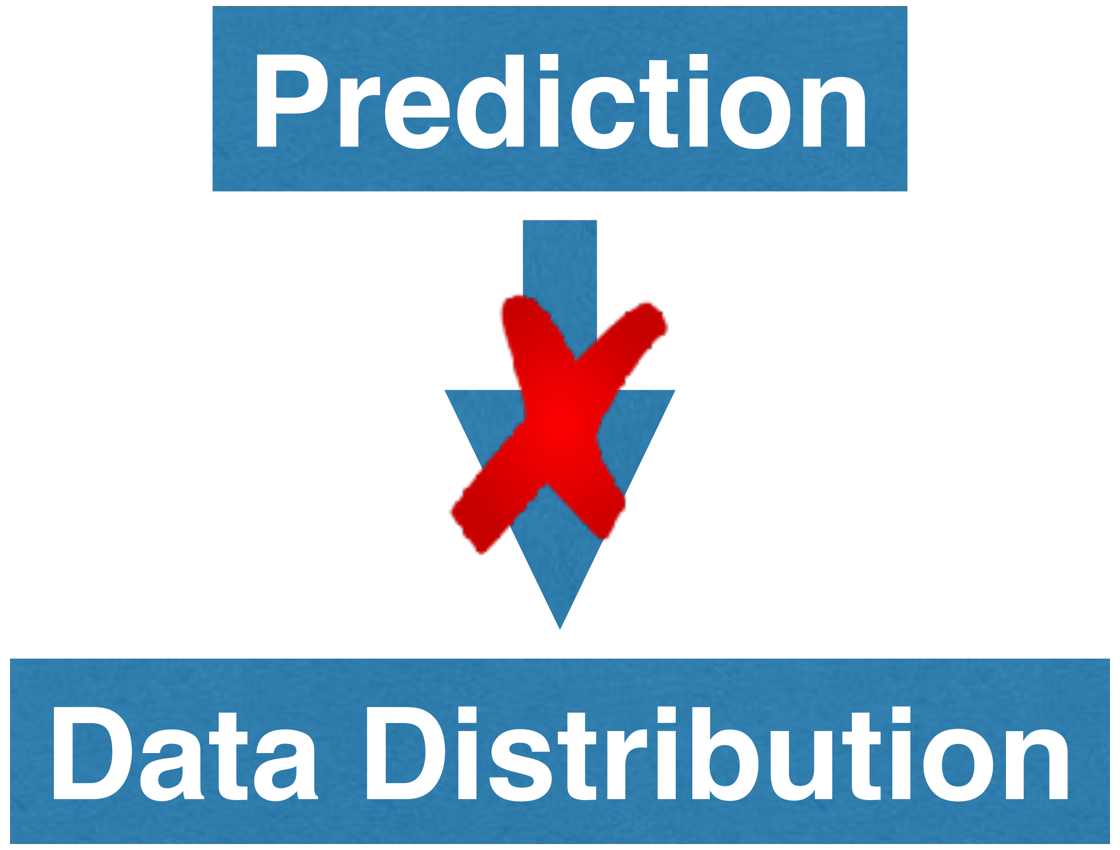
Recap: Supervised Learning

Given i.i.d examples at training:

( ,cat) ( ,cat) ( ,dog)



Passive:




AgentLinear
Selected Actions:
RIGHT
SPEED
Active:
Decisions
Data Distribution

AgentLinear
Selected Actions:

RIGHT SPEED

Active: Decisions → Data Distribution

AgentLinear

Selected Actions:

RIGHT SPEED

Active: Decisions → Data Distribution

Summary so far:

1. In RL, we often start from zero data

Summary so far:

1. **In RL, we often start from zero data**

2. In RL, **decisions/predictions have consequences:**
Future data is determined by our past historical decisions/predictions

Summary so far:

1. **In RL, we often start from zero data**

2. In RL, **decisions/predictions have consequences:**
Future data is determined by our past historical decisions/predictions

3. To solve the task, we often need to make a **long sequence of decisions**

Outlines:

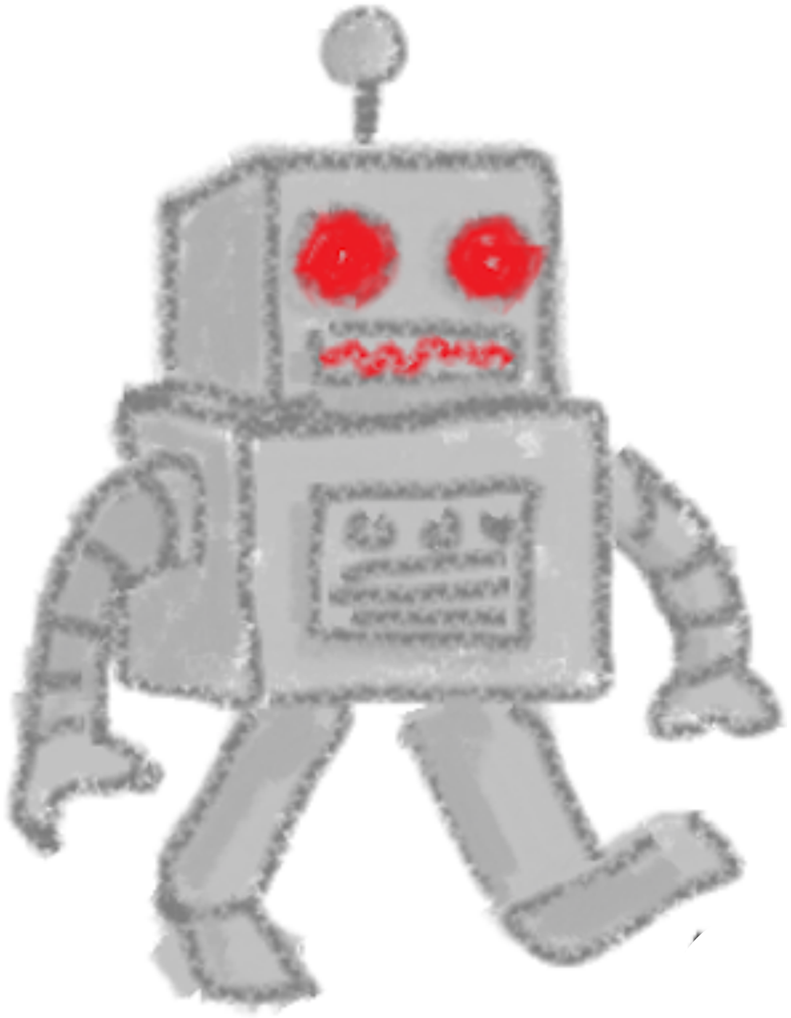
1. Introduction: Applications of RL, RL versus Supervised Learning

2. Basics of Markov Decision Process (MDP): model, example, V & Q functions

The Mathematical framework:

Markov Decision Process

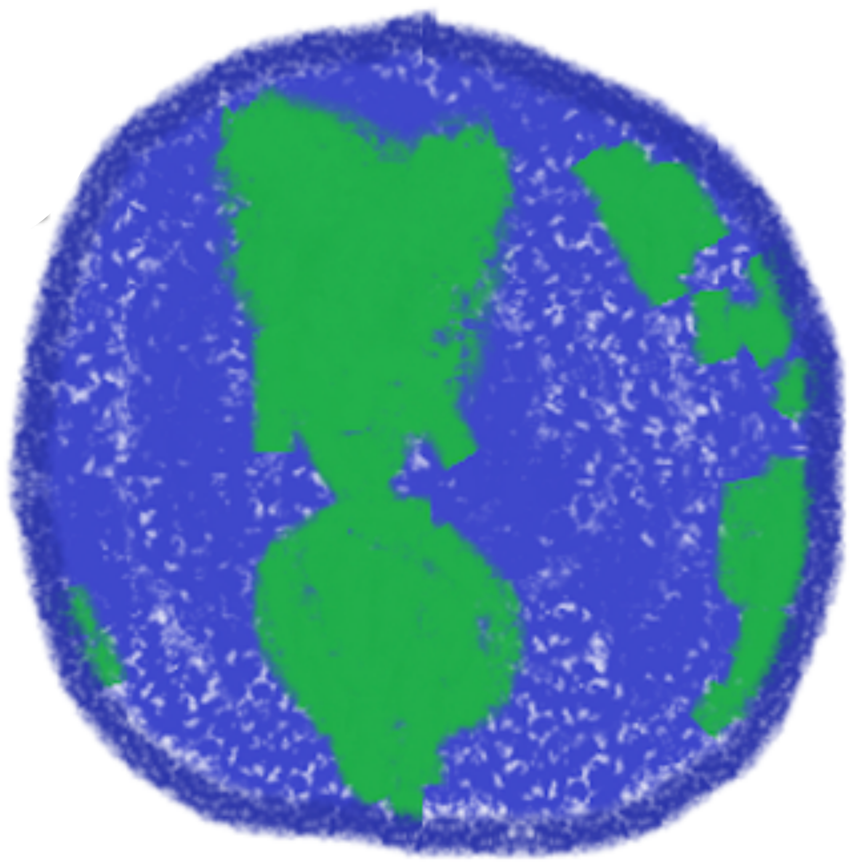
Learning Agent



$$\pi(s) \rightarrow a$$

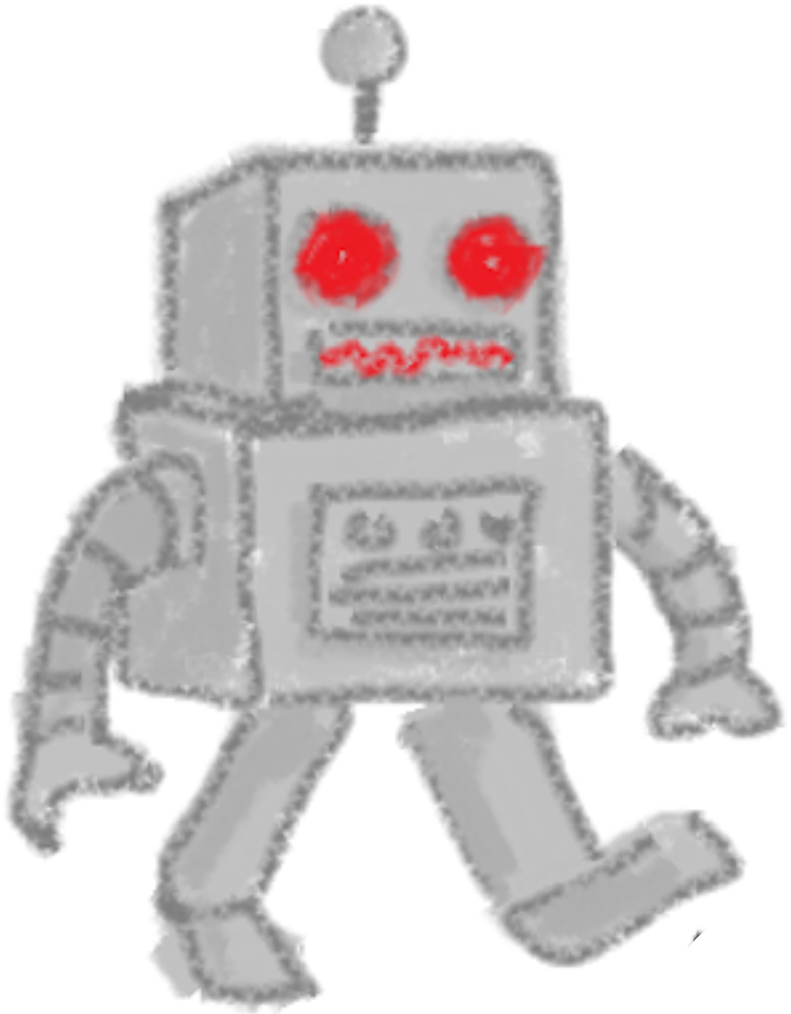
Policy: determine **action** based on **state**

Environment



The Mathematical framework:
Markov Decision Process

Learning Agent



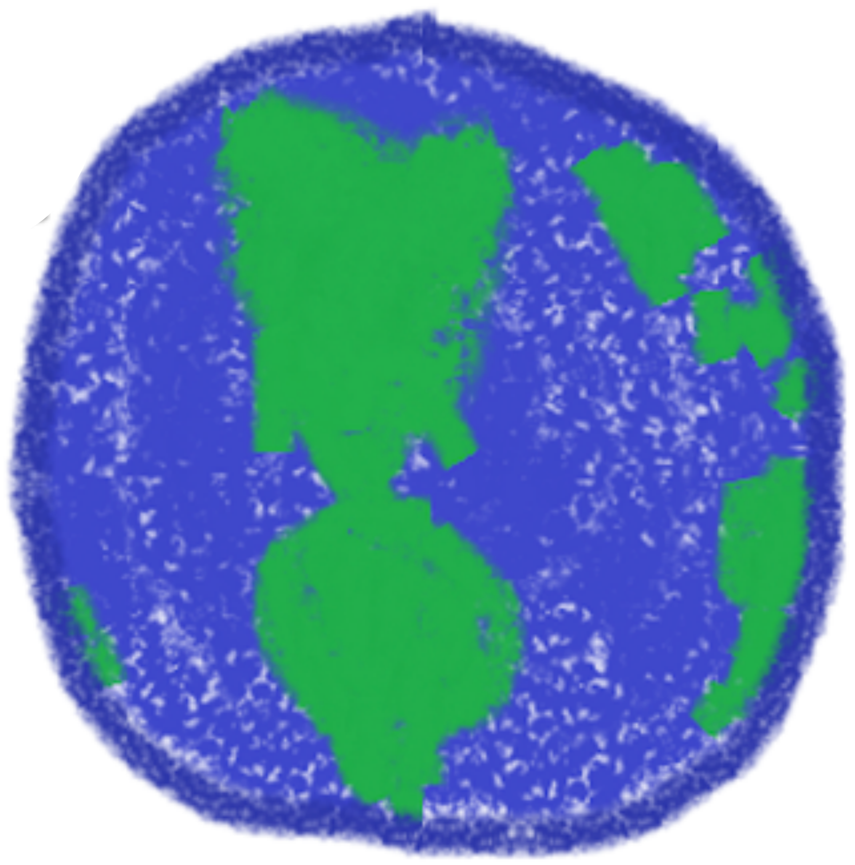
$$\pi(s) \rightarrow a$$

Policy: determine **action** based on **state**

Send **reward** and **next state** from a Markovian transition dynamics

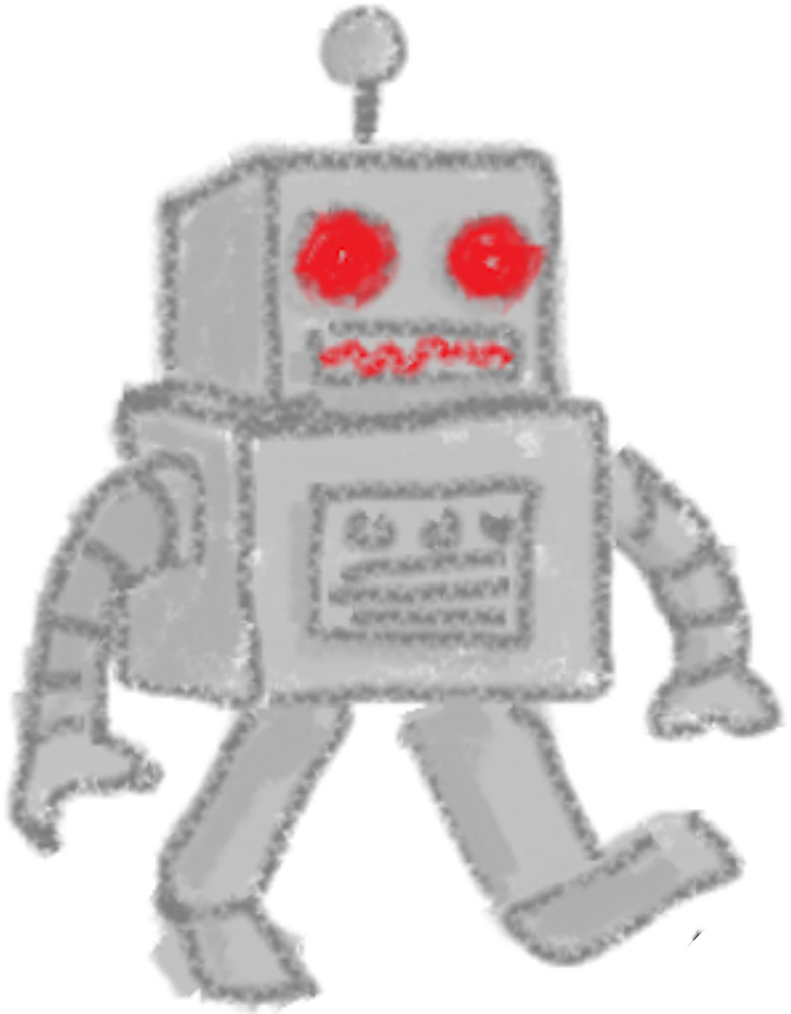
$$r(s,a), s' \sim P(\cdot \mid s,a)$$

Environment



The Mathematical framework:
Markov Decision Process

Learning Agent



$$\pi(s) \rightarrow a$$

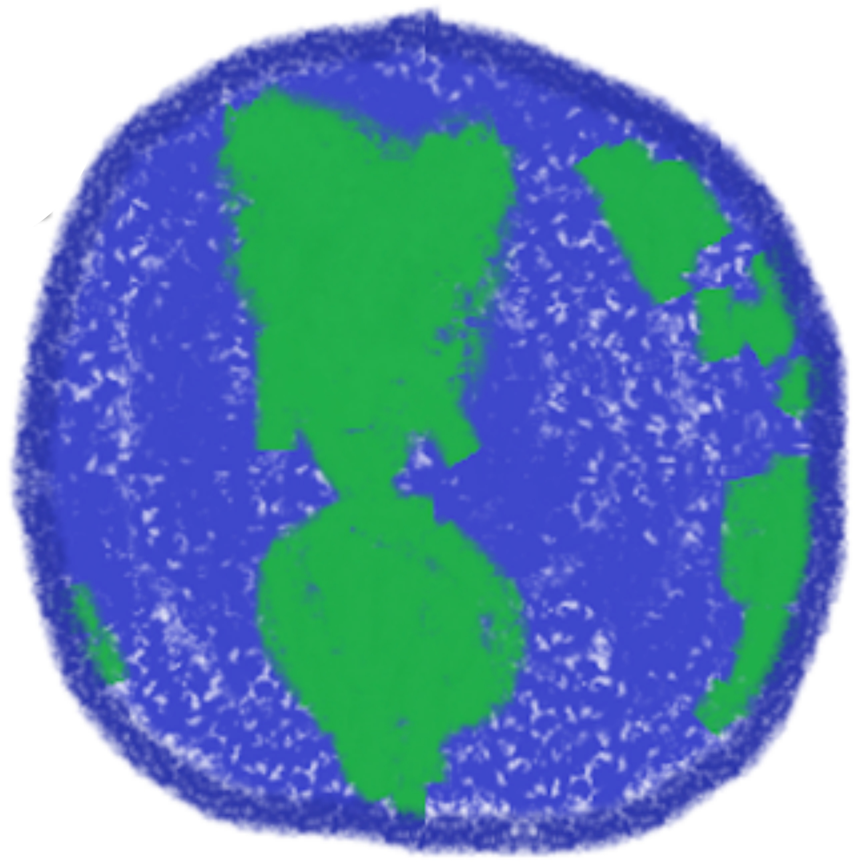
Policy: determine **action** based on **state**

Multiple Steps

Send **reward** and **next state** from a Markovian transition dynamics

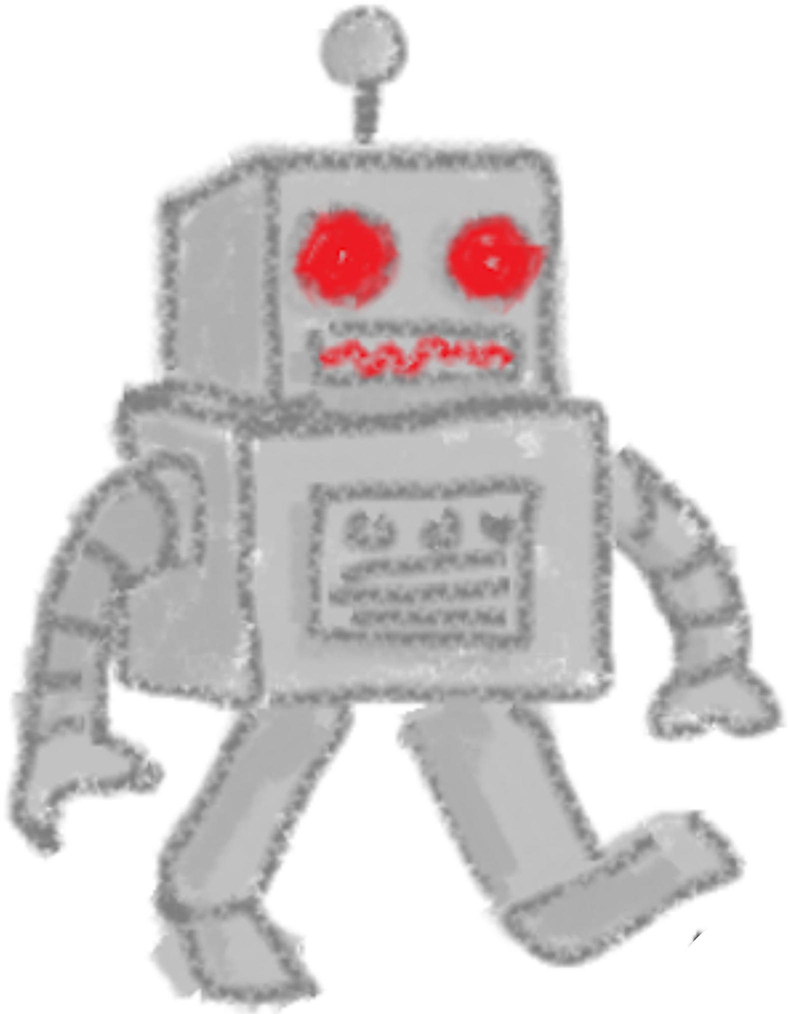
$$r(s, a), s' \sim P(\,\cdot\,|\,s, a)$$

Environment



The Mathematical framework:
Markov Decision Process

Learning Agent



$$\pi(s) \rightarrow a$$

Policy: determine **action** based on **state**

Multiple Steps

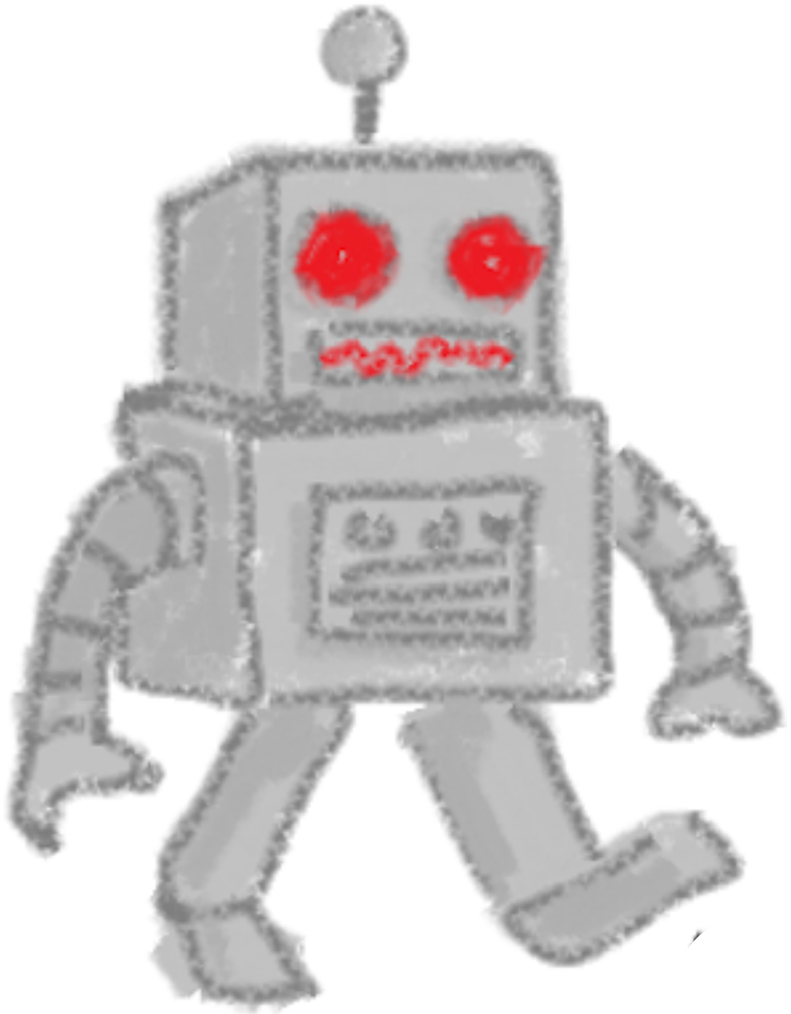
Environment

Send **reward** and **next state** from a Markovian transition dynamics

$$r(s, a), s' \sim P(\cdot \mid s, a)$$

The Mathematical framework:
Markov Decision Process

Learning Agent



$$\pi(s) \rightarrow a$$

Policy: determine **action** based on **state**

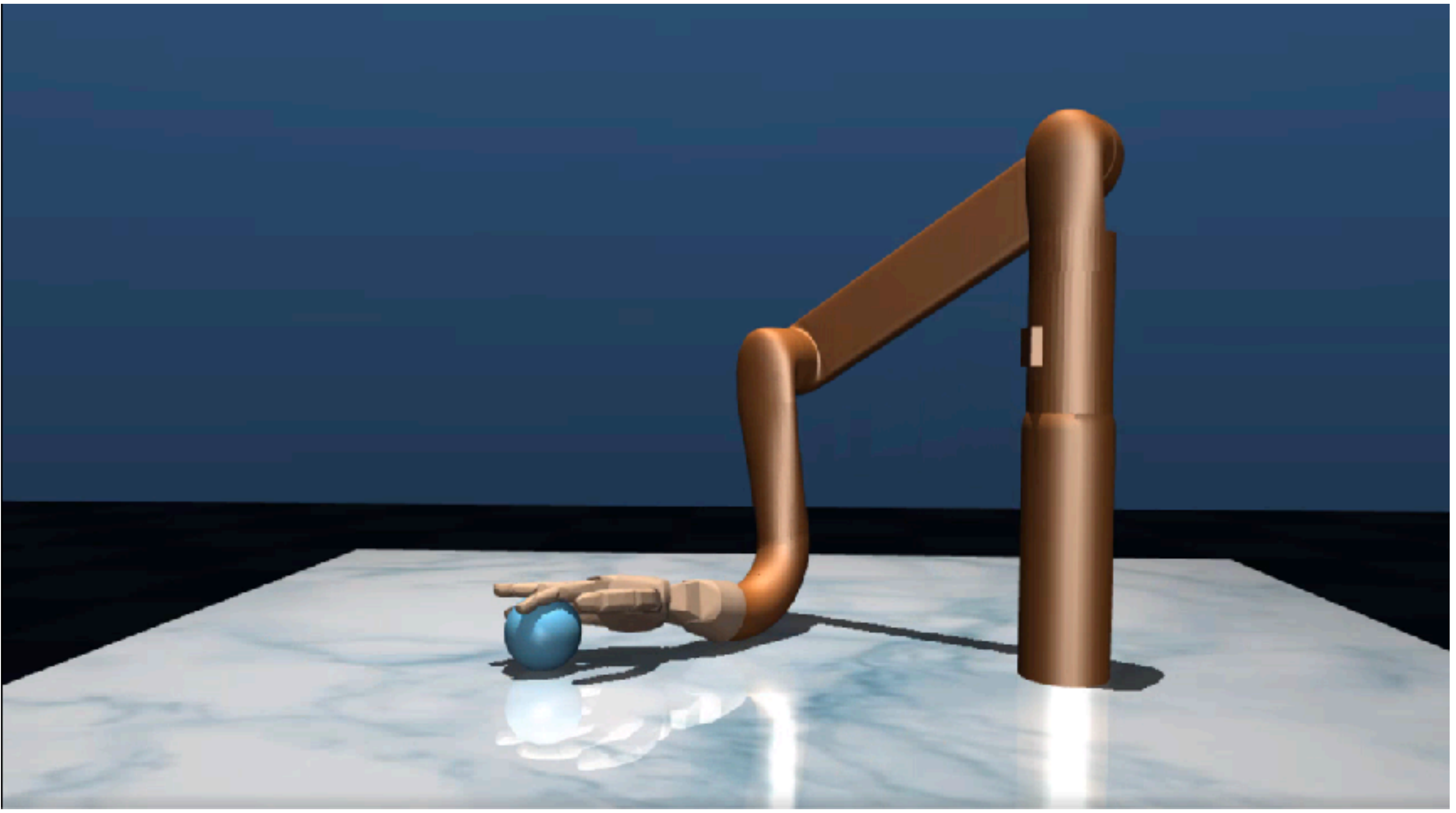
Multiple Steps

Send **reward** and **next state** from a Markovian transition dynamics

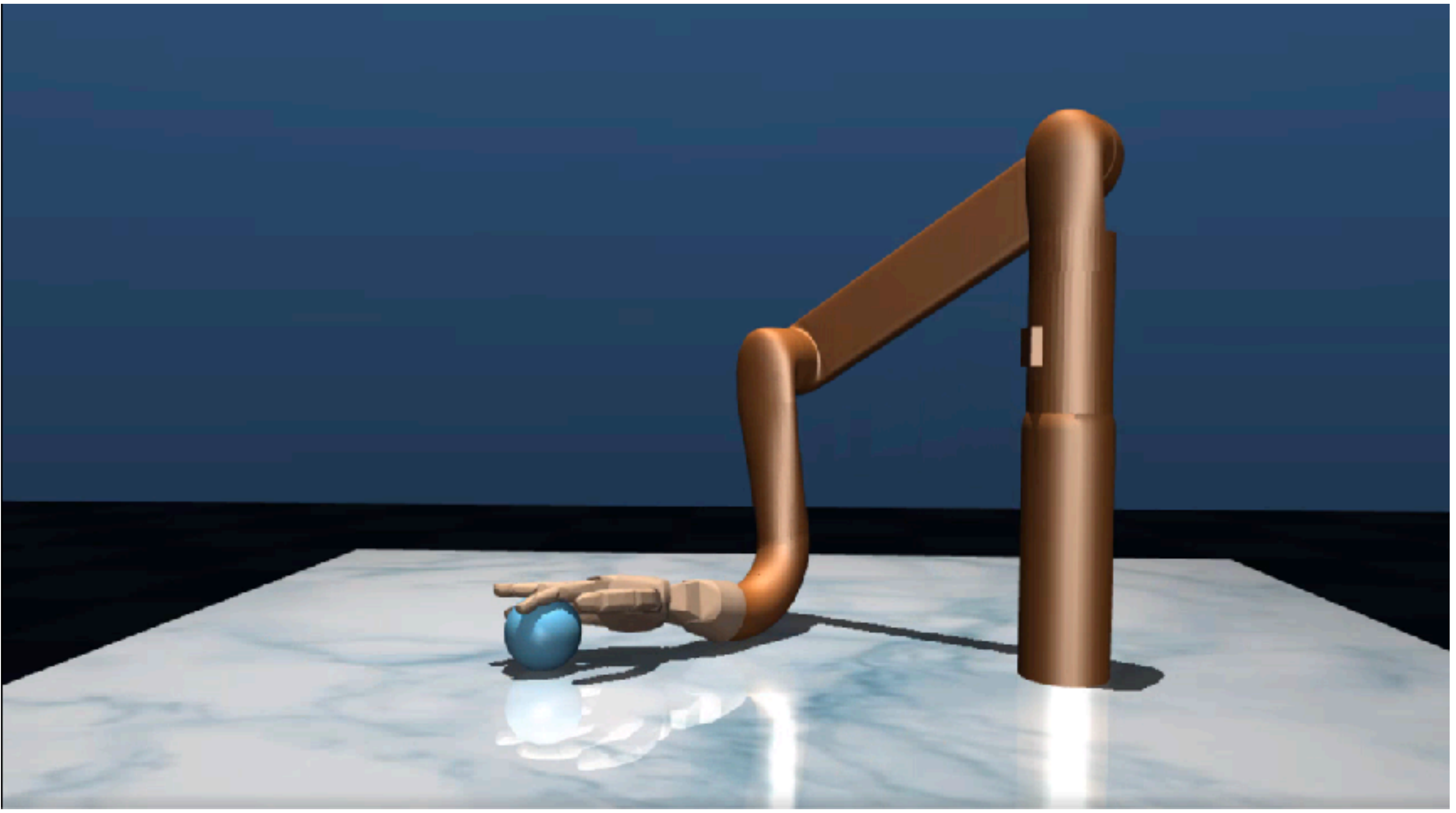
$$r(s, a), s' \sim P(\cdot \mid s, a)$$

Environment

Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position

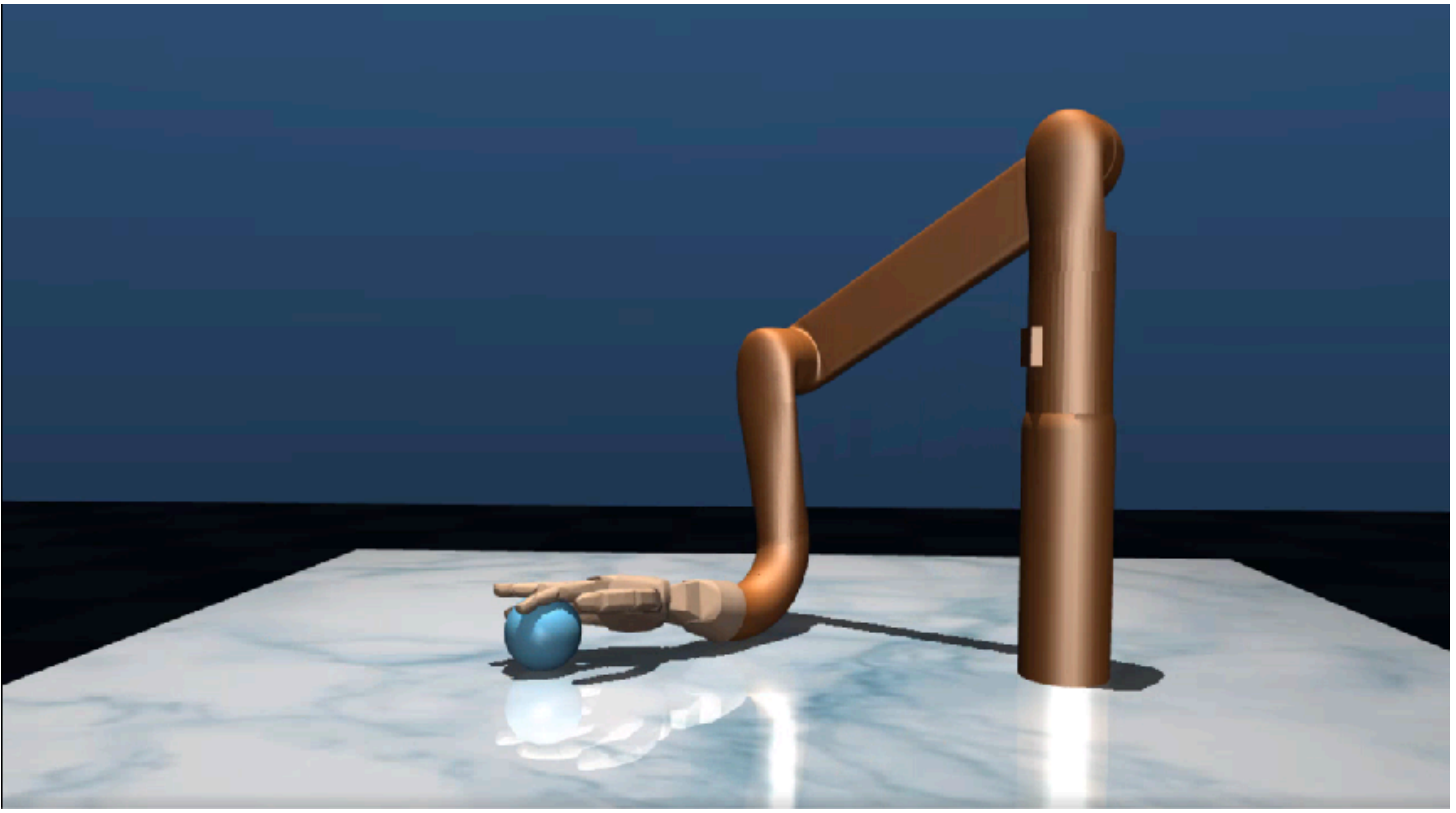


Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position



State s: robot configuration (e.g., joint angles) and the ball's position

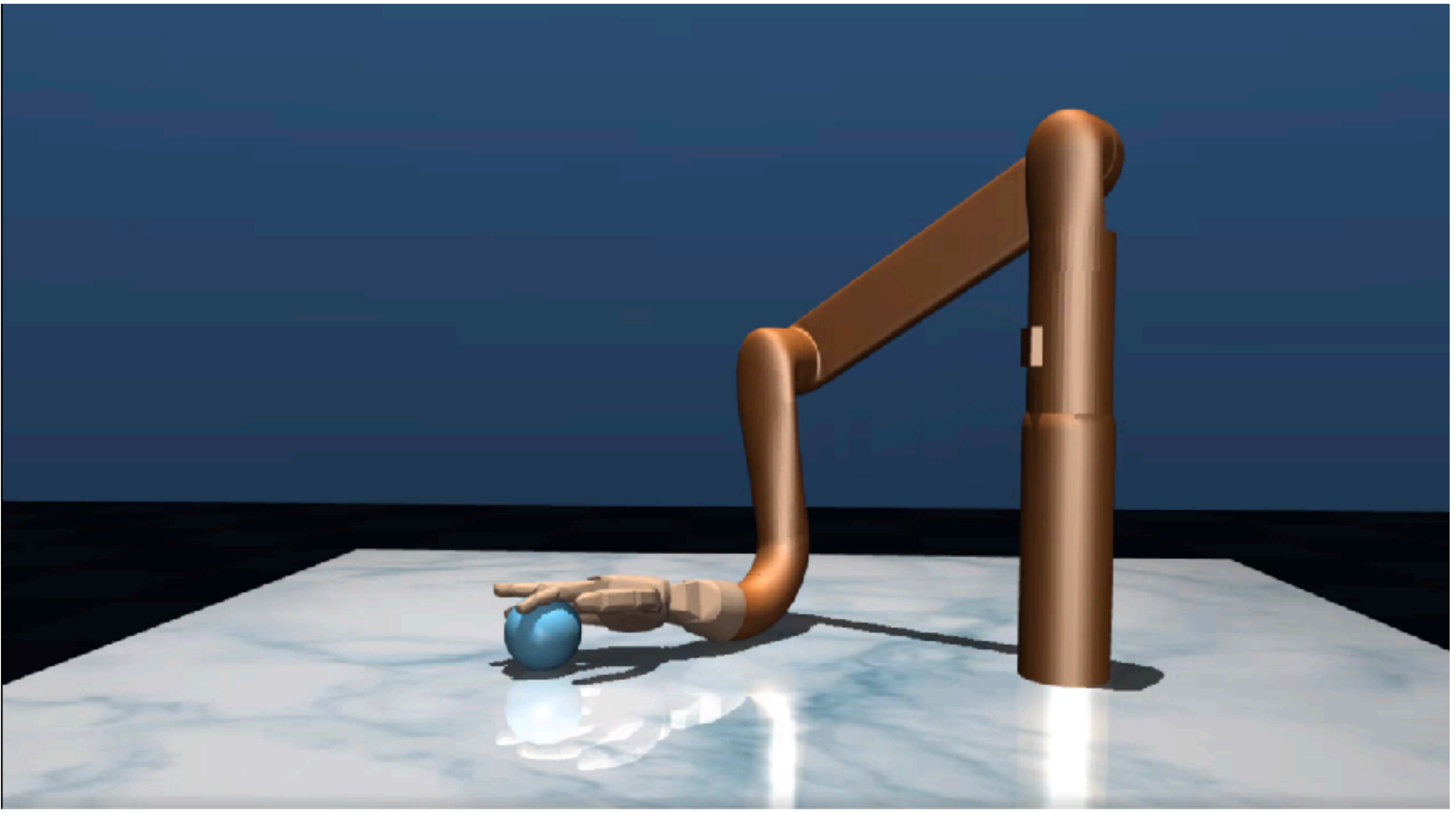
Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position



State s: robot configuration (e.g., joint angles) and the ball's position

Action a: Torque on joints in arm & fingers

Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position

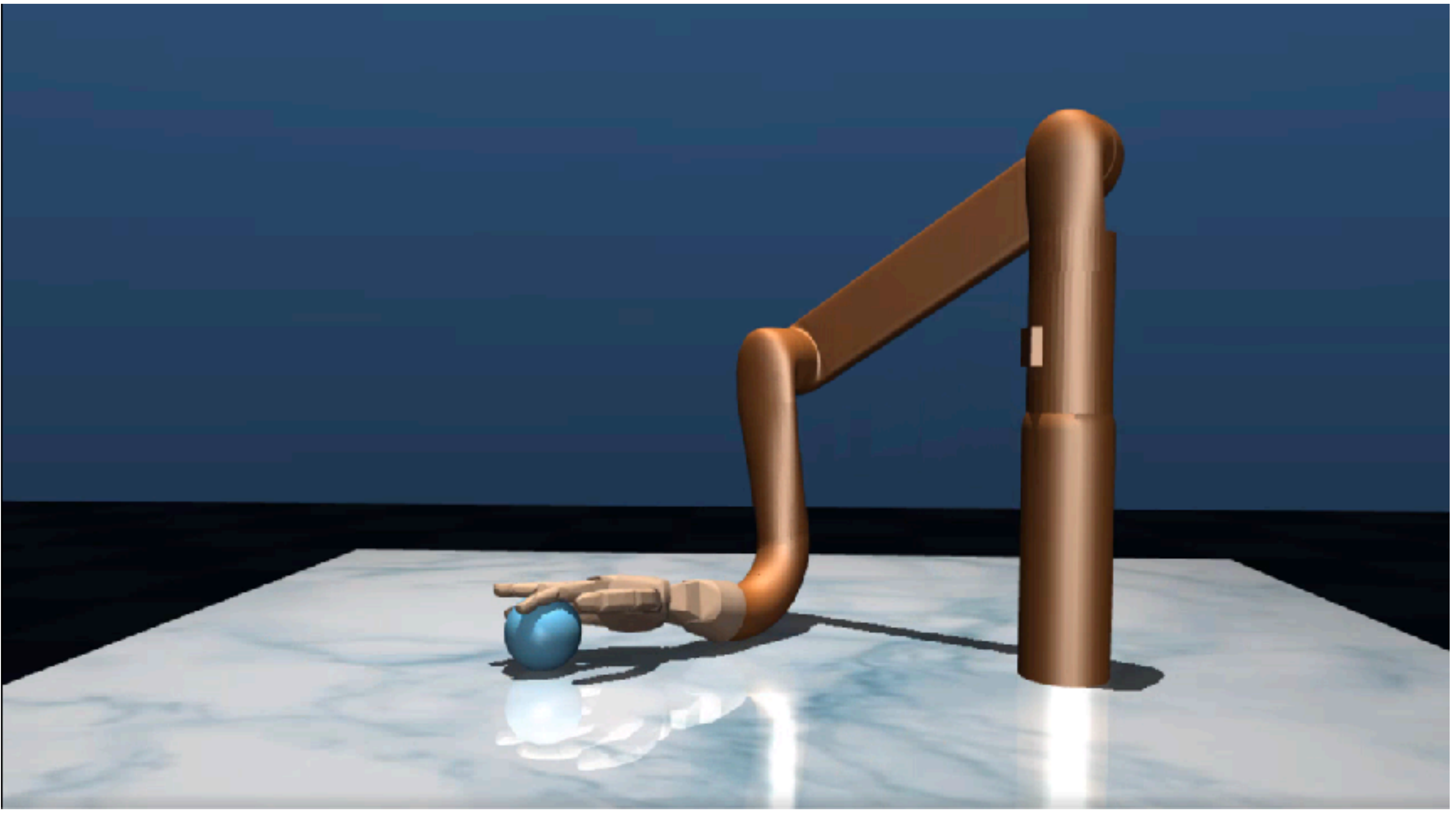


State s: robot configuration (e.g., joint angles) and the ball's position

Action a: Torque on joints in arm & fingers

Transition $s' \sim P(\cdot \,|\, s, a)$: physics + some noise

Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position



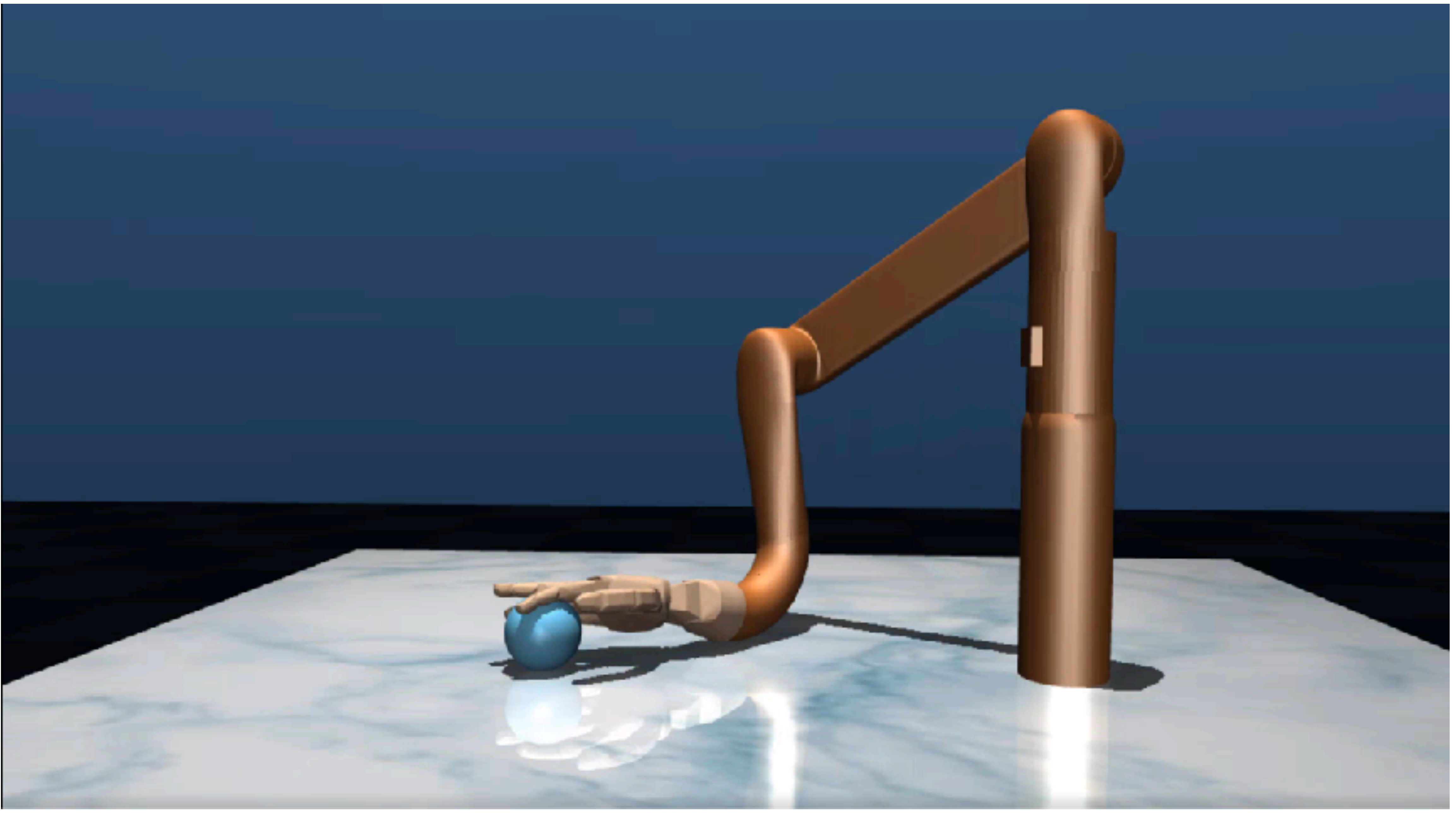
State s: robot configuration (e.g., joint angles) and the ball's position

Action a: Torque on joints in arm & fingers

Transition $s' \sim P(\cdot \mid s, a)$: physics + some noise

policy $\pi(s)$: a function mapping from robot state to action (i.e., torque)

Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position



State s: robot configuration (e.g., joint angles) and the ball's position

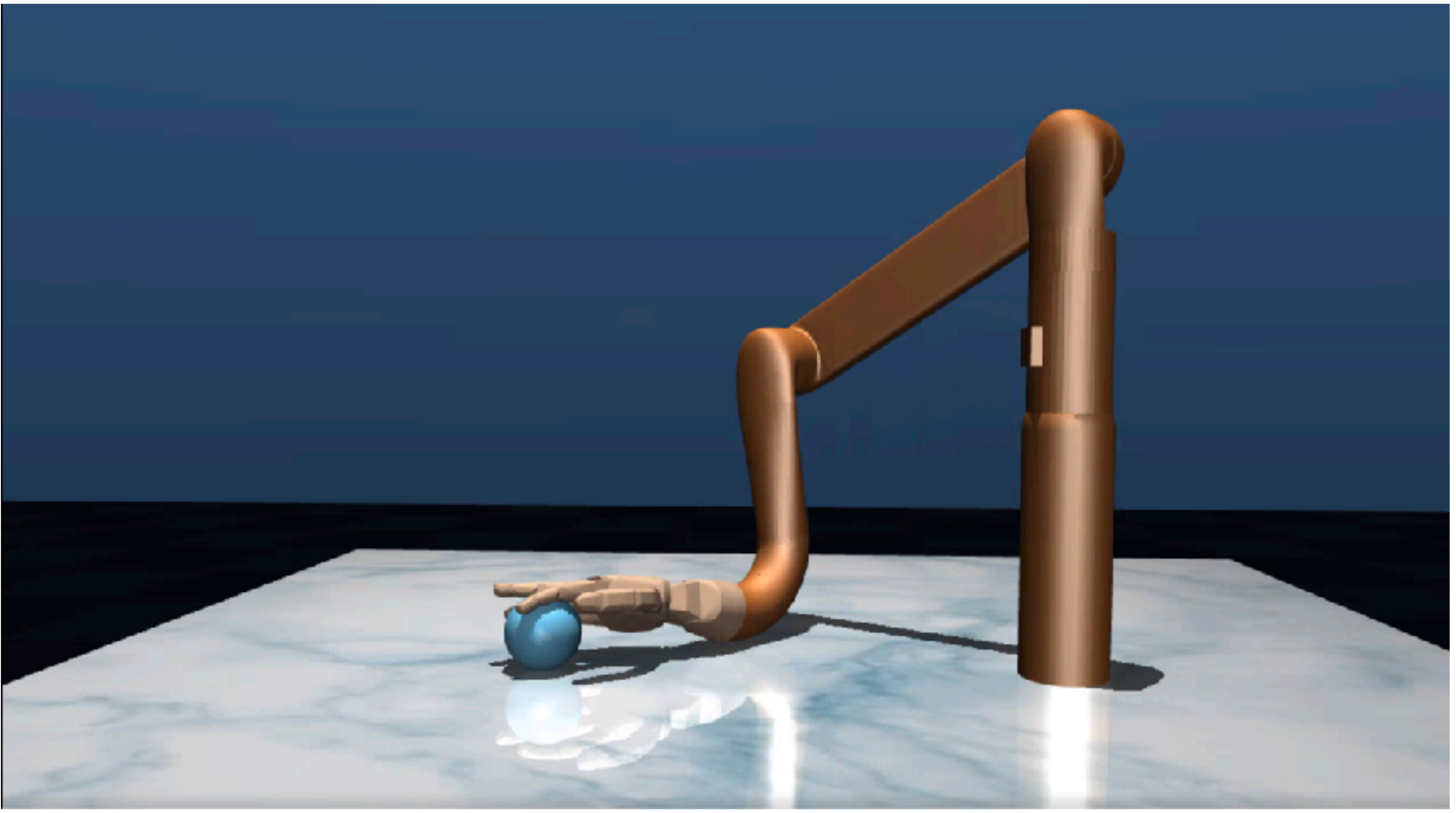
Action a: Torque on joints in arm & fingers

Transition $s' \sim P(\cdot \mid s, a)$: physics + some noise

policy $\pi(s)$: a function mapping from robot state to action (i.e., torque)

<u>**Cost**</u> $c(s, a)$: torque magnitude + dist to goal

Example:
robot hand needs to pick the ball and hold it in a goal (x,y,z) position



State s: robot configuration (e.g., joint angles) and the ball's position

Action a: Torque on joints in arm & fingers

Transition $s' \sim P(\cdot \mid s, a)$: physics + some noise

policy $\pi(s)$: a function mapping from robot state to action (i.e., torque)

Cost $c(s, a)$: torque magnitude + dist to goal

$$\pi^\star = \arg\min_{\pi} \mathbb{E}\left[c(s_0, a_0) + \gamma c(s_1, a_1) + \gamma^2 c(s_2, a_2) + \gamma^3 c(s_3, a_3) + \ldots\ldots \,\middle|\, a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$$

Question:

Assume we have S many states, and A many actions, how many different polices there are?

Question:

Assume we have S many states, and A many actions, how many different polices there are?

(Hint: a policy is a mapping from s to a, we have A many choices per state s)

Infinite horizon Discounted Setting

$$\mathscr{M} = \{S, A, P, r, \gamma\}$$

$$P : S \times A \mapsto \Delta(S), \quad r : S \times A \to [0,1], \quad \gamma \in [0,1)$$

Infinite horizon Discounted Setting

$$\mathcal{M} = \{S, A, P, r, \gamma\}$$

$$P : S \times A \mapsto \Delta(S), \quad r : S \times A \to [0,1], \quad \gamma \in [0,1)$$

Policy $\pi : S \mapsto A$

Infinite horizon Discounted Setting

$$\mathcal{M} = \{S, A, P, r, \gamma\}$$

$$P : S \times A \mapsto \Delta(S), \quad r : S \times A \rightarrow [0,1], \quad \gamma \in [0,1)$$

Policy $\pi : S \mapsto A$

Quantities that allow us to reason policy's long-term effect:

Infinite horizon Discounted Setting

$$\mathcal{M} = \{S, A, P, r, \gamma\}$$

$$P : S \times A \mapsto \Delta(S), \quad r : S \times A \to [0,1], \quad \gamma \in [0,1)$$

Policy $\pi : S \mapsto A$

Quantities that allow us to reason policy's long-term effect:

Value function $V^\pi(s) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \,\middle|\, s_0 = s, a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$

Infinite horizon Discounted Setting

$$\mathscr{M} = \{S, A, P, r, \gamma\}$$

$$P : S \times A \mapsto \Delta(S), \quad r : S \times A \to [0,1], \quad \gamma \in [0,1)$$

Policy $\pi : S \mapsto A$

Quantities that allow us to reason policy's long-term effect:

Value function $V^\pi(s) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| s_0 = s, a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$

Q function $Q^\pi(s, a) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| (s_0, a_0) = (s, a), a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$

Understanding Value function and Q functions

Value function $V^\pi(s) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| s_0 = s, a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$

Understanding Value function and Q functions

Value function $V^\pi(s) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| s_0 = s, a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$

Q function $Q^\pi(s, a) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| (s_0, a_0) = (s, a), a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$

Bellman Equation for V-function:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \middle| s_0 = s, a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$$

Bellman Equation for V-function:

$$V^\pi(s) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \,\middle|\, s_0 = s, a_h = \pi(s_h), s_{h+1} \sim P(\,\cdot\,|\,s_h, a_h)\right]$$

$$V^\pi(s) = r(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} V^\pi(s')$$

Bellman Equation for Q-function:

Bellman Equation for Q-function:

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \,\middle|\, (s_0, a_0) = (s,a), a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$$

Bellman Equation for Q-function:

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \,\middle|\, (s_0, a_0) = (s,a), a_h = \pi(s_h), s_{h+1} \sim P(\cdot \mid s_h, a_h)\right]$$

$$Q^\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} V^\pi(s')$$

Summary:

- **RL is different from Supervised Learning:**
  - Our actions have consequences
  - Need to make sequence of decisions to complete the task

- **Discounted infinite horizon MDP:**
  - State, action, policy, transition, reward (or cost), discount factor
  - **V function and Q function**
  - Key concept: **Bellman equation**