Robot Learning

Lecture 1:
Introduction to Robot Learning

Dr Edward Johns

Monday 16th January 2023

Lecture 1: Introduction to Robot Learning

Contents

Part 1: What is robot learning?

Part 2: Robot learning success stories

Part 3: The different types of robot learning

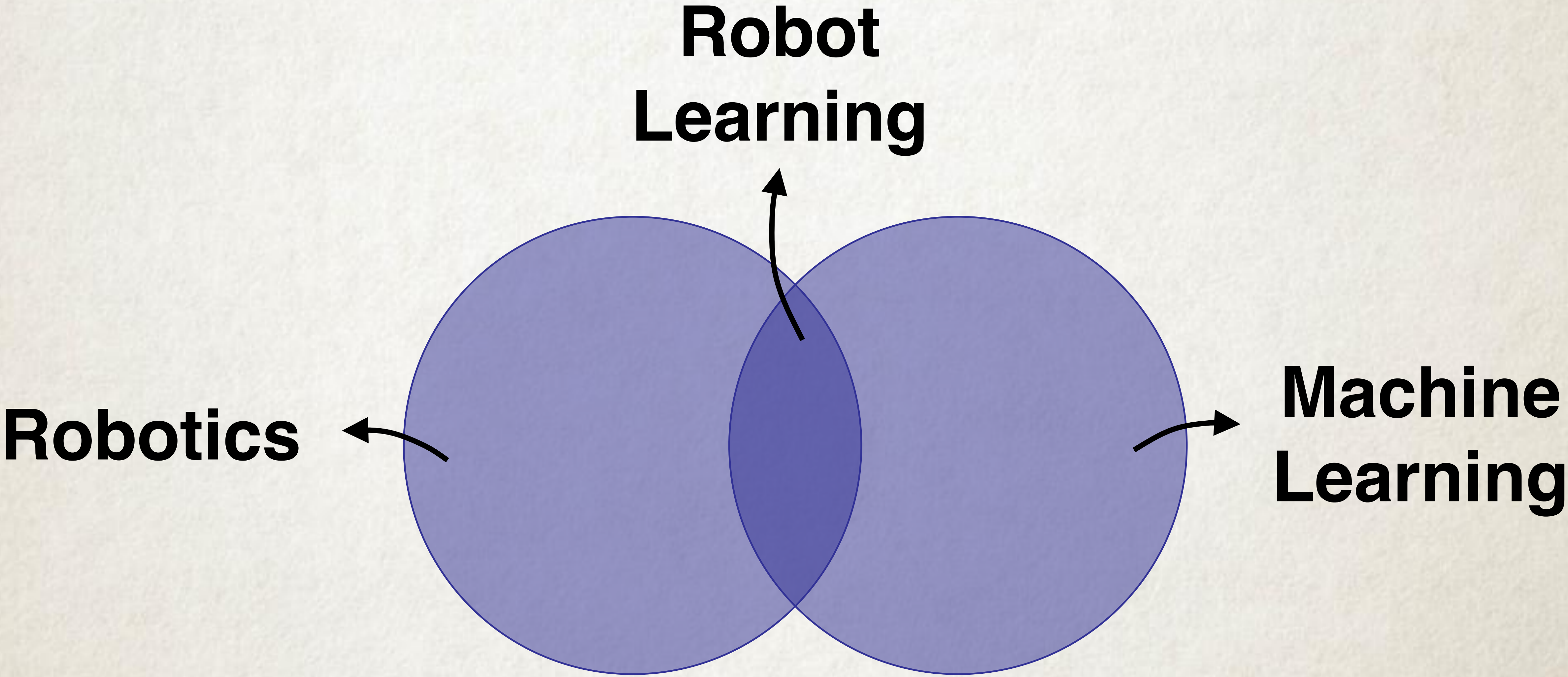
Part 4: Overview of the module

Part 5: States, observations, actions, and policies

Part 6: Task specification

Part 7: Labs and coursework

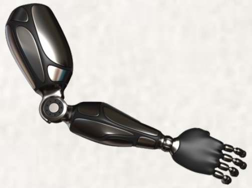
Robot Learning = Robotics + Machine Learning



What is a robot?

A robot can be defined as something which can do the following three things:

- 1) Sensing of the environment.
  - E.g. some hardware converts physical external stimulus into electrical signals.



- 2) Acting in the environment.
  - E.g. some hardware moves in order to physically change the environment.
- 3) Decision making (but not hand-coded for every scenario).
  - E.g. a computer calculates how to convert sensory data to actions.

What is a robot?

		Sensing	Acting	Decision Making
	1	No	No	No
	2	No	No	Yes
	3	No	Yes	No
	4	Yes	No	No
	5	No	Yes	Yes
	6	Yes	No	Yes
	7	Yes	Yes	No
	8	Yes	Yes	Yes

What is a robot?

		Sensing	Acting	Decision Making
Lightbulb	1	No	No	No
ChatGPT	2	No	No	Yes
Dishwasher	3	No	Yes	No
Camera	4	Yes	No	No
???	5	No	Yes	Yes
Amazon Alexa	6	Yes	No	Yes
Vending machine	7	Yes	Yes	No
C-3PO	8	Yes	Yes	Yes

What is machine learning?

First five responses from ChatGPT:

- *Machine learning is a method of teaching computers to learn from data, without being explicitly programmed.*
- *Machine learning is a branch of artificial intelligence that involves the development of algorithms and statistical models that enable machines to learn from data and make predictions or decisions without being explicitly programmed to do so.*
- *Machine learning is a type of artificial intelligence that allows systems to learn and improve from experience without being explicitly programmed.*
- *Machine learning is a type of artificial intelligence (AI) that allows systems to learn and improve from experience without being explicitly programmed.*
- *Machine learning is a subfield of artificial intelligence that involves the development of algorithms and statistical models that enable a computer to learn from data and improve its performance on a task without being explicitly programmed.*

What is machine learning?

ChatGPT's response for "*Write one sentence that describes what is common amongst the following five sentences: (1) … (2) … (3) … (4) … (5) …* " :

> *Machine learning is a subfield of artificial intelligence that involves the development of **algorithms and statistical models** that enable a **computer to learn from data** and improve its performance on a task **without being explicitly programmed**.*

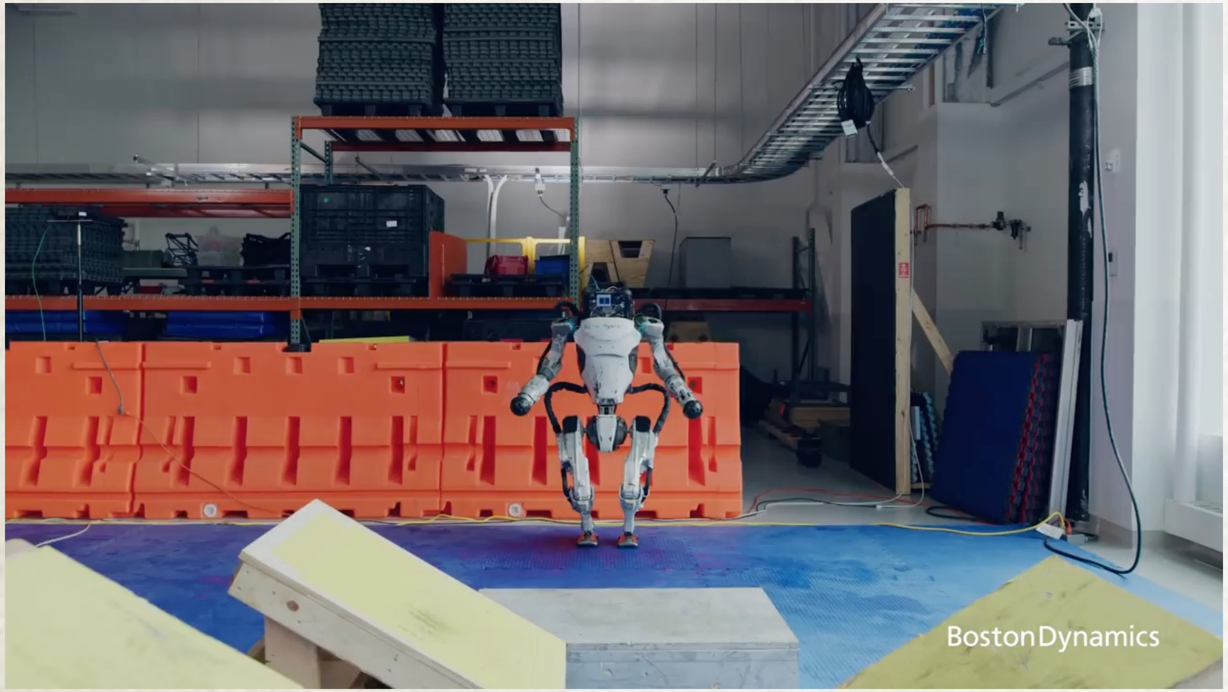
Finally… What is robot learning?

Machine learning: *a subfield of artificial intelligence that involves the development of* ***algorithms and statistical models*** *that enable a* ***computer to learn from data*** *and improve its performance on a task* ***without being explicitly programmed****.*

Robot learning: *the development of* ***algorithms and statistical models*** *that enable a* ***robot to learn from data*** *and improve its performance on a task* ***without being explicitly programmed*** *… where a robot is something which can* ***sense, act, and make decisions****.*

Why do we need robot learning?

We don't always need robot learning …
Some robots can be explicitly programmed.



Boston Dynamics Atlas robot

BMW car factory

Why do we need robot learning?

But explicit programming is often unsuitable when:

- The task is too complex for human engineers to manually design a controller. E.g. the object's dynamics are complex, the robot's sensors are noisy, or the robot's motors are inconsistent.
- It is too time consuming to manually design a controller for every feasible scenario. E.g. grasping any object in your home, stacking dishes in any dishwasher, chopping up any carrot.

If it is easier to design a learning method and then collect data, then robot learning is more optimal than explicit programming.

Lecture 1: Introduction to Robot Learning

Contents

Part 1: What is robot learning?

Part 2: Robot learning success stories

Part 3: The different types of robot learning

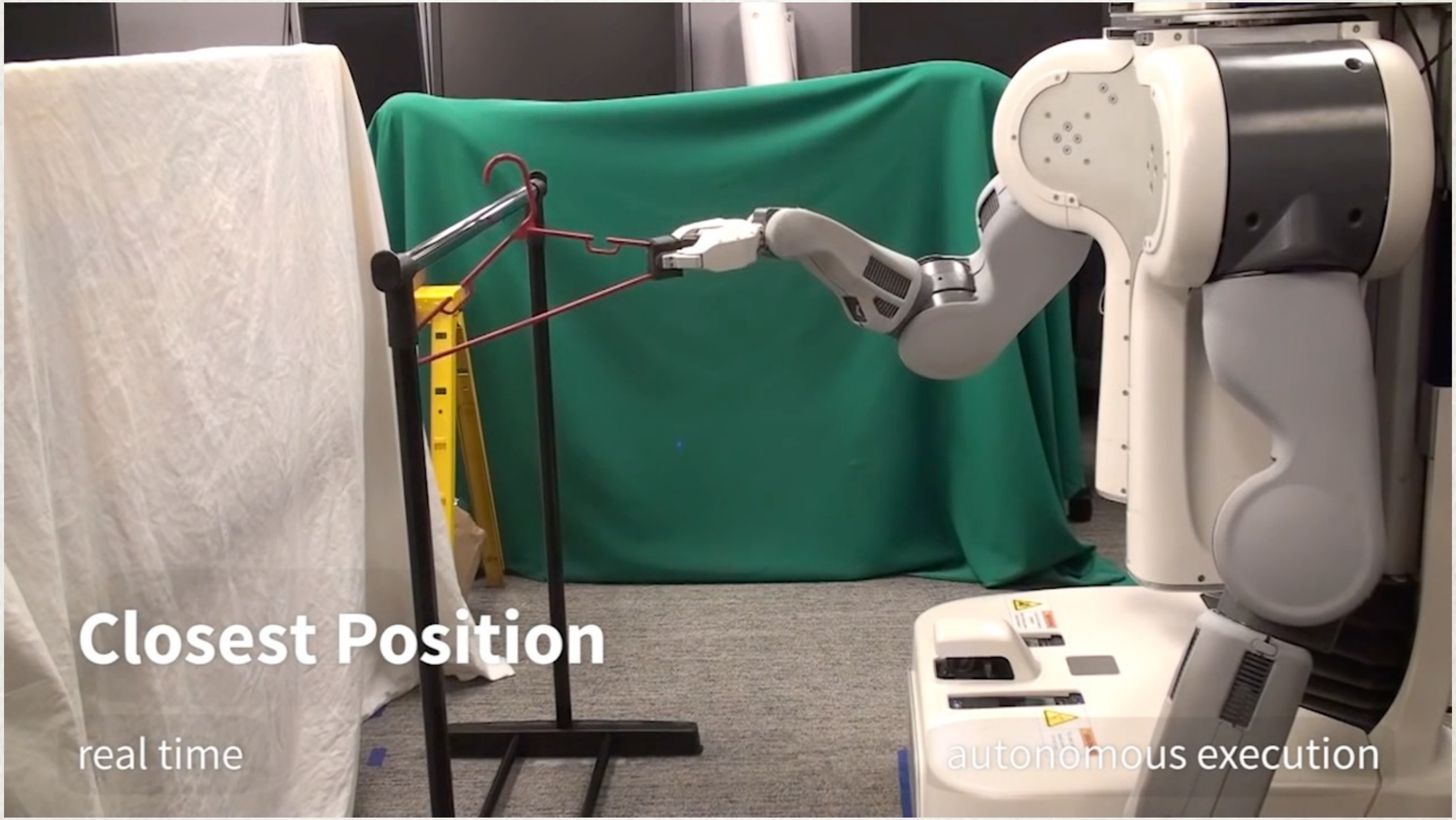
Part 4: Overview of the module

Part 5: States, observations, actions, and policies

Part 6: Task specification

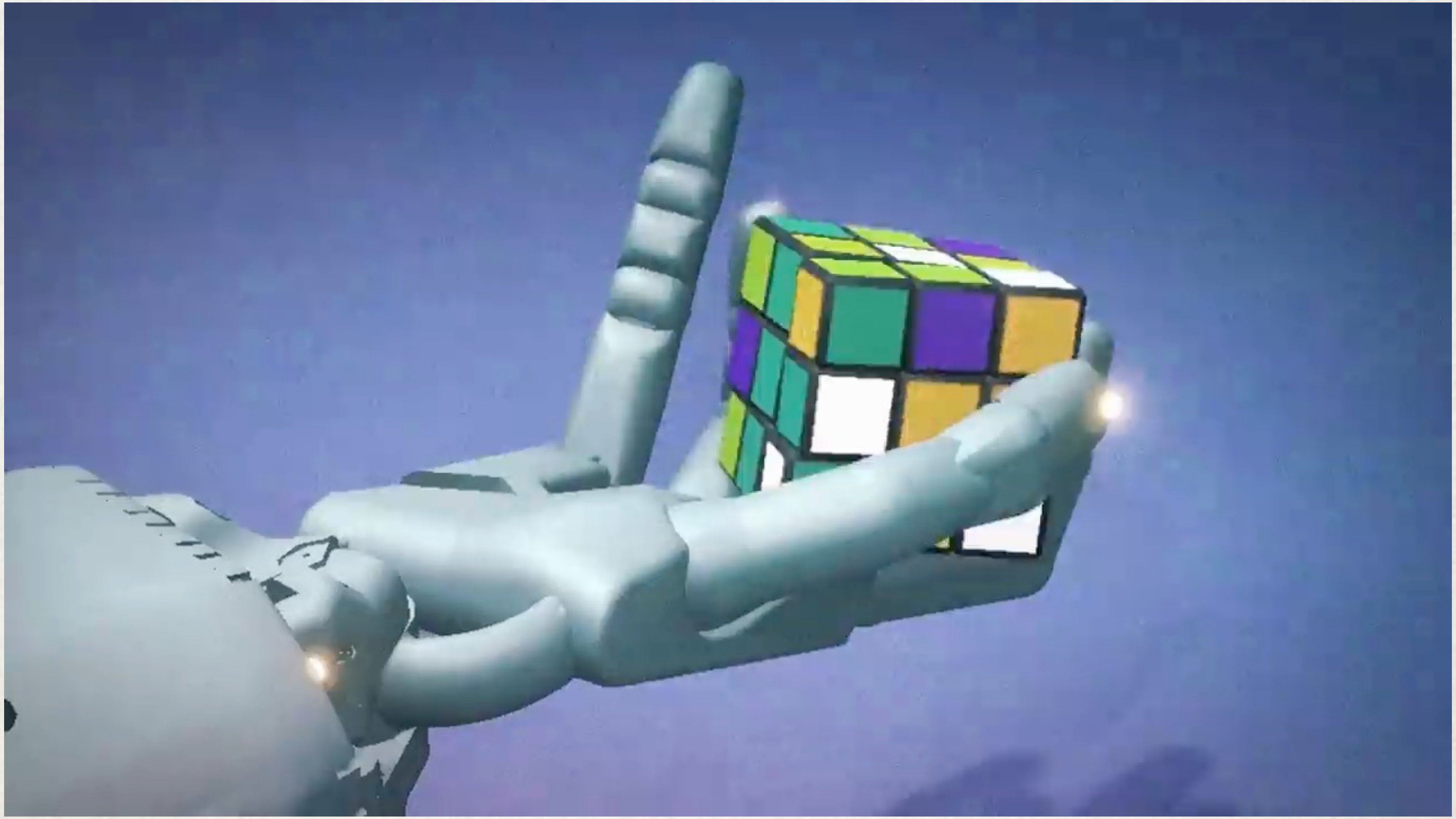
Part 7: Labs and coursework

Reinforcement learning



Levine et al., "End-to-end training of deep visuomotor policies", 2016

Sim-to-real

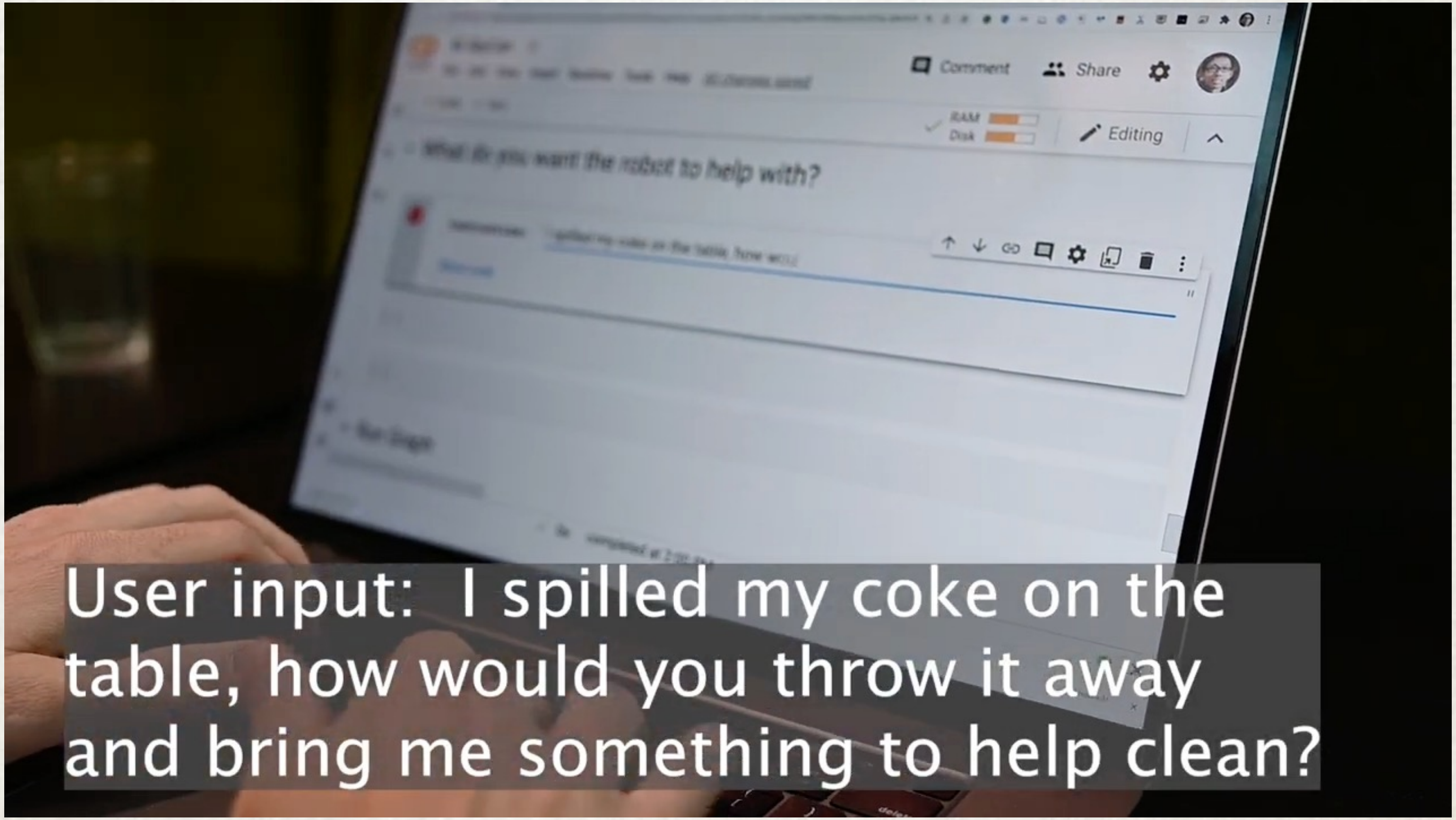


OpenAI, “Learning Dexterous In-Hand Manipulation”, 2018

Locomotion

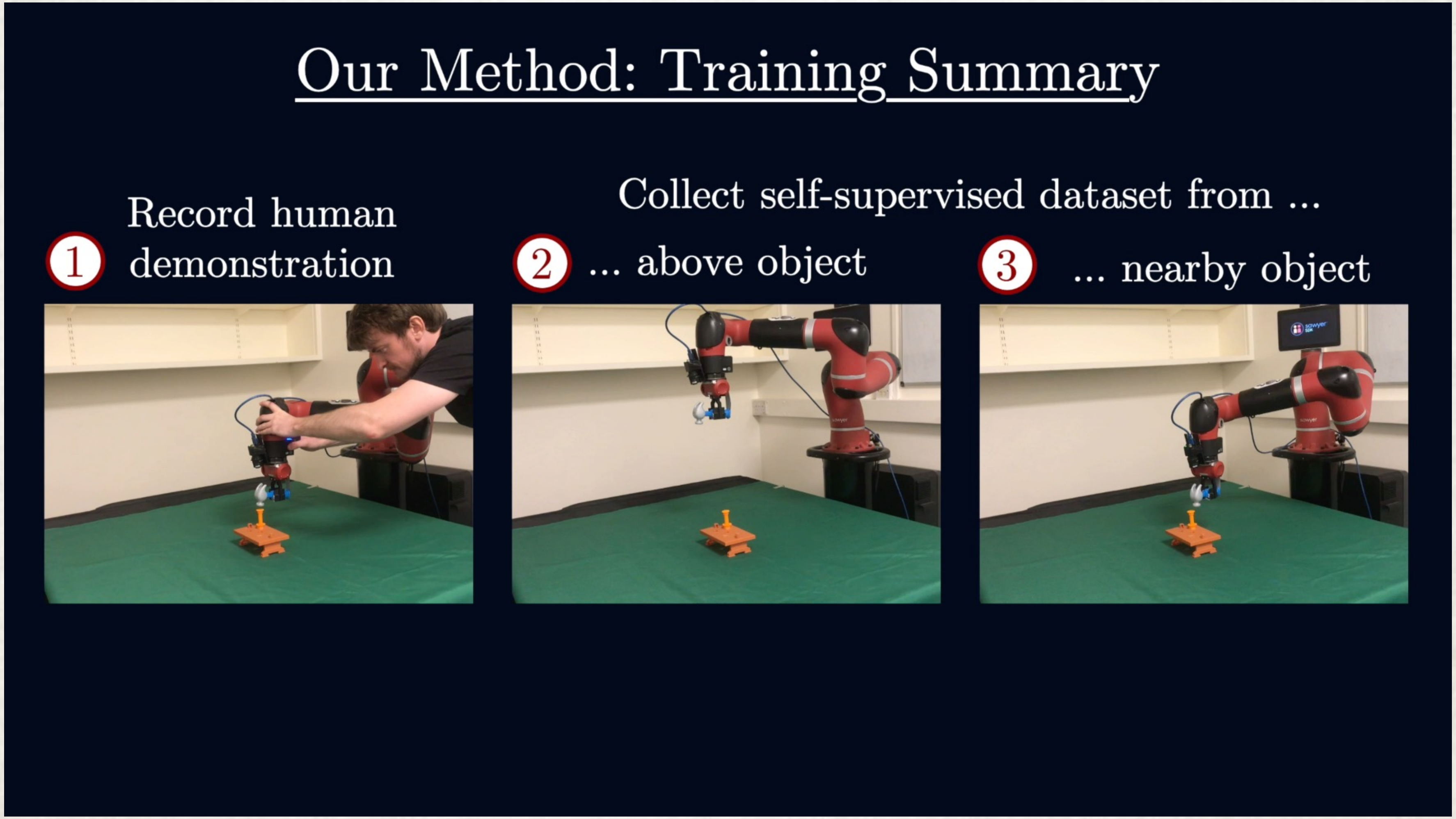
Miki et al., "Learning robust perceptive locomotion for quadrupedal robots in the wild", 2022

Language



Robotics at Google, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances", 2022

Imitation Learning



Johns, "Coarse-to-Fine Imitation Learning: Robot Manipulation from a Single Demonstration", 2021

Lecture 1: Introduction to Robot Learning

Contents

Part 1: What is robot learning?

Part 2: Robot learning success stories

Part 3: The different types of robot learning

Part 4: Overview of the module

Part 5: States, observations, actions, and policies

Part 6: Task specification

Part 7: Labs and coursework

The different types of machine learning

Supervised Learning	Unsupervised Learning	Reinforcement Learning

The different types of robot learning

Supervised Learning

Human provides robot with data for a specific task, defining what action should be taken in each scenario. (Often referred to as imitation learning.)

Unsupervised Learning

Human provides high-level algorithm for how a robot should autonomously learn new tasks, without any information about the specific task.

Reinforcement Learning

Human provides feedback to robot for a specific task, on how good each action was, each time a robot tries an action.

The different types of robot learning

Each has a relationship with human learning.

Supervised Learning	Unsupervised Learning	Reinforcement Learning

Lecture 1: Introduction to Robot Learning

Contents

Part 1: What is robot learning?

Part 2: Robot learning success stories

Part 3: The different types of robot learning

Part 4: Overview of the module

Part 5: States, observations, actions, and policies

Part 6: Task specification

Part 7: Labs and coursework

Lecture Content

Lecture 1: Introduction to Robot Learning

Lecture 2: Analytical Methods

Lecture 3: Real-world Reinforcement Learning

Lecture 4: Imitation Learning

Lecture 5: Unsupervised Robot Learning

Lecture 6: Applications

Lecture 7: Competition and Revision

Timetable

Week	Monday Lecture (Huxley 311)	Wednesday Lab (Huxley Lab)	Coursework
1	No lectures or labs		
2	January 16th, Lecture 1	January 18th, Lab 1	
3	January 23rd, Lecture 2	January 25th, Lab 2	Coursework 1 released on January 25th
4	January 30th, Lecture 3	February 1st, Lab 3	
5	February 6th, Lecture 4	February 8th, Lab 4	Coursework 1 due on February 7th Coursework 2 released on February 8th
6	February 13th, Lecture 5	February 15th, Lab 5	
7	February 20th, Lecture 6	February 22nd, Lab 6	
8	No lecture	March 1st, Lab 7	Coursework 2 due on March 1st
9	March 6th, Lecture 7	No lab	
10	No lectures or labs		
11	Exam Week		

Lecture Logistics

- All lectures will be held in-person in Huxley 311.
- All lectures will be recorded and uploaded to Panopto.
- All lecture, lab, and coursework material, will be available on Scientia.
- Lecture slides will be available for each lecture.

Lab Logistics

- All labs will be held in-person in the Huxley computer lab.
- For some labs there will be exercises, and for some labs you will be free to continue working on the courseworks.
- GTAs will be available during the labs to help you.

EdStem

- Please use EdStem to ask any questions, including both technical questions and logistical questions.
- Please “heart” questions you like.
- Please attempt to answer each other’s questions if you can.
- If a question has not been answered by another student within 24 hours, I will answer it myself shortly afterwards.
- Please use sections, e.g. “Lectures”, “Labs”, “Courseworks”, etc.

Lecture 1: Introduction to Robot Learning

Contents

Part 1: What is robot learning?

Part 2: Robot learning success stories

Part 3: The different types of robot learning

Part 4: Overview of the module

Part 5: States, observations, actions, and policies

Part 6: Task specification

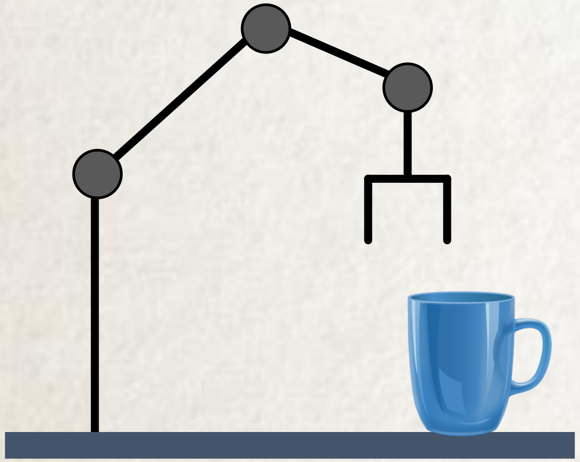
Part 7: Labs and coursework

States

- The state, s, is a sufficient but minimal representation of the physical environment.
- It should contain all information necessary for the robot to solve the task.
- But it should not contain any extra information information that is irrelevant to the task.

States

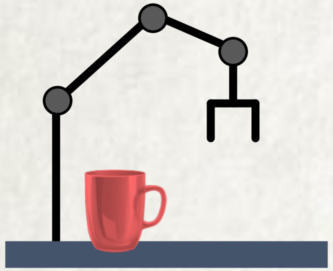
Example

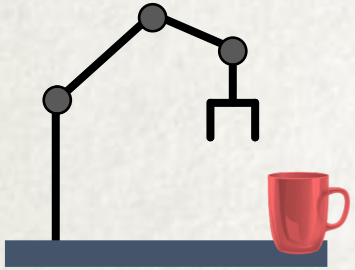


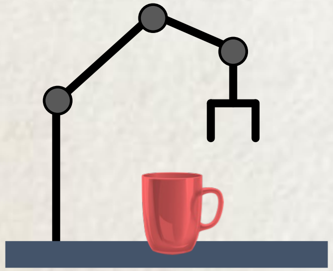
- If the task is for the robot to grasp the mug, the robot needs to know where the gripper is, and where the mug is.
- It might also need to know the mug's mass and the mug's coefficient of friction.
- But it probably does not need to know the colour of the mug, the mass of the table, where the nearest window is, what time it is in Australia …

States

Example



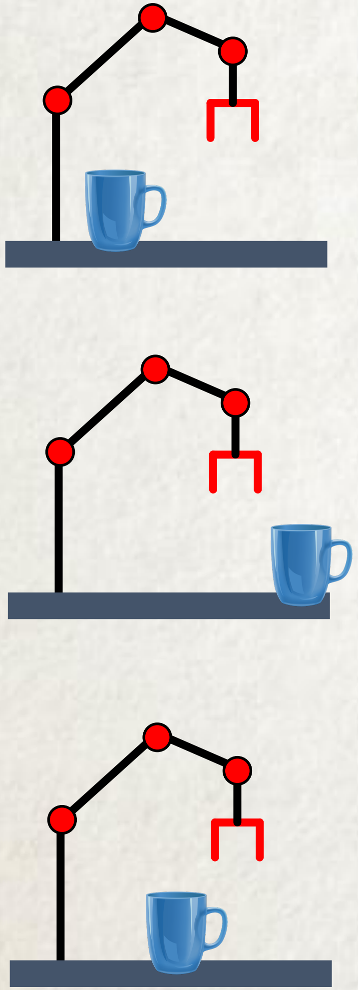




- Options for the mug's state:
  - 3D position and 3D orientation.
  - 3D position.
  - 1D horizontal position along table.

States

Example



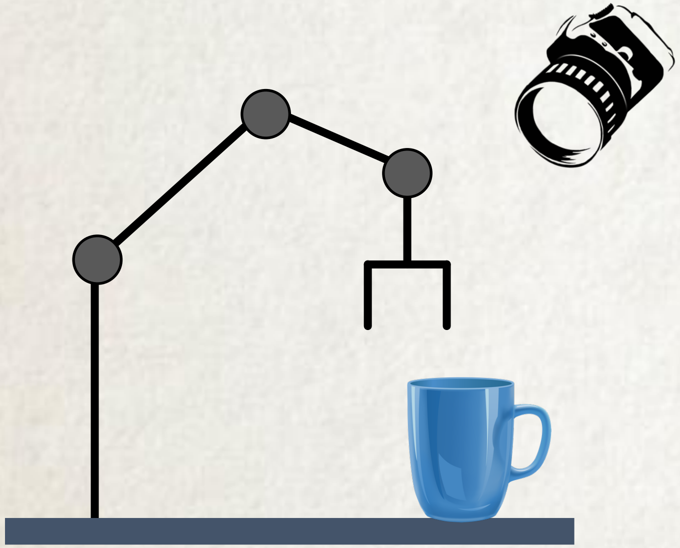
- Options for the robot's state:
  - Joint angles.
  - Gripper's position.
  - Gripper's position and orientation.

Observations

- The observation, *o*, is the information available from the robot's (noisy) sensors.
- If an environment is *fully observable*, then it is possible (although not necessarily easy) to infer the state from the observation.
- If an environment is *partially observable*, then it is impossible to infer the state just from the observation. Further information would be required (e.g. extra sensors, or multiple observations over sequential time steps).

Observations

Example



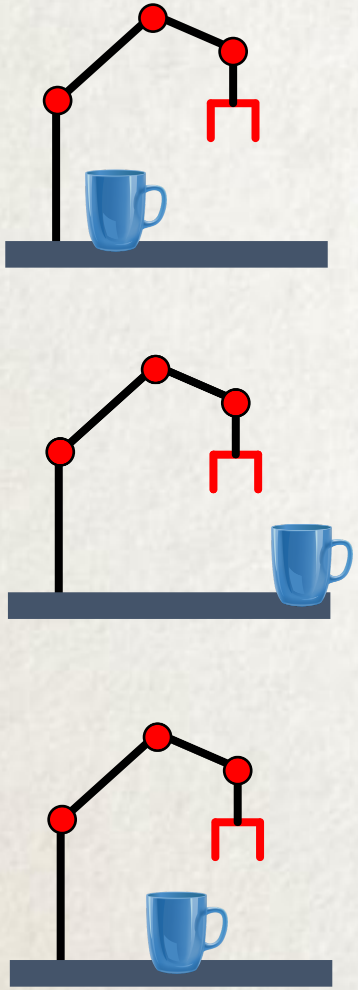
- Options for the observation:
  - An image of the mug.
  - The (noisy) measurement of the joint angles according to the robot's joint encoders.
  - Both of the above.
  - An image of the mug and the robot.

Actions

- The action, *a*, is what the robot executes in an attempt to achieve the desired state.
- Often, a robot learning algorithm predicts a high-level action (e.g. position to move the gripper to), which is then converted to a low-level action (e.g. joint torques), such as via a PID controller.
- Therefore, a full robot learning algorithm often involves both machine learning components, and analytical components. Robots are rarely trained fully end-to-end in the real world.

Actions

Example



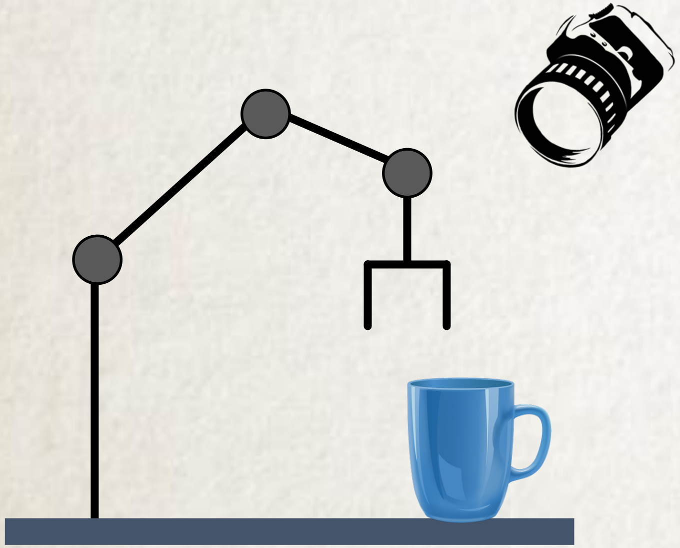
- Options for the robot's action:
  - Joint torques.
  - Joint velocities.
  - Joint positions.
  - Gripper velocity.
  - Gripper position.

Policies

- The policy is a function which takes the state or observation as input, and returns an action as output:
  - $a = \pi(s)$ or $a = \pi(o)$
- The function $\pi(.)$ could be:
  - A neural network.
  - Any other machine learning model, e.g. a Gaussian Process.
  - A combination of a learned model and an analytical controller.

Policies

Example



- Options for the policy:
  - A neural network which predicts gripper velocity from image observation.
  - A Gaussian Process which predicts joint torques from joint positions and mug position.
  - An analytical controller which move the gripper close to the mug, and then switches to one of the above learned policies.

Lecture 1: Introduction to Robot Learning

Contents

Part 1: What is robot learning?

Part 2: Robot learning success stories

Part 3: The different types of robot learning

Part 4: Overview of the module

Part 5: States, observations, actions, and policies

Part 6: Task specification

Part 7: Labs and coursework

Human engineer specifies task to robot

State space

Observation space

Action space

Task specification

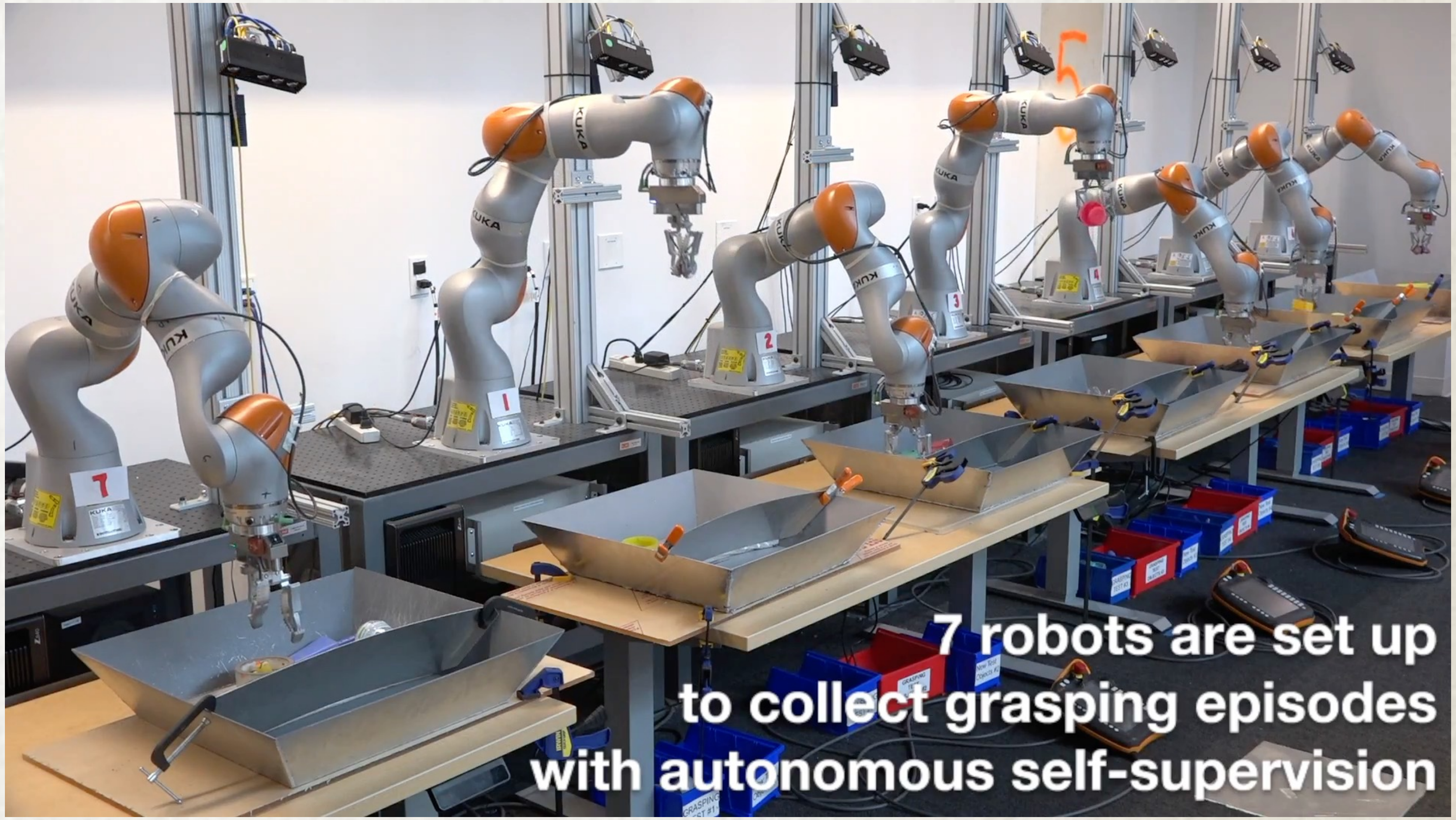
Engineer

Robot

Different ways to specify a task

A task could be specified by providing a:

- Reward function.



Different ways to specify a task

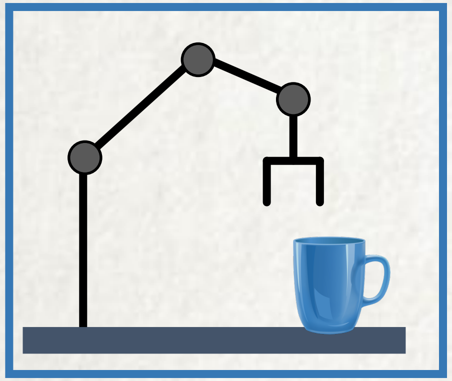
A task could be specified by providing a:

- Reward function.
- Cost function (cost ~ inverse reward).
- Goal state (e.g. a position of an object).
- Goal observation (e.g. an image of the scene).
- Demonstration (e.g. by physically moving the robot).

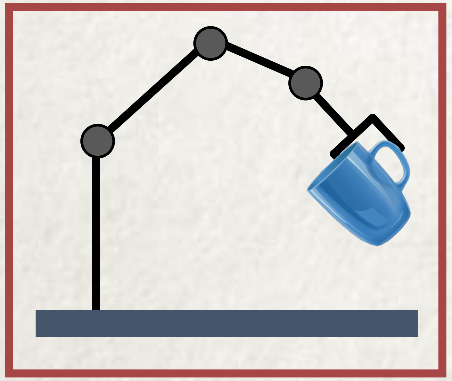
Reward functions

Example

Current state



Goal state

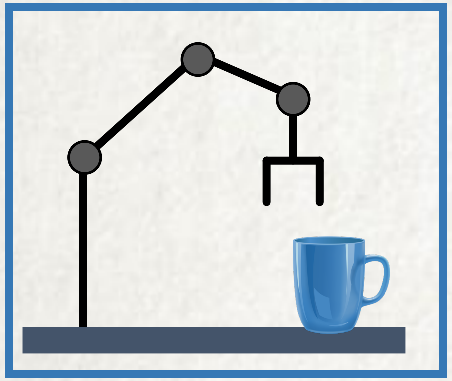


- Options for the reward function:
  - Sparse reward: 1 when goal state is reached, 0 otherwise.
  - Dense reward: negative distance between mug's current and goal pose (pose = position and orientation).
  - Dense reward as above + negative penalty per time step (to encourage fast execution of task).

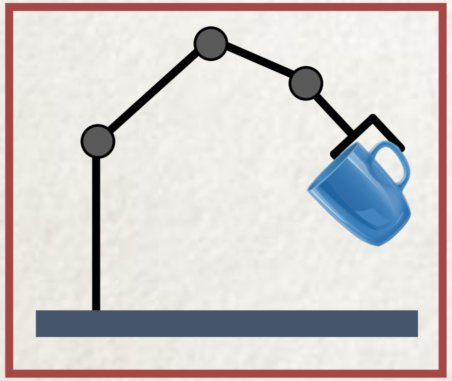
Goal States / Observations

Example

Current state



Goal state

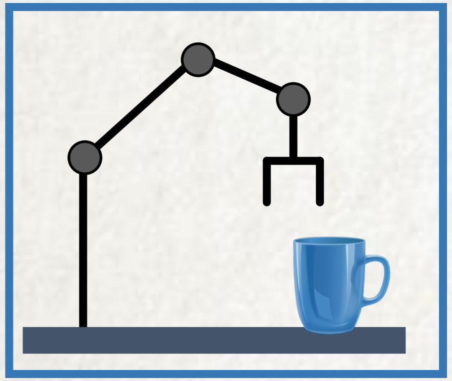


- Options for goals:
  - State of desired object (goal state).
  - Image of desired scene (goal observation).
- Goal states / observations are usually then converted into reward / cost functions:
  - E.g. based on distance between current and goal state.

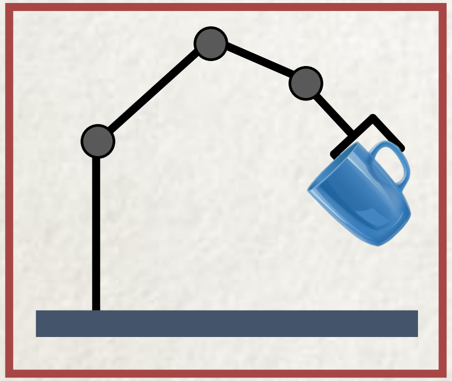
Imitation learning

Example

Current state



Goal state



- Options for demonstrations:
  - Physically moving the robot (kinesthetic teaching).
  - Remotely controlling the robot (teleoperation).
  - Robot observes human performing task (third-person imitation learning).

Lecture 1: Introduction to Robot Learning

Contents

Part 1: What is robot learning?

Part 2: Robot learning success stories

Part 3: The different types of robot learning

Part 4: Overview of the module

Part 5: States, Observations, Actions, and Policies

Part 6: Task Specification

Part 7: Labs and Coursework

Real-world robot learning

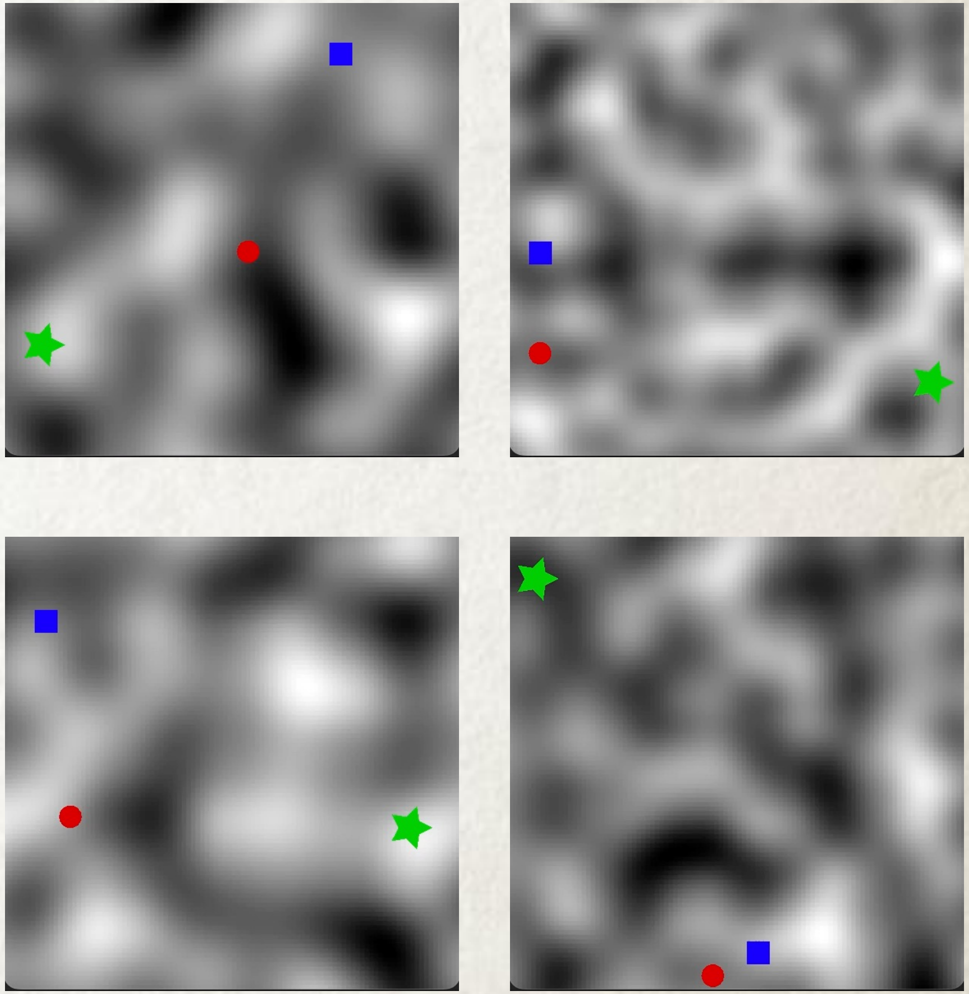
- Real-world robot learning is great fun, but very data inefficient.
- A typical robot learning algorithm would takes weeks to train, possibly with several robots learning simultaneously.
- This is not possible for a lecture course with ~100 students!

Simulated robot learning

- Simulated robot learning is much more appropriate.
- However, even in simulation, high-dimensional robot learning (e.g. image observations, full 3D scenes), is still too data inefficient.

Our 2D navigation environment

- State = 2D position of robot.
- Action = 2D movement of robot.
- Similar to a maze navigation task.
- In 2D, we can visualise interesting information, e.g. the policy, the reward function, or the dynamics.
- The competition at the end of the course will involve you racing your robot against your classmates!



Next Week:

Analytical Methods