CS4278/CS5478 Intelligent Robots: Algorithms and Systems

Lin Shao

NUS

Today's Plan

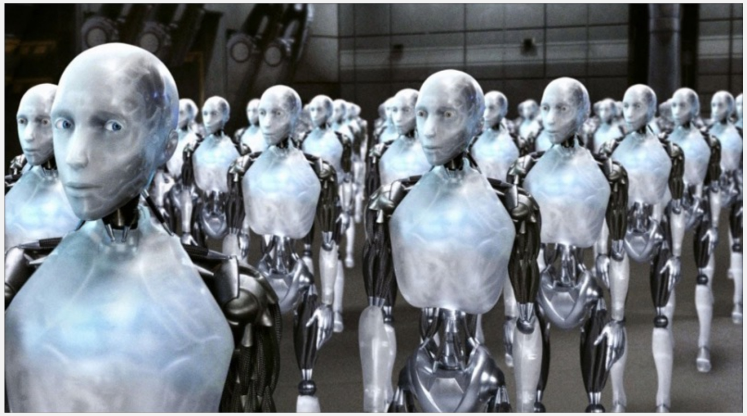
- Overview of the course
- Course logistics
- Introduction to robots

Today's Plan

- **Overview of the course**
- Course logistics
- Introduction to robots

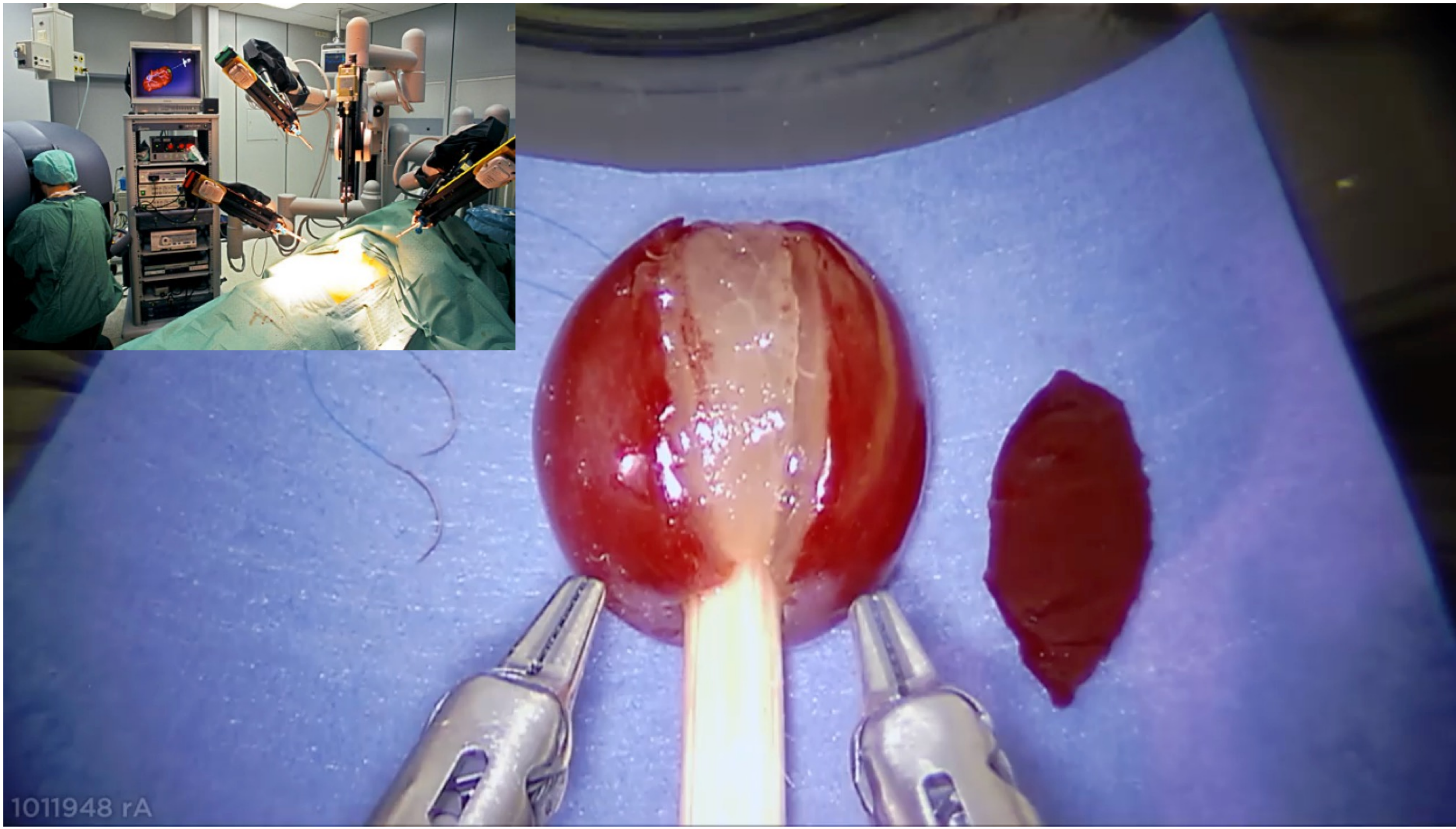
What is a robot?

Transformers

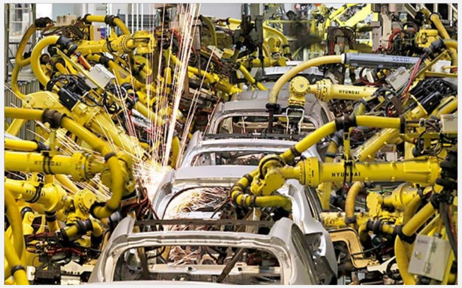


I, Robot

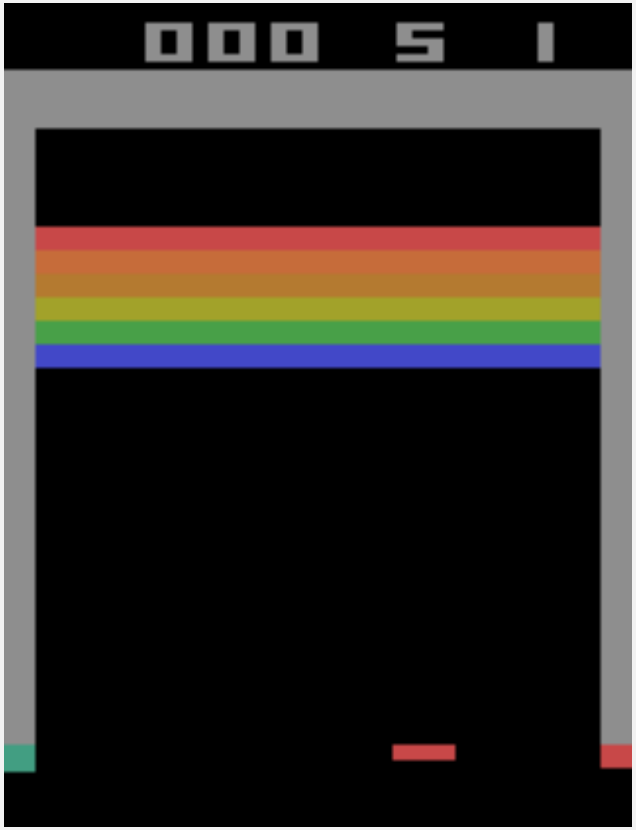
da Vinci surgical robot



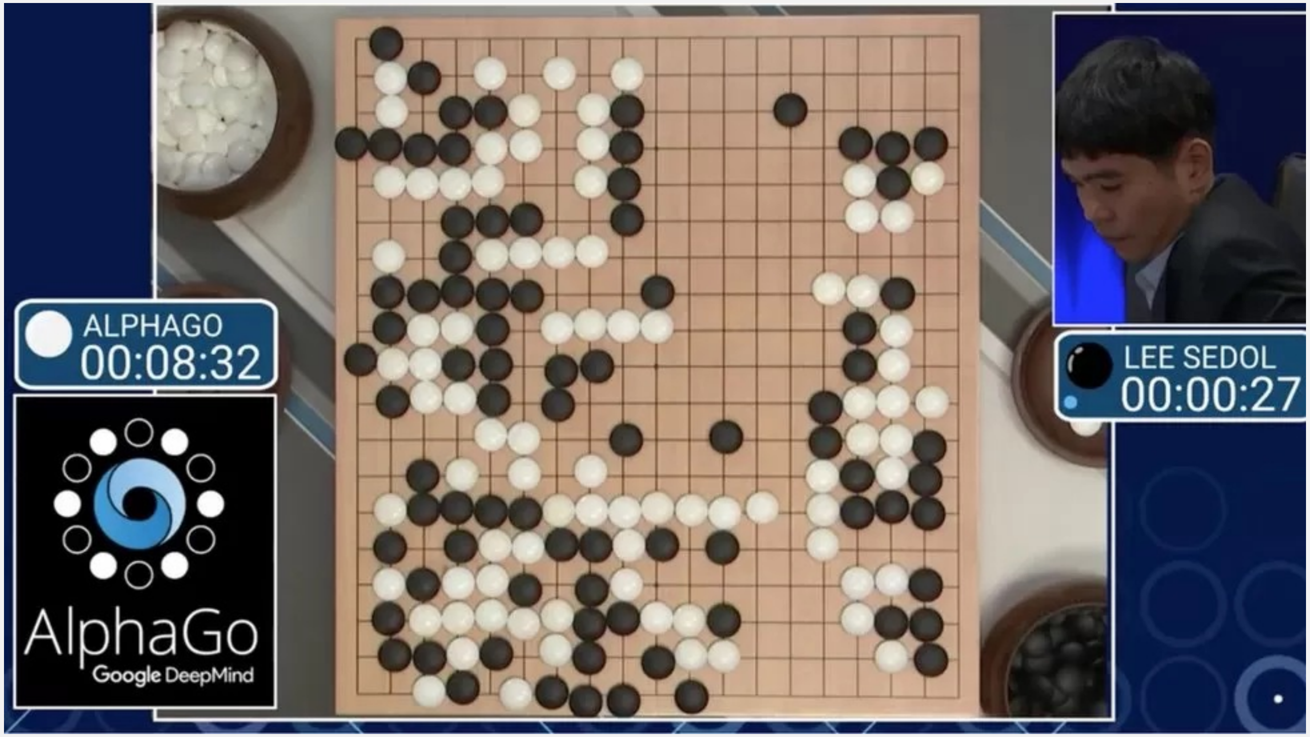
“Intelligent” Robots



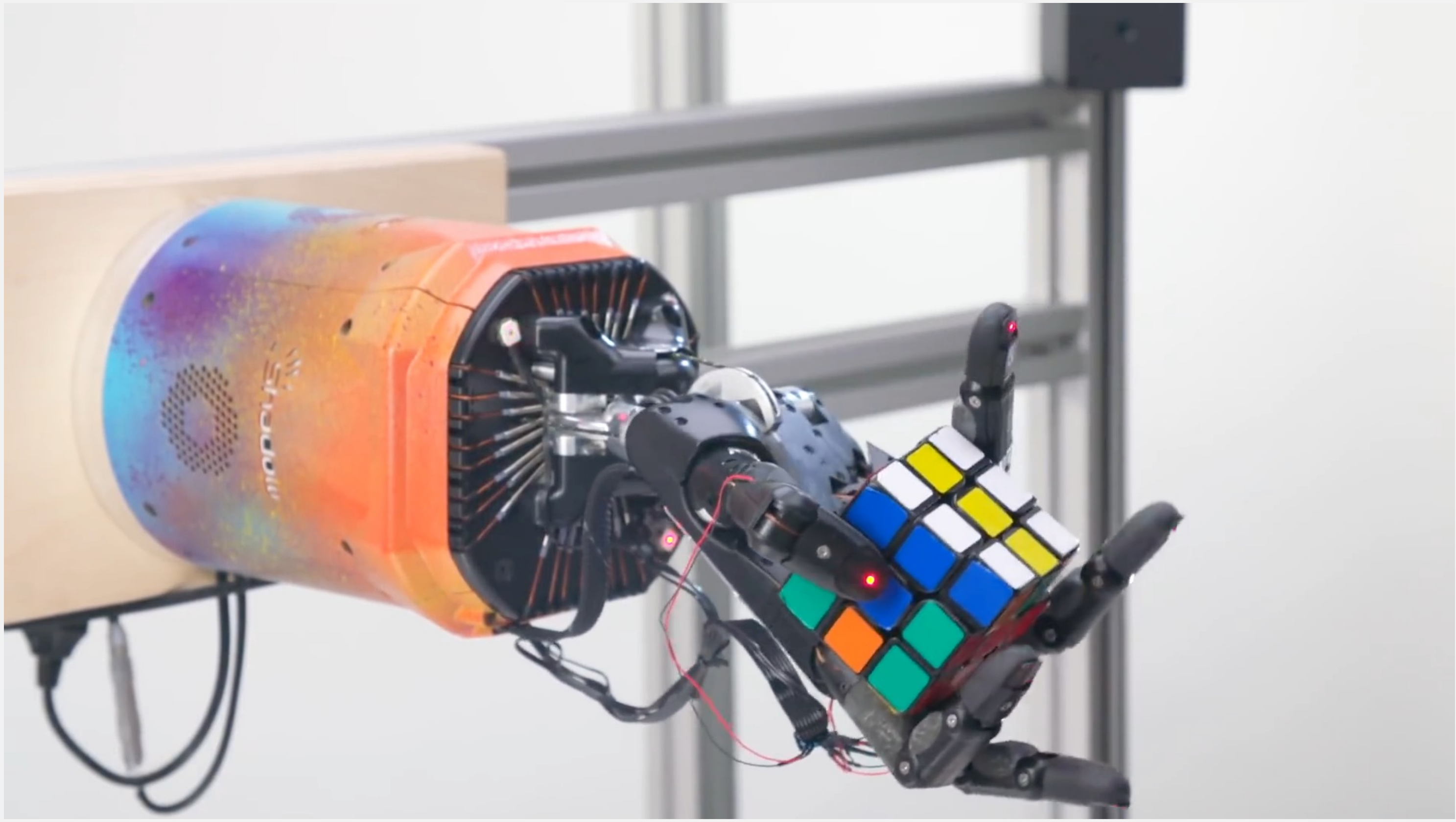
Atari (~2015)



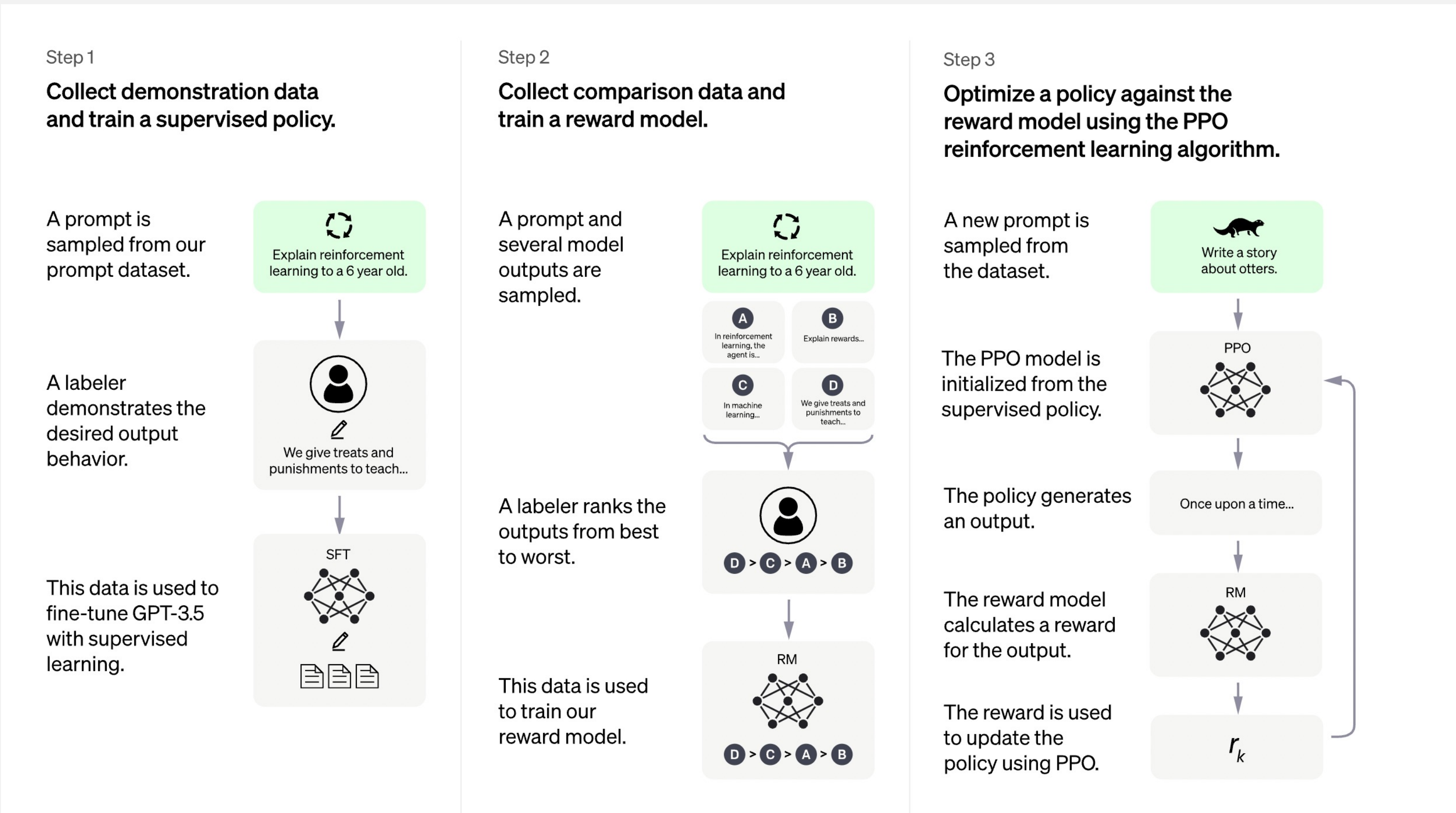
AlphaGo (2016)



Rubik's cube(2019)

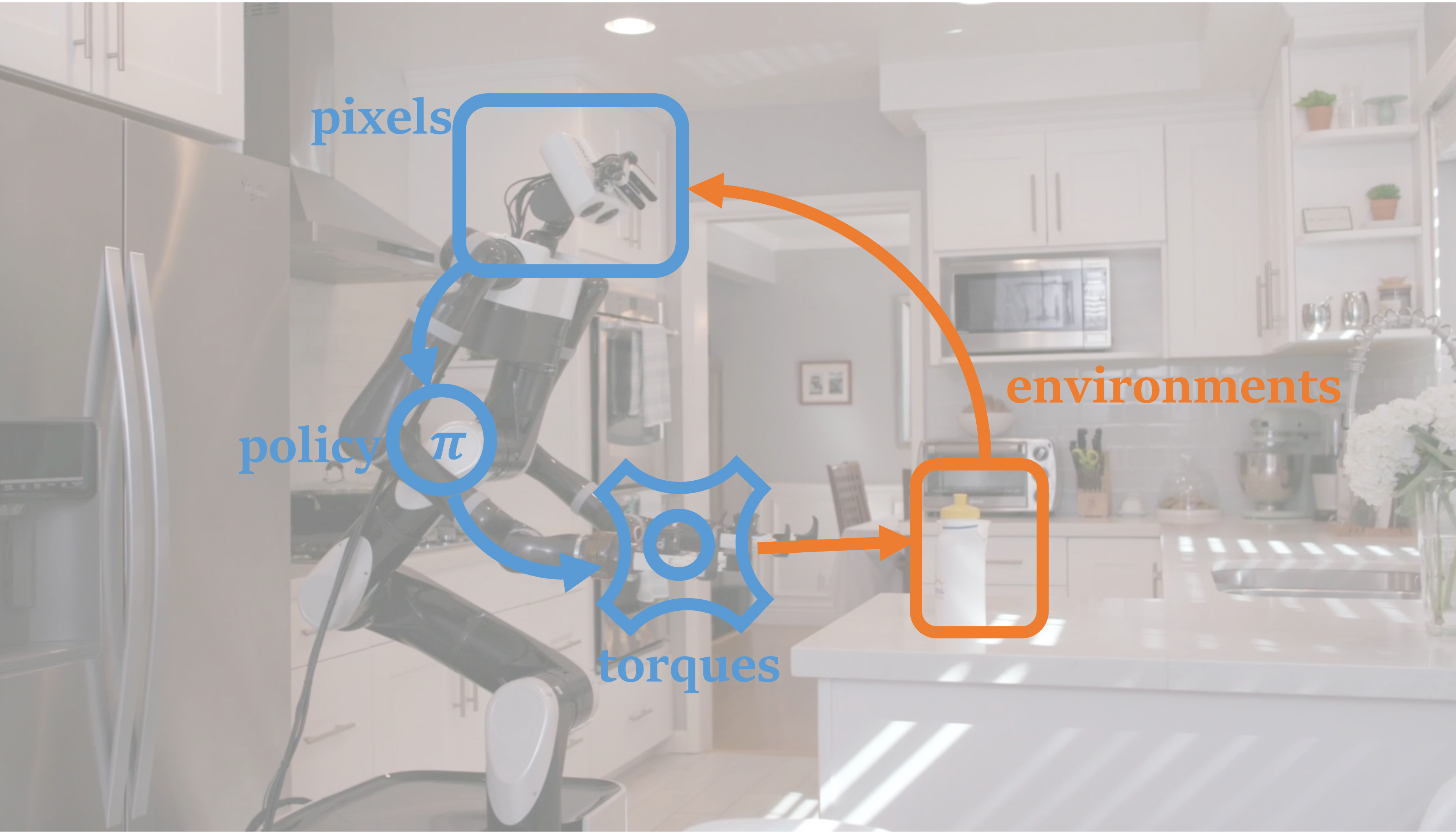


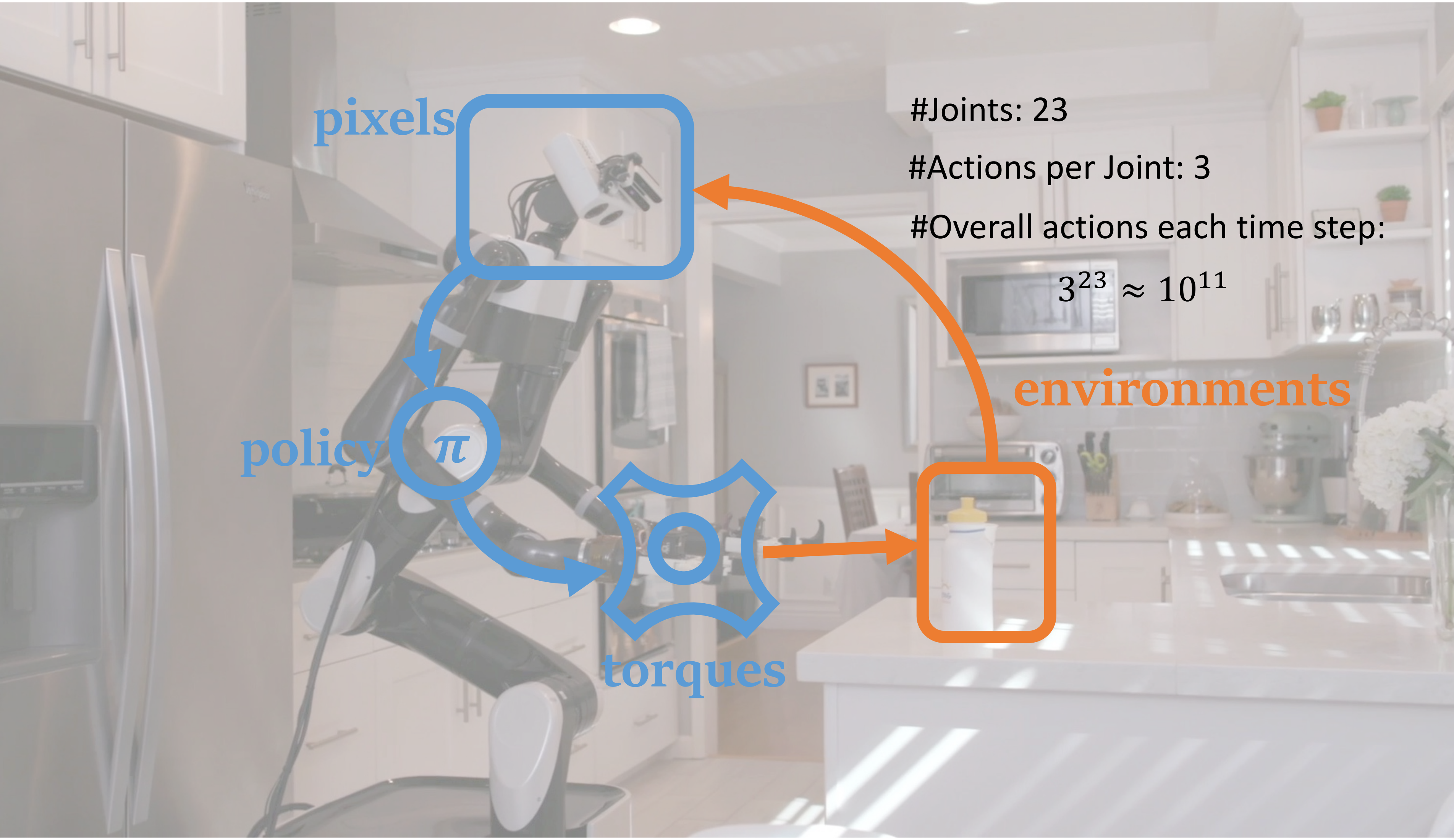
ChatGPT(2022)



(https://openai.com/blog/chatgpt/)

Intelligent Robots


pixels
policy
π
torques
environments

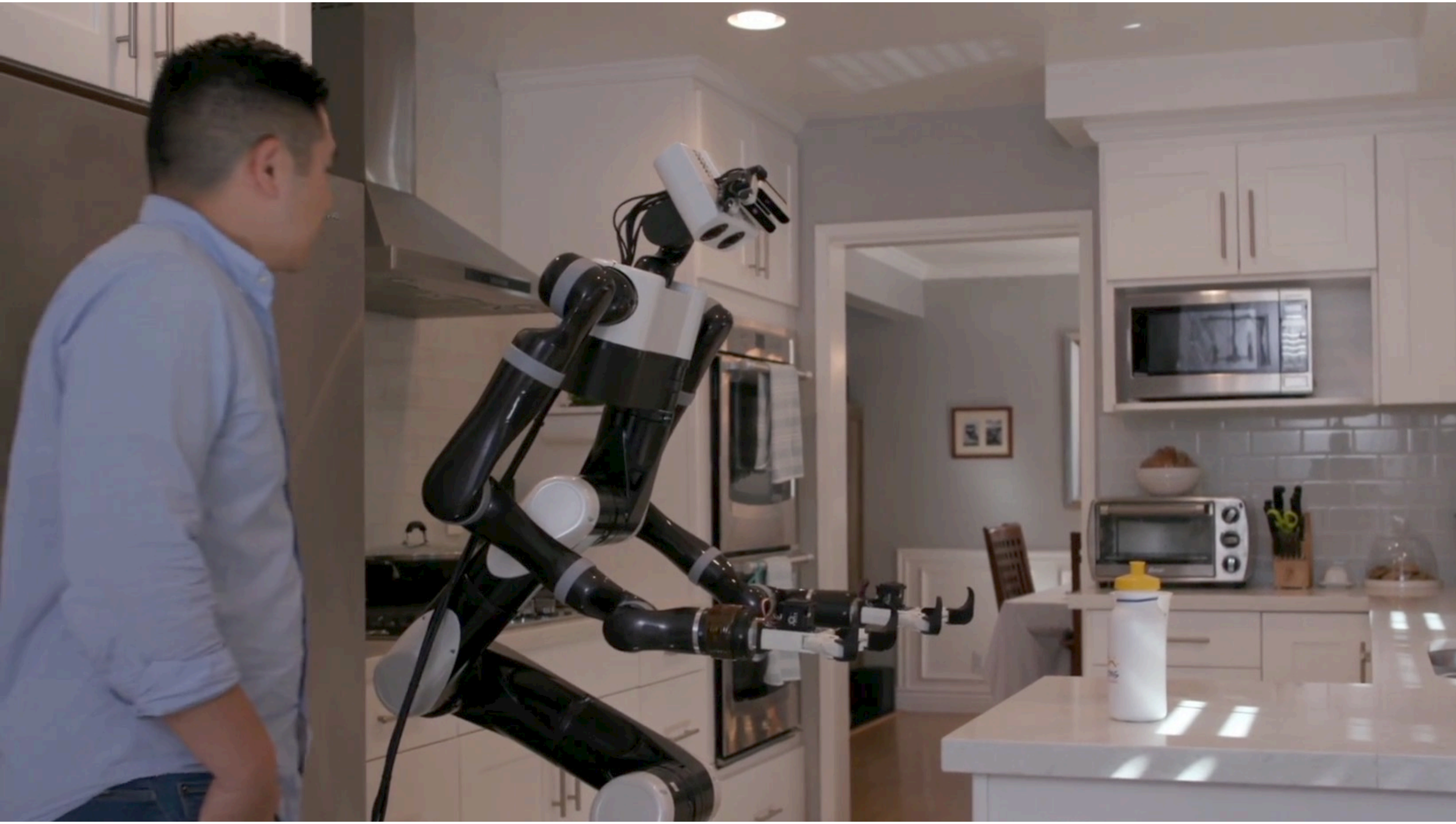


#Joints: 23

#Actions per Joint: 3

#Overall actions each time step:

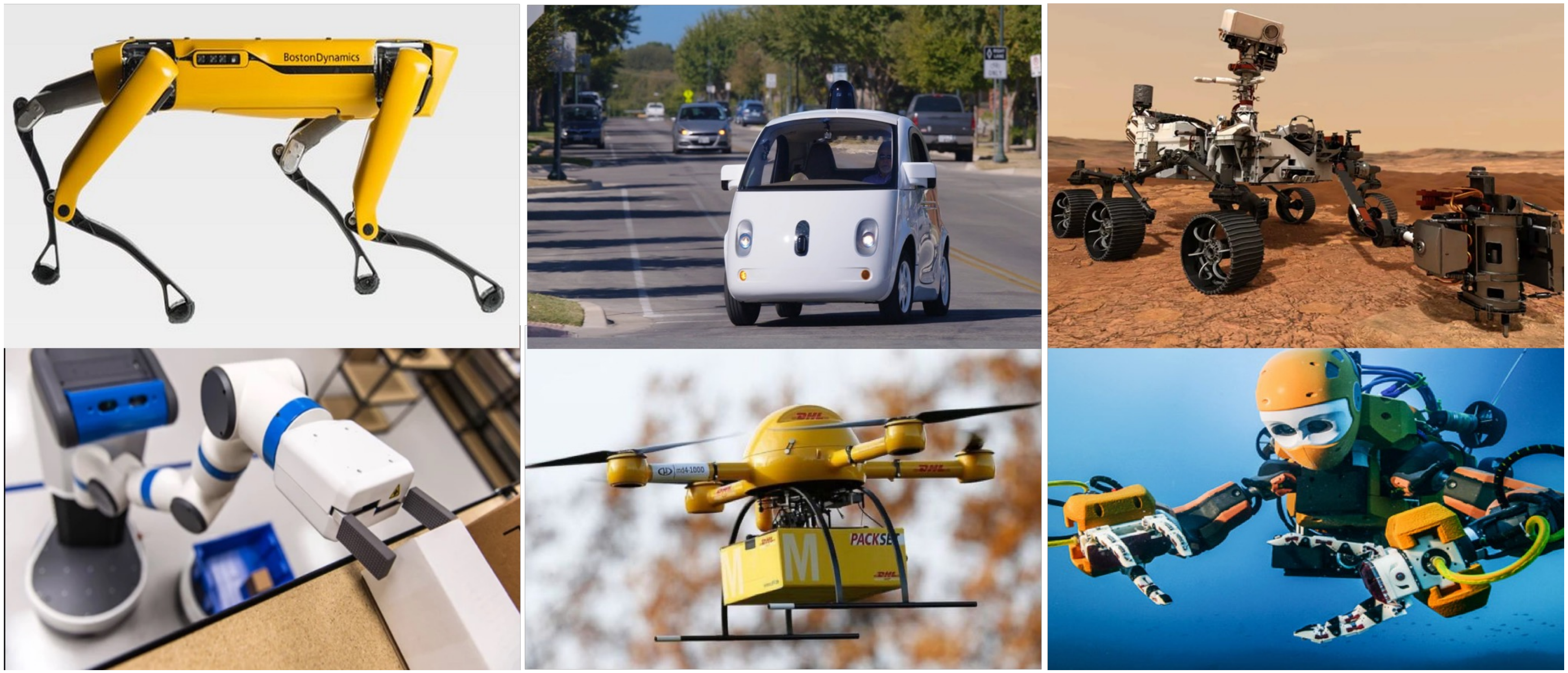
$$3^{23} \approx 10^{11}$$



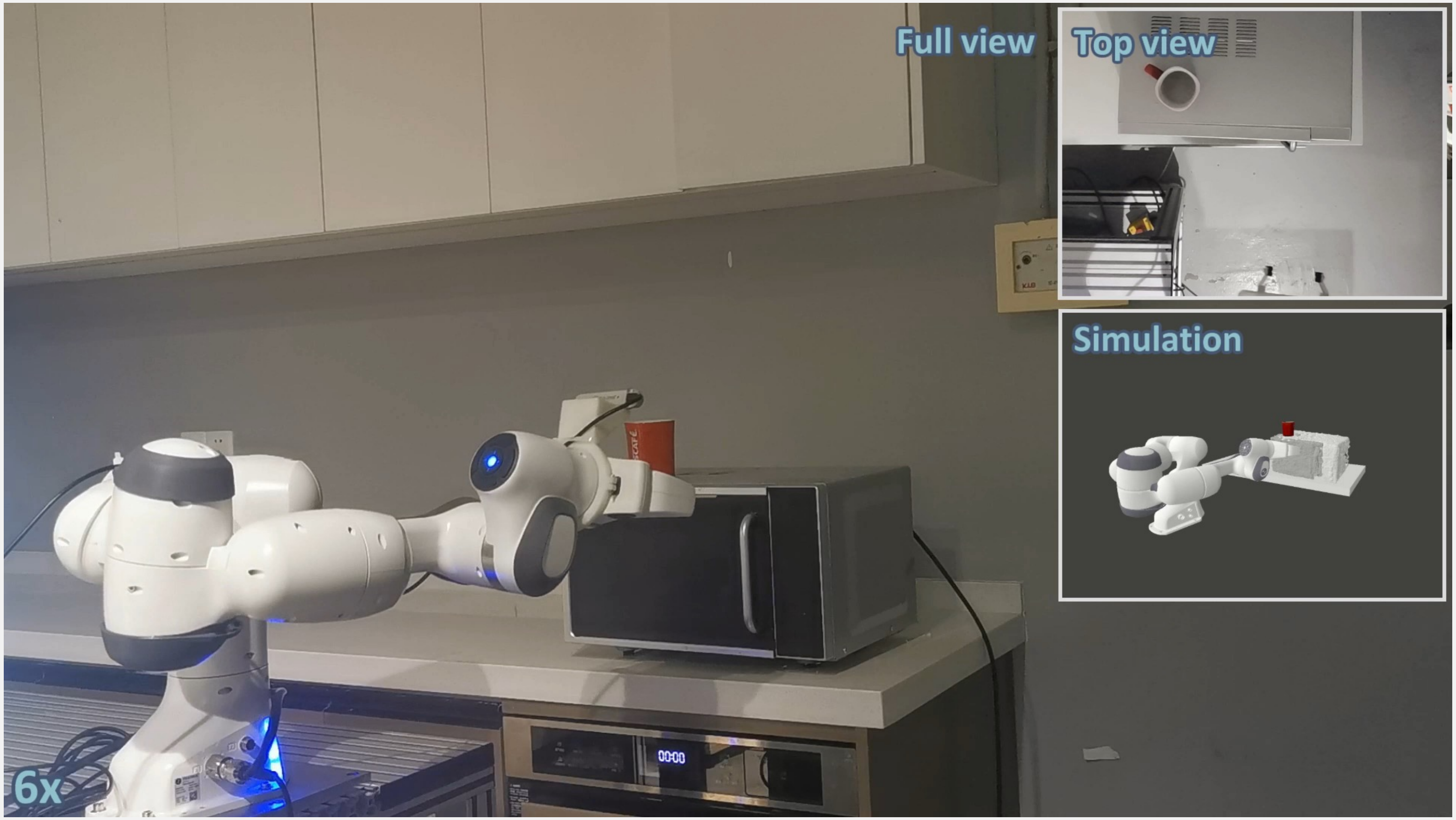
Topics

- Mathematical Models
  - Kinematics
  - Dynamics
- Planning
  - C-Space
  - Planning in low-Dim and high-Dim
- Sensing & Control
  - Sensors & Robot Vision
  - State Estimation
  - PID, LQR, MPC
- Grasping & Manipulation
- Decision Making
  - MDP & Deep RL
  - POMDP
- Advanced Topics
  - Diff Simulators
  - Large Language Models

Robot



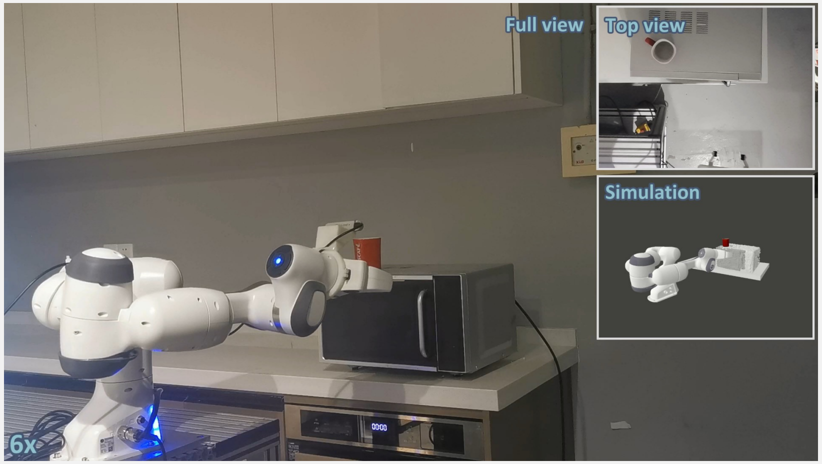
Manipulator



Mathematical Models

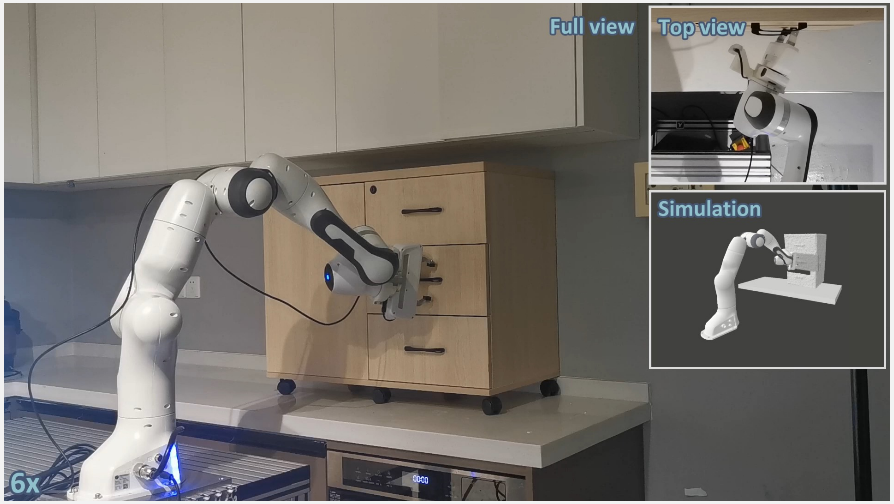
- Spatial Descriptions
- Kinematics
- Dynamics

- Planning and Control
  - Motion Planning
  - Motion Control
  - Force Control



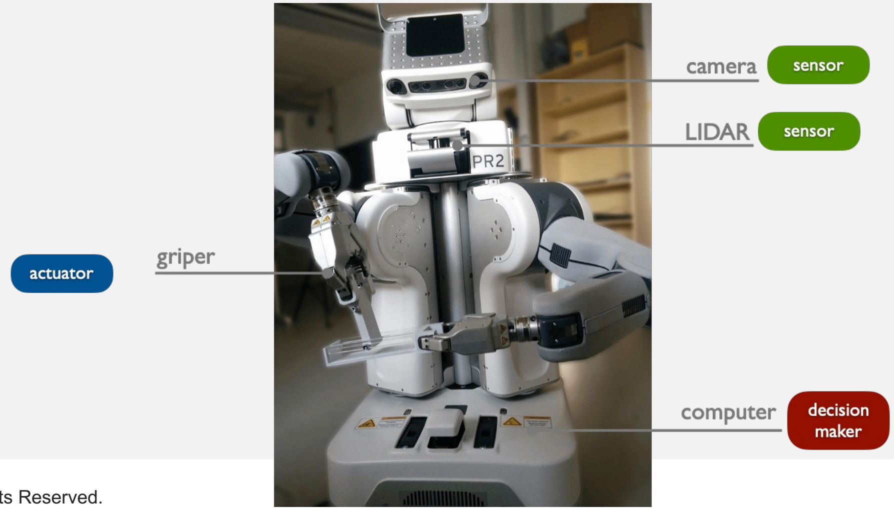
Planning

- Planning enables the robot to choose a sequence of actions and reach a specified goal.



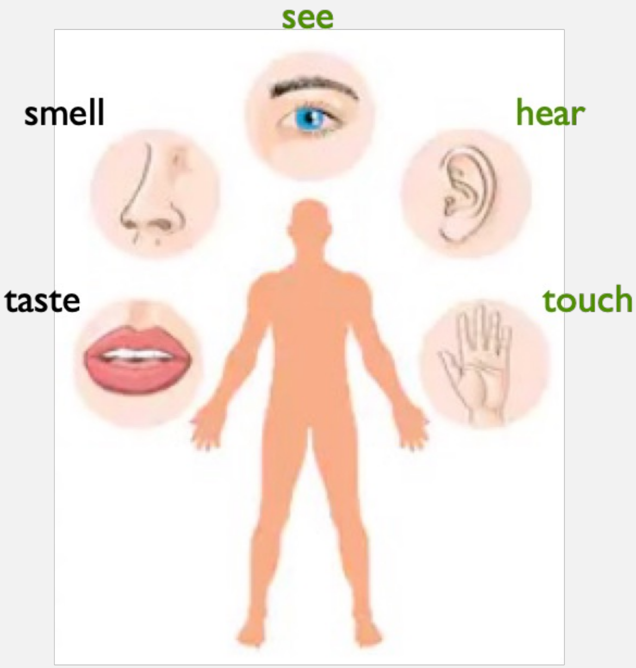
Sensors

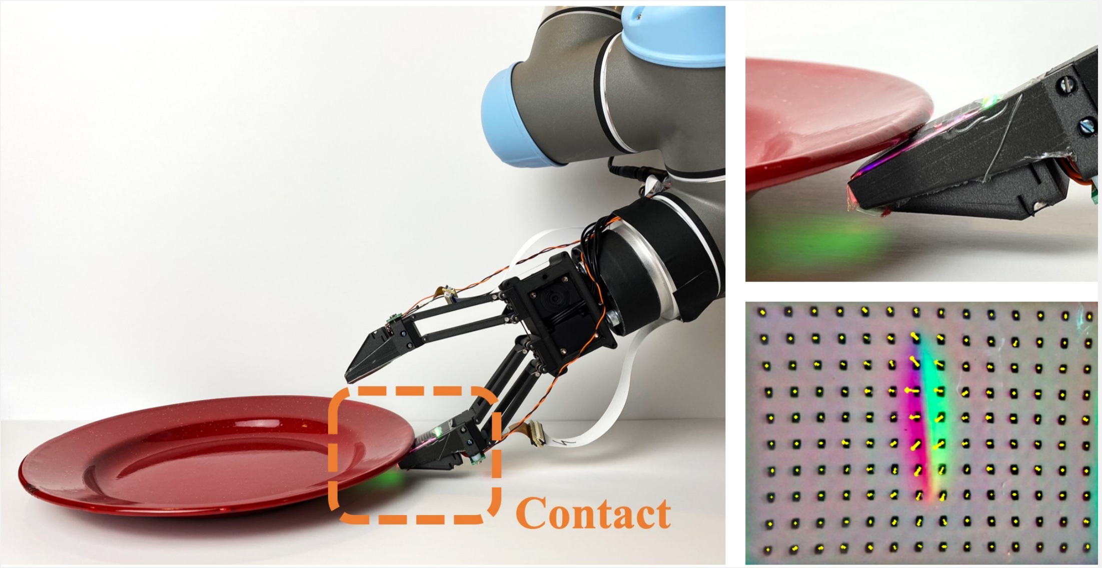
sensor
IMU
decision maker
computer
sensor
wheel encoder
LIDAR
sensor
wheels
actuator



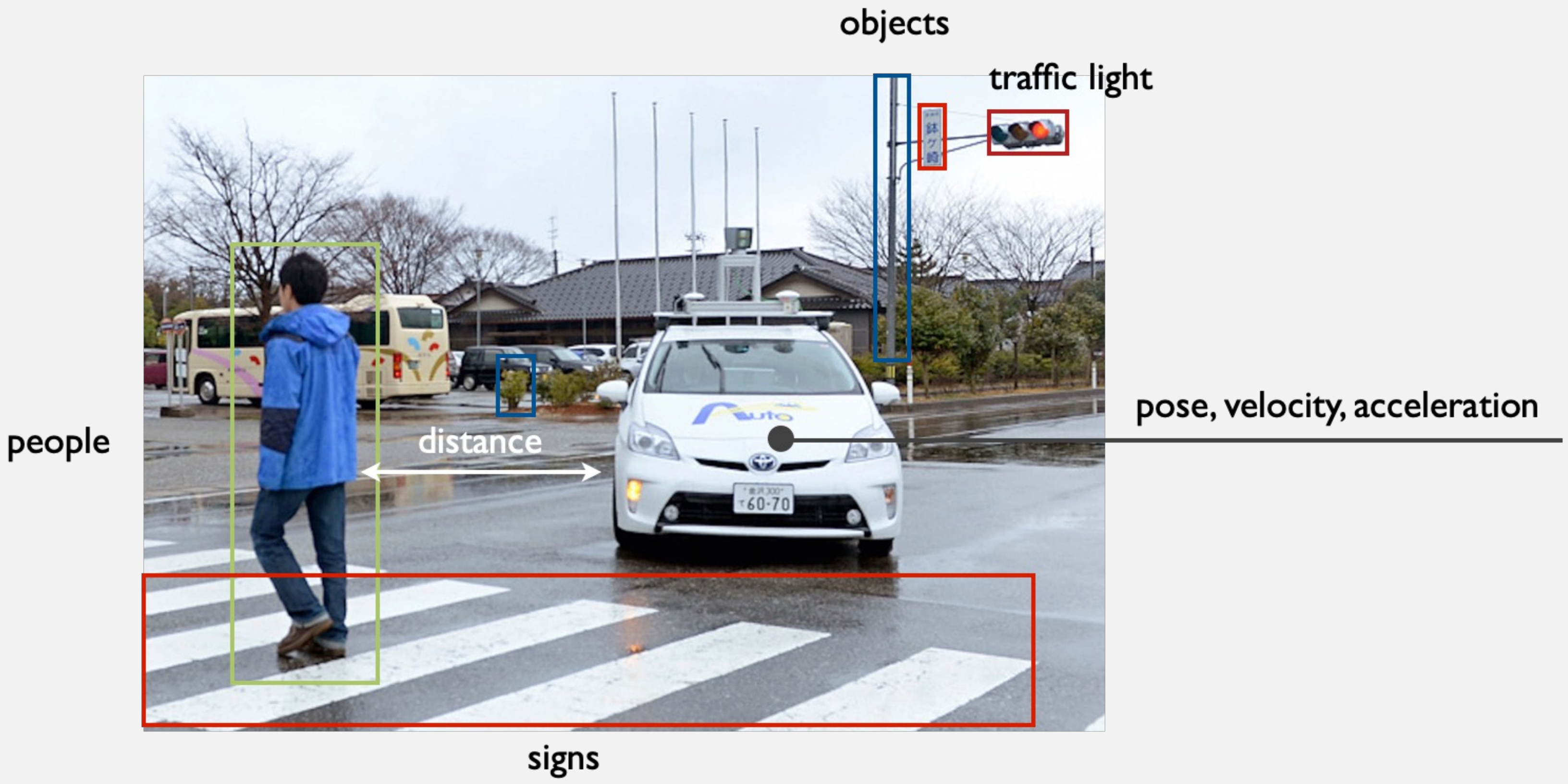
camera
sensor
LIDAR
sensor
PR2
actuator
griper
computer
decision maker

Sensors

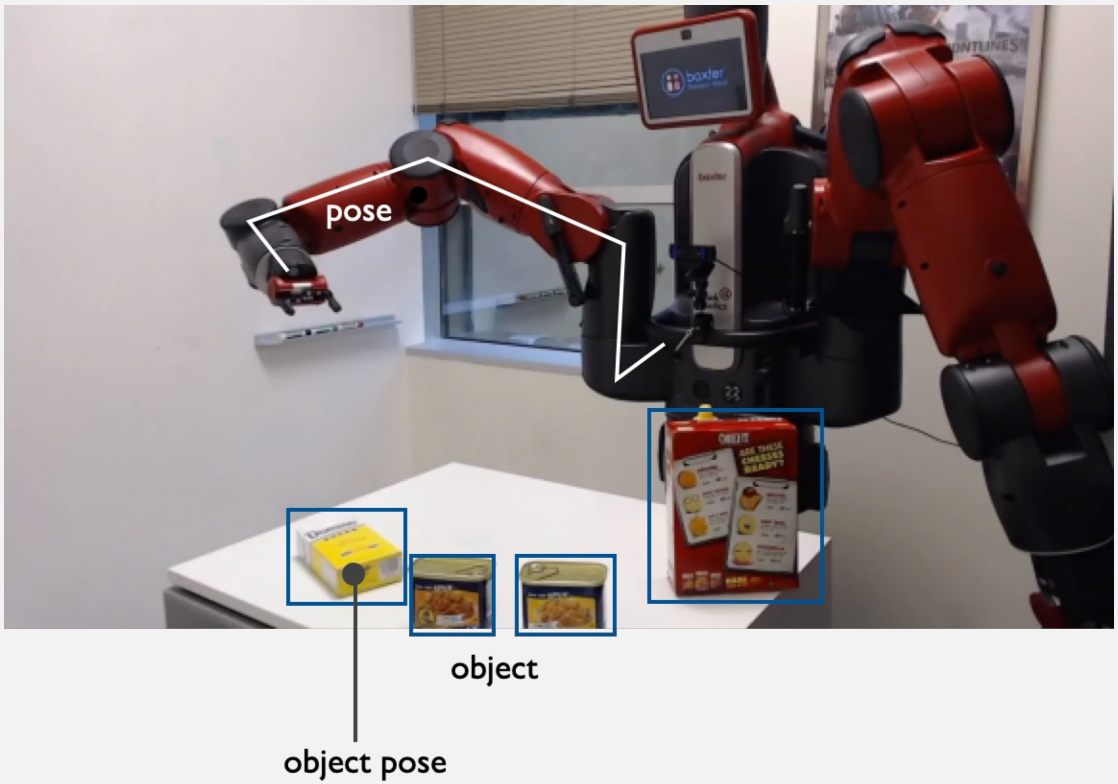




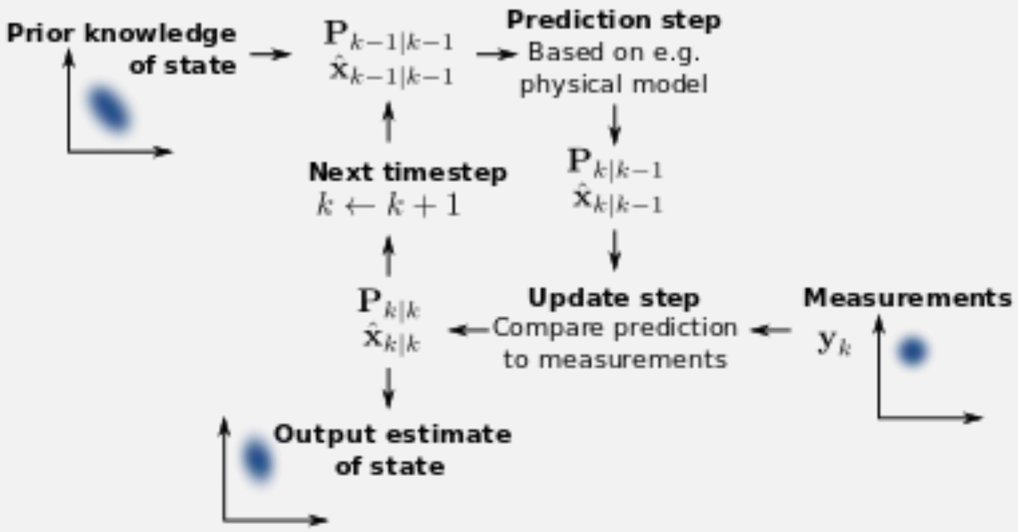
What does the robot sense?



What does the robot sense?



State Estimation



Optimal Control

Find an *admissible control* $\mathbf{u}^*$ which causes the system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$$

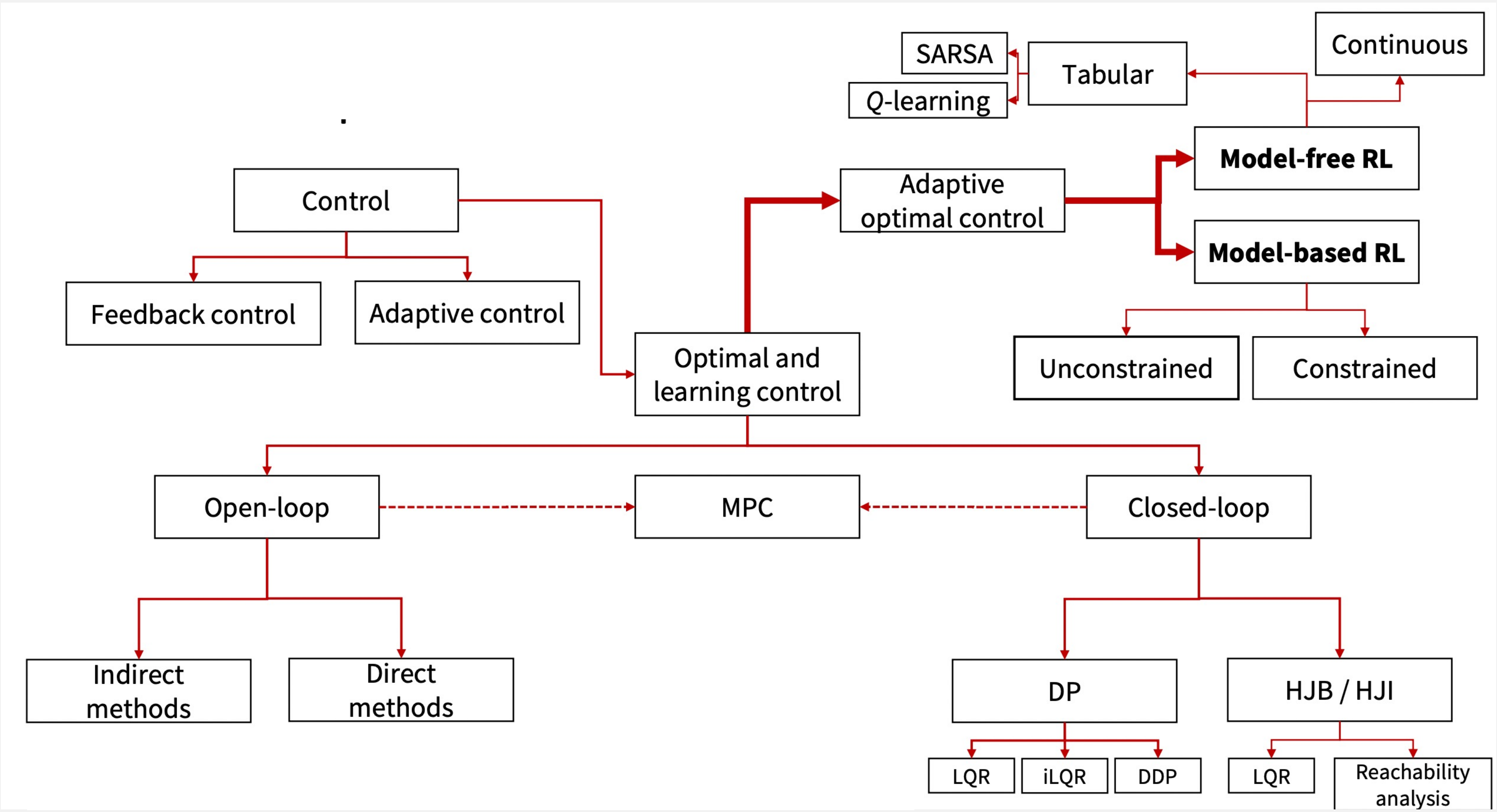
to follow an *admissible trajectory* $\mathbf{x}^*$ that minimizes the performance measure

$$J = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(t), \mathbf{u}(t), t)\, dt$$

- h (terminal cost) and g (stagewise/running cost) are scalar functions
- t_f may be specified or free

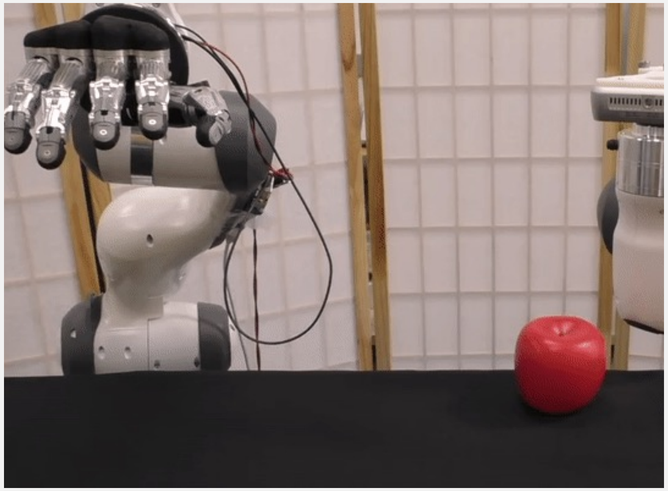
from Marco Pavone's course

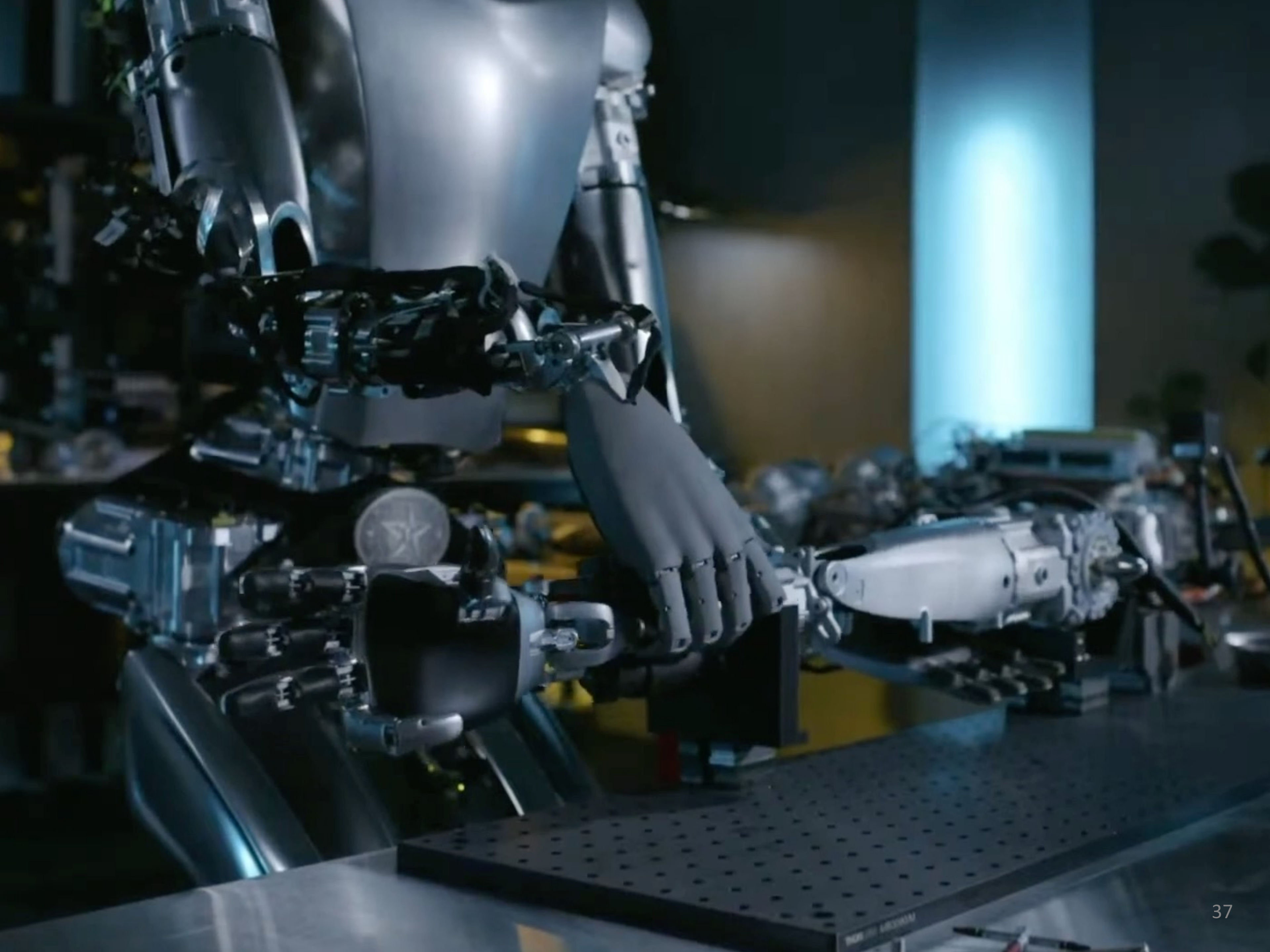
Optimal Control



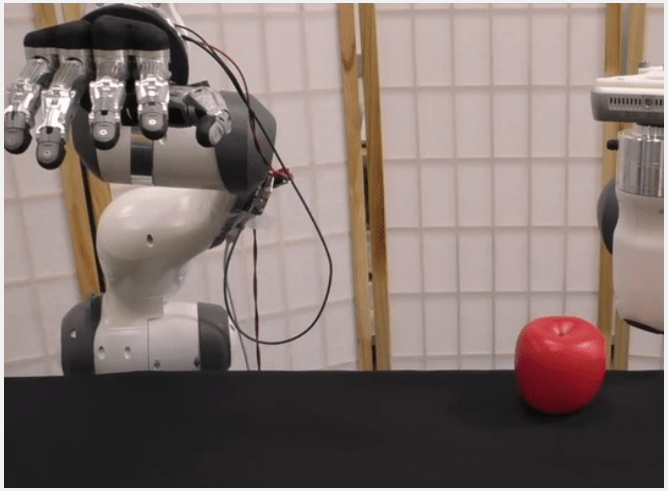
from Marco Pavone's course

Grasping





Grasping



What is a Grasp?

Why is grasping challenging?

How to model a Grasp?

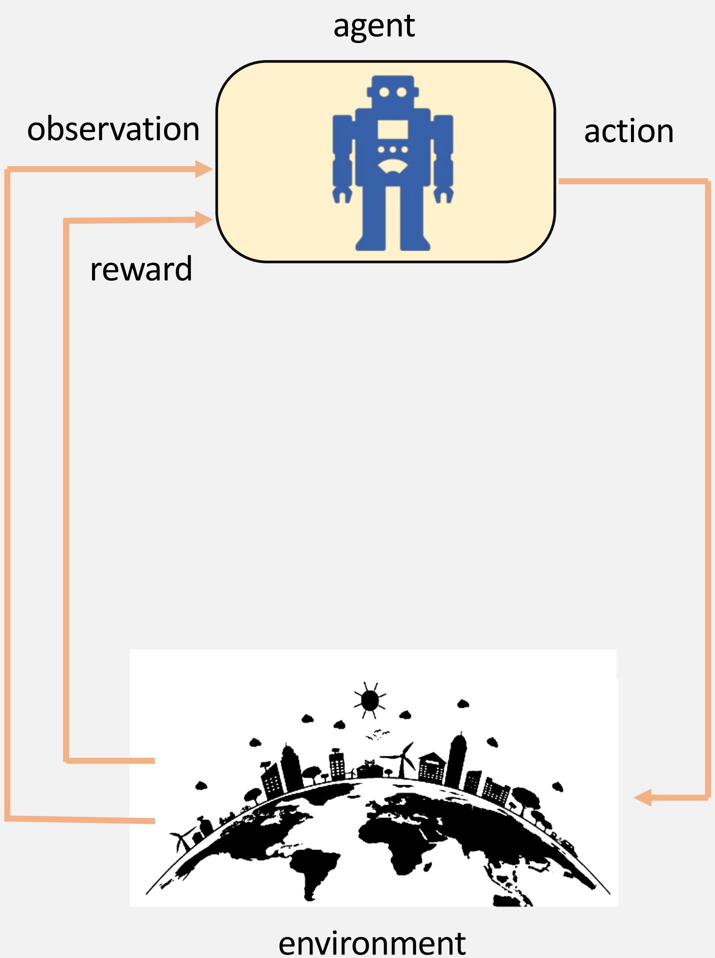
How to evaluate a Grasp?

Form Closure

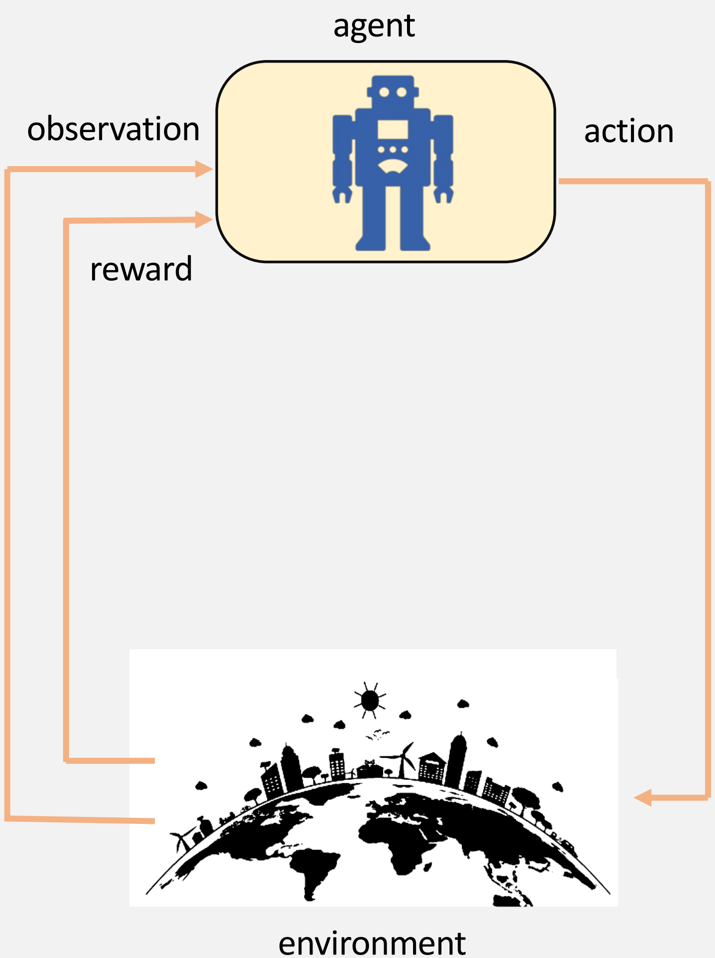
Force Closure

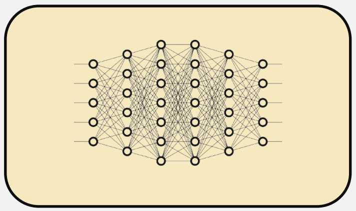
How to generate a Grasp?

Decision Making



Model-free RL



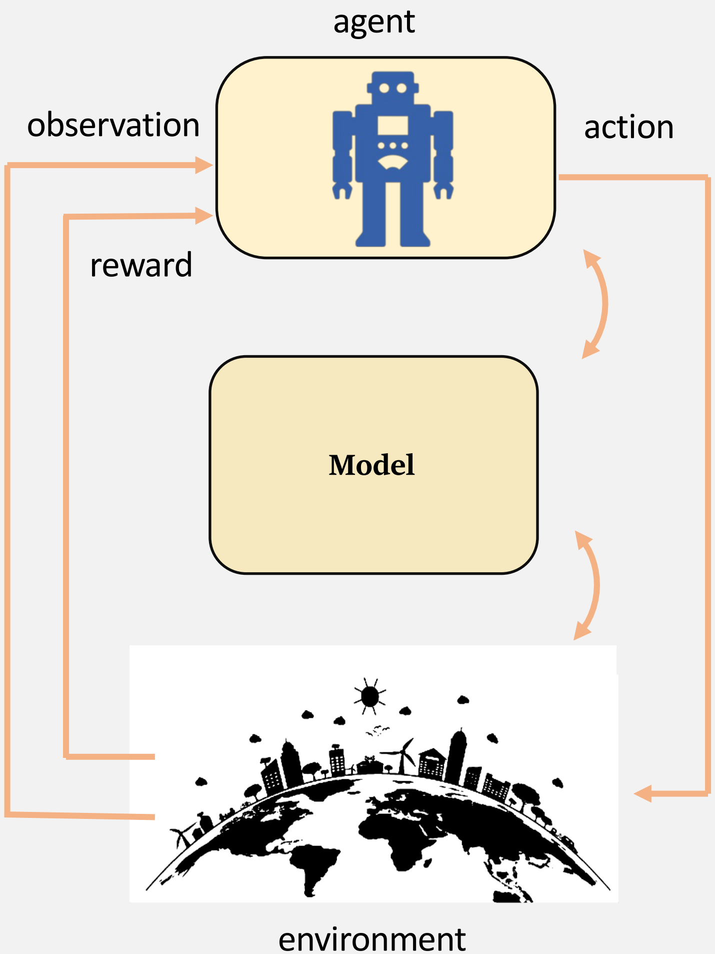


- ✔ Work directly with raw sensory inputs (images)
- ✔ Learn automatically and reduce human engineering efforts

Policy gradient
Deep RL with Q-functions
Actor-critic
…

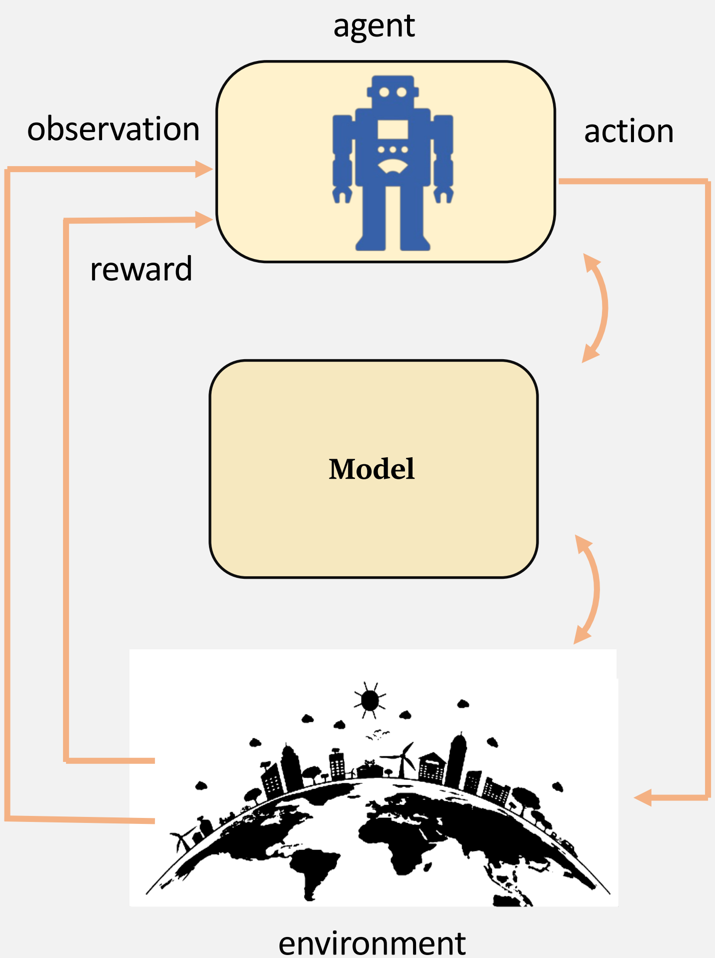
Overestimation issue
Trust region
Exploration vs. Exploitation
Generalization
Hindsight experience replay

Model-based RL



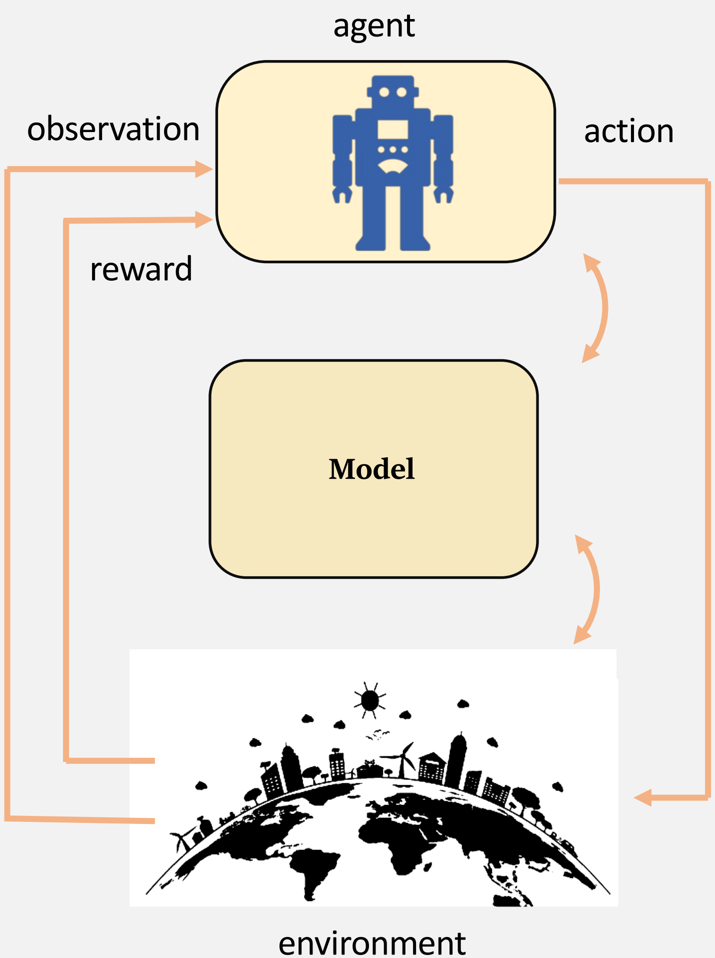
- Potentially sample-efficiency
- Modeling errors degenerate performances

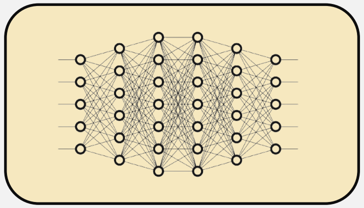
Model-based RL



- Engineering efforts from human experts
- Do not scale up large numbers of tasks

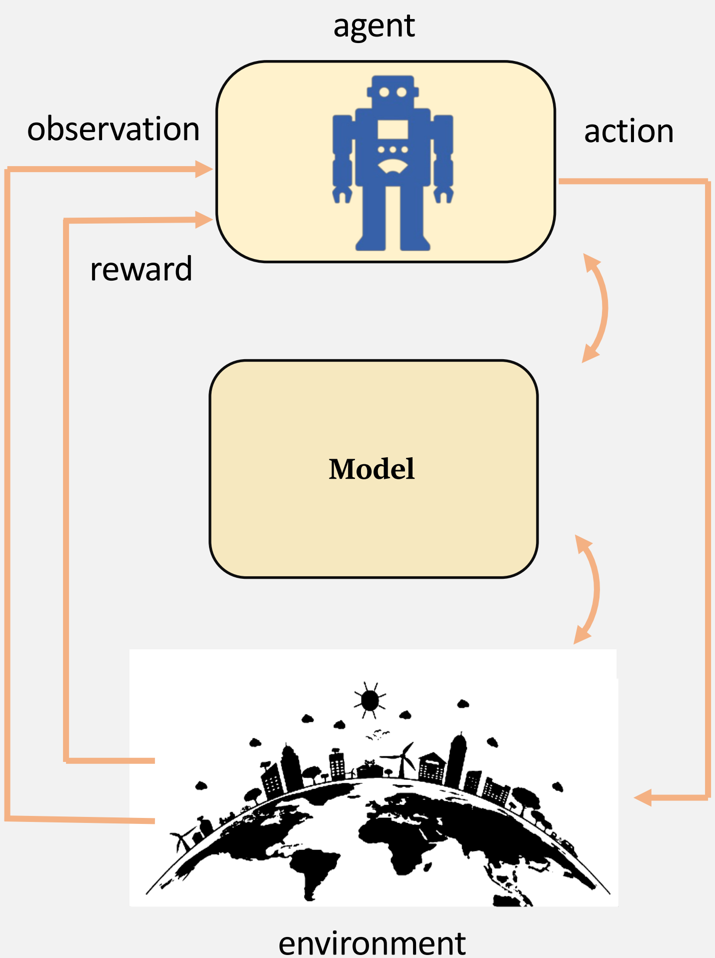
Model-based RL





- Require a large amount of training data
- Might not satisfy physical dynamics
- May degenerate beyond the training data distribution when testing in the wild

Model-based RL



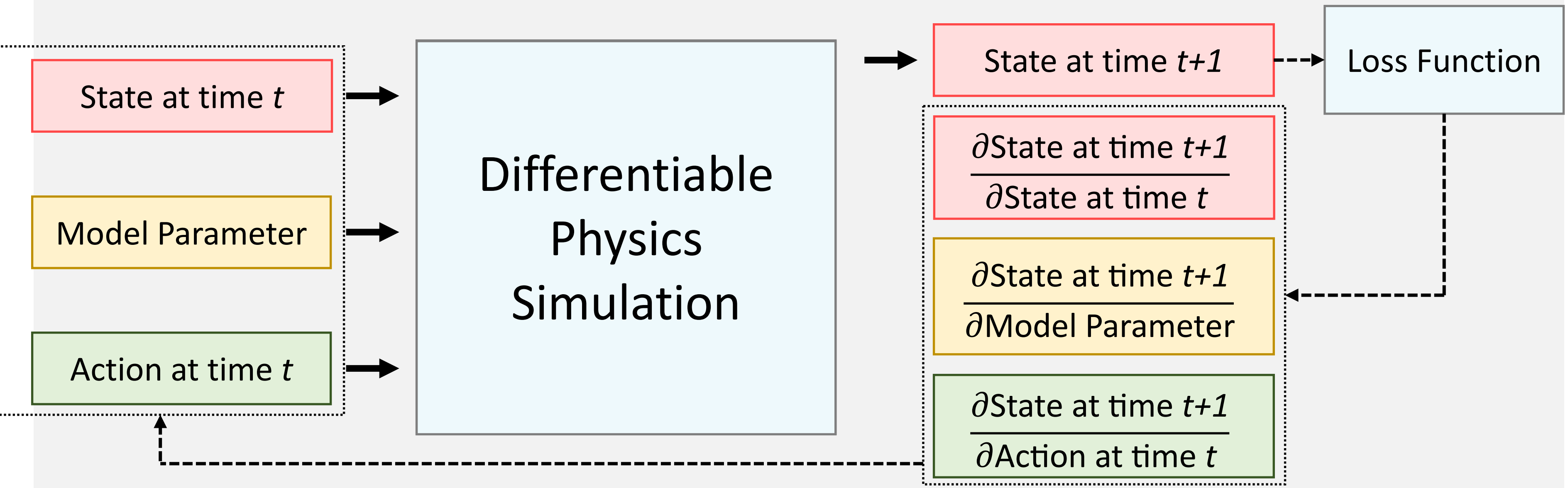
Takes raw sensory data as inputs

Satisfies Physics laws

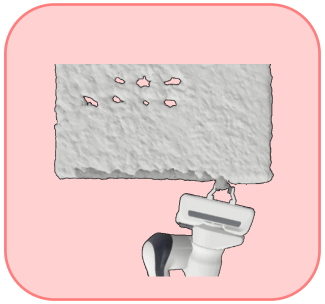
Self-update/learn and reduce modeling errors automatically

Significantly sample-efficient

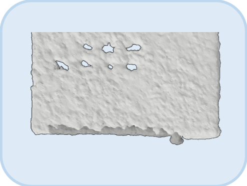
Differentiable Physic Simulation

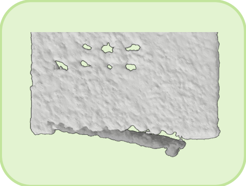


Robot-Centric Setting



Initial State





Goal State

$$\mathcal{L}^r(\mathcal{T}) = \sum_{t=0}^{T-1} l^r(s_t^{sim}, a_t^{sim}; \{x_t^*\}_{t=0}^{T-1}, \{u_t^*\}_{t=0}^{T-1}, \mathcal{T})$$

$$\begin{aligned} \text{s.t.} \quad s_{t+1}^{sim} &= \mathcal{F}^{sim}(s_t^{sim}, a_t^{sim}) \\ s_0^{sim} &= s_{\text{init}}^{o} \\ a_t^{sim} &= \pi(s_t^{sim}) \end{aligned}$$

object state sequence

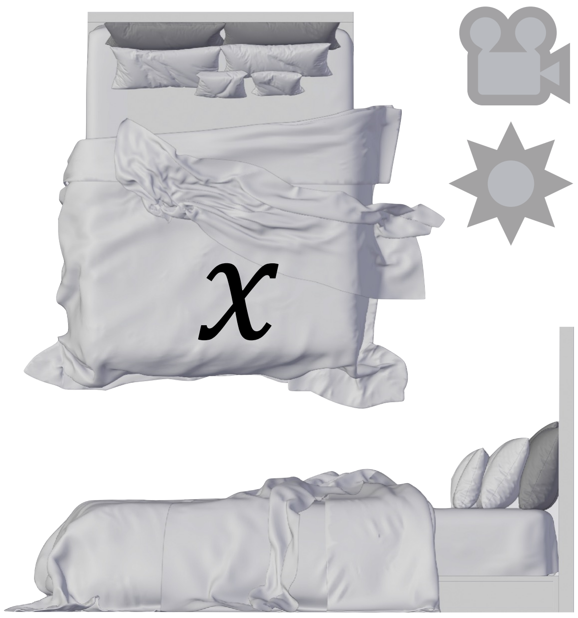
object action sequence

$$l^r(s_t^{sim}, a_t^{sim}; \{x_t^*\}_{t=0}^{T-1}, \{u_t^*\}_{t=0}^{T-1}, \mathcal{T}) = \begin{cases} \|a_t^{sim}\|^2 + \|x_t - x_t^*\| + \mathscr{C} + \Psi_t & t < T-1 \\ \|x_{T-1} - \mathcal{G}(\mathcal{T})\|^2 + \mathscr{C} + \Psi_t & t = T-1 \end{cases}$$

robot state

action state

Differentiable Rendering



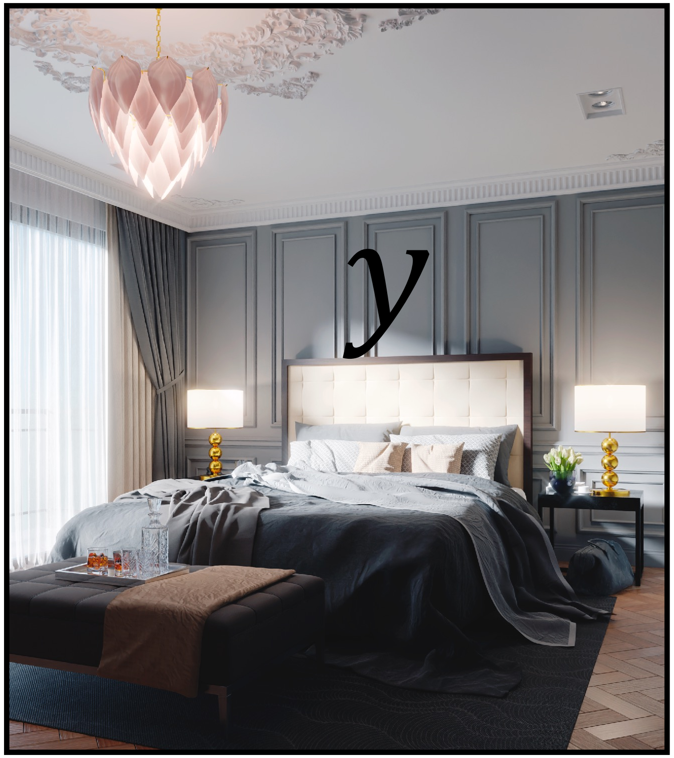
geometry, materials, emitters, camera, ...

Rendering

$$f(x) = y$$

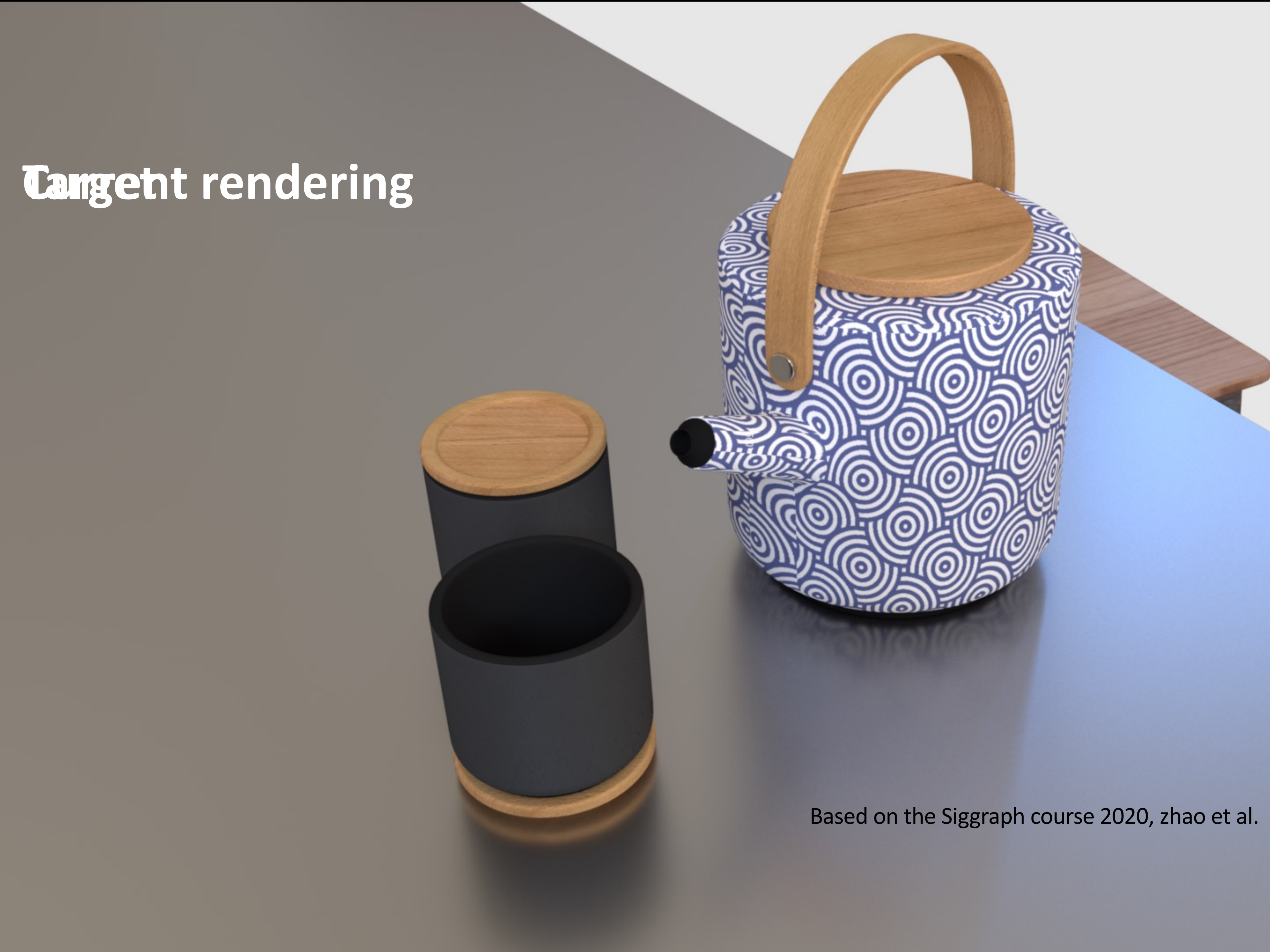
Differentiable rendering

$$\frac{dy}{dx}$$

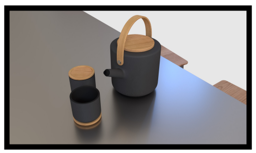
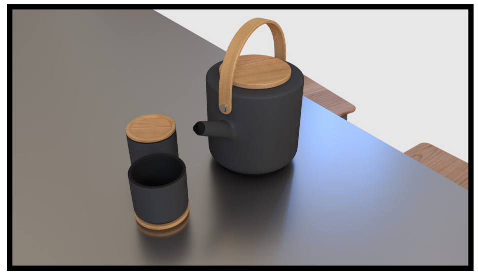
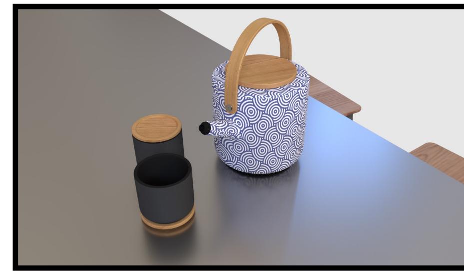


Scene: "bed classic" from jiraniano

Based on the Siggraph course 2020, zhao et al.


Target rendering
Current rendering
Based on the Siggraph course 2020, zhao et al.

Loss Function

$L\Big($  $\Big) = \Big\|$  $-$  $\Big\|^2$

Rendering

Target

scene parameter

The problem: $$\min_x L(f(x))$$

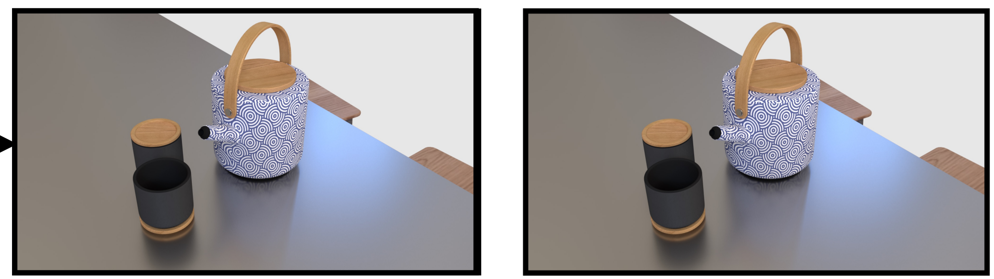
loss function

rendering

Based on the Siggraph course 2020, zhao et al

The problem: $\min_{x} L(f(x))$

x $\xrightarrow{f(x)}$



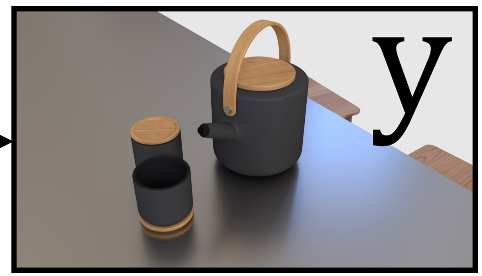
L $\xrightarrow{\frac{\partial L}{\partial x}}$ x

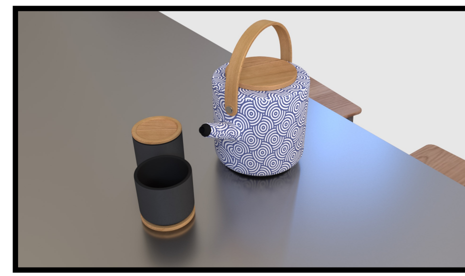
- meshes
- textures, etc.
- volumes, light sources, …
- material (BSDF) parameters
- parameters of procedural models

Based on the Siggraph course 2020, zhao et al

The problem: $\min_x L(f(x))$

$$x \xrightarrow{f(x)} y$$

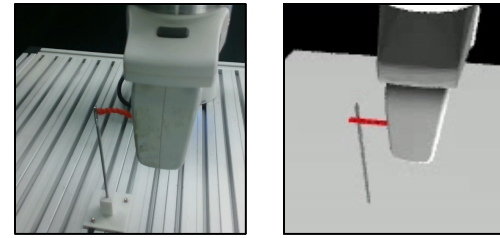




$$L \xrightarrow{\frac{\partial L}{\partial x}} x$$

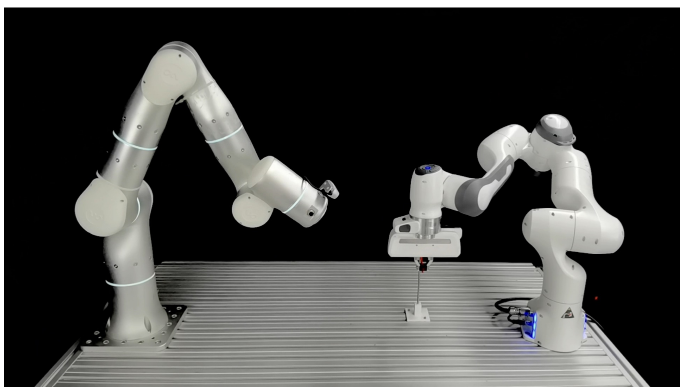
- meshes
- textures, etc.
- volumes, light sources, …
- material (BSDF) parameters
- parameters of procedural models

Based on the Siggraph course 2020, zhao et al

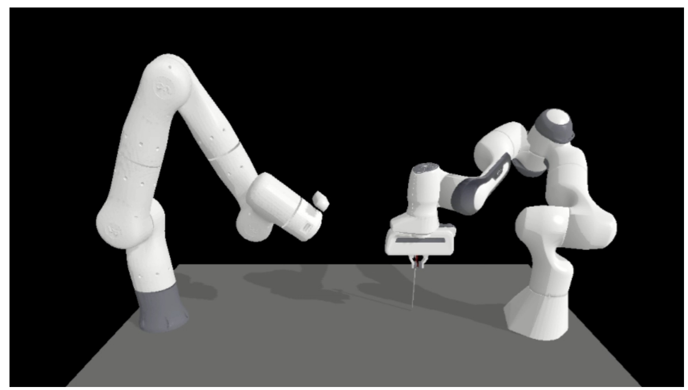


Differentiable Rendering

Real2Sim



Real World



Model

Learn@Sim

Sim2Real

Needle-Threading

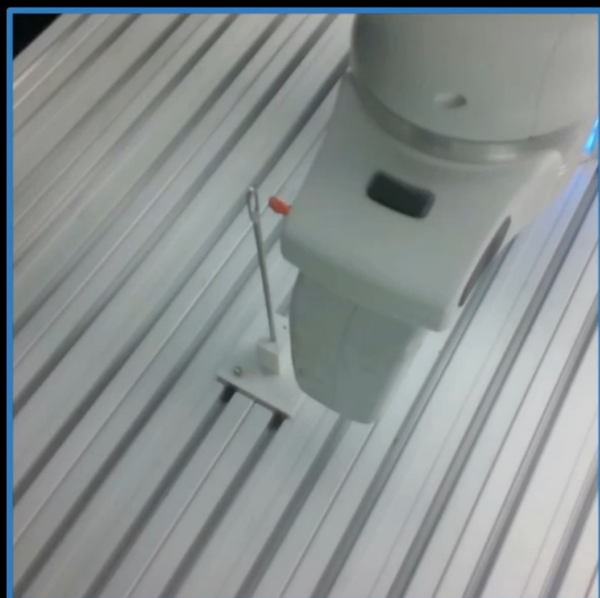
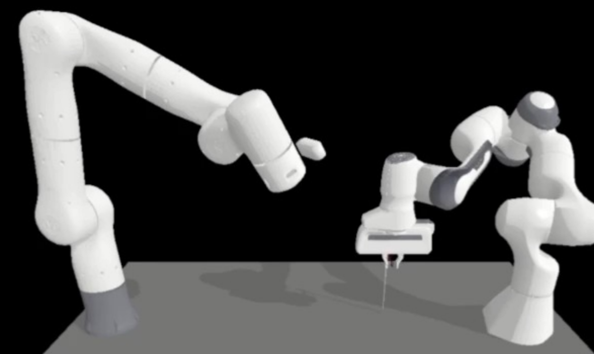
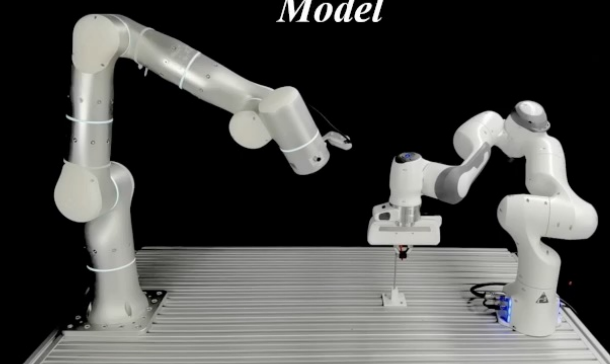


Image from RealSense every time step during manipulation



Model



Real World

Large Language Models

Topics

Spatial Descriptions
Kinematics
Dynamics

Configuration Space
Motion Planning in low-dim Spaces
Motion Planning in high-dim Spaces

Sensors& Robot Vision
Control
State Estimation

Grasping & Manipulation

MDP & Deep RL
POMDP

Advanced Topics

Today's Plan

- Overview of the course
- **Course logistics**
- Introduction to Robots

Staff

- Instructor: Lin Shao
- Office Hour:

  Time: Tuesday after the class

  Location: COM2-03-03
- Contact:

  Email: linshao@nus.edu.sg

  Canvas

Learning Objectives

- Understand, implement, and analyze robot algorithms. Apply them in specific robot task domains.
  - State estimation
    “Where am I?”
  - Reasoning and decision making
    “What shall I do?”
  - Control
    “How do I do it?”
- Understand the common architectures of intelligent robot systems.

Learning Objectives

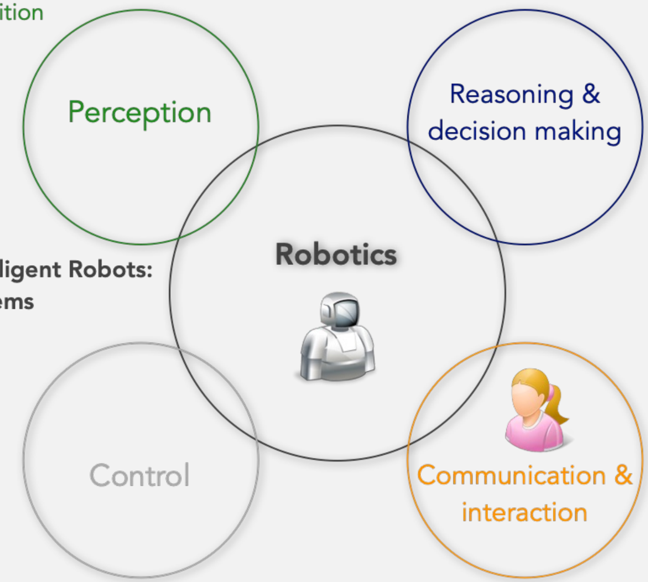
- Gain practical experiences in developing and debugging robot systems on common platforms, e.g., ROS.
- Gain exposure to one or more application domains of robotics.
  - Autonomous driving
  - Robot Grasping
  - Dexterous manipulation
  - Automated warehouse
  - ….

Related Modules

CS4243 Computer Vision & Pattern Recognition
CS4277/5477 3D Computer Vision

CS3244 Machine Learning
CS4246/5446 AI Planning and Decision Making
CS5242 Neural Networks & Deep Learning
CS5340 Uncertainty Modelling in AI

CS4278/5478 Intelligent Robots: Algorithms & Systems



Check offerings from BME, ECE, ME

CS4248 Natural Language Processing

CS6208 Advanced Topics in AI
CS6216 Advanced Topics in Machine Learning
CS6244 Advanced Topics in Robotics

Prerequisites

- Calculus
  - e.g., “gradient”
- Linear Algebra
  - e.g., “rotation matrix”
- Probability
  - e.g., “Bayes rule”
- AI
- Machine Learning

Prerequisites

- Prerequisites experiences. Proficient in at least one programming language
  - Python
  - C/C++

Working Load

- Homework 4
- Project
- No Midterm

Textbooks

- Supplementary Textbook
  - *Reinforcement Learning: An Introduction*, R.S. Sutton and A.G. Barto
  - *Probabilistic Robotics*. S. Thrun, W. Burgard, and D. Fox

Course website

- Canvas

Timeline

Date	Topics	Deadline
Week1 Aug 15	Introduction	
Week2 Aug 22	Kinematics	Hw1 Out
Week3 Aug 29	Dynamics	
Week4 Sept 05	C-Space Motion Planning	Hw1 DDL Hw2 Out
Week5 Sept 12	Motion Planning in low-Dim	
Week6 Sept 19	Motion Planning in high-Dim	
Week7 Oct 03	Sensors & Robot Vision	Hw2 DDL & HW3 Out Project Proposal

Timeline

Date	Topics	Deadline
Week8 Oct 10	Control	
Week9 Oct 17	State Estimation	
Week10 Oct 24	Grasping	Hw3 DDL & Hw4 Out Project Milestone
Week11 Oct 31	MDP&Deep RL	
Week12 Nov 07	POMDP	
Week13 Nov 14	Advanced Topics	Hw4 DDL
Dec 05		Project Report & Video

Grading Metrics

Component	Contribution to Grade
Homework	60% = 15% x 4
Final Project	40%
Total	100%

Component	Contribution to Grade
Project Proposal	5%
Project Milestone	5%
Project Presentation	10%
Final Project Report	20%
Total	40%

Grading Policies

- **Final Project (40%)**:

  Each student is required to work individually or in groups of up to three people on a research project.

  The project requires a 1-page proposal, a 2-page milestone report, a 5-8 page final report, and a final presentation video.

  All the page limits exclude references. Students are recommended to use the NeurIPS conference style.

Project Instructions

- **Project Proposal Reports (5%)**

  1-page proposal containing the problem definition, a literature review, a potential solution with the evaluation metric, and the timeline. You should schedule a meeting with me to discuss the project before the submission ddl.

- **Project Milestone Reviews (5%)**

  2-page writeup summarizing progress so far, if there need to be any changes, and the updated timeline.

- **Presentation Video, Possibly with Demo (10%)**

  a short (~10-min) presentation.

- **Final Project Report (20%)**:

  5-8 page project report.

Late Policy

- Each student will have 7 days for the late submission.
- You may use up to 2 late days per assignment with no penalty
- You may use late days for the project proposals, and project milestones but not for the final project reports and videos
- Once you have used all your free late days, we will deduct 25% of the total points per additional late day.

Collaboration Policy

- You may brainstorm, discuss ideas, etc. verbally with other students on problem sets and programming assignments.
- You must write up your solutions **independently** without resorting to any external help. In particular, “cut-and-paste” of any form and any extent is strictly forbidden and is treated as plagiarism.
- You should take reasonable precautions to prevent your work from being used for plagiarism.

Today's Plan

- Overview of the course
- Course logistics
- **Introduction to Robots**