Deep Reinforcement Learning

CS 224R

The Plan for Today

1. Course goals & logistics
2. Why study deep reinforcement learning?

Key learning today: what is deep reinforcement learning??

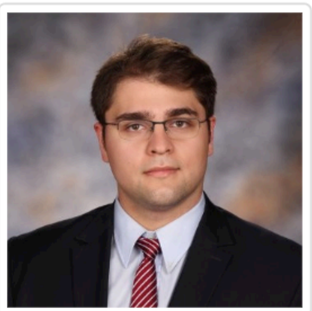
Introductions

Prof. Chelsea Finn

Instructor

Dr. Karol Hausman

Instructor



Rafael Rafailov

Head Teaching Assistant

Dilip Arumugam

Teaching Assistant

Annie Xie

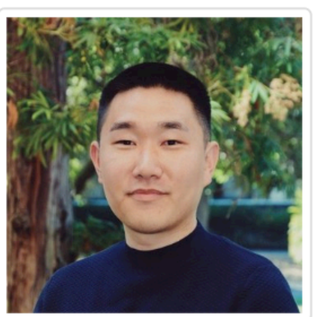
Teaching Assistant

Regina Wang

Teaching Assistant

Amelie Byun

Course Manager



John Cho

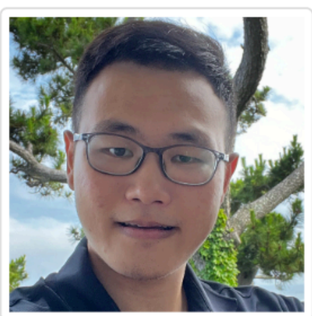
Course Coordinator

Ansh Khurana

Teaching Assistant

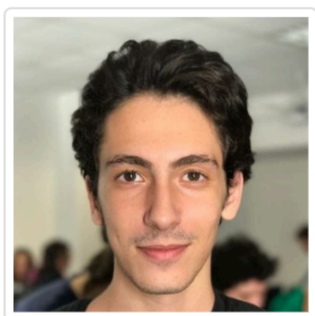
Saurabh Kumar

Teaching Assistant



Jonathan Yang

Teaching Assistant



Max Sobol Mark

Teaching Assistant

Welcome!

First question: How are you doing?

(answer by raising hand)

Information & Resources

Course website: http://cs224r.stanford.edu/ ← We have put a lot of info here Please read it. :)

Ed: Connected to Canvas

Staff mailing list: cs224r-spr2223-staff@lists.stanford.edu

Office hours: Check course website & Canvas, ***start on Weds***.

OAE letters can be sent to staff mailing list or in private Ed post.

Lectures & Office Hours

Lectures
- In-person, livestreamed, & recorded
- A few guest lectures (Jie Tan, Archit Sharma, one TBD)

Ask questions!
- by raising your hand

Office hours
- mix of in-person and remote

What do we mean by deep reinforcement learning?

Sequential decision-making problems

A system needs to make *multiple* decisions based on stream of information.

observe, take action, observe, take action, ...

AND the solutions to such problems

- imitation learning
- model-free & model-based RL
- offline & online RL
- multi-task & meta RL

and more!

Emphasis on solutions that scale to deep neural networks

How does deep RL differ from other ML topics?

Supervised learning

Given labeled data: $\{(x_i, y_i)\}$, learn $f(x) \approx y$

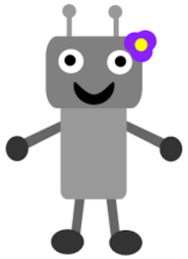
- directly told what to output
- inputs x are independently, identically distributed (i.i.d.)

Reinforcement learning

Learn *behavior* $\pi(a \mid s)$.

- from experience, indirect feedback
- data **not** i.i.d.: actions a affect the future observations.

Behavior can include:



motor control

dialog

game playing

driving

We can’t cover everything in deep RL.

We’ll focus on:

- methods and implementation
- examples in robotics & control (but techniques generalize broadly)
- topics that we think are most useful & exciting!

For more theory & other applications, see CS234!

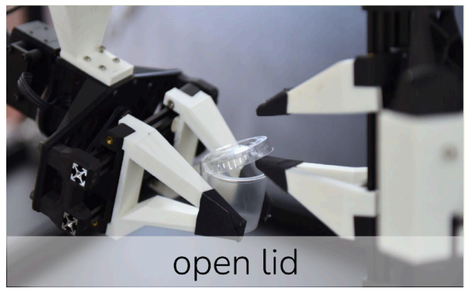
Topics

1. Imitation learning
   (behavior cloning, inverse RL)
2. Model-free deep RL algorithms
   (policy gradients, actor-critic methods, Q-learning)
3. Model-based deep RL algorithms
4. Offline RL methods
   (e.g. conservative methods, decision transformers)
5. Multi-task and meta RL topics
   (e.g. hindsight relabeling, learning to explore)
6. Advanced topics: hierarchical RL, sim2real, reset-free RL

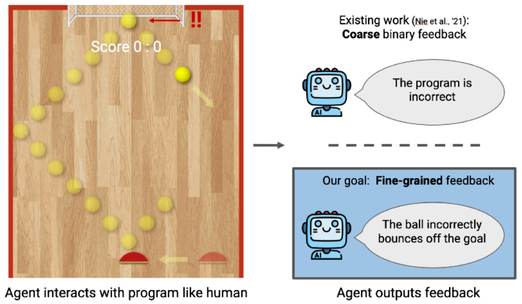
Emphasis on **deep learning** techniques.

Case studies of interesting & timely applications

- Fine-grained manipulation skills
- RL with human feedback (RLHF) for language models
- Educational feedback on interactive assignments



Zhao et al. Learning Fine-Grained Bi-Manual Manipulation. 2023



Liu et al. Giving Feedback on Interactive Student Programs with Meta-Exploration. NeurIPS 2022

What will you learn in this course?

1. The foundations of modern deep learning methods for sequential decision making
2. How to implement and work with practical deep RL systems (in PyTorch)
3. A glimpse into the scientific and engineering process of building and understanding new algorithms

Pre-Requisites

Machine learning: CS229 or equivalent.

e.g. we'll assume knowledge of SGD, cross-val, calculus, probability theory, linear algebra

Some familiarity with deep learning:

- We'll build on concepts like backpropagation, convolutional networks, recurrent networks
- Assignments will require training networks in **PyTorch**.
- Annie will hold a PyTorch review session on Thursday, April 6, 4:30 pm PT in Gates B3.

Some familiarity with reinforcement learning:

- We will go quickly over the basics.
- See Sutton & Barto or CS 221 for intro RL content

Assignments

Homework 1: Imitation learning

Homework 2: Online reinforcement learning

Homework 3: Offline reinforcement learning

Homework 4: Goal-conditioned & meta reinforcement learning

Grade of lowest-scoring HW worth only **5% of grade**.

Rest are worth **15% of grade**.

Grading: 50% homework, 50% project

6 late days total across: homeworks, project-related assignments
maximum of 2 late dates per assignment

Collaboration policy: Please read course website & honor code.
Document collaborators & write up HW solutions on your own.

AI tools (e.g. ChatGPT, Copilot) not allowed for homework, allowed for final project.

Final Project

Research-level project of your choice

- in groups of **1-3 students**
- if applicable, encouraged to **use your research**!
- can share with other classes, with slightly higher expectation
- same late day policy as HWs
  (but no late days for poster)

Poster presentation on June 7th (4-7 pm)

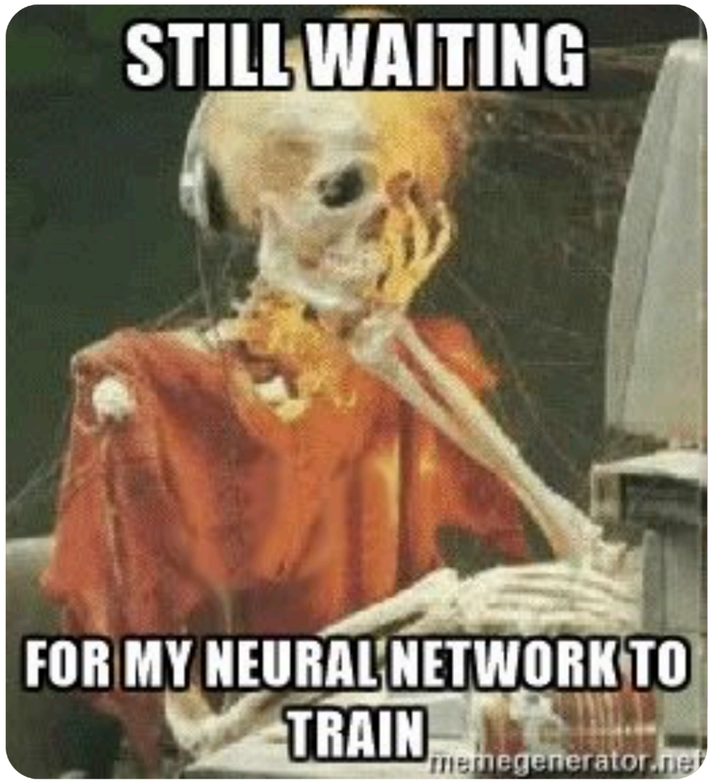
A word of warning

Deep RL methods take a long time to learn behavior!

We are trying to make homeworks fast to train.

(e.g. by using simple environments)

But, they will still take some time & you may choose to be more ambitious in your project.



We recommend that you don’t start HWs/project deliverables the night before the deadline. :)

One more thing

This course is new!

We have been working hard to develop a great course!

But, we will probably make mistakes.

We would **love** your feedback both for this iteration & future iterations.

—> **high-resolution feedback form** sent weekly to subset of students.

Initial Steps

1. Homework 1 coming out on Weds — due Weds 4/19 at 11:59 pm PT
2. Start forming final project groups if you want to work in a group

The Plan for Today

1. Course goals & logistics
2. **Why study deep reinforcement learning?**

Some of Chelsea's Research

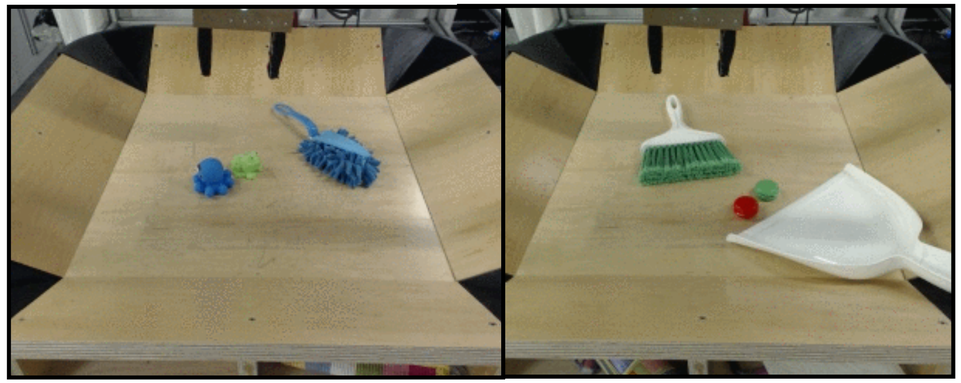
(and why I care about deep reinforcement learning)

How can we enable agents to develop broadly intelligent behavior?

Robots.

Zhou, Kim, Wang, Florence, Finn. CVPR ‘23

Chen, Nair, Finn. RSS ‘21



Xie, Ebert, Levine, Finn, RSS ‘19

Why robots? Robots can teach us things about intelligence.

faced with the **real world**

must **generalize** across tasks, objects, environments, etc

need some **common sense understanding** to do well

supervision can’t be taken for granted

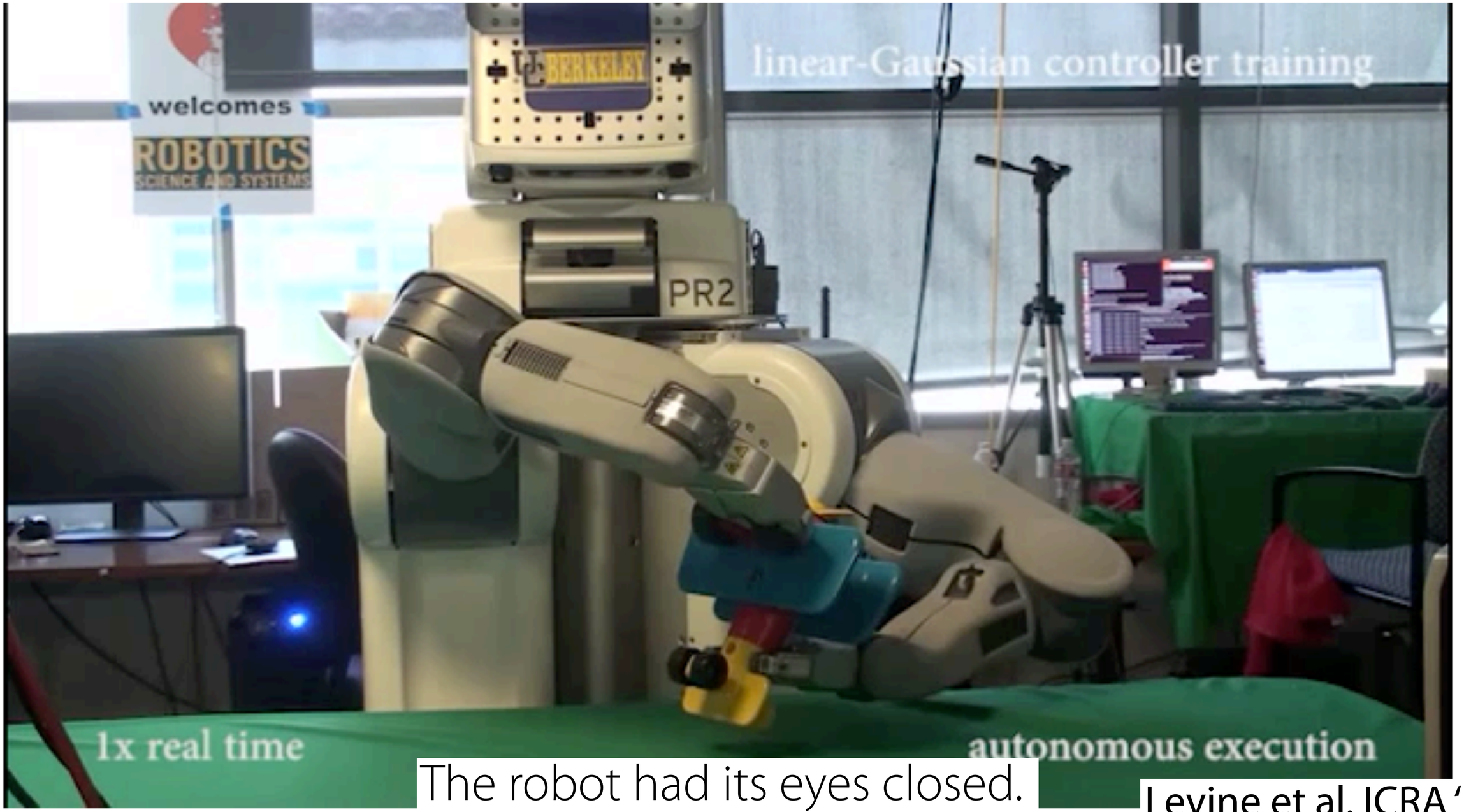
Sequential decision making & embodiment seem fundamental to our intelligence.

Robots have tremendous potential for positive societal impact

Recent examples: labor shortages, helping care for aging populations

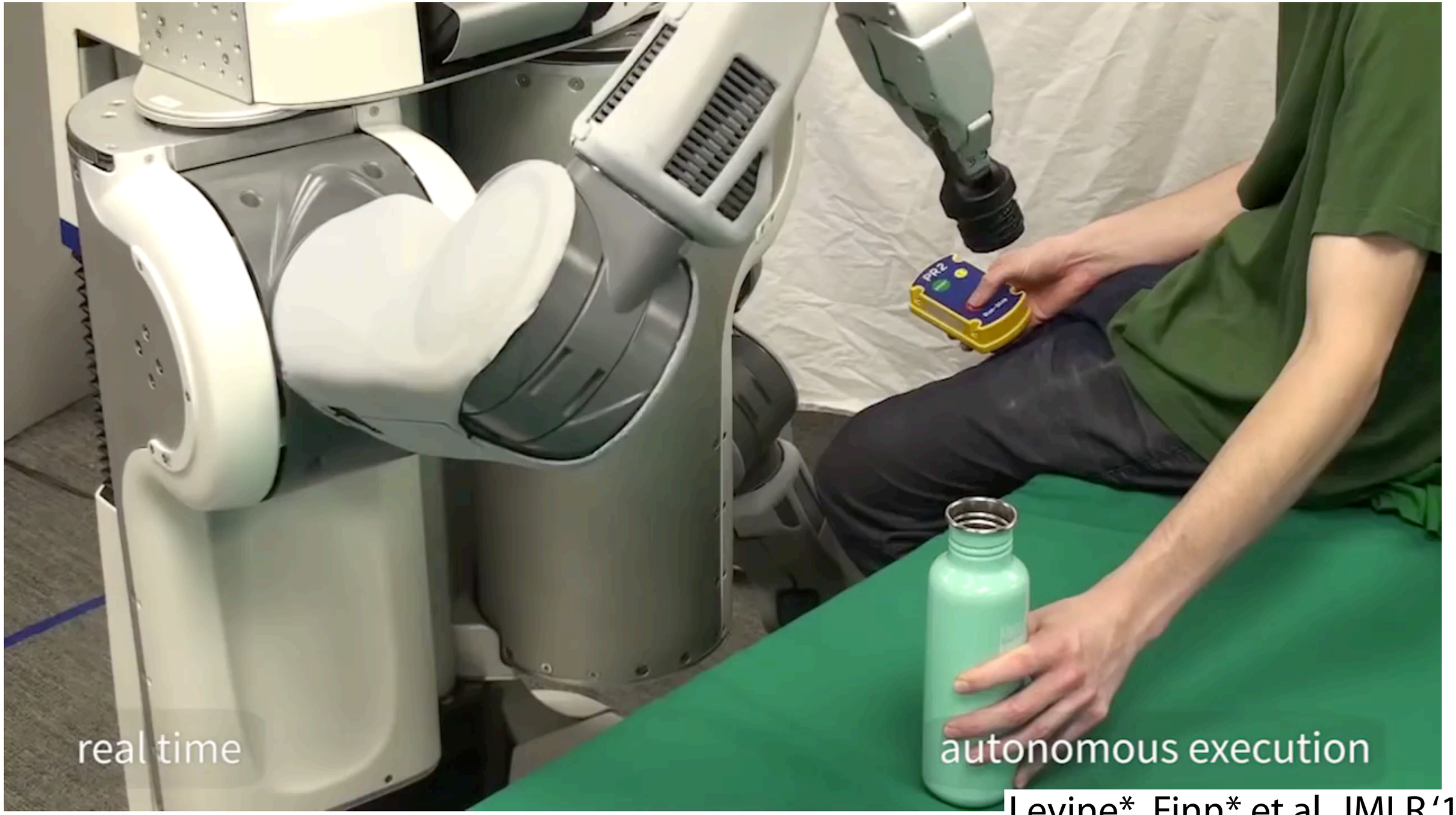
- **search and rescue** after natural disasters
- helping perform **dangerous & tedious jobs**
- assist in **surgery**
- **drive trucks and cars** without distraction
- **space exploration**
- **agriculture**
- and more

Beginning of my PhD

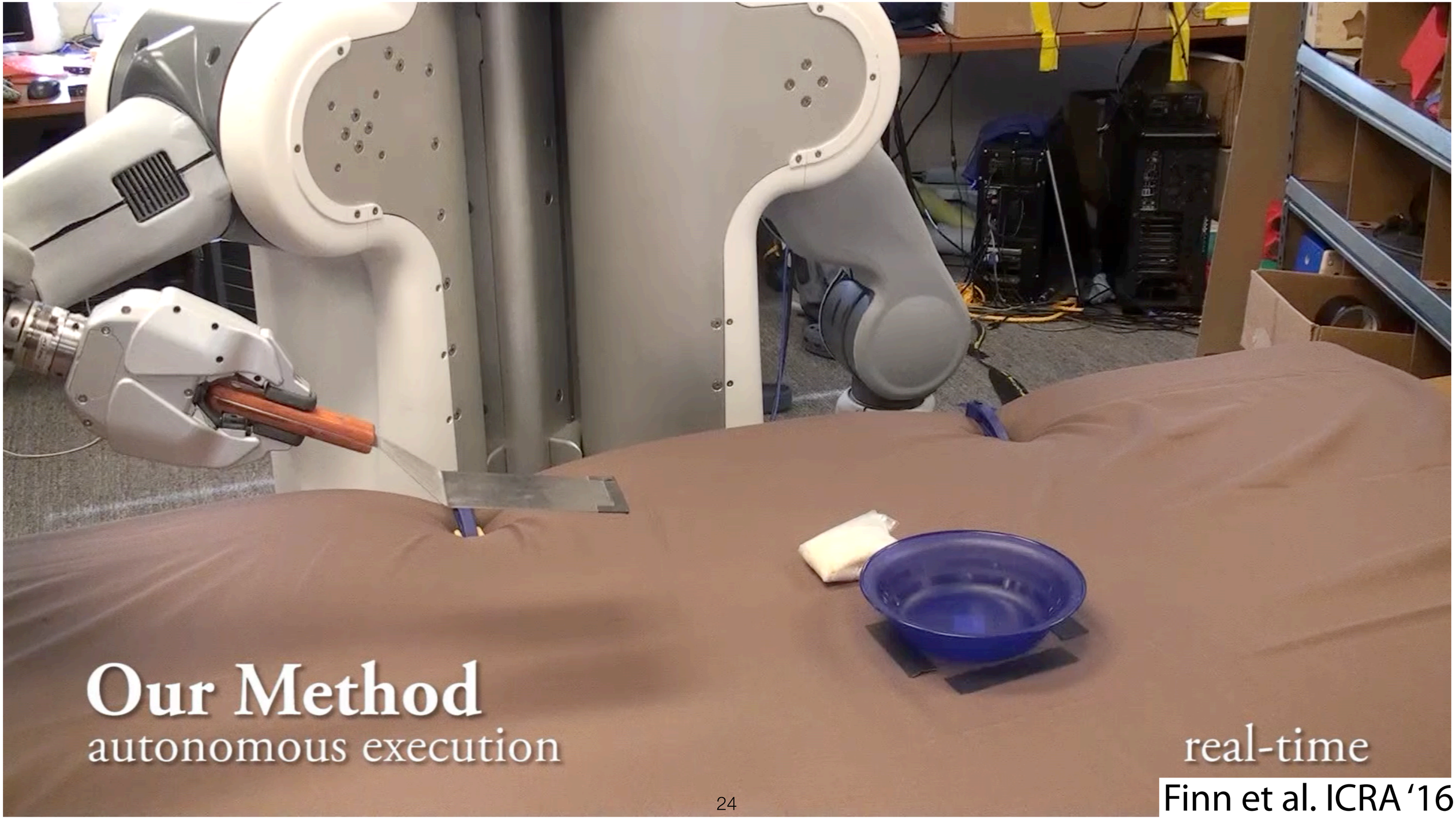


The robot had its eyes closed.

Levine et al. ICRA ‘15

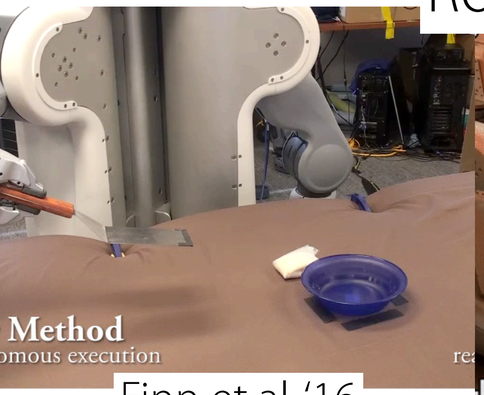


Levine*, Finn* et al. JMLR ‘16



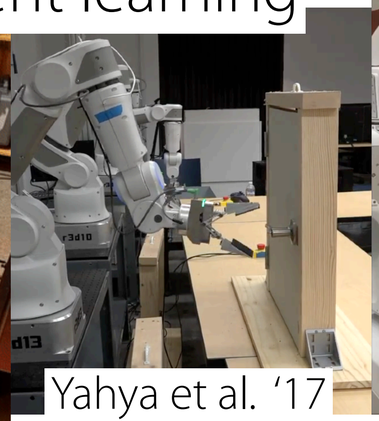
Finn et al. ICRA ‘16

Robot reinforcement learning

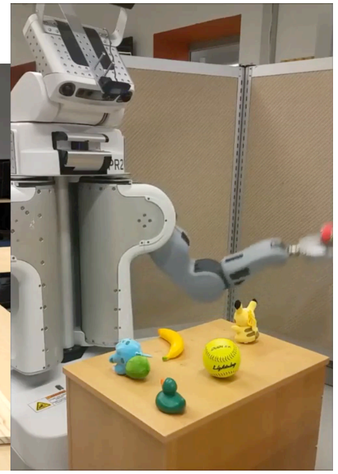


Finn et al.'16

Chebotar et al. '17

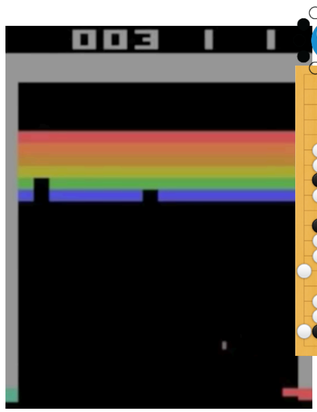


Yahya et al. '17

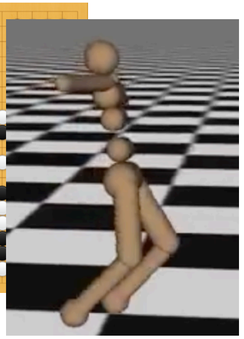


Ghadirzadeh et al. '17

Reinforcement learning



Atari



locomotion

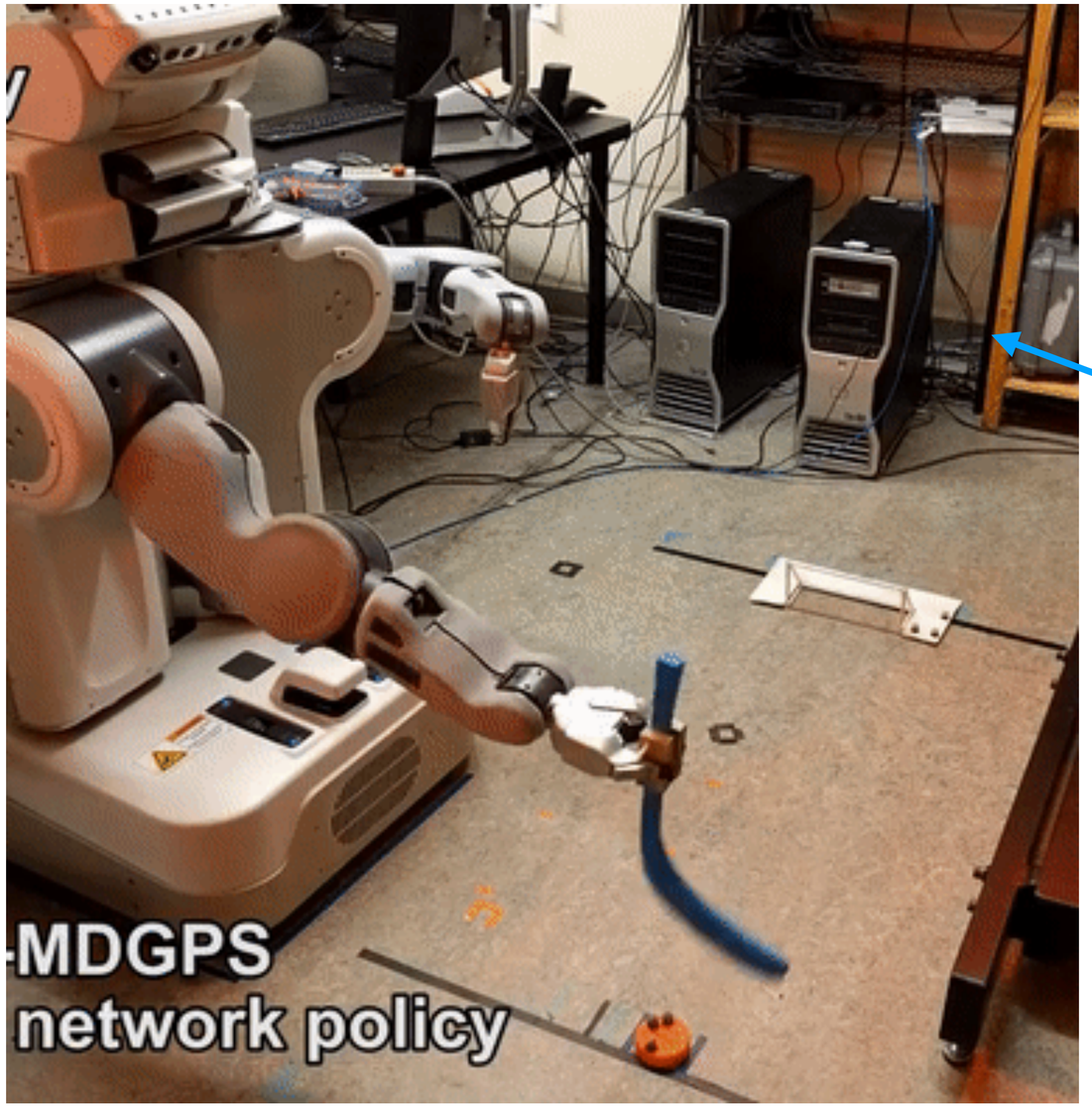
Reinforcement learning is a powerful tool for robotics & game playing!

But, there are also a lot of *big, open* problems.

1. Can robots learn ***generalizable*** behavior? e.g. from large offline datasets —> offline RL
   e.g. by leveraging experience across tasks —> multi-task, meta RL

2. Can robots learn behavior ***autonomously***?

Behind the scenes…

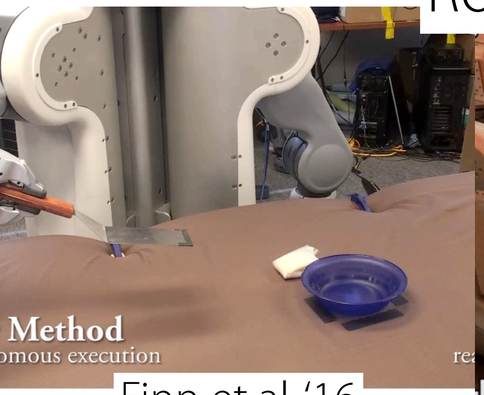


Yevgen

Yevgen is doing more work than the robot!

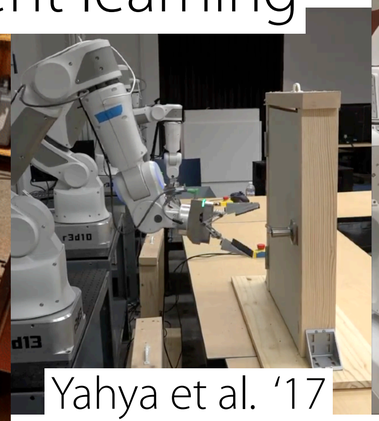
It’s not practical to collect a lot of data this way.

Robot reinforcement learning

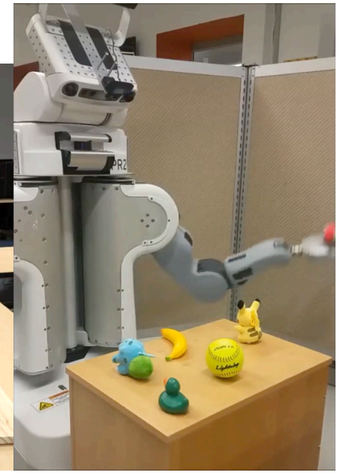


Finn et al.'16

Chebotar et al. '17

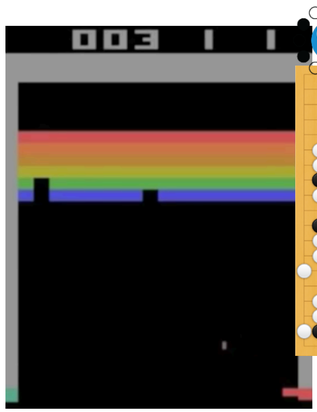


Yahya et al. '17

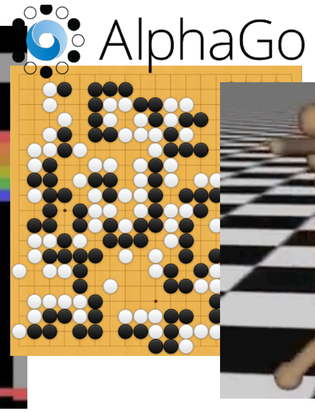


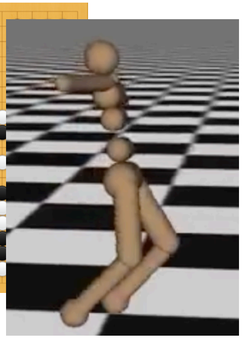
Ghadirzadeh et al. '17

Reinforcement learning



Atari





locomotion

Reinforcement learning is a powerful tool for robotics & game playing!

But, there are also a lot of *big, open* problems.

1. Can robots learn ***generalizable*** behavior? e.g. from large offline datasets —> offline RL
   e.g. by leveraging experience across tasks —> multi-task, meta RL

2. Can robots learn behavior ***autonomously***? ("Reset-free RL" —> guest lecture)

Some of Karol's Research

(and why Karol cares about deep reinforcement learning)


23:30:02 06/06/2015 UTC

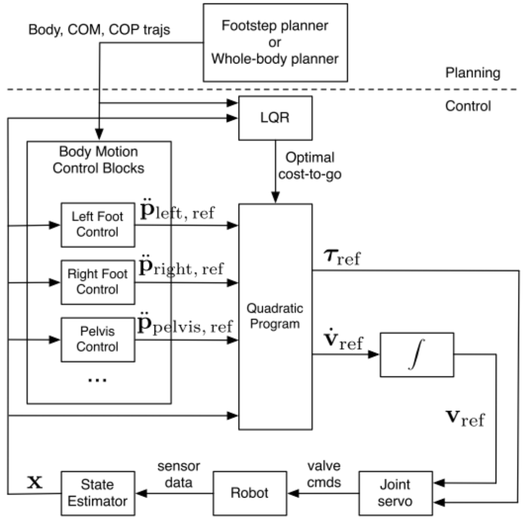
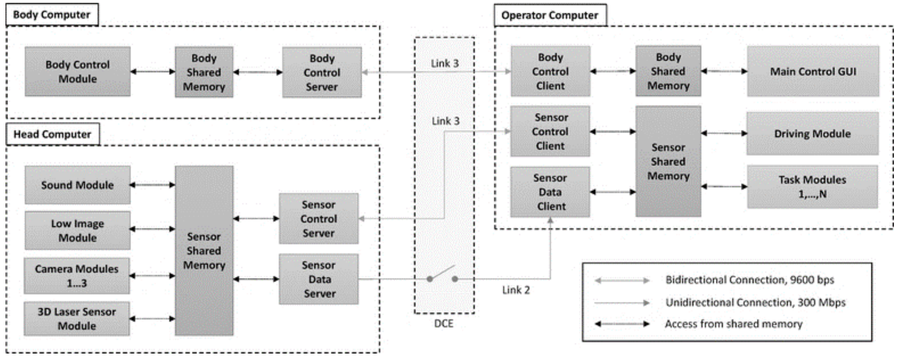
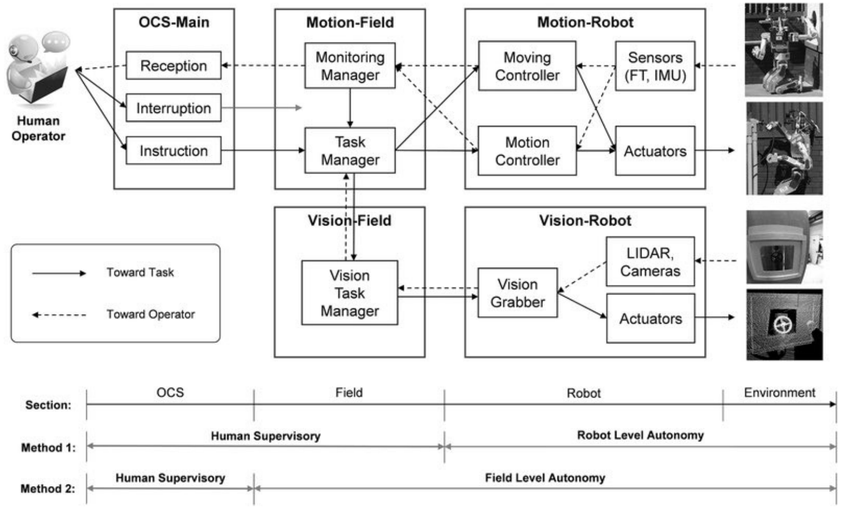
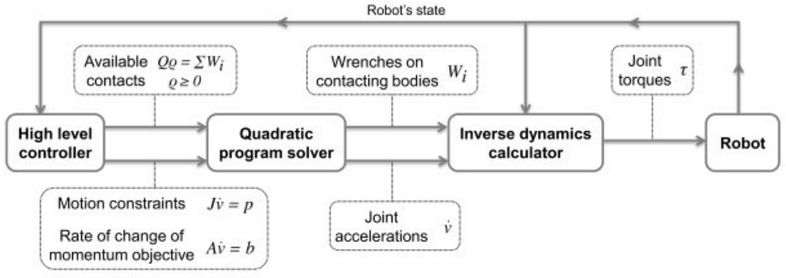


Fig. 6 Block diagram illustrating the flow of signals through the system. The footstep planner (Section 3.1) or the whole-body motion planner (Section 3.2) provide input desired trajectories to the control system. The controller (Section 4) runs in a closed loop with the state estimator (Section 5) at approximately 800 Hz. LQR solutions can be recomputed online (typically in a separate thread) using the current state of the robot to reduce the systems sensitivity to deviations from the nominal walking trajectory.







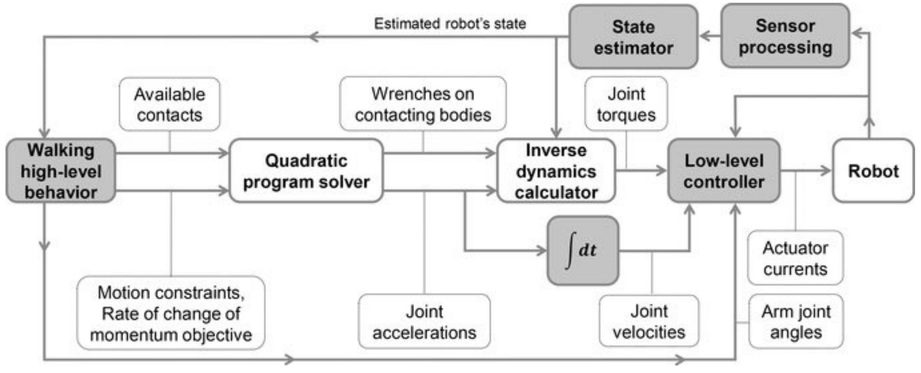


Fig. 5. Overview of information flow in the lower level parts of the controller

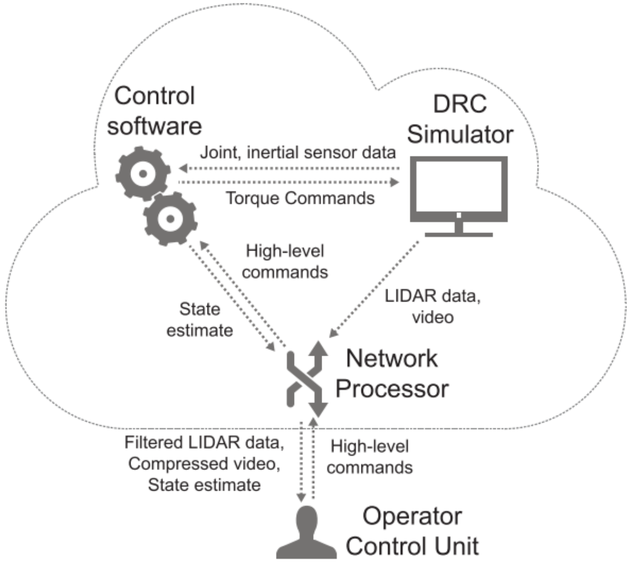
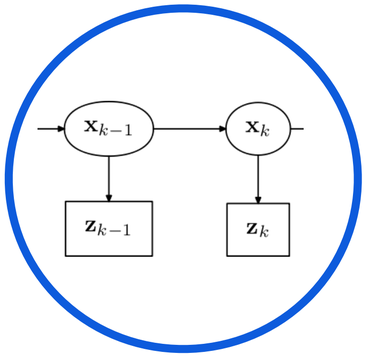
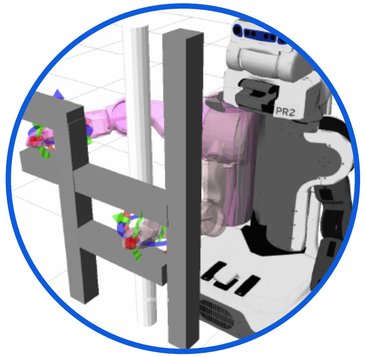


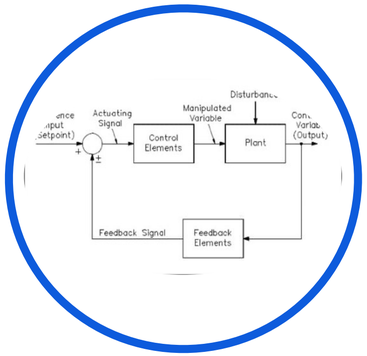
Fig. 6. Network layout for the IHMC VRC entry. The control software communicates with the DRC simulator through a high bandwidth connection at a rate of 1000 Hz. The Operator Control Unit communicates through a network processor at low bandwidth and high latency.

Traditional Robotic Pipeline

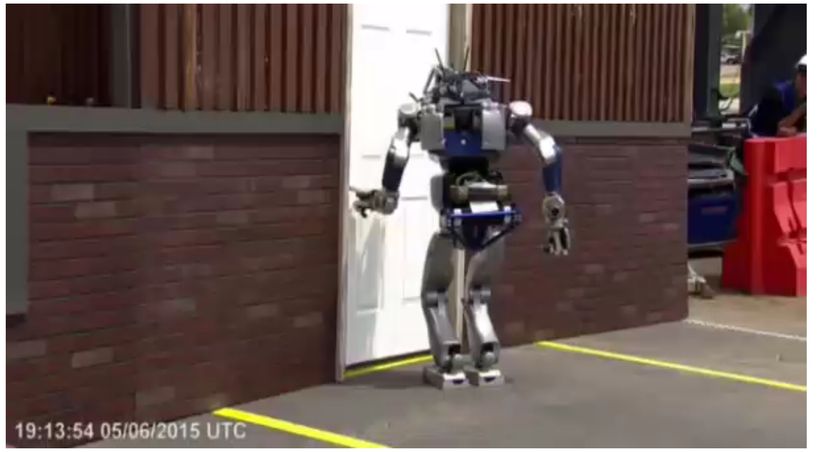
observations → state estimation → planning → low-level control → motor commands



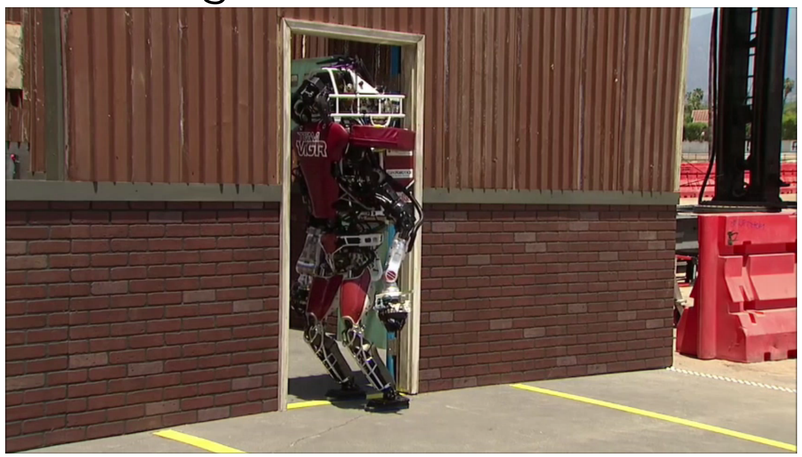


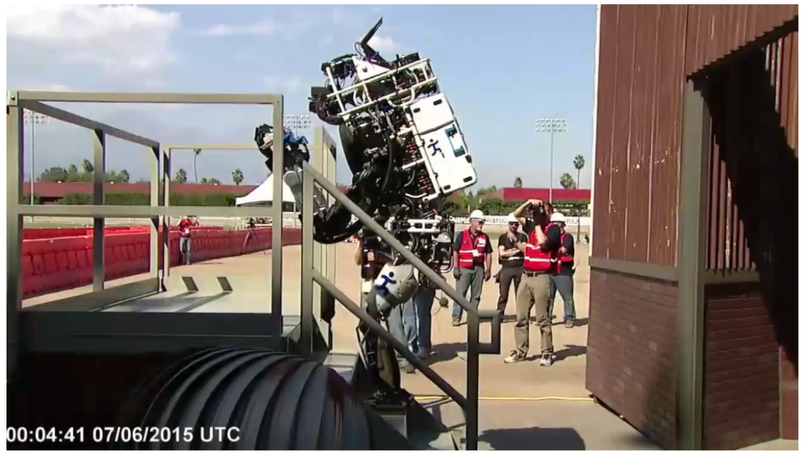


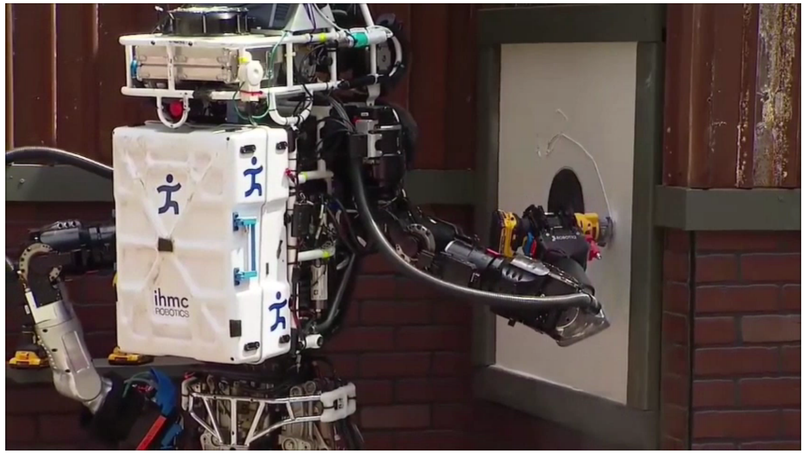
problems with contact



avoiding contact







Traditional Robotic Pipeline

perception

action

observations

state estimation

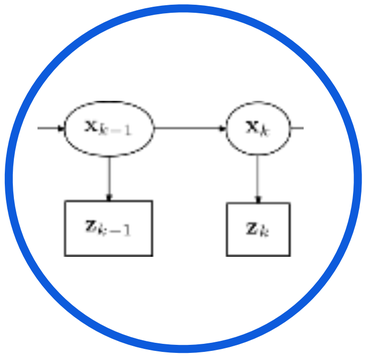
plannin

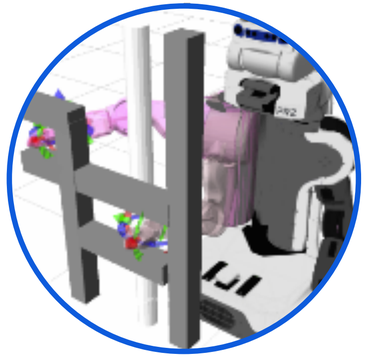
-lev

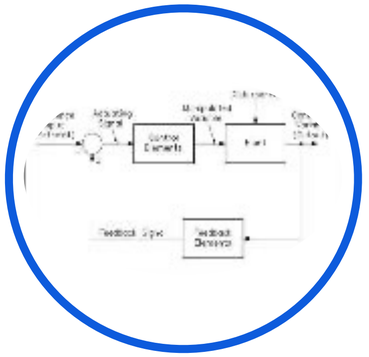
rol

tor

nmands





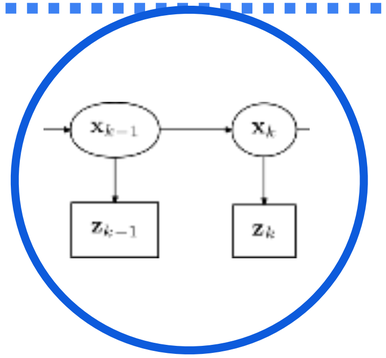


Traditional Robotic Pipeline

perception

observations

state estimation

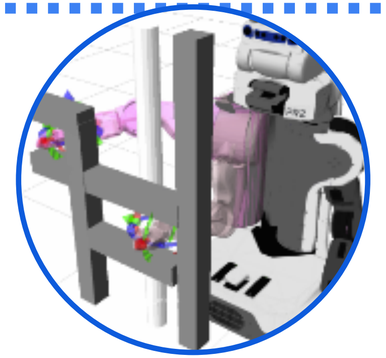


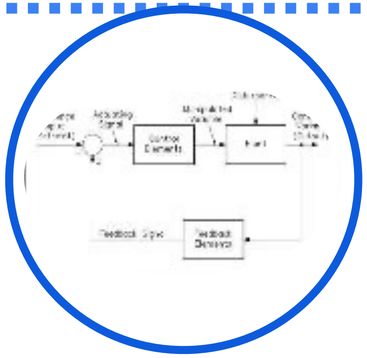
action

planning

low-level control

motor commands





perception

action

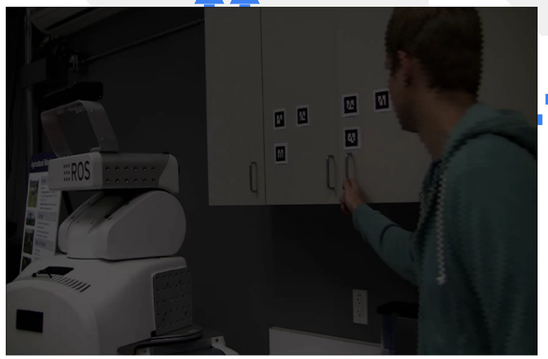
observations

state estimation

planning

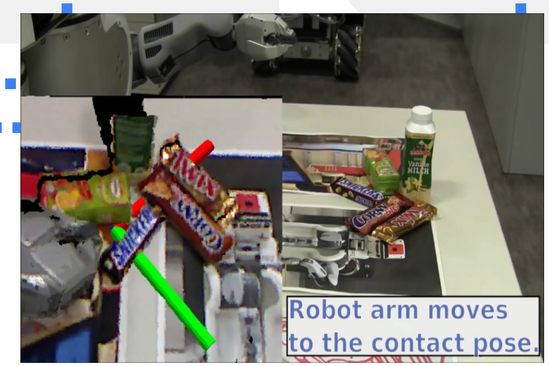
low-level control

motor commands

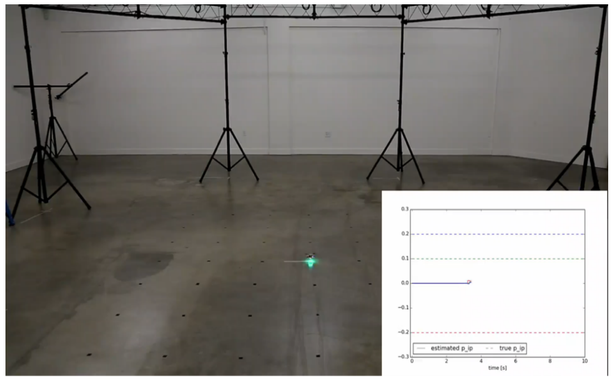


[H. et al., ICRA 2015]

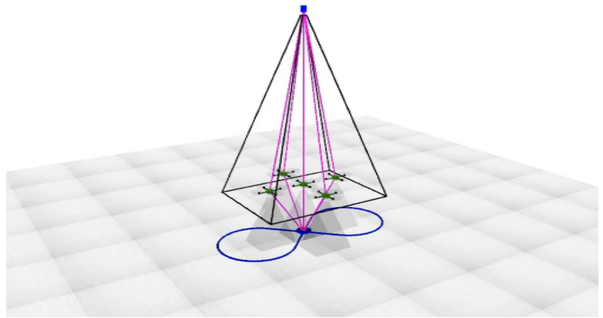
[H. et al., IROS, ISER 2016]



[H. et al., ICRA 2013]



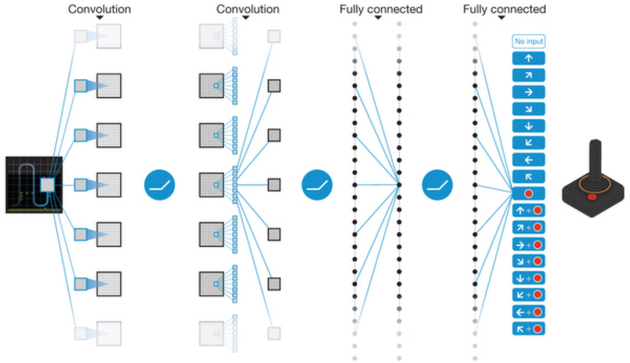
[H. et al., ICRA, RSS, IJRR 2017]



[H. et al., IROS, IJRR 2016]

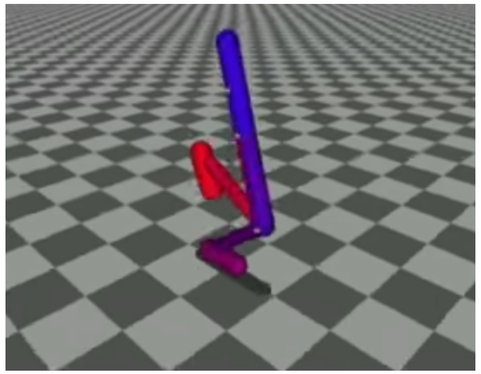
Atari games

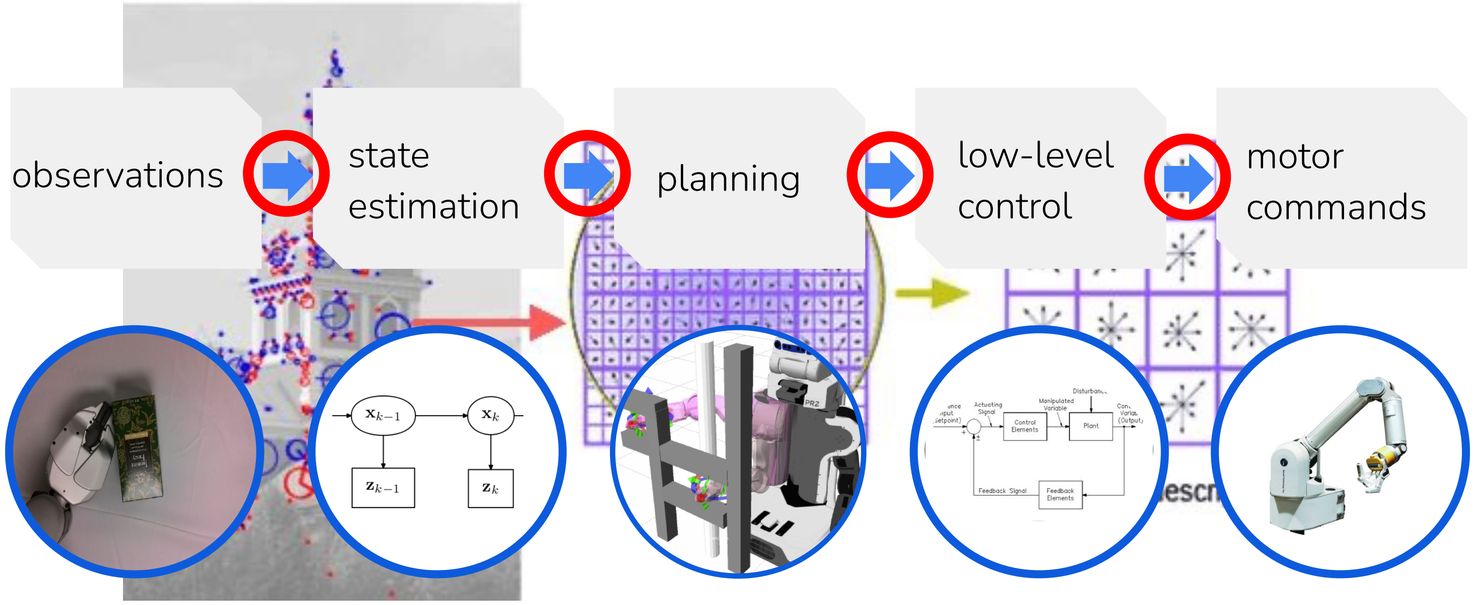
DQN, Mnih et al. 2013

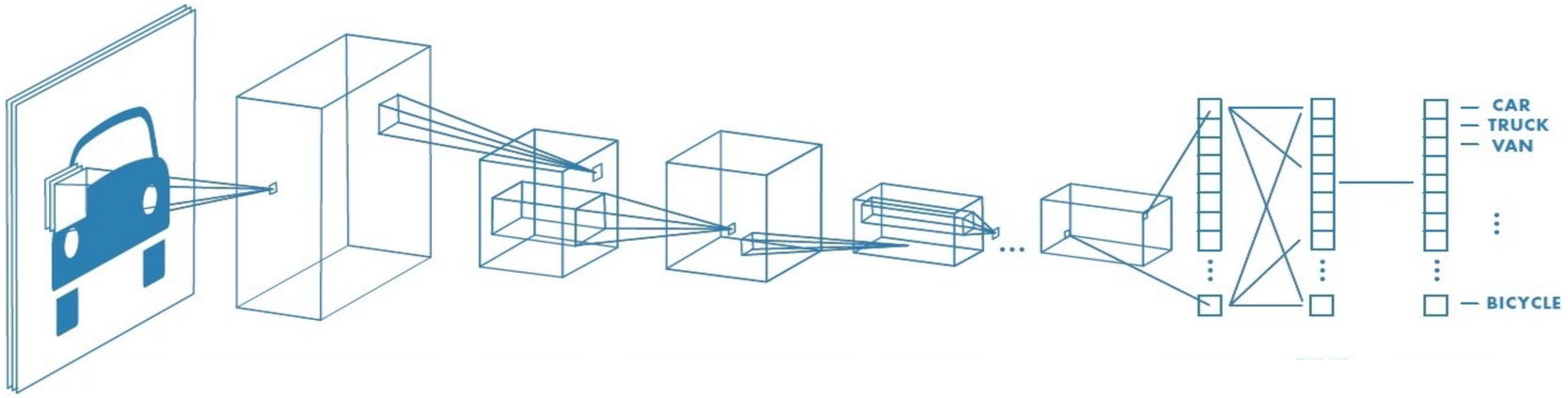


Continuous control

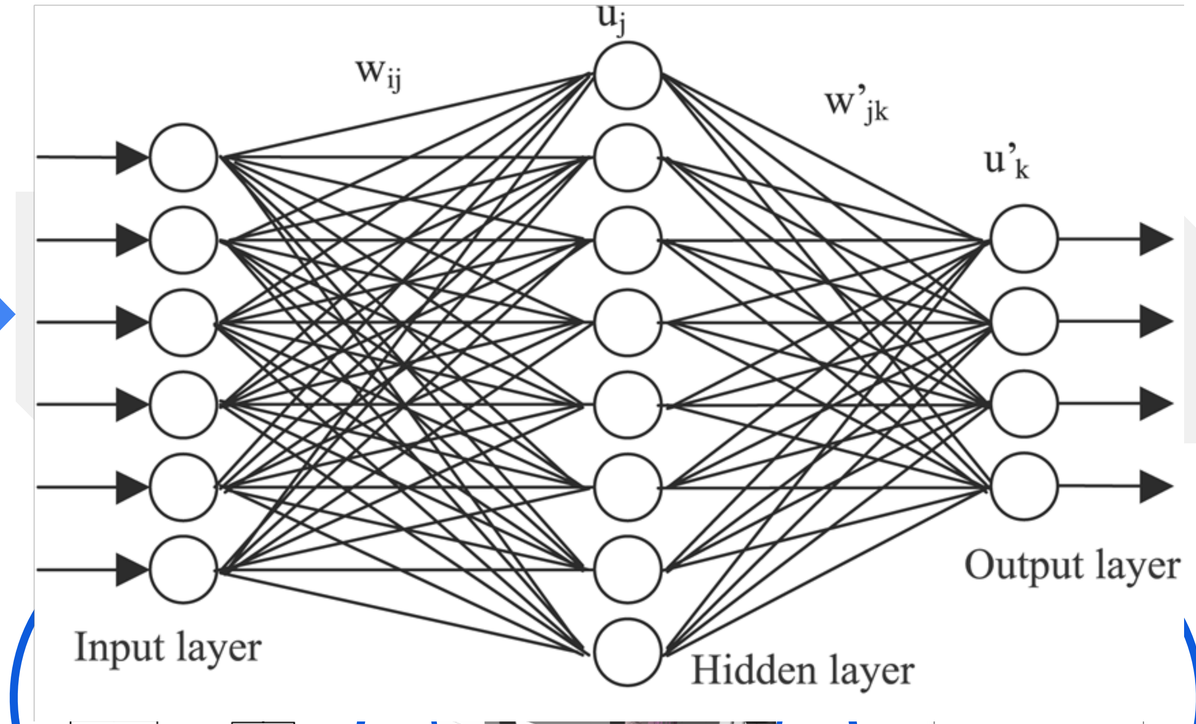
TRPO, Schulman et al. 2015



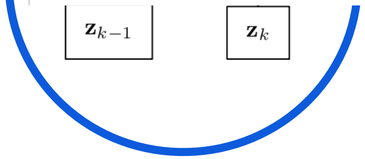

observations
state estimation
planning
low-level control
motor commands
$\mathbf{x}_{k-1}$
$\mathbf{x}_k$
$\mathbf{z}_{k-1}$
$\mathbf{z}_k$

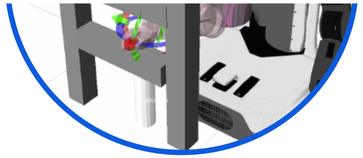

CAR
TRUCK
VAN
BICYCLE

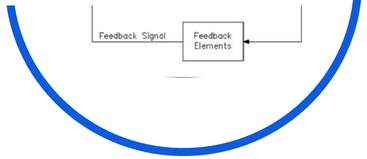
observations



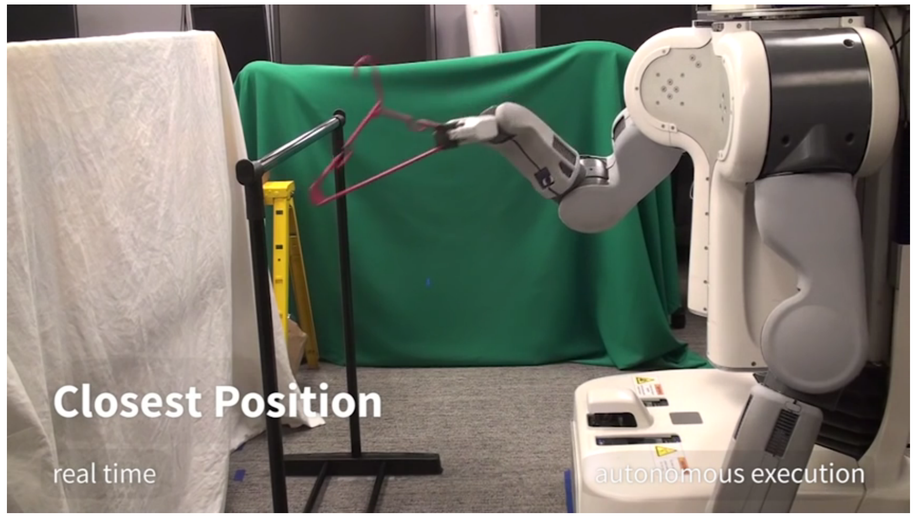
motor commands





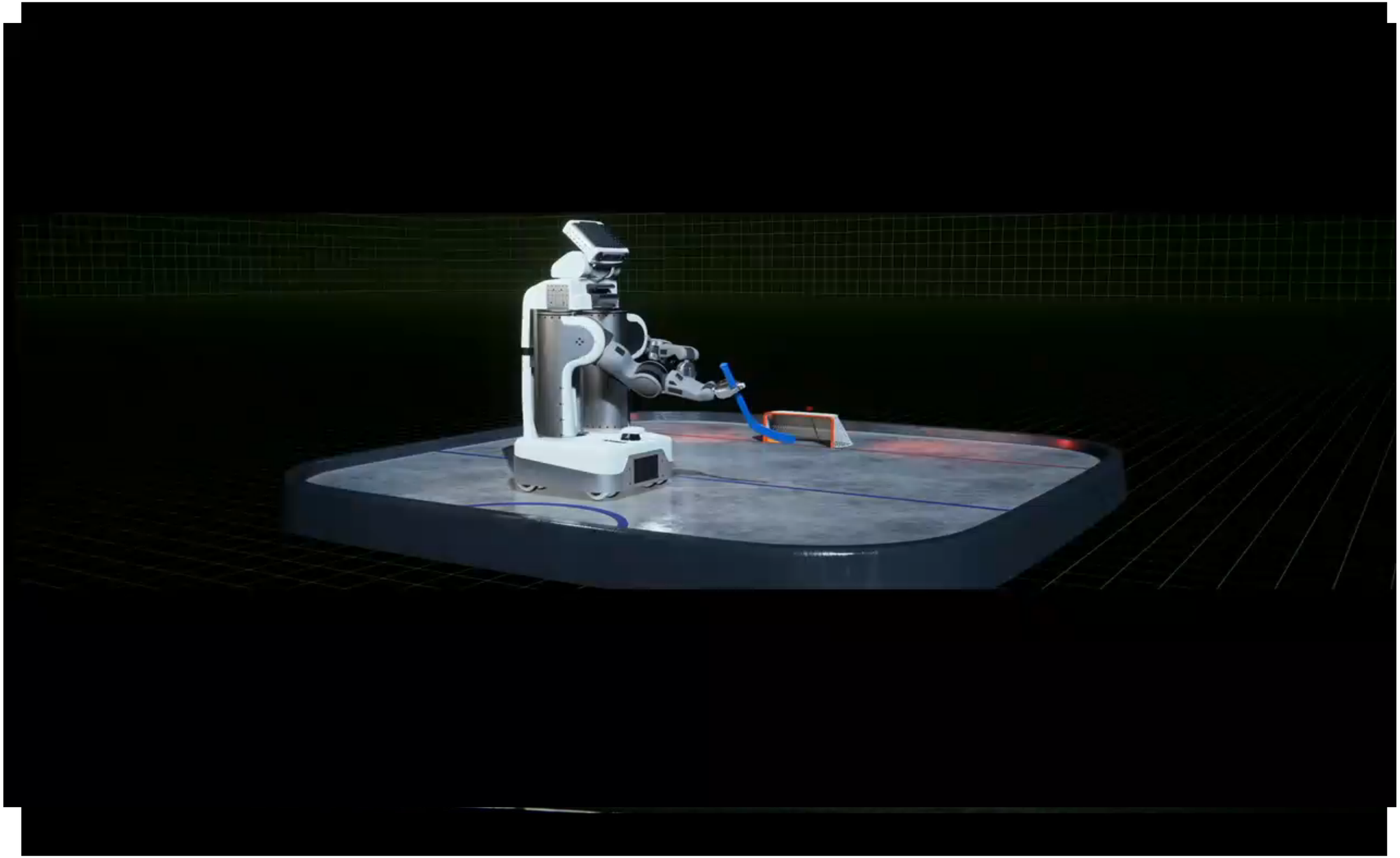


Guided Policy Search (GPS)
Levine, Finn, et al. 2015



PIGPS
Chebotar et al. 2016

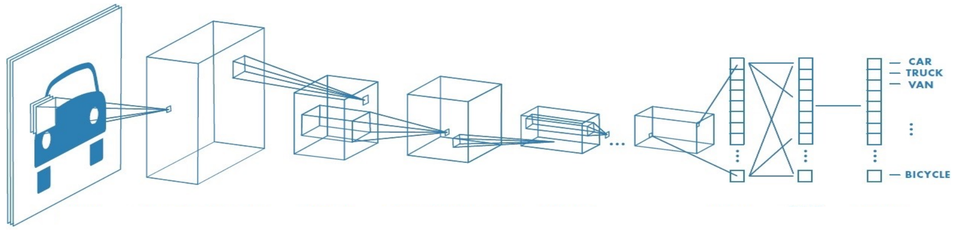
Combining Model-Based and Model-Free Updates for Trajectory-Centric Reinforcement Learning

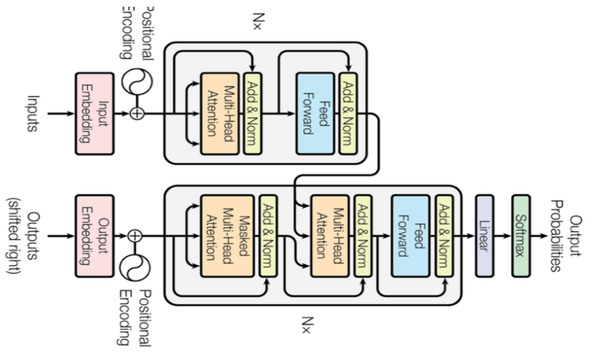


The recipe that has worked in other fields so far:

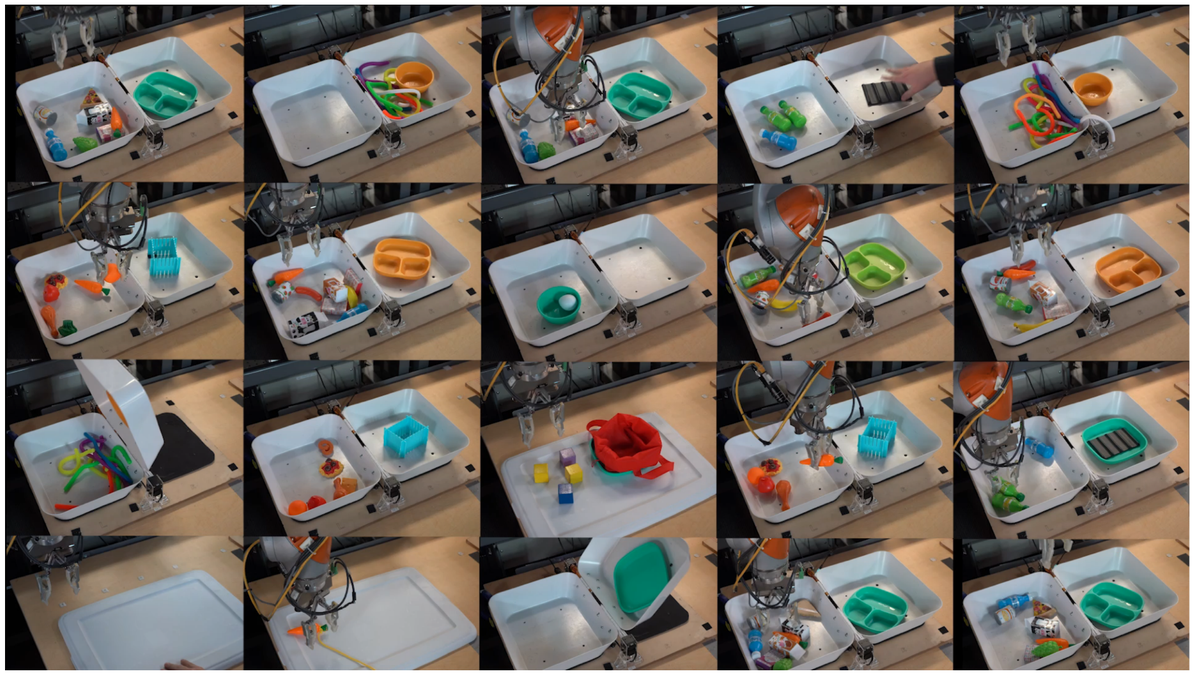
A lot of diverse, multi-task data

Expressive, capable models





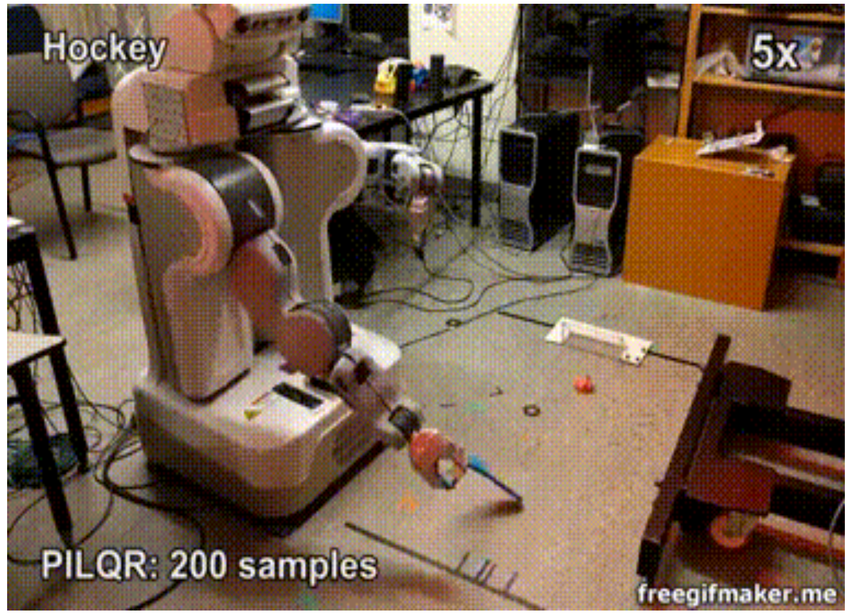
A lot of diverse, multi-task data



MT-Opt
Kalashnikov et al. 2021

RT-1
Brohan et al. 2022

Multi-task robot learning: the real reason



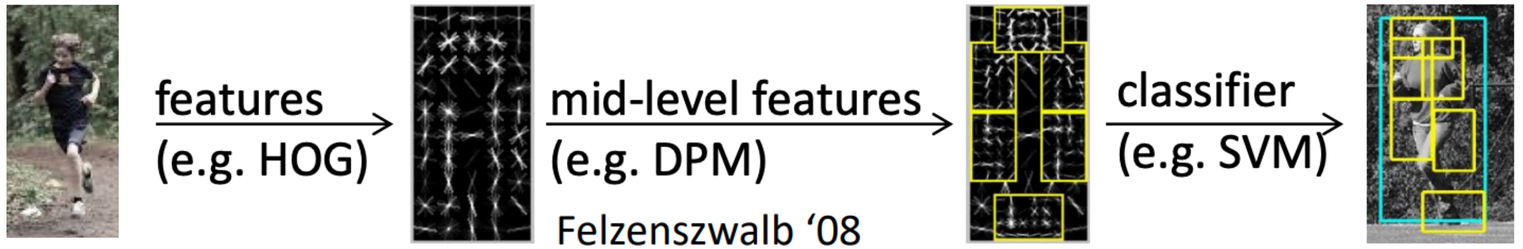
Why else study deep reinforcement learning?

1. Sequential decision-making problems are everywhere!

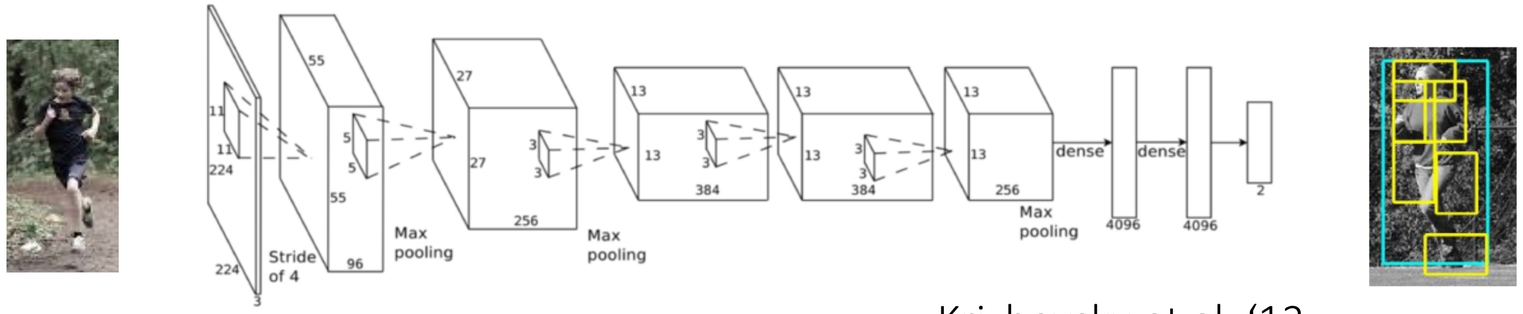
   a. Controlling robots & autonomous vehicles
   b. What if you want your AI system to interact with people?
   c. What if deploying your system affects future outcomes & observations? “feedback loops”
   d. What if your objective isn’t just accuracy? (and isn’t differentiable)

Why else study deep reinforcement learning?

Standard computer vision:
hand-designed features



Modern computer vision:
end-to-end training

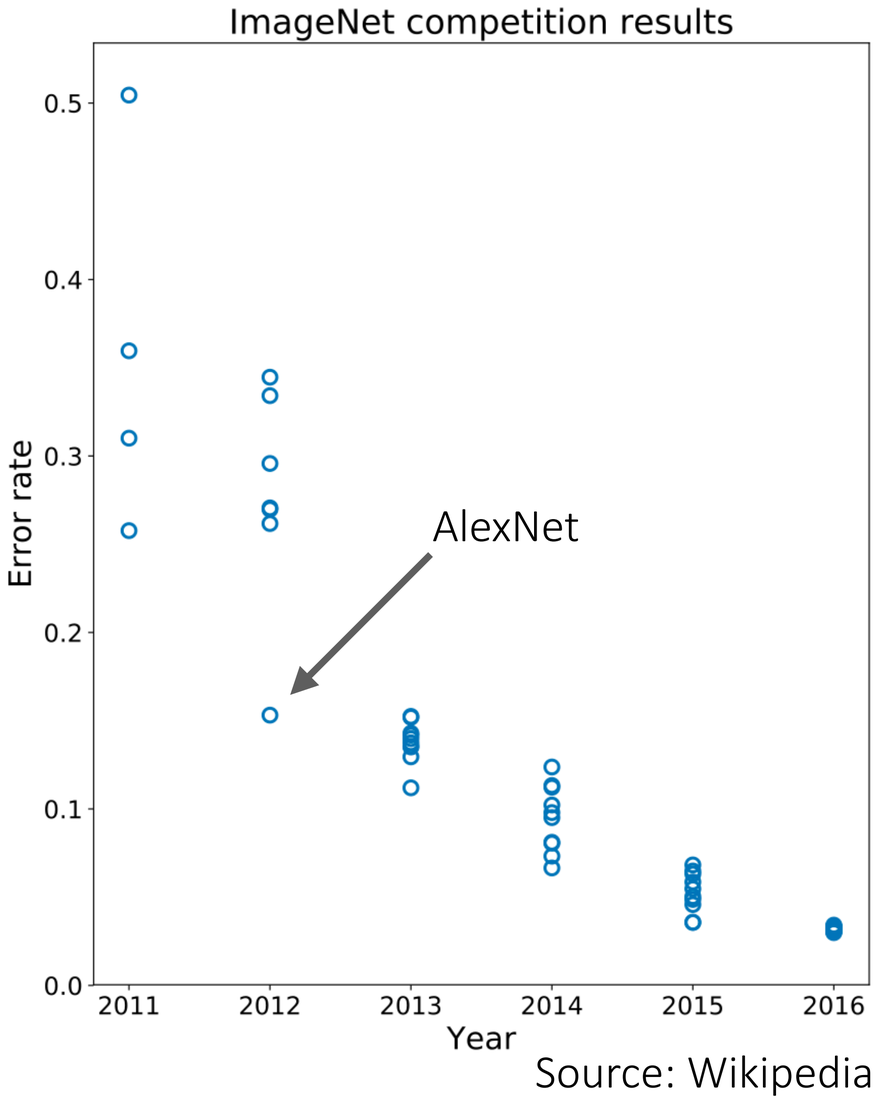


Krizhevsky et al. '12

Deep learning allows us to handle *unstructured inputs* (pixels, language, sensor readings, etc.) without hand-engineering features, with less domain knowledge

Slide adapted from Sergey Levine

Deep learning for object classification



Source: Wikipedia

Deep learning for machine translation

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Table 10: Mean of side-by-side scores on production data

	PBMT	GNMT	Human	Relative Improvement
English → Spanish	4.885	5.428	5.504	87%
English → French	4.932	5.295	5.496	64%
English → Chinese	4.035	4.594	4.987	58%
Spanish → English	4.872	5.187	5.372	63%
French → English	5.046	5.343	5.404	83%
Chinese → English	3.694	4.263	4.636	60%

Human evaluation scores on scale of 0 to 6

PBMT: Phrase-based machine translation

GNMT: Google's neural machine translation (in 2016)

Why study deep reinforcement learning *now*?

1992 PhD thesis by Long-Ji Lin (CMU)

This dissertation demonstrates how we can possibly overcome the slow learning problem and tackle non-Markovian environments, making reinforcement learning more practical for realistic robot tasks:

- Reinforcement learning can be naturally integrated with artificial neural networks to obtain high-quality generalization, resulting in a significant learning speedup. Neural networks are used in this dissertation, and they generalize effectively even in the presence of noise and a large number of binary and real-valued inputs.
- Reinforcement learning agents can save many learning trials by using an action model, which can be learned on-line. With a model, an agent can mentally experience the effects of its actions without actually executing them. Experience replay is a simple technique that implements this idea, and is shown to be effective in reducing the number of action executions required.

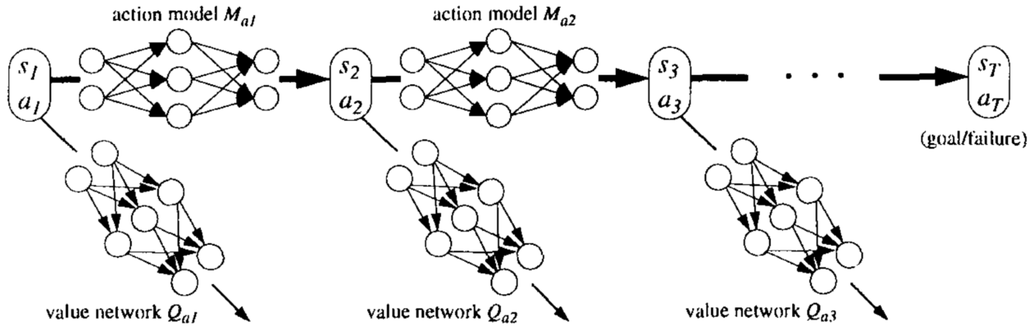
...

- Reinforcement learning agents can significantly reduce learning time by hierarchical learning— they first solve elementary learning problems and then combine solutions to the elementary problems to solve a complex problem. Simulation experiments indicate that a robot with hierarchical learning can solve a complex problem, which otherwise is hardly solvable within a reasonable time.

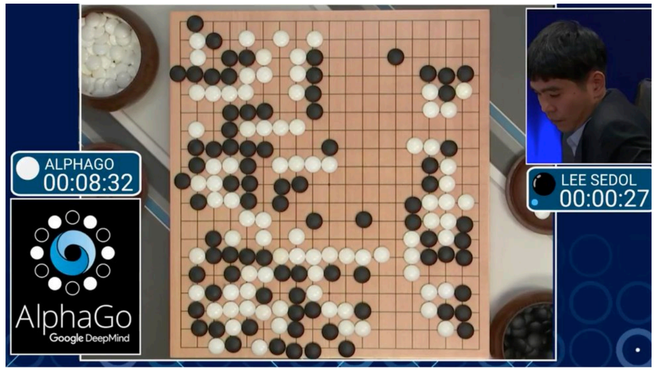
1995 paper by Sebastian Thrun

***Abstract*—Designing robots that learn by themselves to perform complex real-world tasks is a still-open challenge for the field of Robotics and Artificial Intelligence. In this paper we present the robot learning problem as a lifelong problem, in which a robot faces a collection of tasks over its entire lifetime. Such a scenario provides the oppor-**

Since in this paper we are interested in learning robot control, we will describe EBNN in the context of Q-Learning [28]. Q-



Why study deep reinforcement learning now?

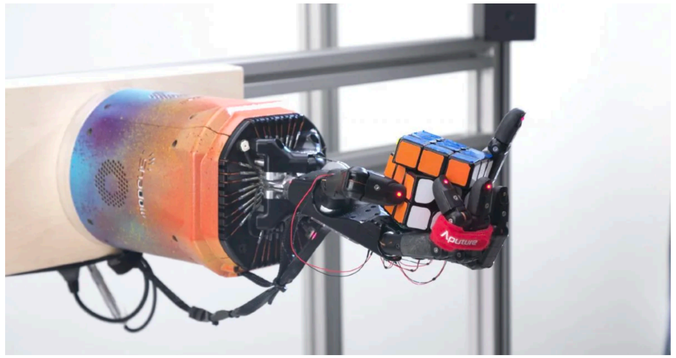


AlphaGo (’16)

ChatGPT (’22)

Stratospheric balloon navigation (’20)

Racing in Gran Turismo (’22)



Dexterous manipulation (’19)

But, we also still have many open questions and challenges!

Course Reminders

Your Initial Steps:

Homework 1 comes out Weds, due Weds 4/19 at 11:59 pm PT

Start forming final project groups if you want to work in a group

Coming Up Next:

Imitation Learning Lecture (Weds)

PyTorch Tutorial (Thurs)