Supervised Learning of Behaviors: Deep Learning, Dynamical Systems, and Behavior Cloning

CS 294-112: Deep Reinforcement Learning

Week 2, Lecture 1

Sergey Levine

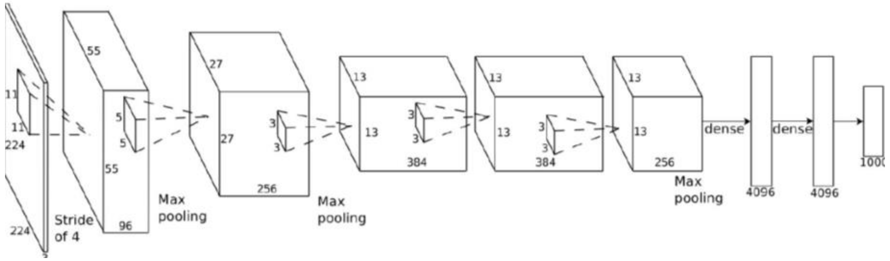
Today’s Lecture

1. Definition of sequential decision problems
2. Imitation learning: supervised learning for decision making
   a. Does direct imitation work?
   b. How can we make it work more often?
3. Case studies of recent work in (deep) imitation learning
4. What is missing from imitation learning?

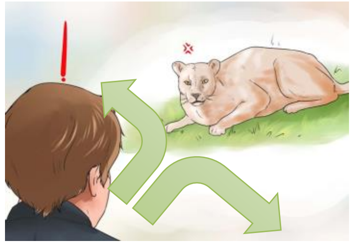
- Goals:
  - Understand definitions & notation
  - Understand basic imitation learning algorithms
  - Understand their strengths & weaknesses

Terminology & notation

$\mathbf{o}_t$



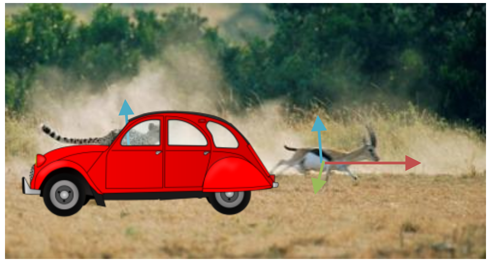
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$



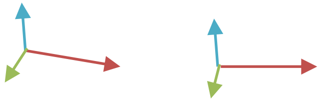
$\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

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

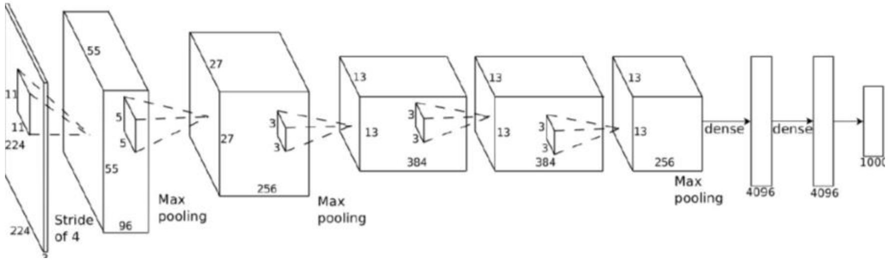


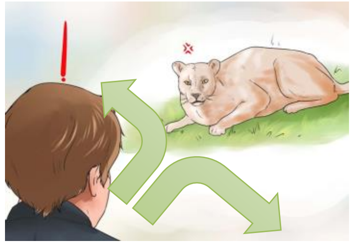
$\mathbf{o}_t$ – observation



$\mathbf{x}_t$ – state

Terminology & notation

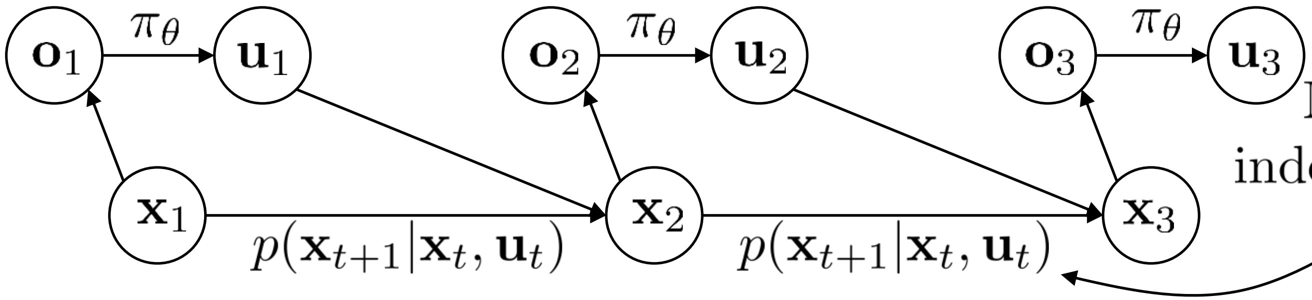




$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

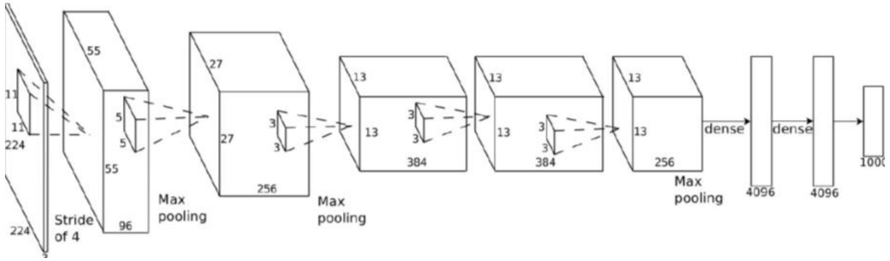
$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

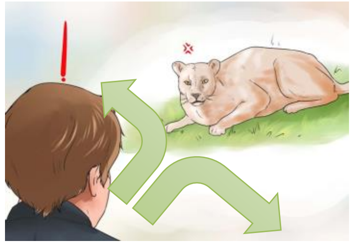
$\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ – policy



Markov property
independent of $\mathbf{x}_{t-1}$

Terminology & notation





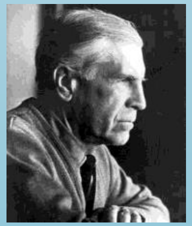
$\mathbf{o}_t$ $\qquad$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\qquad$ $\mathbf{u}_t$

$\mathbf{x}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{u}_t$ – action

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

a bit of history…

$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action
управление

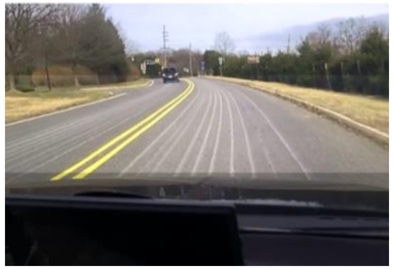


Lev Pontryagin

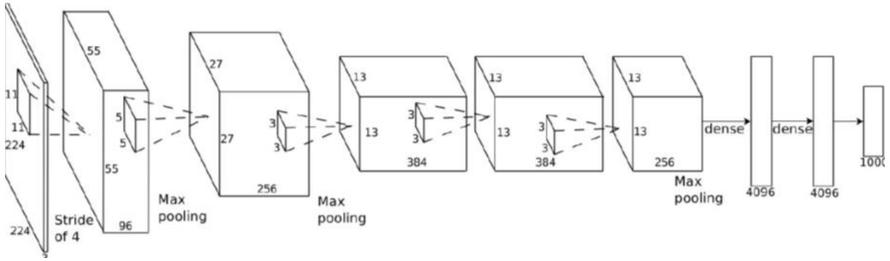
$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action

Richard Bellman

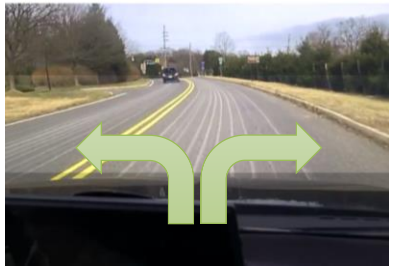
Imitation Learning



$\mathbf{o}_t$



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



$\mathbf{u}_t$

$\mathbf{o}_t$

$\mathbf{u}_t$

training data

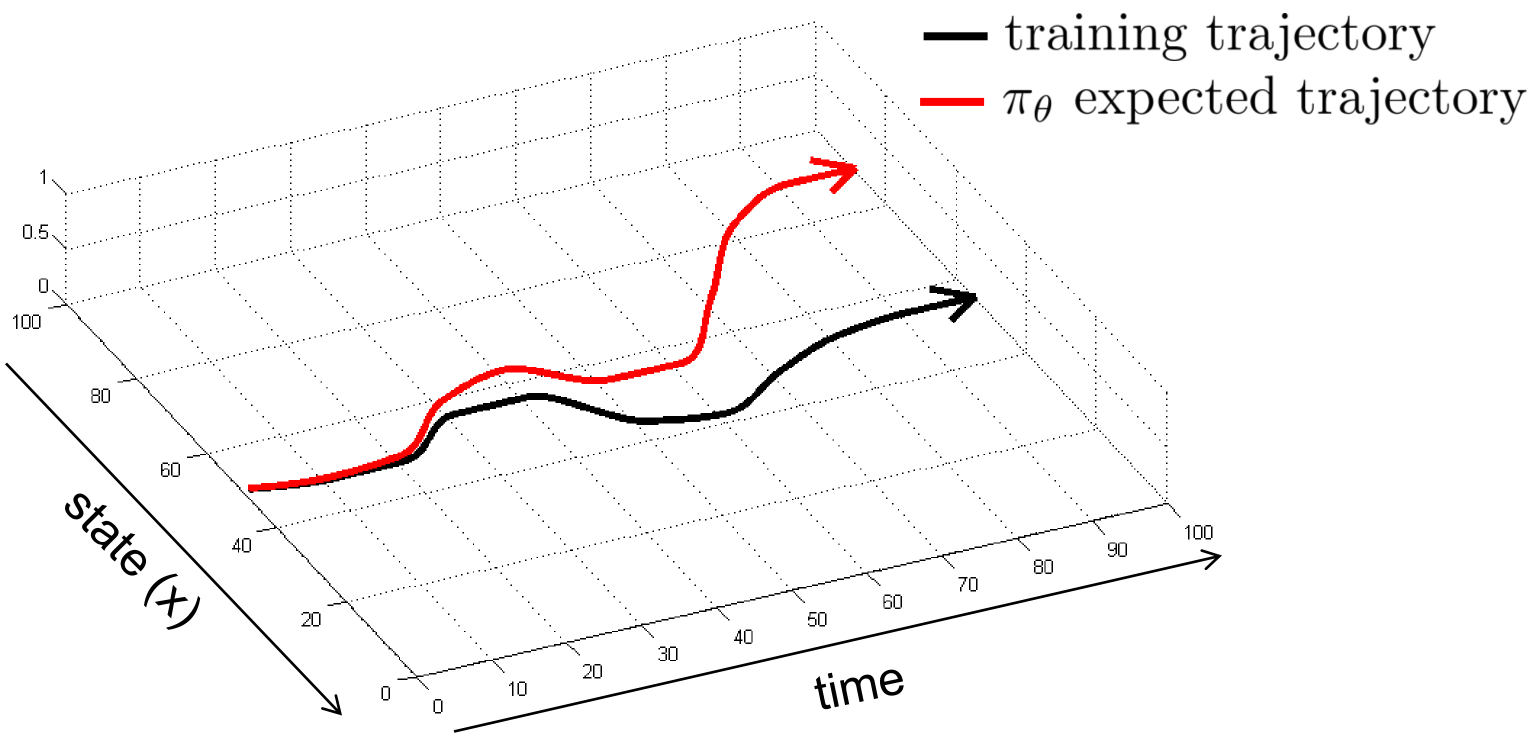
supervised learning

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

Images: Bojarski et al. '16, NVIDIA

Does it work?

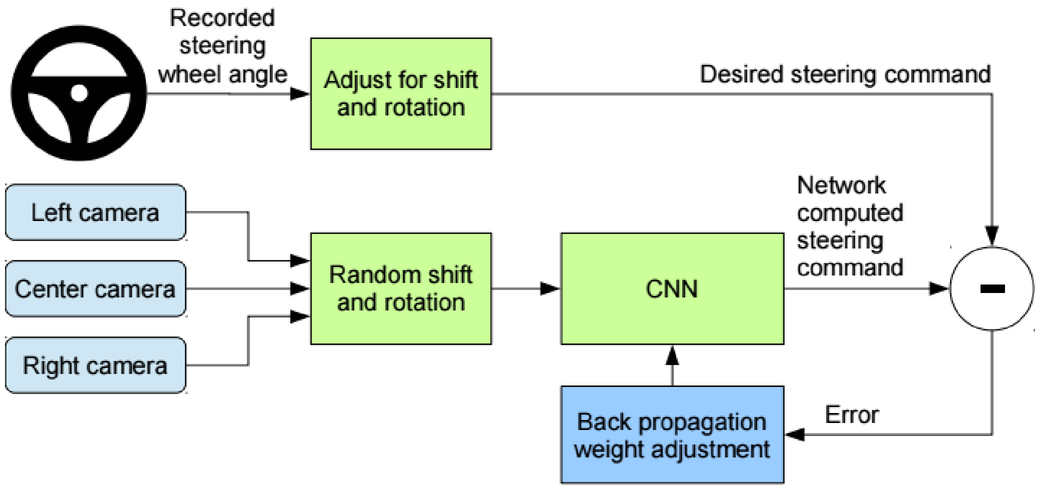
No!

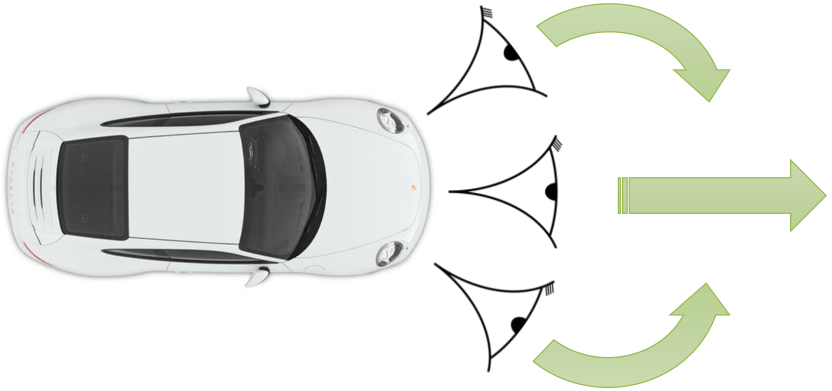


Does it work? Yes!

Video: Bojarski et al. ‘16, NVIDIA

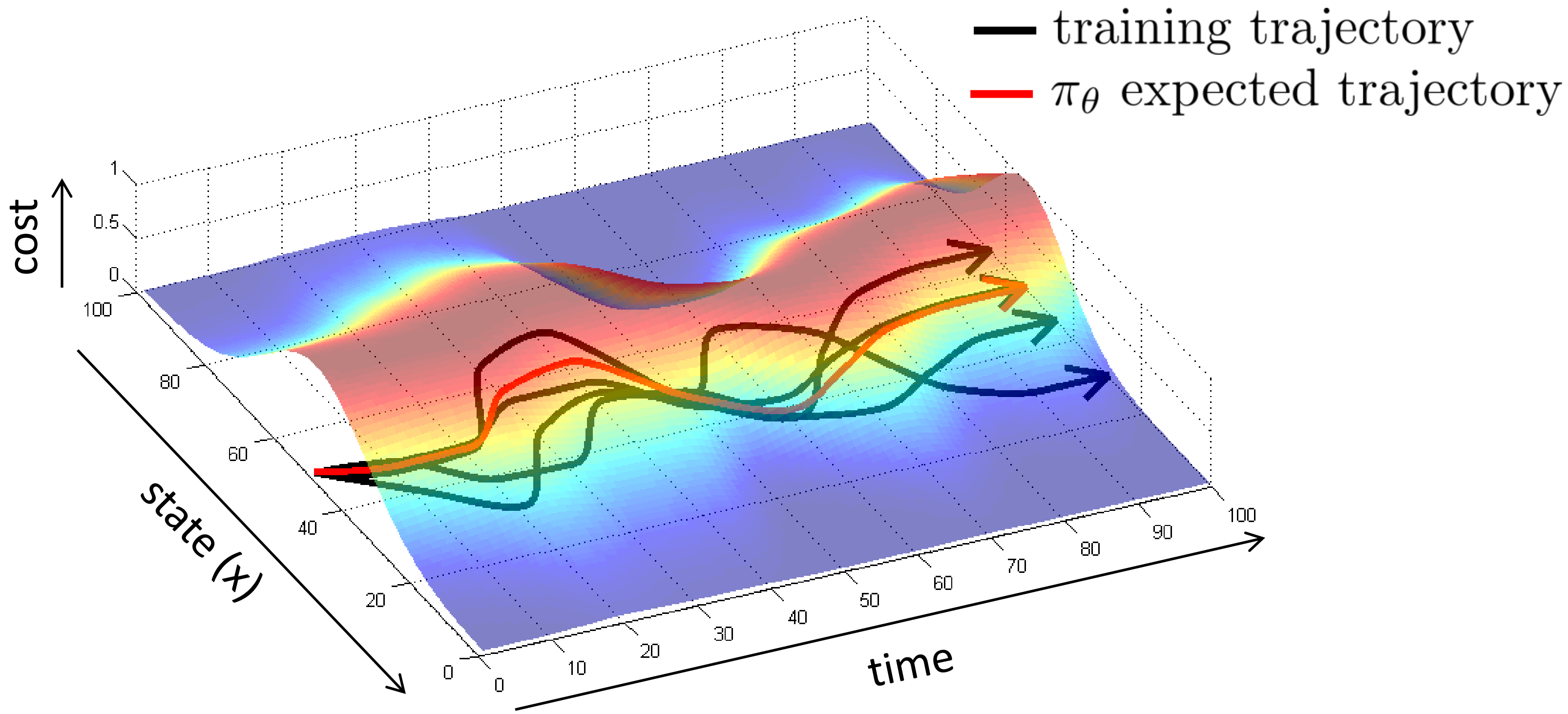
Why did that work?





Bojarski et al. ‘16, NVIDIA

Can we make it work more often?

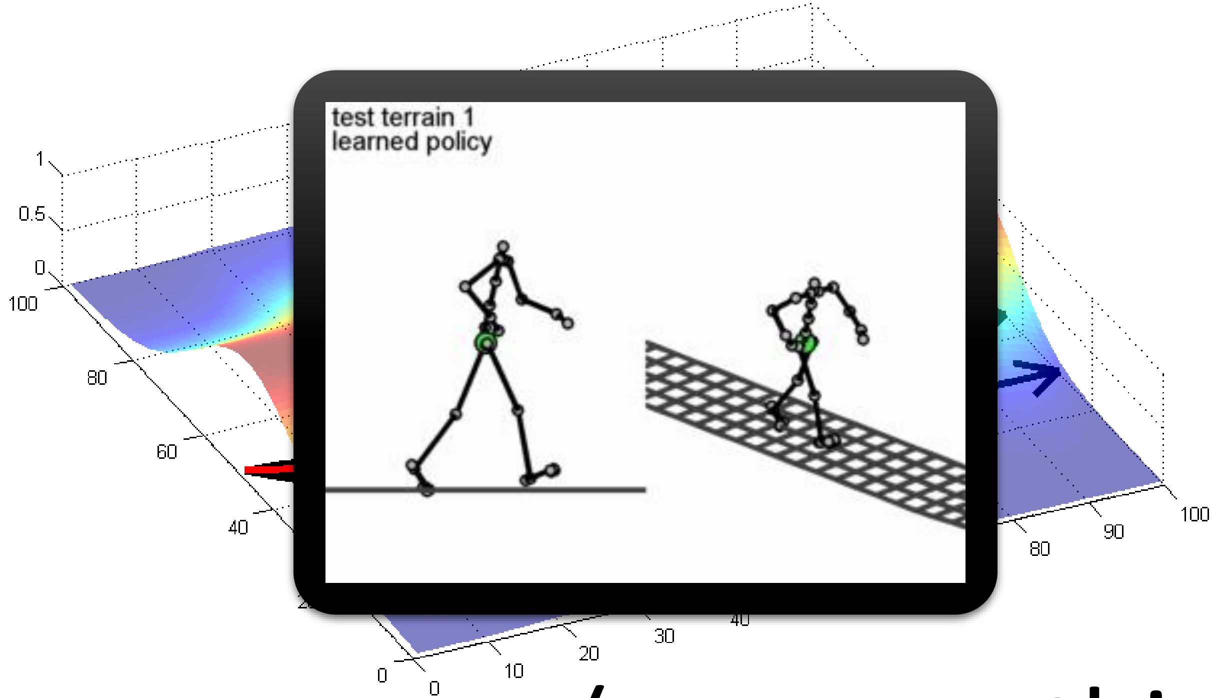


stability

Learning from a stabilizing controller

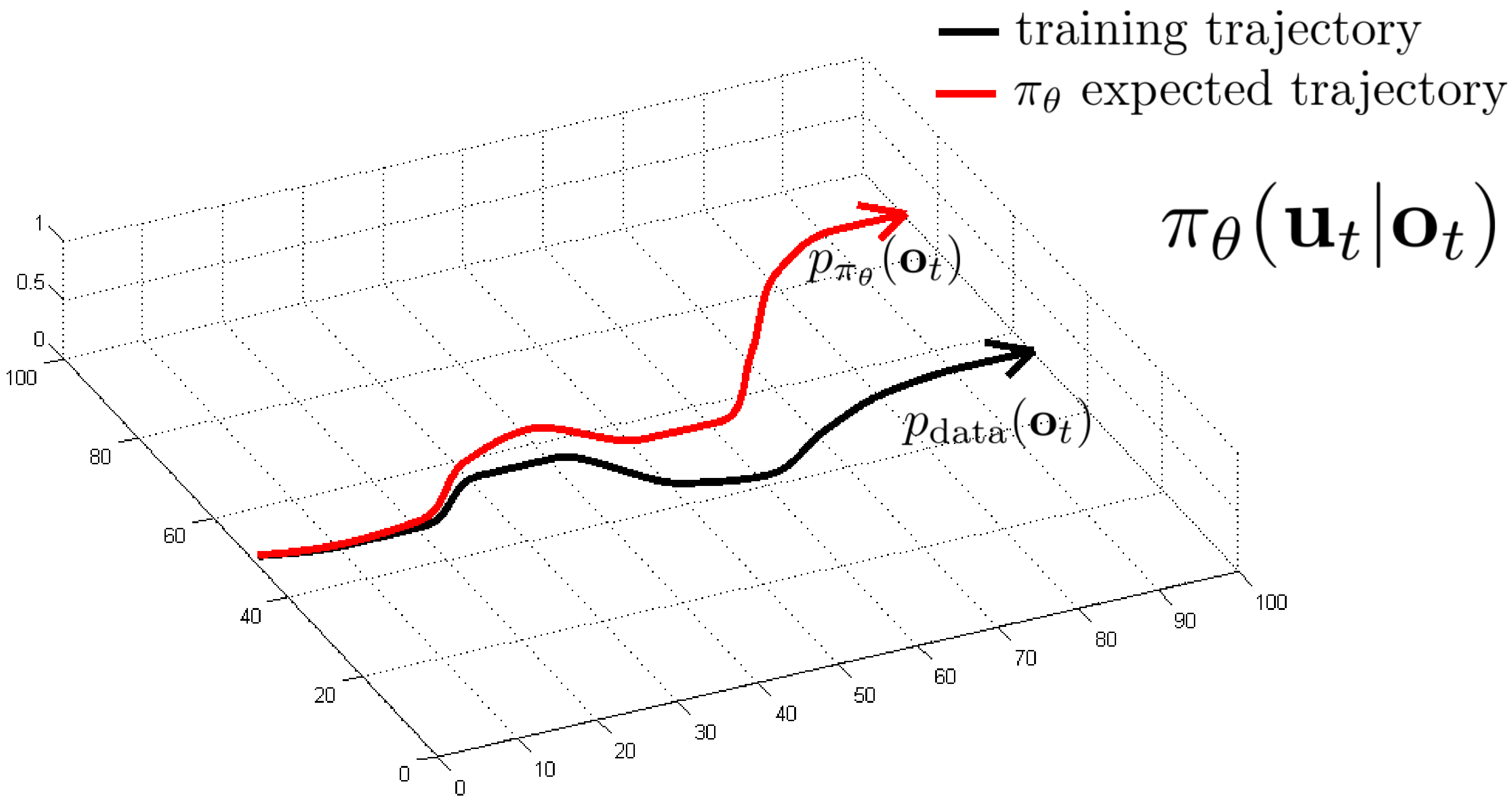
$$\underbrace{p(\mathbf{x}_1, \mathbf{u}_1, \mathbf{x}_2, \mathbf{u}_2, \ldots, \mathbf{x}_T, \mathbf{u}_T)}_{\tau}$$

Gaussian distribution obtained using variant of iterative LQR



(more on this later)

Can we make it work more often?



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

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

Can we make it work more often?

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

DAgger: **D**ataset **Agg**regation

goal: collect training data from $p_{\pi_\theta}(\mathbf{o}_t)$ instead of $p_{\text{data}}(\mathbf{o}_t)$

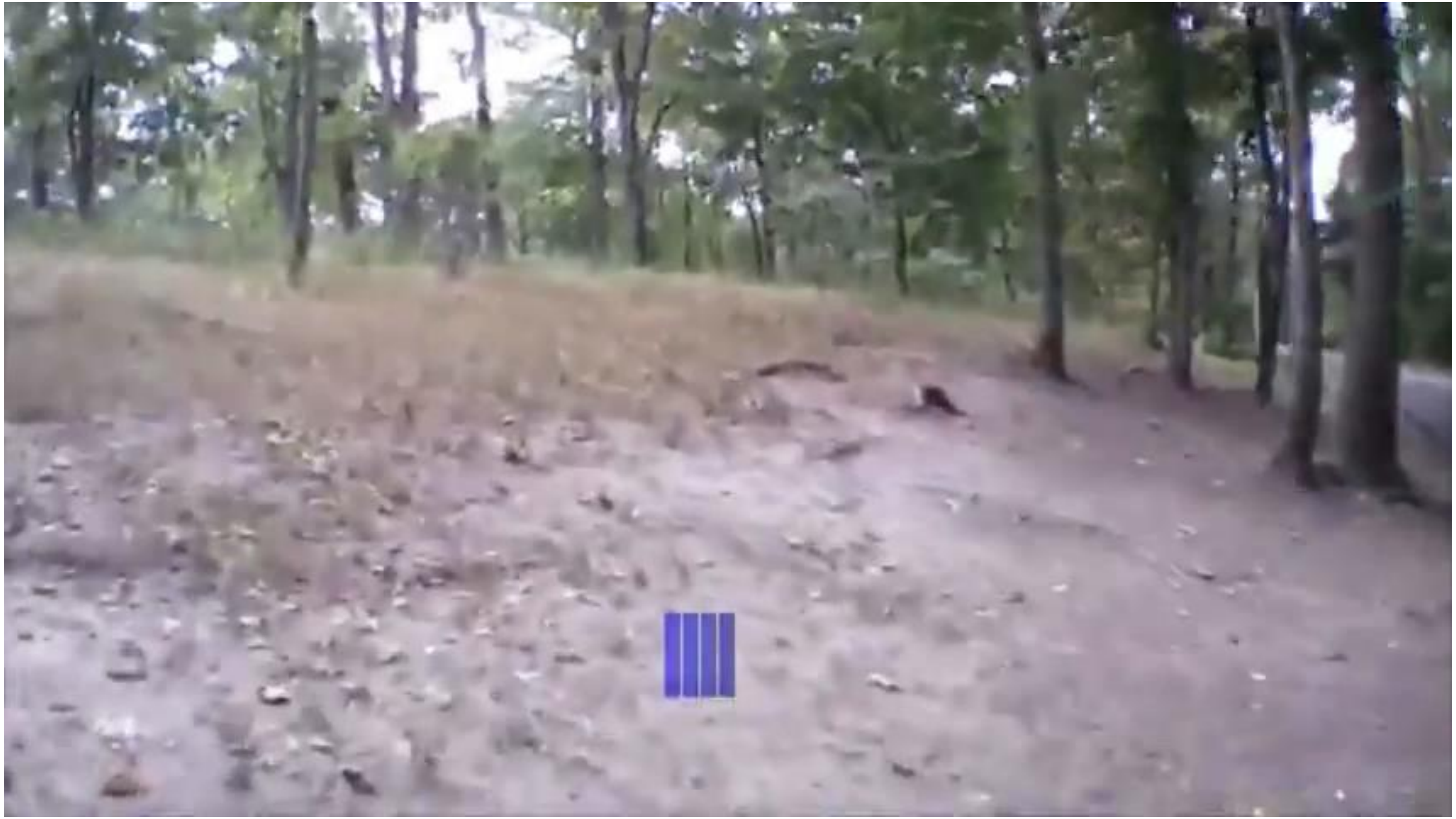
how? just run $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

but need labels $\mathbf{u}_t$!

1. train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{u}_1, \ldots, \mathbf{o}_N, \mathbf{u}_N\}$
2. run $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{u}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Ross et al. '11

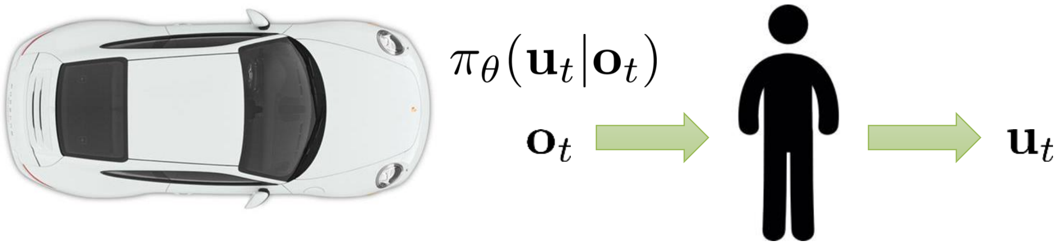
DAgger Example



Ross et al. '11

What’s the problem?

1. train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{u}_1, \dots, \mathbf{o}_N, \mathbf{u}_N\}$
2. run $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \dots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{u}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

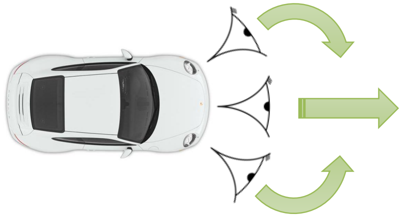


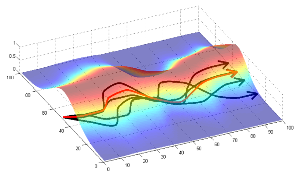
Ross et al. ‘11

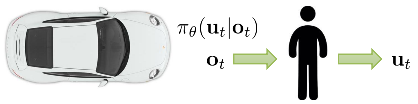
Imitation learning: recap

$\mathbf{o}_t$
$\mathbf{u}_t$ → training data → supervised learning → $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$

- Often (but not always) insufficient by itself
  - Distribution mismatch problem
- Sometimes works well
  - Hacks (e.g. left/right images)
  - Samples from a stable trajectory distribution
  - Add more **on-policy** data, e.g. using DAgger



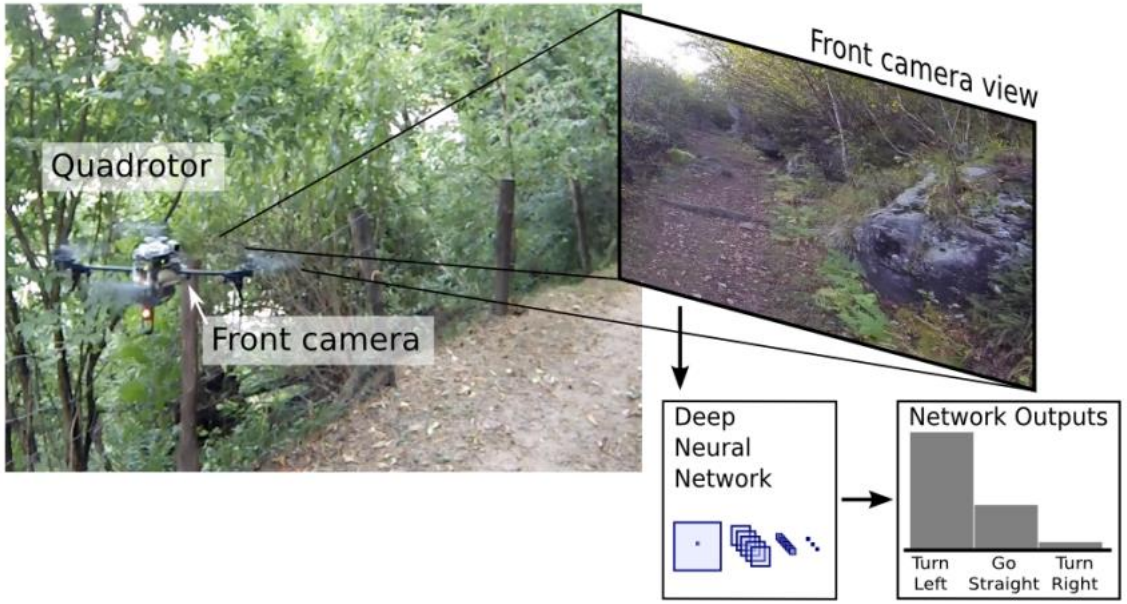




Case study 1: trail following as classification

A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots

Alessandro Giusti[1], Jérôme Guzzi[1], Dan C. Cireşan[1], Fang-Lin He[1], Juan P. Rodríguez[1]
Flavio Fontana[2], Matthias Faessler[2], Christian Forster[2]
Jürgen Schmidhuber[1], Gianni Di Caro[1], Davide Scaramuzza[2], Luca M. Gambardella[1]

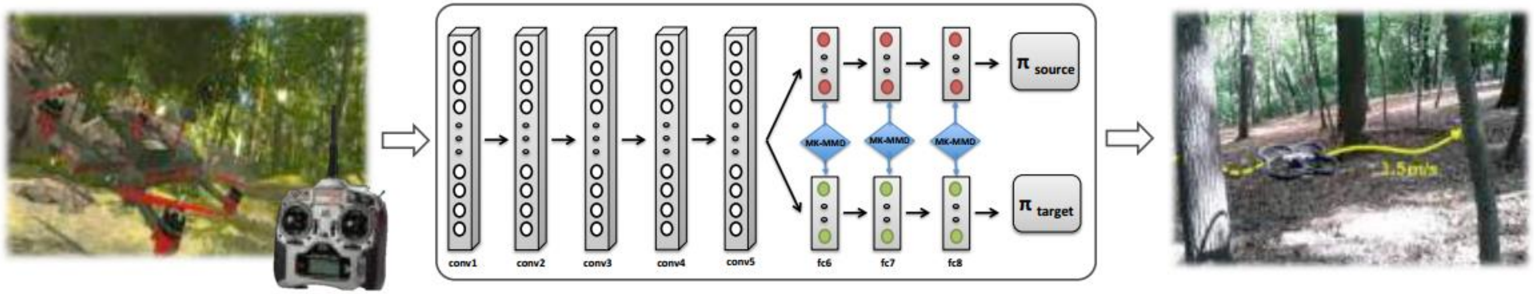


Case study 2: DAgger & domain adaptation

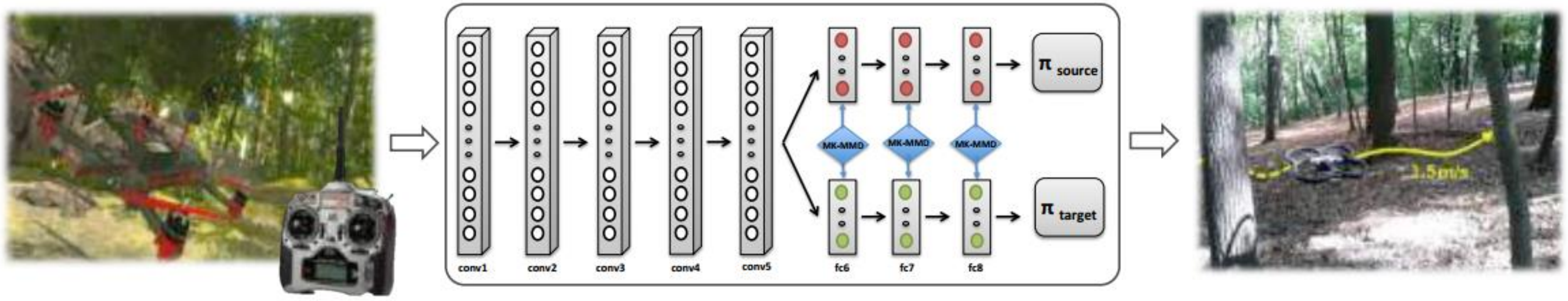
Learning Transferable Policies for Monocular Reactive MAV Control

Shreyansh Daftry, J. Andrew Bagnell, and Martial Hebert

Robotics Institute, Carnegie Mellon University, Pittsburgh, USA
{daftry,dbagnell,hebert}@ri.cmu.edu



1. train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{u}_1, \ldots, \mathbf{o}_N, \mathbf{u}_N\}$
2. run $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{u}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$



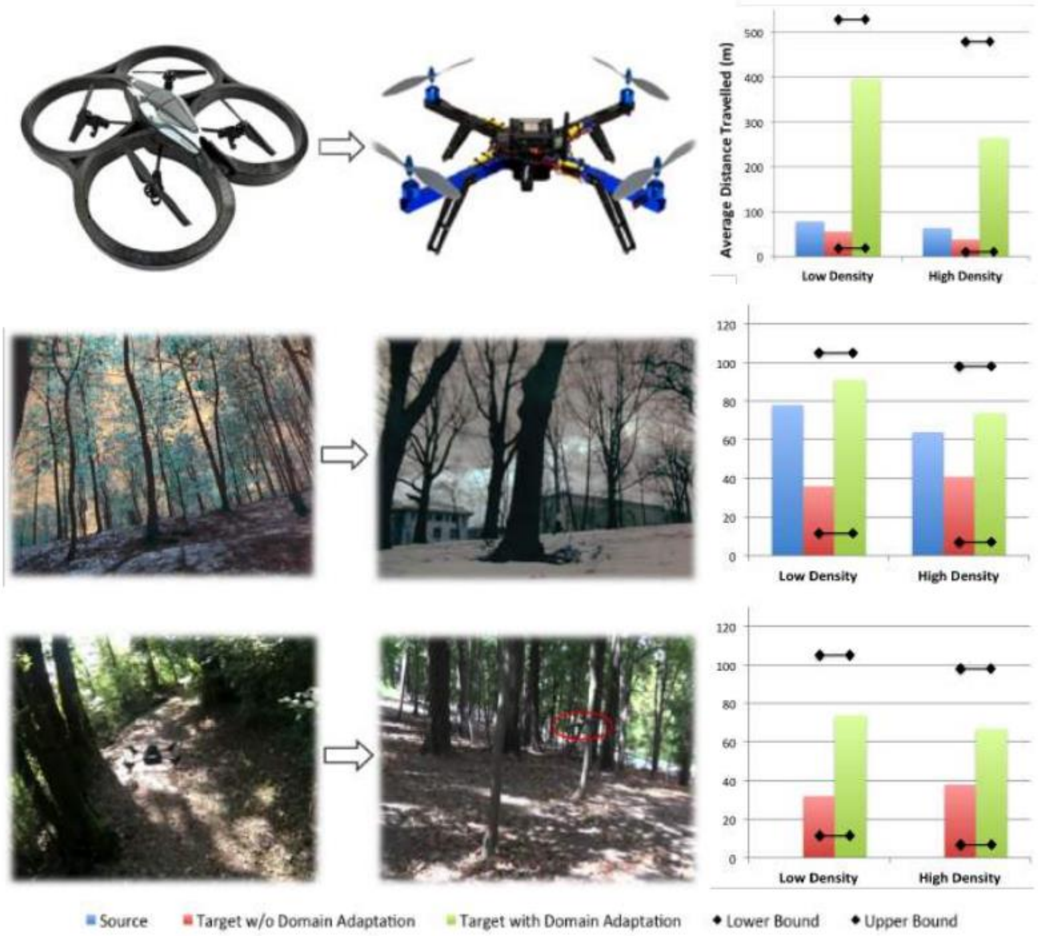
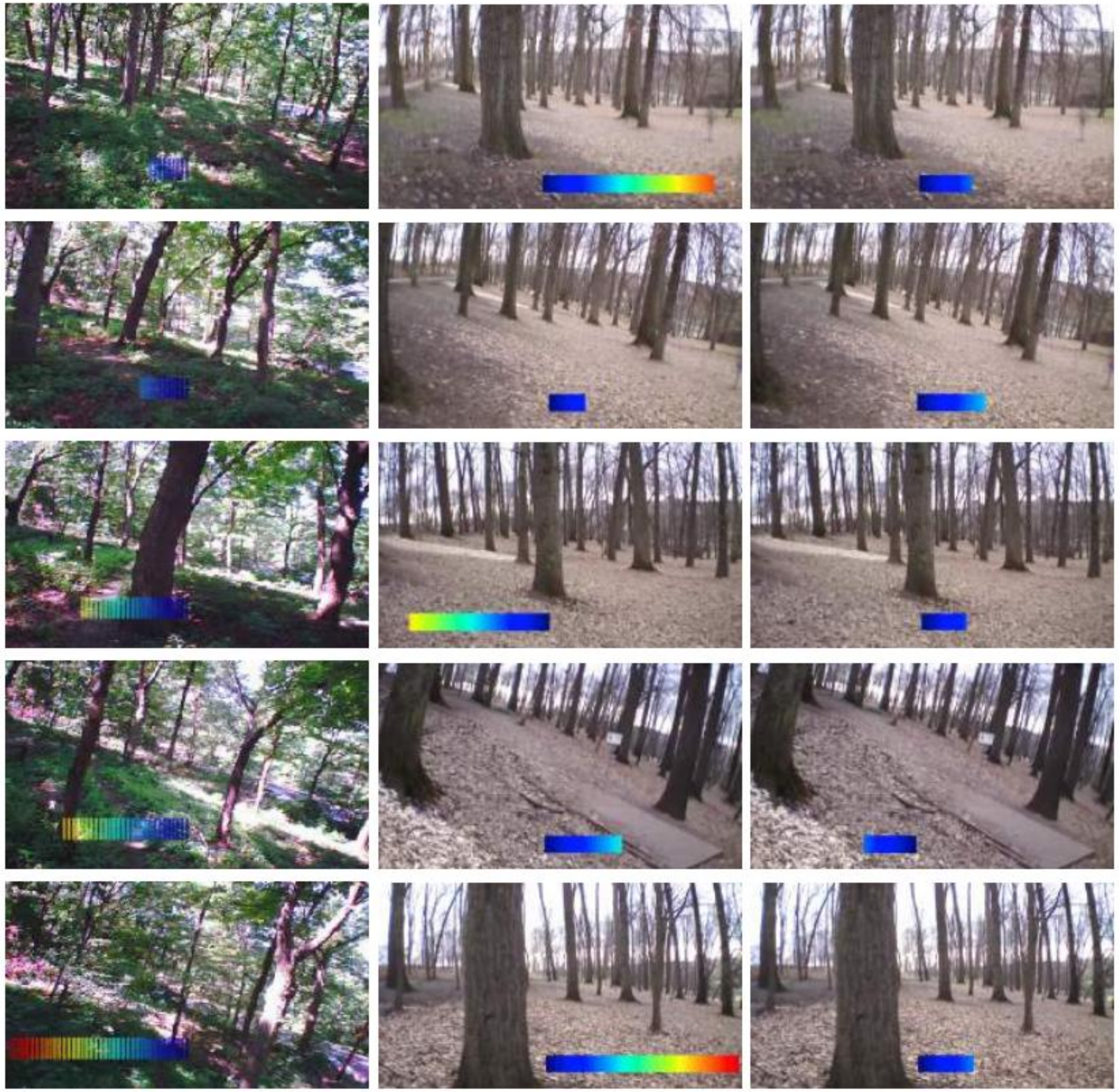


Fig. 2. Experiments and Results for (Row-1) Transfer across physical systems from ARDrone to ArduCopter, (Row-2) Transfer across weather conditions from summer to winter and (Row-3) Transfer across environments from Univ. of Zurich to CMU.

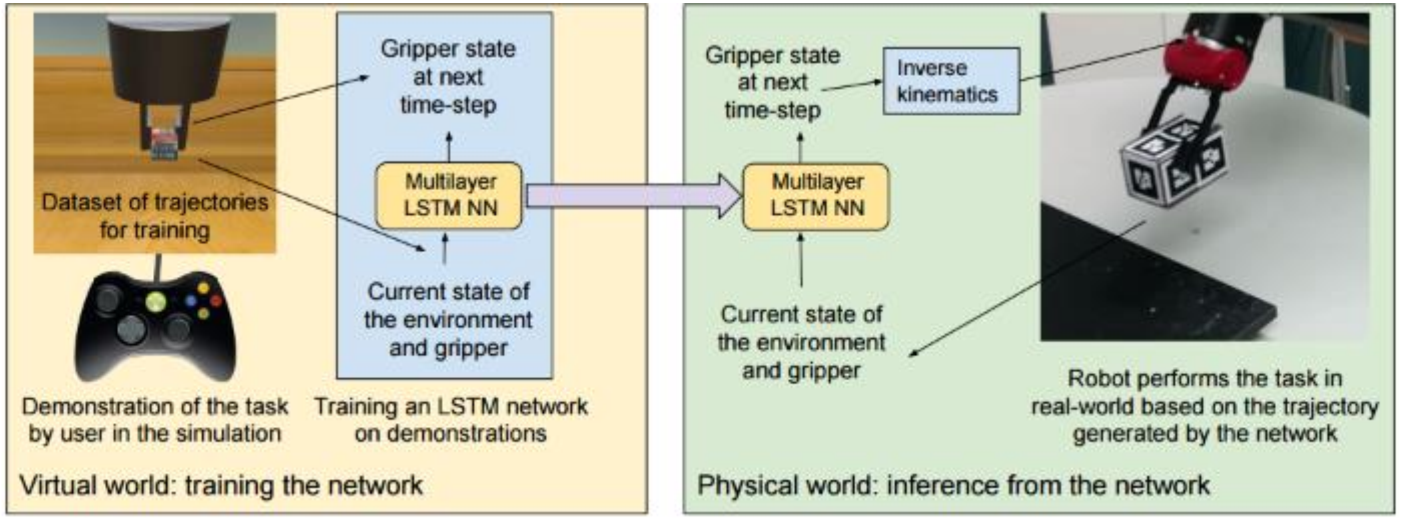


(a) (b) (c)

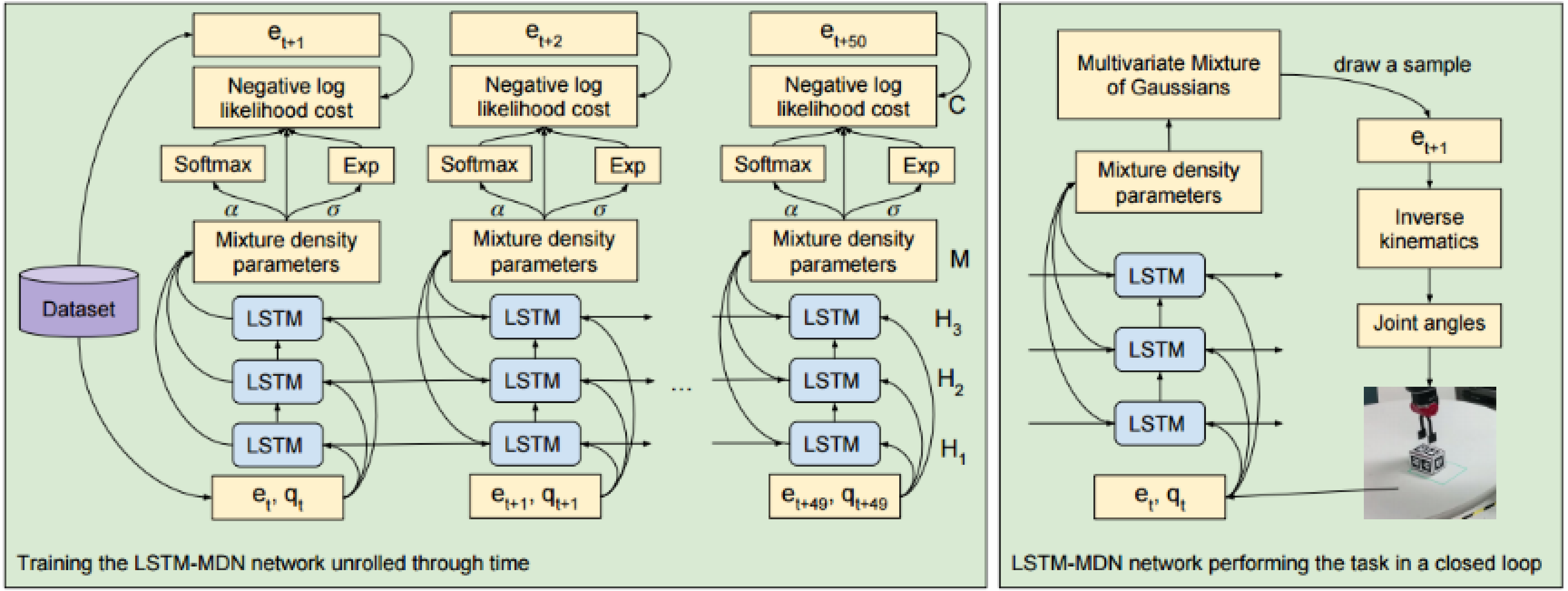
Case study 3: Imitation with LSTMs

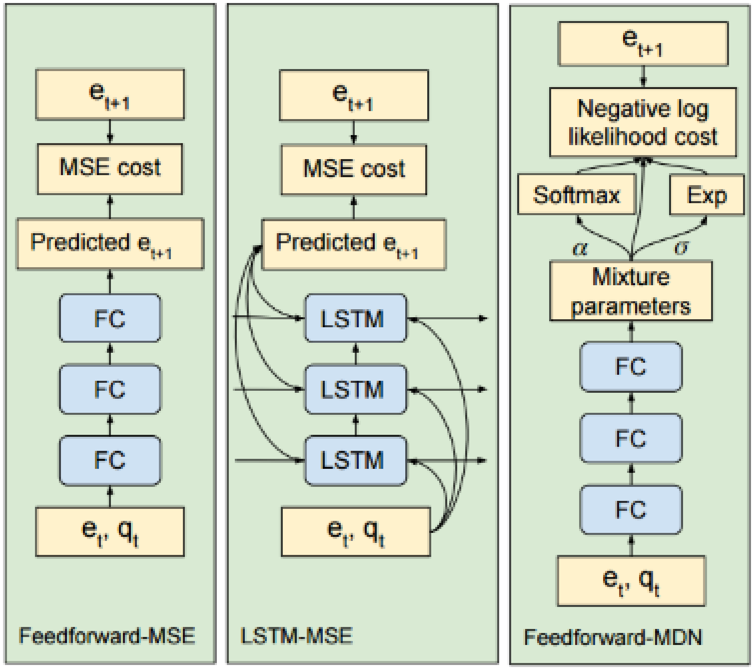
Learning real manipulation tasks from virtual demonstrations using LSTM

Rouhollah Rahmatizadeh[1], Pooya Abolghasemi[1], Aman Behal[2] and Ladislau Bölöni[1]



Learning Manipulation Trajectories Using Recurrent Neural Networks





Controller	Pick and place	Push to pose
Feedfoward-MSE	0%	0%
LSTM-MSE	85%	0%
Feedforward-MDN	95%	15%
LSTM-MDN	**100%**	**95%**

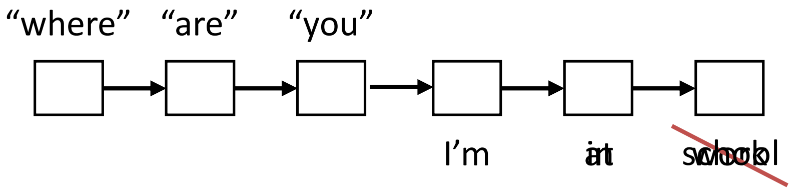
Environment	Pick and place	Push to pose
Virtual world	100%	95%
Physical world	80%	60%

Other topics in imitation learning

- Structured prediction

  x: where are you

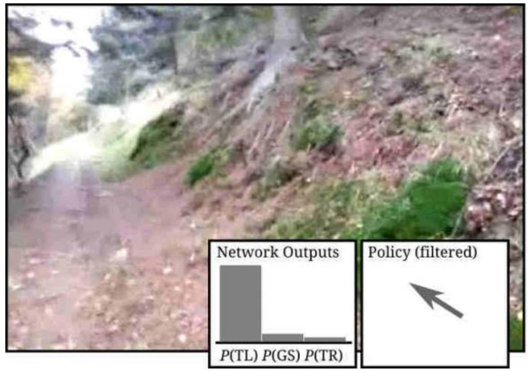
  y: I’m at work

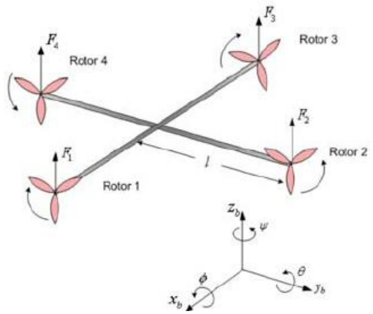


  - See Mohammad Norouzi’s lecture in April!
- Interaction & active learning
- Inverse reinforcement learning
  - Instead of copying the demonstration, figure out the *goal*
  - Will be covered later in this course

Imitation learning: what's the problem?

- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful
- Humans are not good at providing some kinds of actions





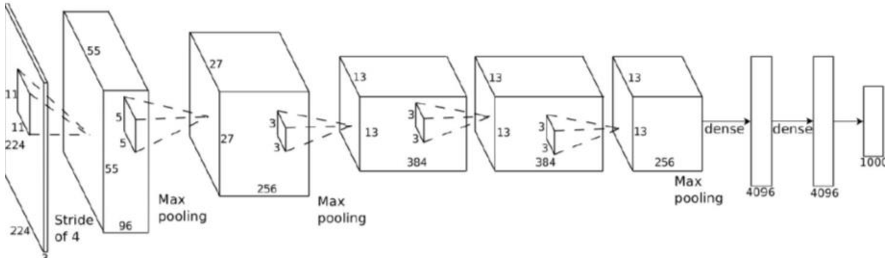
- Humans can learn autonomously; can our machines do the same?
  - Unlimited data from own experience
  - Continuous self-improvement

Next time: learning without humans

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

$\mathbf{o}_t \rightarrow$  $\rightarrow \mathbf{u}_t$

Terminology & notation



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

$\mathbf{x}_t$ – state

$\mathbf{o}_t$ – observation

$\mathbf{u}_t$ – action

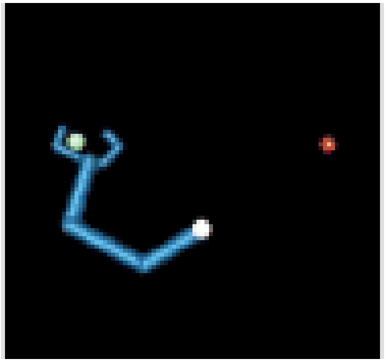
$c(\mathbf{x}_t, \mathbf{u}_t)$ – cost function

$r(\mathbf{x}_t, \mathbf{u}_t)$ – reward function

$$\min_{\mathbf{u}_1,\dots,\mathbf{u}_T} \sum_{t=1}^{T} c(\mathbf{x}_t, \mathbf{u}_t) \text{ s.t. } \mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$$

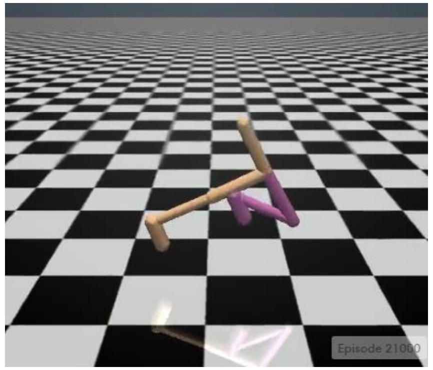
$$\min_{\mathbf{u}_1,\dots,\mathbf{u}_T} \log p(\text{eaten by tiger}|\mathbf{u}_1,\dots,\mathbf{u}_T)$$

Cost/reward functions in theory and practice



$$r(\mathbf{x},\mathbf{u}) = \begin{cases} 1 \text{ if object at target} \\ 0 \text{ otherwise} \end{cases}$$

$$\begin{aligned} r(\mathbf{x},\mathbf{u}) = & -w_1 \|p_{\text{gripper}}(\mathbf{x}) - p_{\text{object}}(\mathbf{x})\|^2 + \\ & -w_2 \|p_{\text{object}}(\mathbf{x}) - p_{\text{target}}(\mathbf{x})\|^2 + \\ & -w_3 \|\mathbf{u}\|^2 \end{aligned}$$



$$r(\mathbf{x},\mathbf{u}) = \begin{cases} 1 \text{ if walker is running} \\ 0 \text{ otherwise} \end{cases}$$

$$\begin{aligned} r(\mathbf{x},\mathbf{u}) = & w_1 v(\mathbf{x}) + \\ & w_2 \delta(|\theta_{\text{torso}}(\mathbf{x})| < \epsilon) + \\ & w_3 \delta(h_{\text{torso}}(\mathbf{x}) \geq h) \end{aligned}$$

A cost function for imitation?

$\mathbf{o}_t$

$\mathbf{u}_t$

training data

supervised learning

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

$$c(\mathbf{x},\mathbf{u}) = -\log p(\mathbf{u} = \pi^\star(\mathbf{x})|\mathbf{x})$$

1. train $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{u}_1, \ldots, \mathbf{o}_N, \mathbf{u}_N\}$
2. run $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{u}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Ross et al. '11

The trouble with cost & reward functions

reward

Mnih et al. '15

reinforcement learning agent

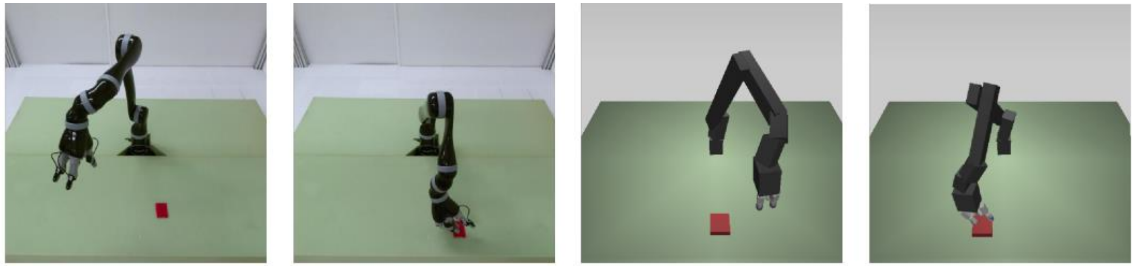
what is the reward?

Sim-to-Real Robot Learning from Pixels with Progressive Nets

Andrei A. Rusu, Matej Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, Raia Hadsell

Google DeepMind
London, UK

{andreirusu, matejvecerik, tcr, heess, razp, raia}@google.com



Rewards are given automatically by tracking the colored target

More on this later…

A note about terminology…

the “R” word

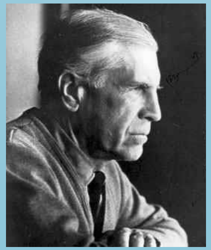
a bit of history…

reinforcement learning
(the **problem** statement)

$$\min \sum_{t=1}^{T} E[c(\mathbf{x}_t, \mathbf{u}_t)] \qquad \mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$$

reinforcement learning
(the **method**)

without using the **model** $\mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$



Lev Pontryagin

Richard Bellman

Andrew Barto

Richard Sutton