Supervised Learning of Behaviors

CS 294-112: Deep Reinforcement Learning

Sergey Levine

Class Notes

1. Make sure you sign up for Piazza!
2. Homework 1 is now out
   - Milestone due soon – good way to check your TensorFlow knowledge
3. Remember to start forming final project groups
4. Waitlist

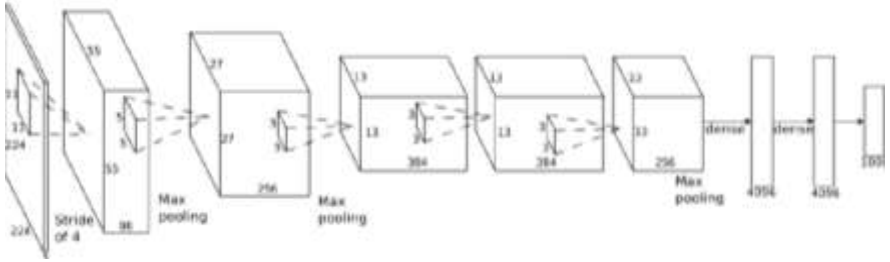
Today’s Lecture

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

$\mathbf{o}_t$

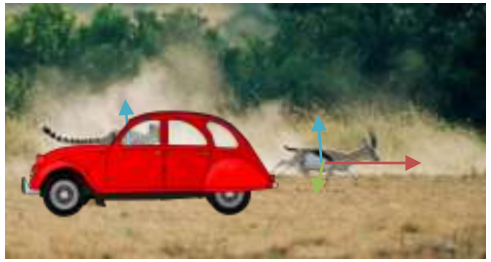


$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

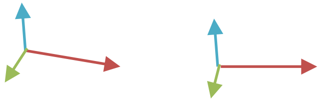
$\mathbf{a}_t$

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

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)



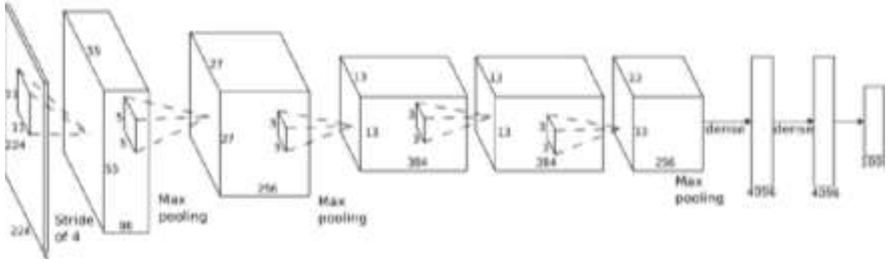
$\mathbf{o}_t$ – observation



$\mathbf{s}_t$ – state

Terminology & notation

$\mathbf{o}_t$

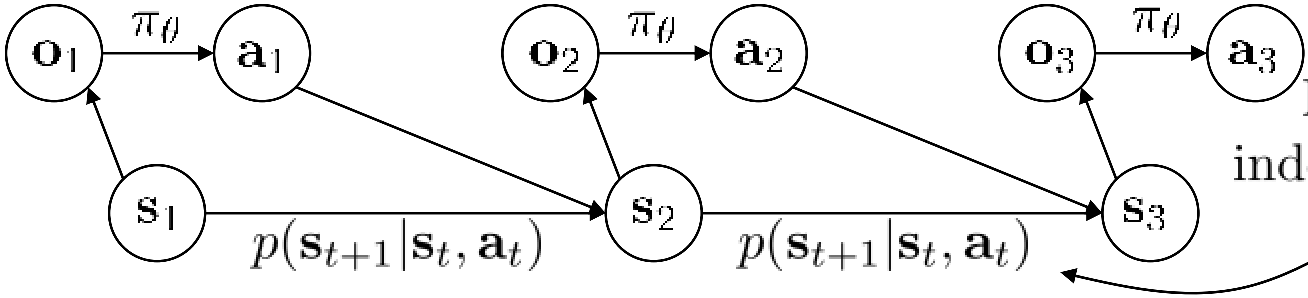


$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

$\mathbf{a}_t$

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

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)



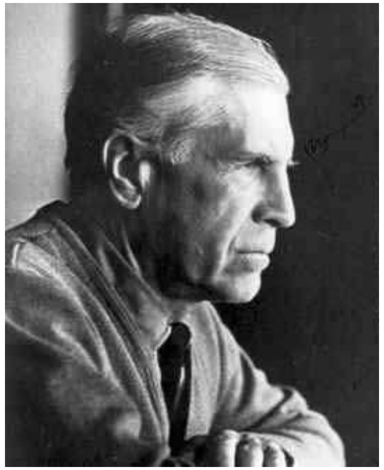
Markov property
independent of $\mathbf{s}_{t-1}$

Aside: notation

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

Richard Bellman

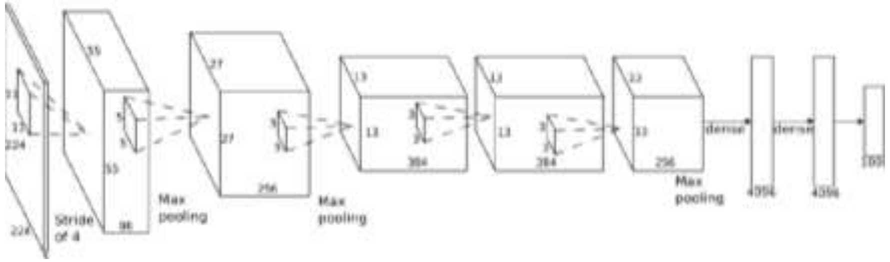
$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action   управление



Lev Pontryagin

Imitation Learning

$\mathbf{o}_t$



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

$\mathbf{a}_t$

$\mathbf{o}_t$

$\mathbf{a}_t$

training data

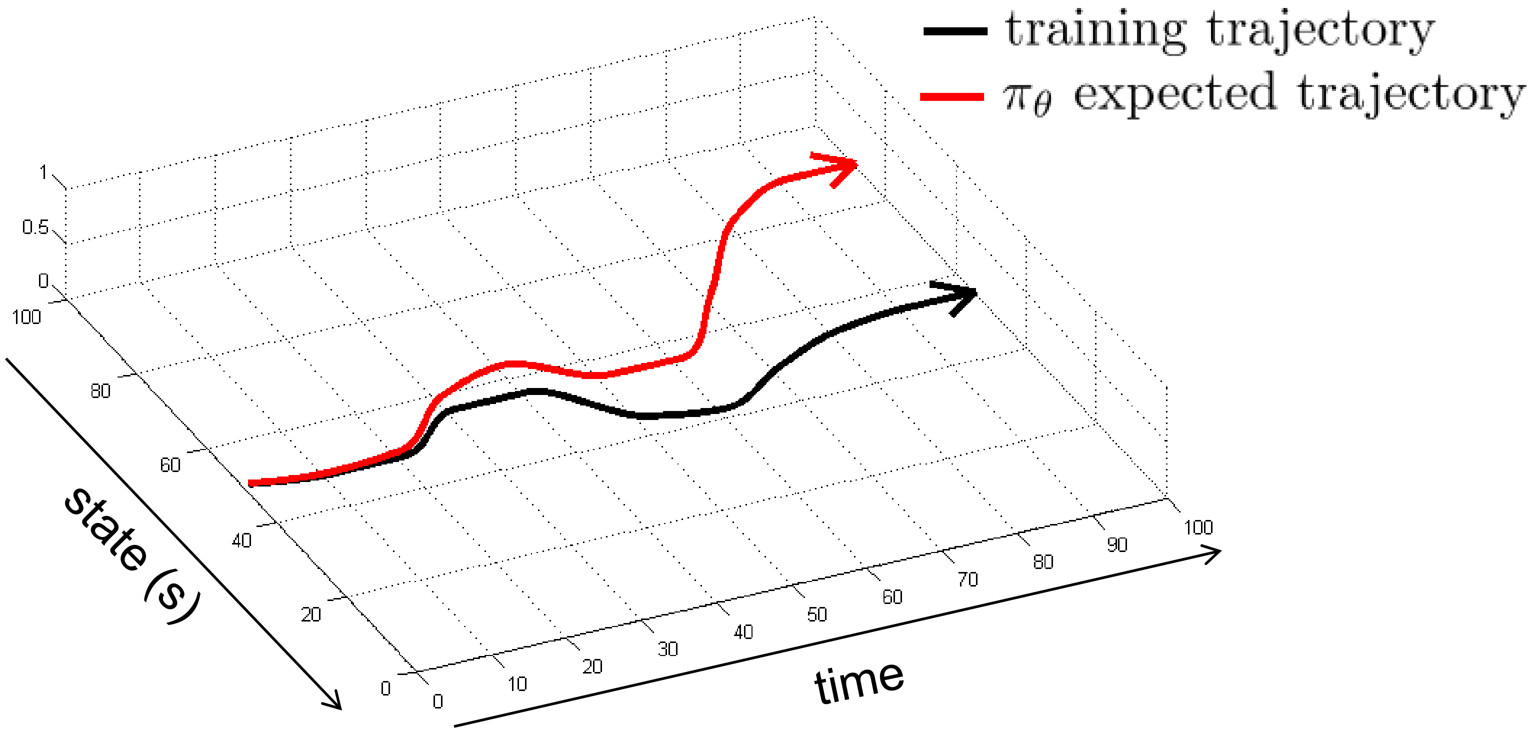
supervised learning

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

Images: Bojarski et al. '16, NVIDIA

Does it work?

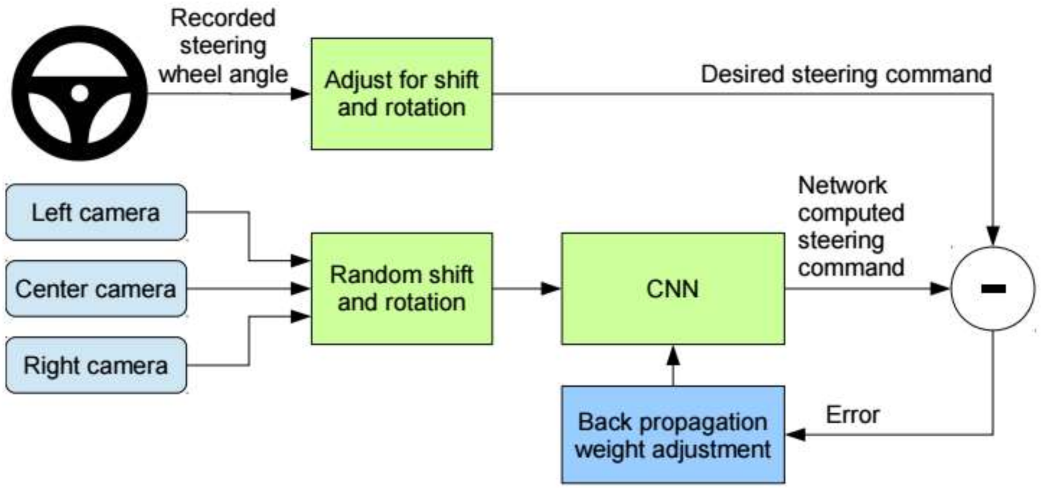
No!

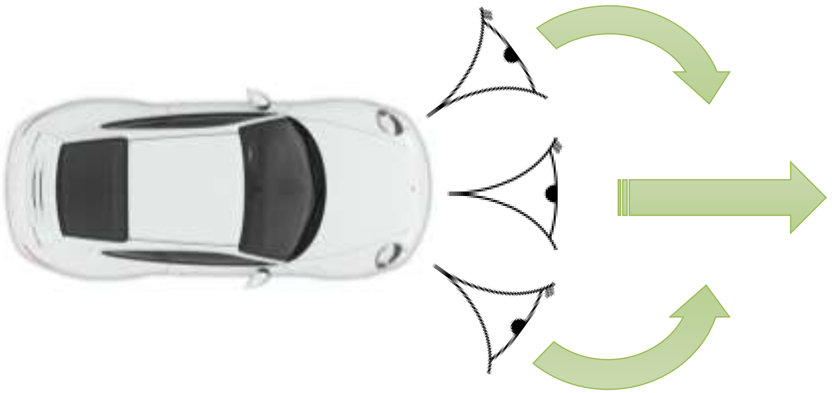


Does it work? Yes!

Video: Bojarski et al. ‘16, NVIDIA

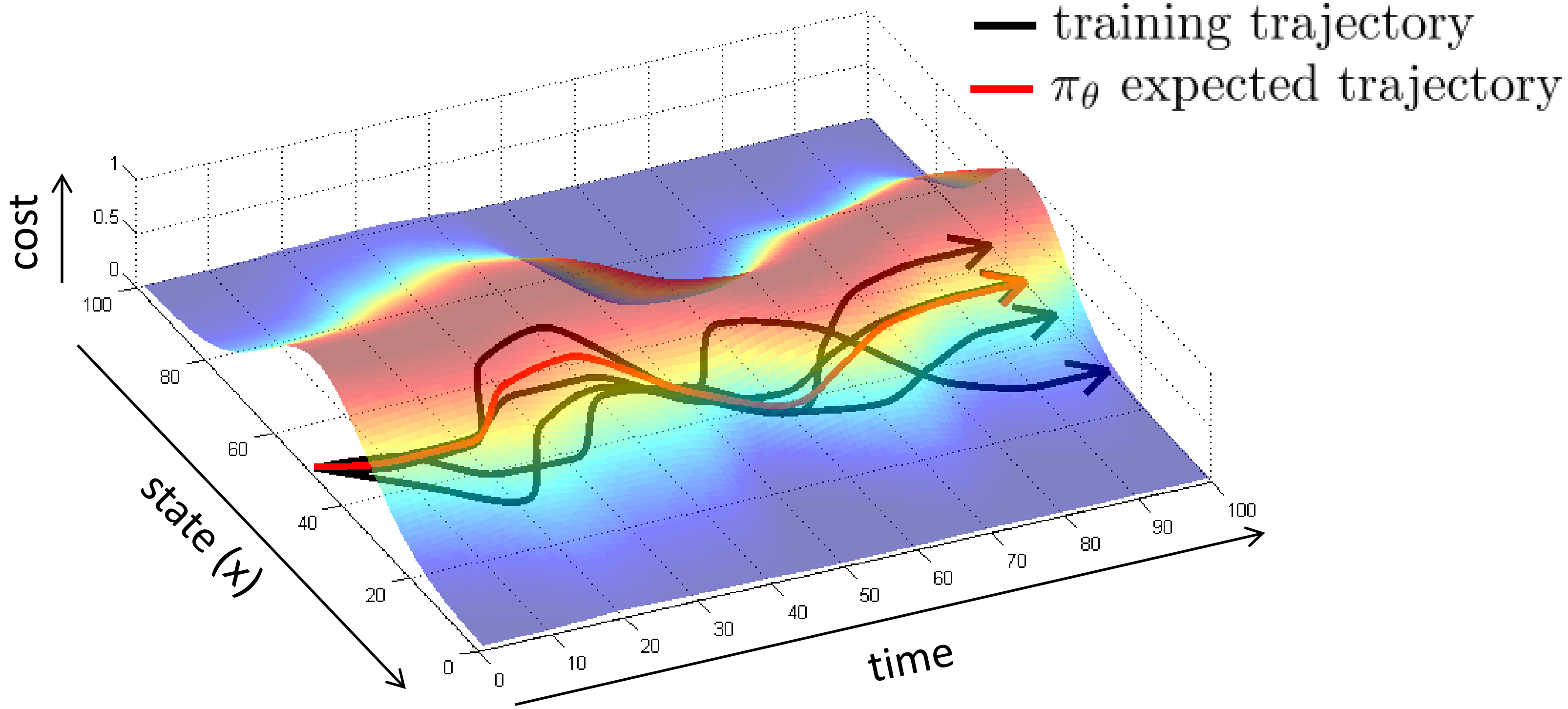
Why did that work?





Bojarski et al. ‘16, NVIDIA

Can we make it work more often?

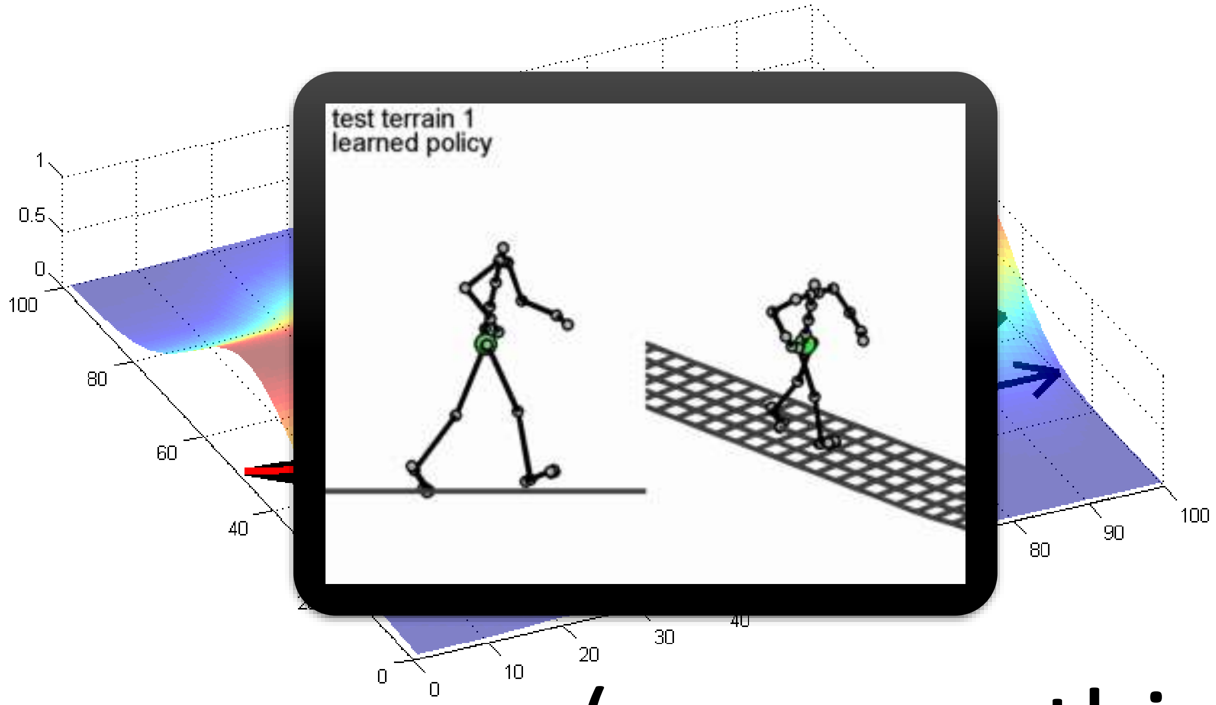


stability

Learning from a stabilizing controller

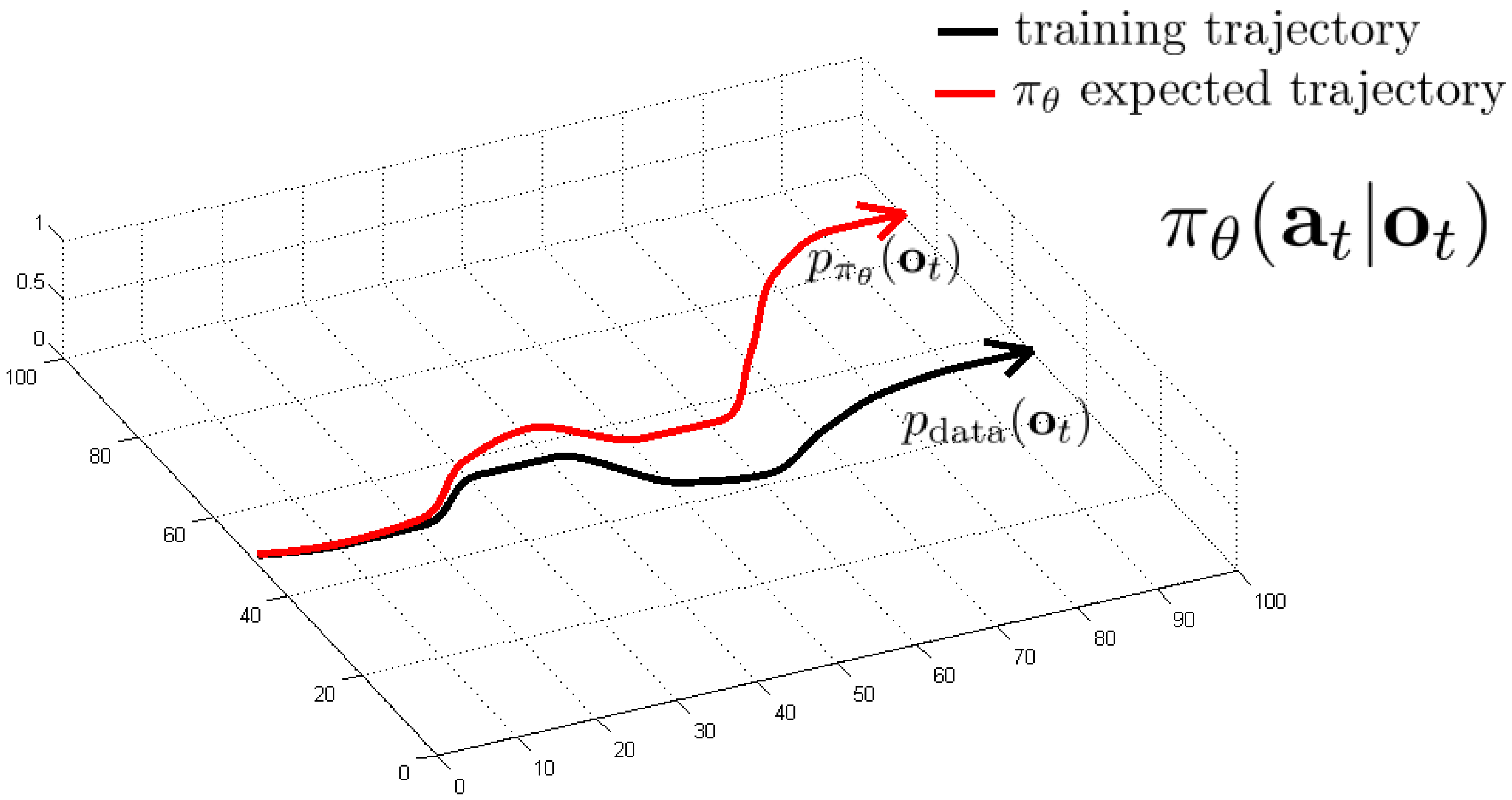
$\underbrace{p(\mathbf{s}_1, \mathbf{a}_1, \mathbf{s}_2, \mathbf{a}_2, \ldots, \mathbf{s}_T, \mathbf{a}_T)}_{\tau}$

Gaussian distribution obtained using variant of iterative LQR



(more on this later)

Can we make it work more often?



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

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

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

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

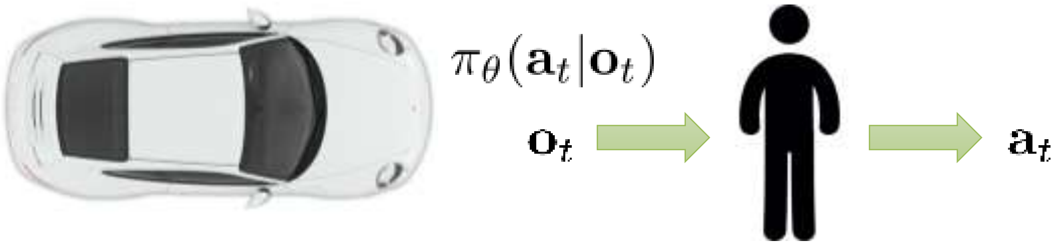
Ross et al. '11

DAgger Example

Ross et al. '11

What's the problem?

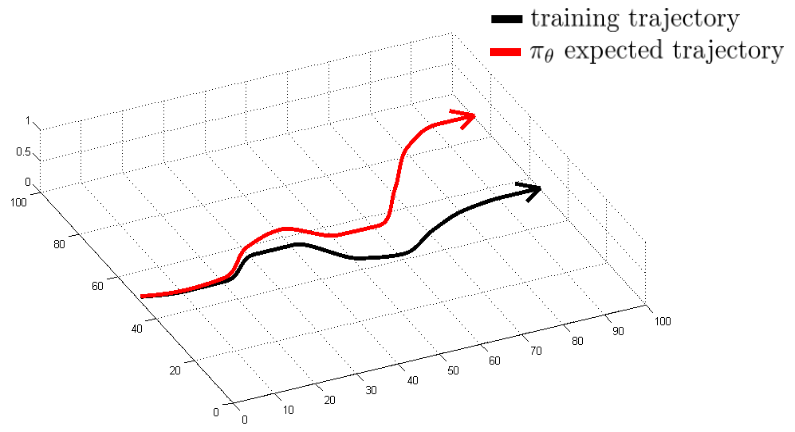
1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. $\boxed{\text{Ask human to label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



Ross et al. '11

Can we make it work without more data?

- DAgger addresses the problem of distributional “drift”
- What if our model is so good that it doesn’t drift?
- Need to mimic expert behavior very accurately
- But don’t overfit!



Why might we fail to fit the expert?

1. Non-Markovian behavior
2. Multimodal behavior

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

behavior depends only on current observation

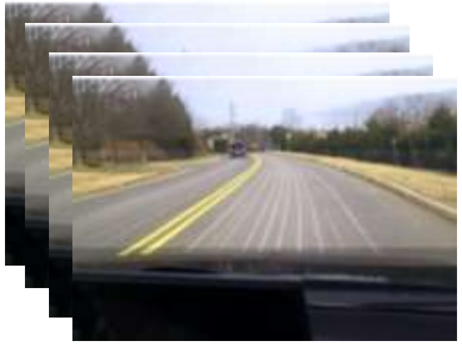
$$\pi_\theta(\mathbf{a}_t|\mathbf{o}_1, ..., \mathbf{o}_t)$$

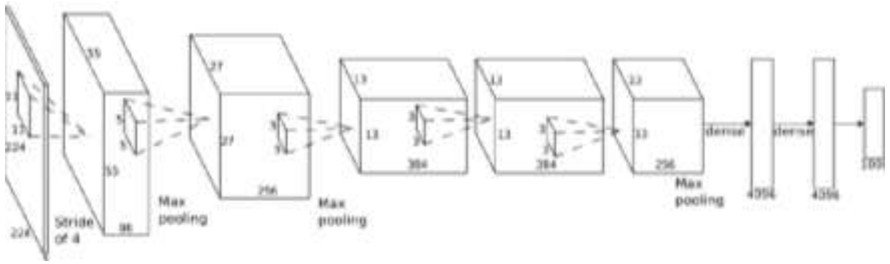
behavior depends on all past observations

If we see the same thing twice, we do the same thing twice, regardless of what happened before

Often very unnatural for human demonstrators

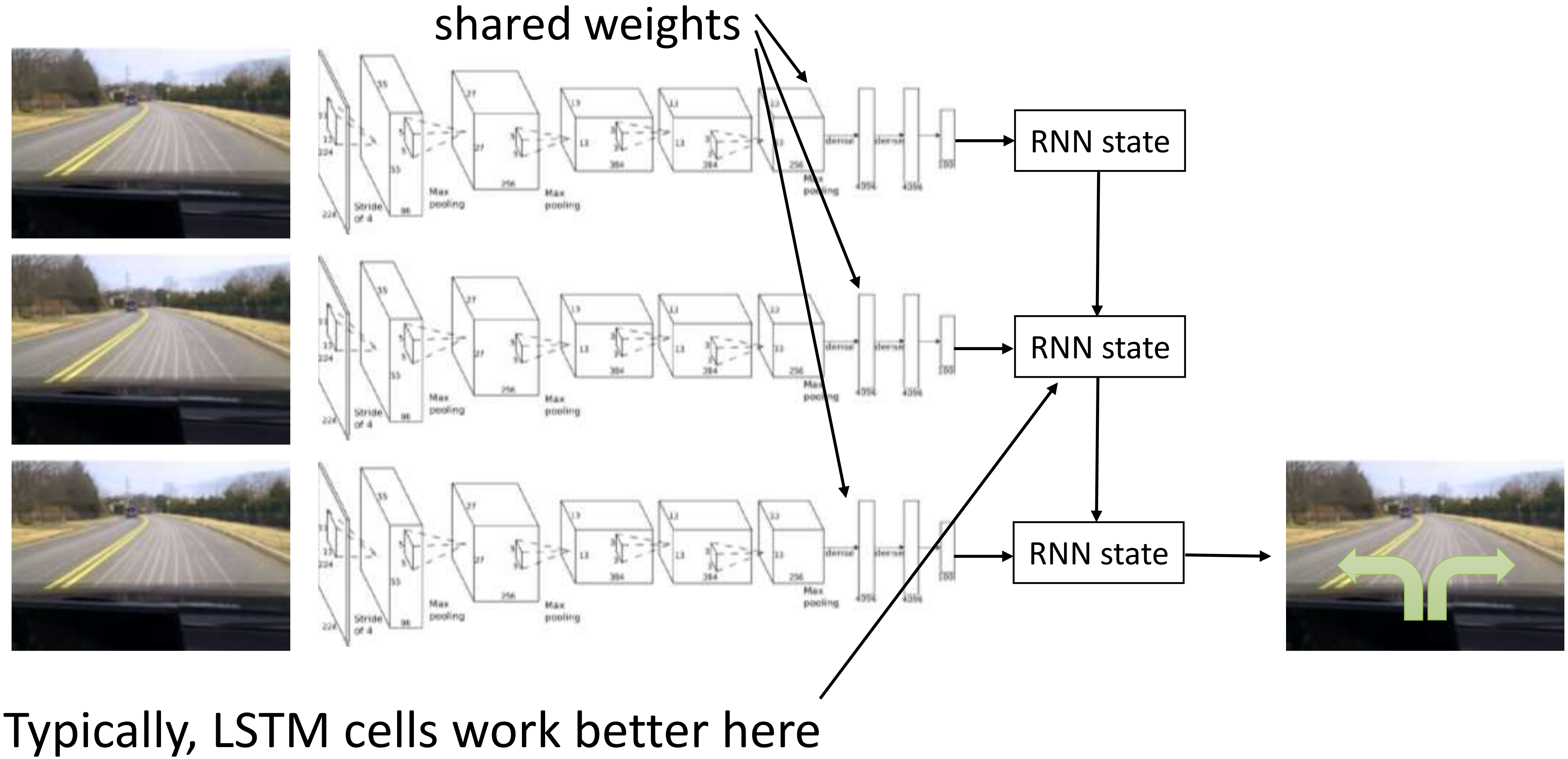
How can we use the whole history?





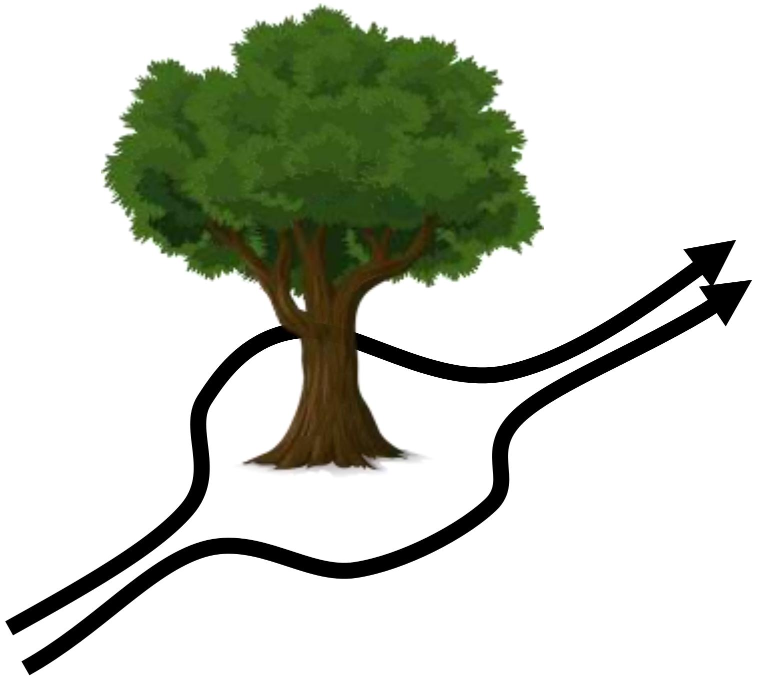
variable number of frames,
too many weights

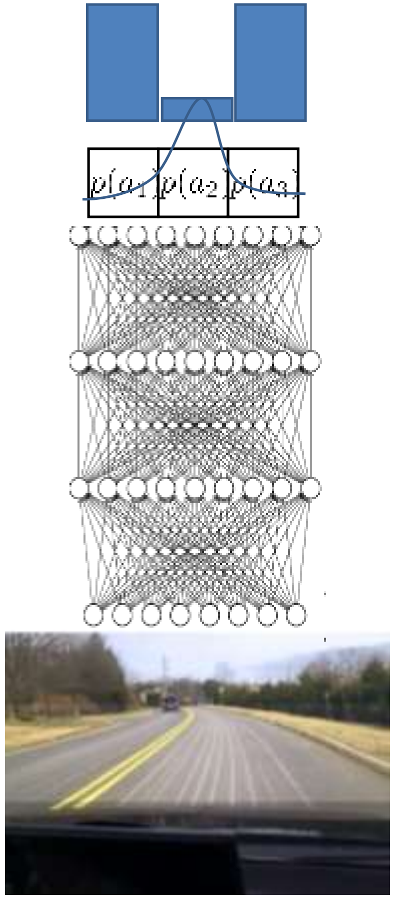
How can we use the whole history?



Why might we fail to fit the expert?

1. Non-Markovian behavior
2. Multimodal behavior



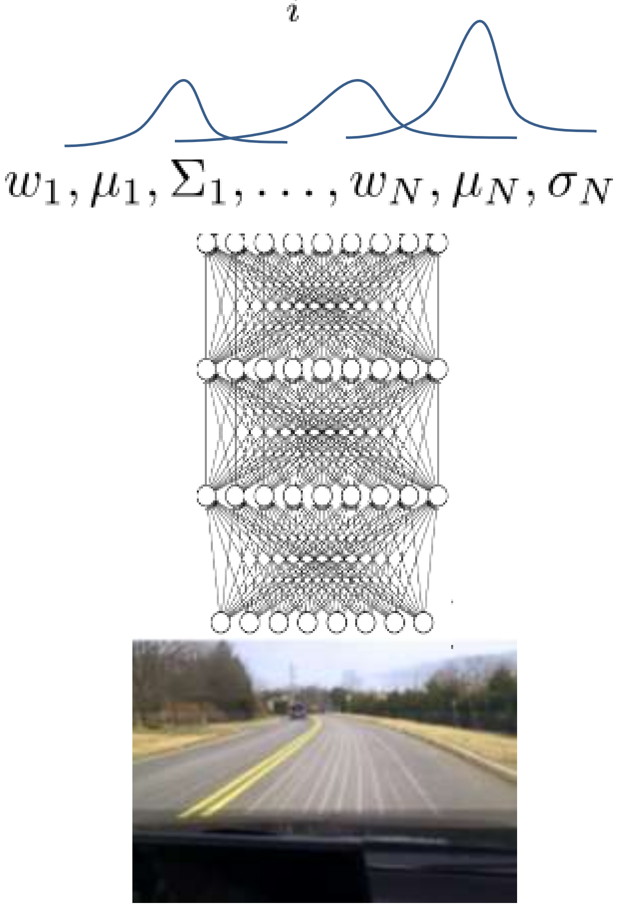


1. Output mixture of Gaussians
2. Implicit density model
3. Autoregressive discretization

Why might we fail to fit the expert?

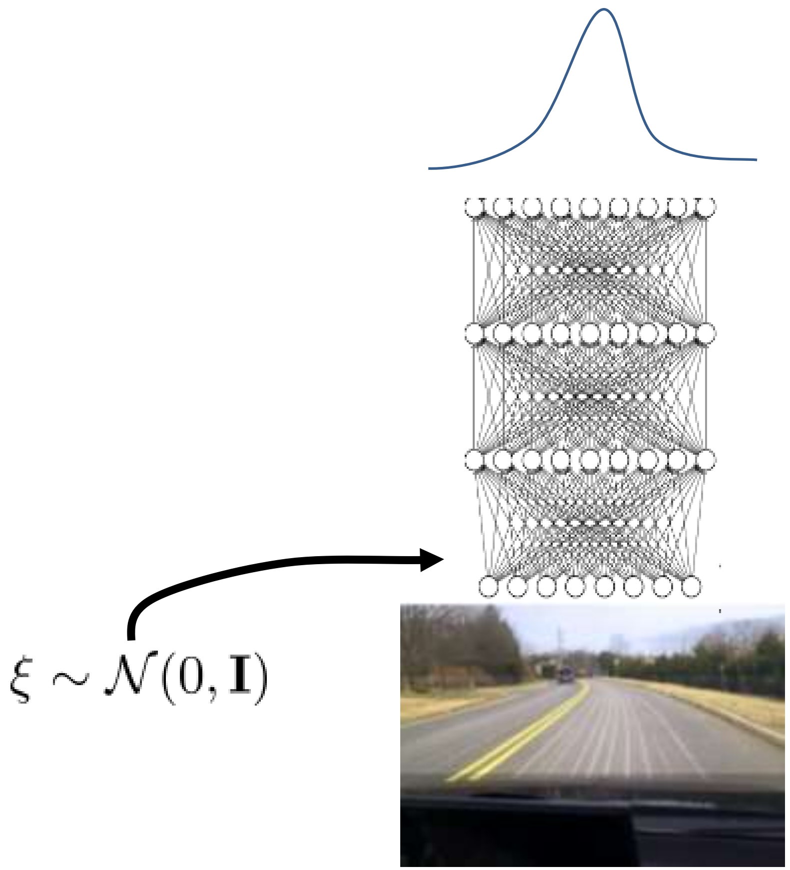
1. Output mixture of Gaussians
2. Implicit density model
3. Autoregressive discretization

$$\pi(\mathbf{a}|\mathbf{o}) = \sum_i w_i \mathcal{N}(\mu_i, \Sigma_i)$$



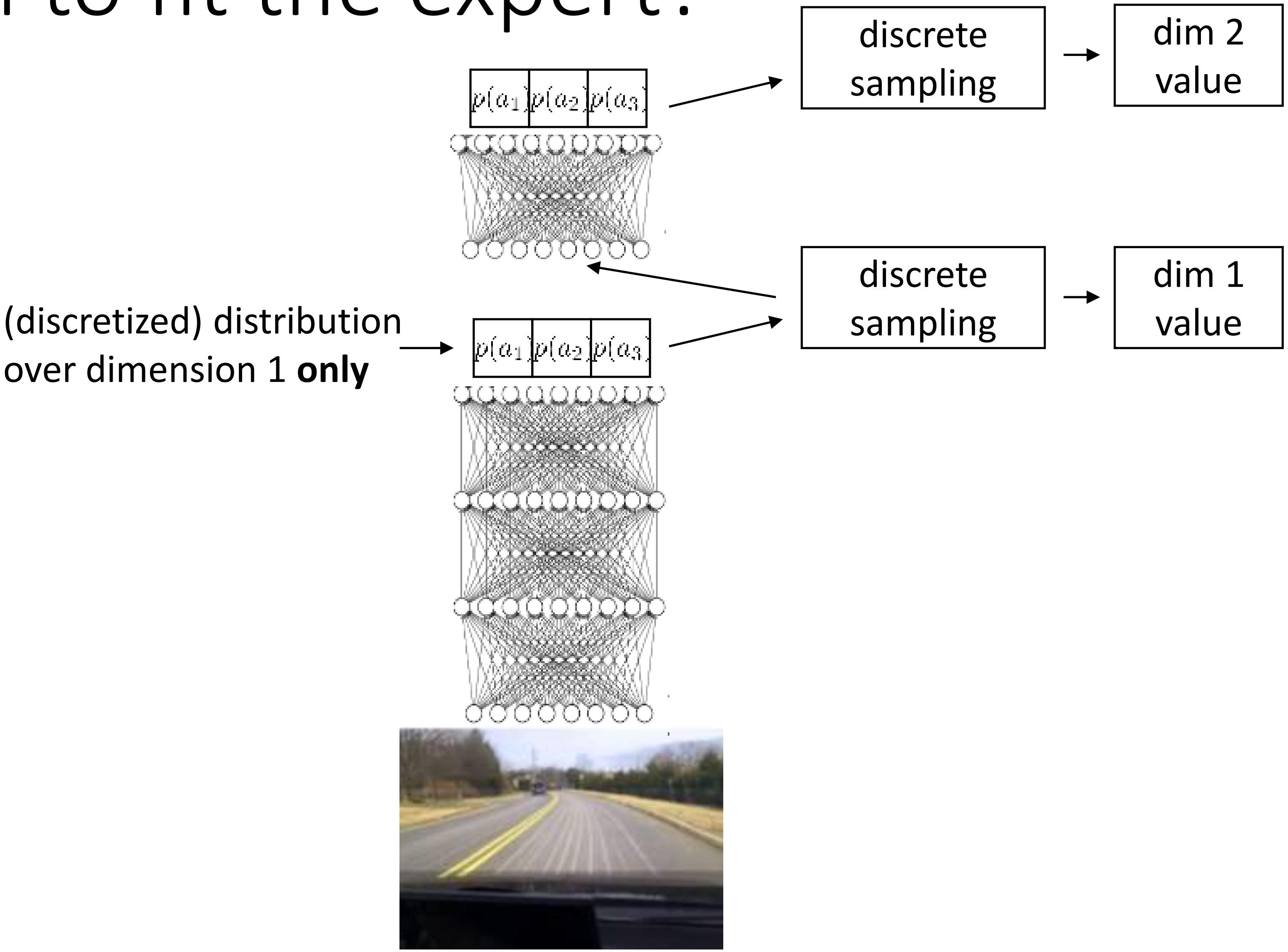
Why might we fail to fit the expert?

1. Output mixture of Gaussians
2. Implicit density model
3. Autoregressive discretization



Why might we fail to fit the expert?

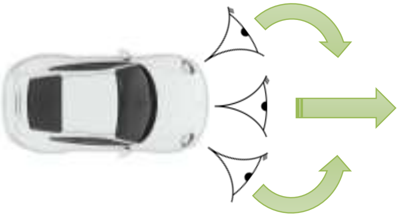
1. Output mixture of Gaussians
2. Implicit density model
3. Autoregressive discretization

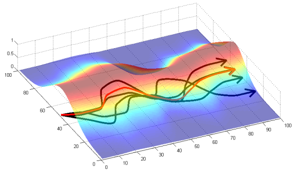


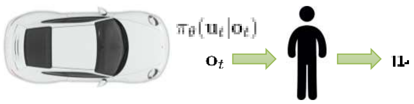
Imitation learning: recap

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

- 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
  - Better models that fit more accurately



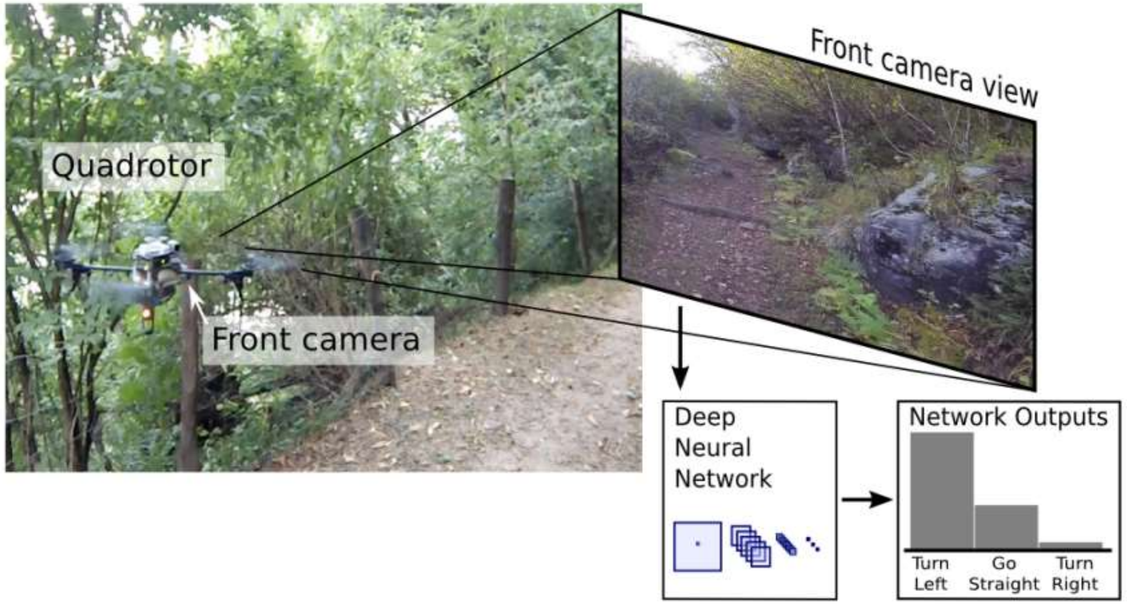




Case study 1: trail following as classification

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]

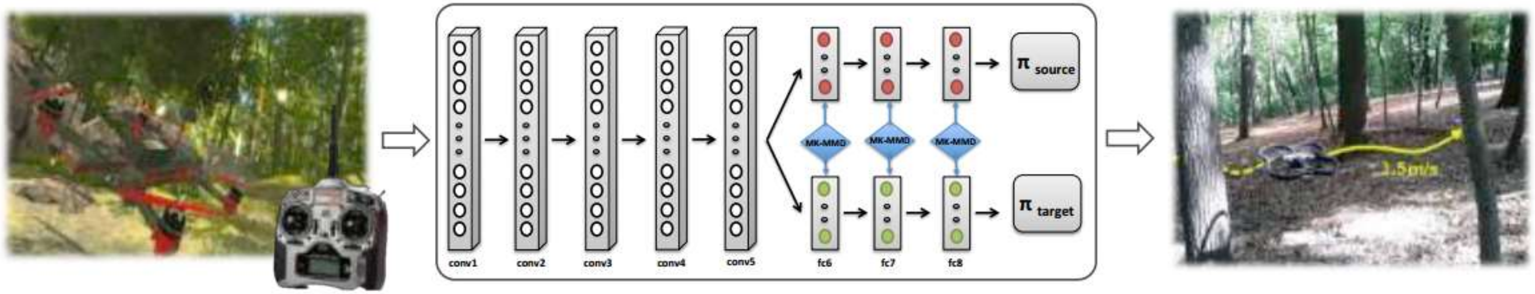


Case study 2: DAgger & domain adaptation

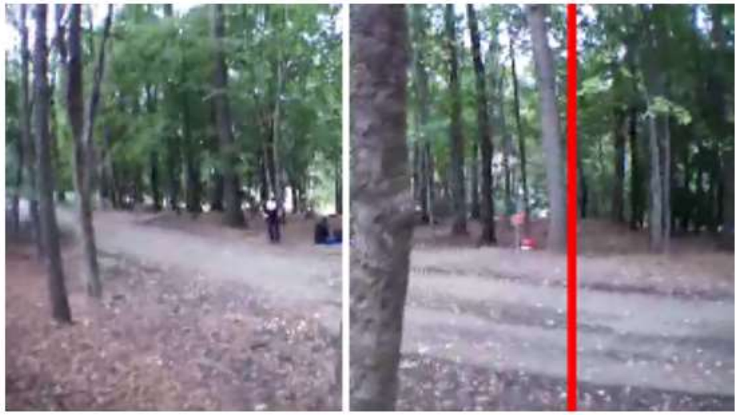
Learning Transferable Policies for Monocular Reactive MAV Control

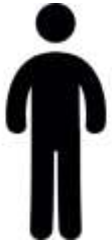
Shreyansh Daftry, J. Andrew Bagnell, and Martial Hebert

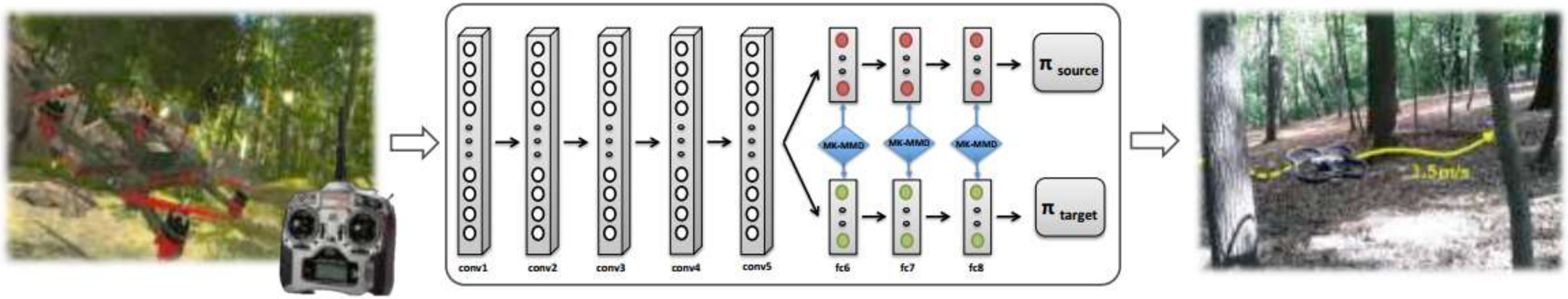
Robotics Institute, Carnegie Mellon University, Pittsburgh, USA
{daftry,dbagnell,hebert}@ri.cmu.edu



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







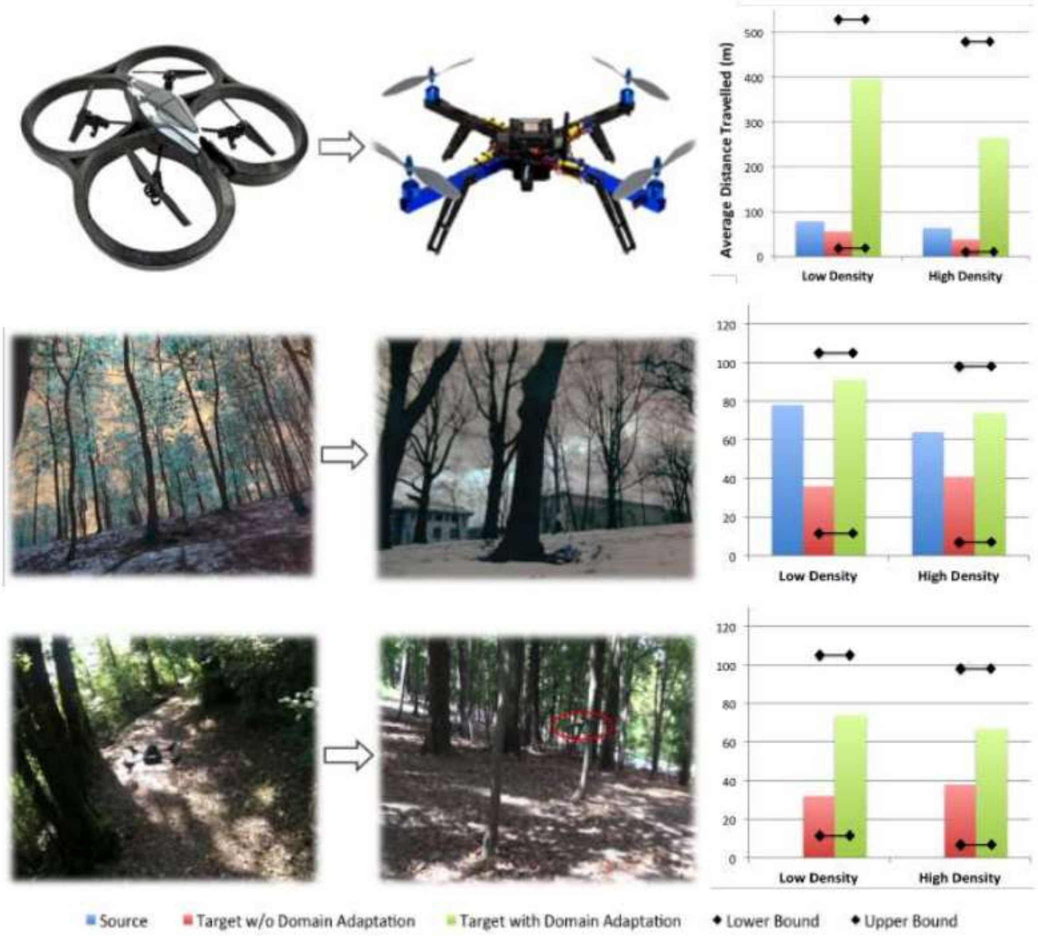


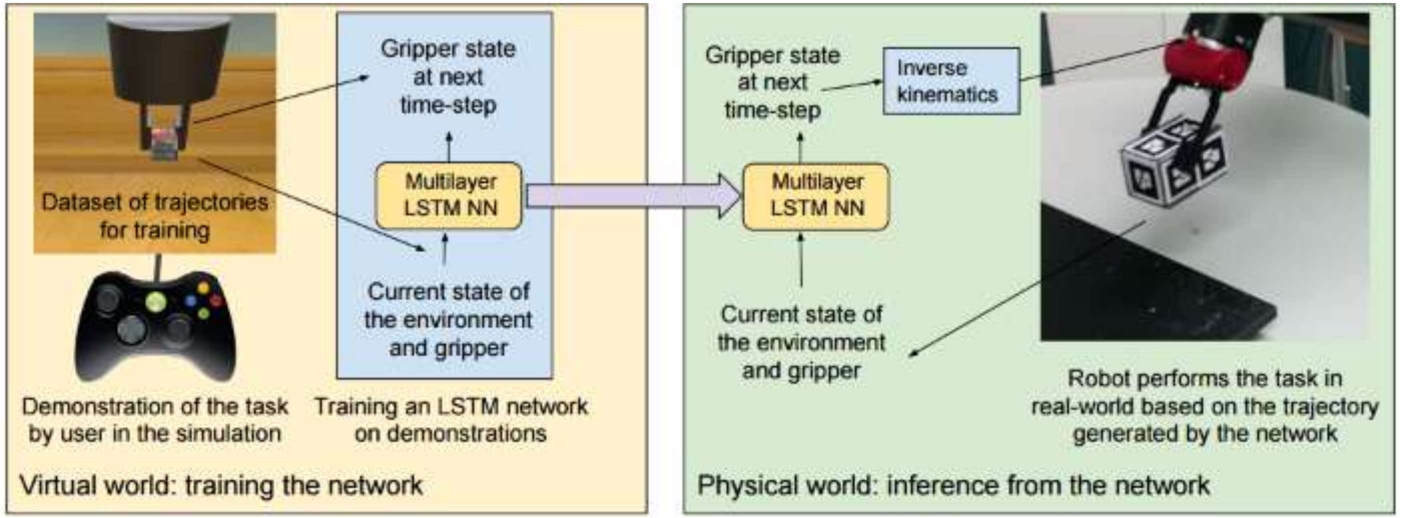
Fig. 2. Experiments and Results for (Row-1) Transfer across physical systems from ARDrone to ArduCopter, (Row-2) Transfer across weather conditions from summer to winter and (Row-3) Transfer across environments from Univ. of Zurich to CMU.

(a) (b) (c)

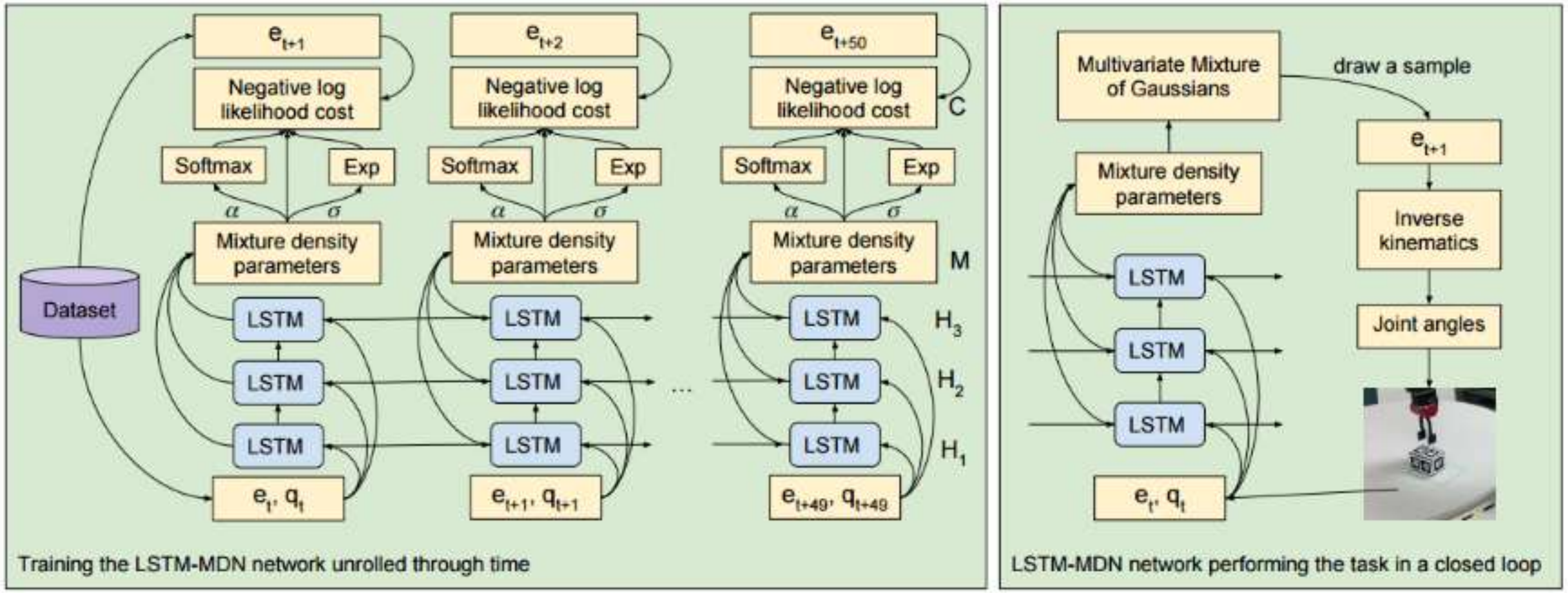
Case study 3: Imitation with LSTMs

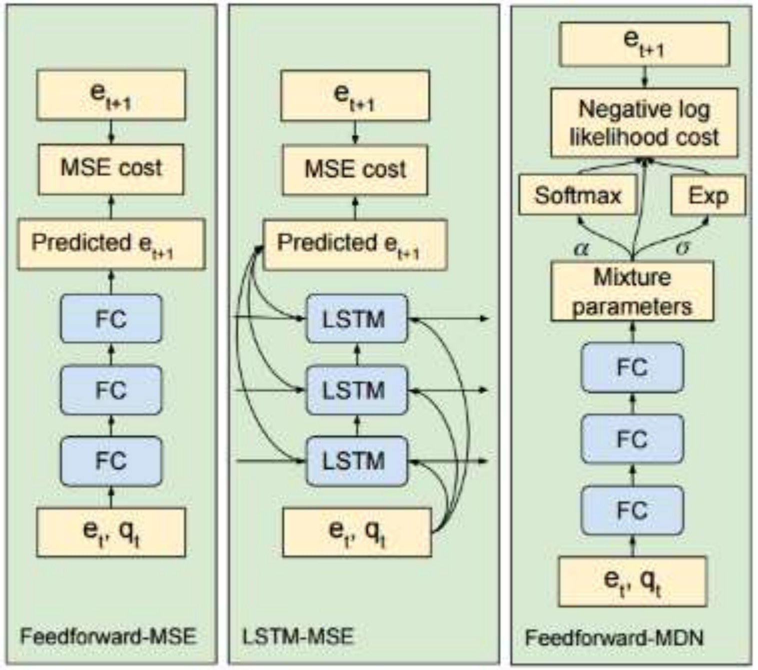
Learning real manipulation tasks from virtual demonstrations using LSTM

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



Learning Manipulation Trajectories Using Recurrent Neural Networks



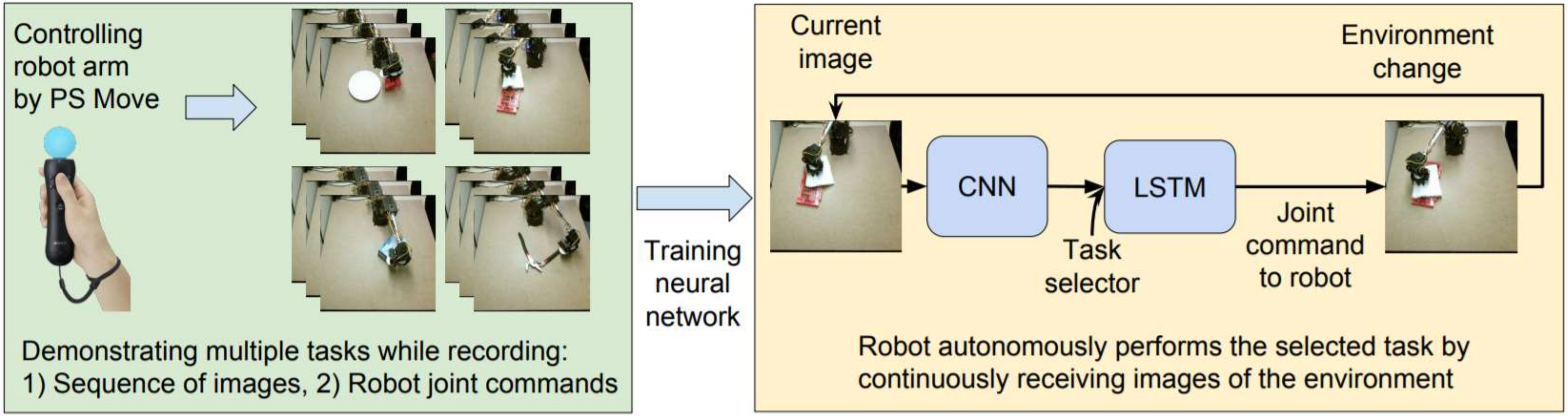


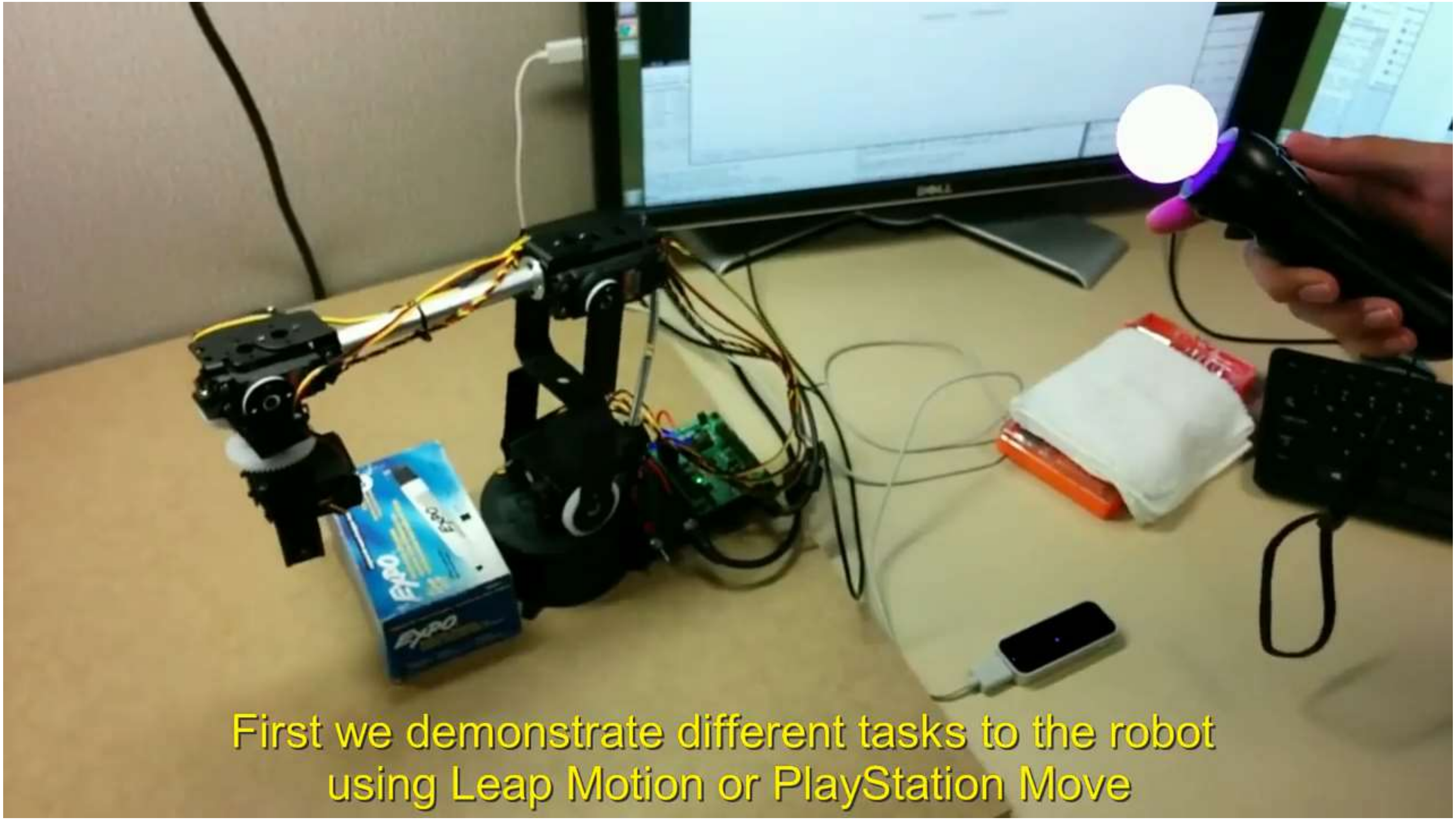
Controller	Pick and place	Push to pose
Feedfoward-MSE	0%	0%
LSTM-MSE	85%	0%
Feedforward-MDN	95%	15%
LSTM-MDN	**100%**	**95%**

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

Follow-up: adding vision

Vision-Based Multi-Task Manipulation for Inexpensive Robots Using End-To-End Learning from Demonstration



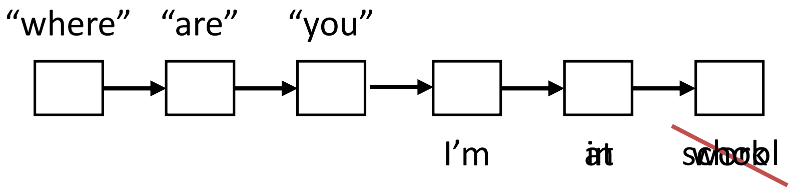

EXPO
First we demonstrate different tasks to the robot
using Leap Motion or PlayStation Move

Other topics in imitation learning

- Structured prediction

  x: where are you

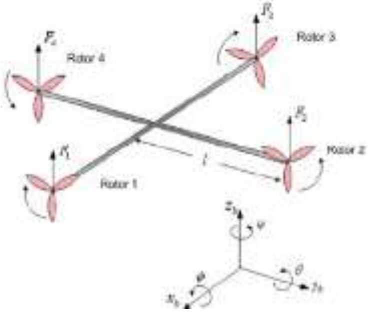
  y: I’m at work



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

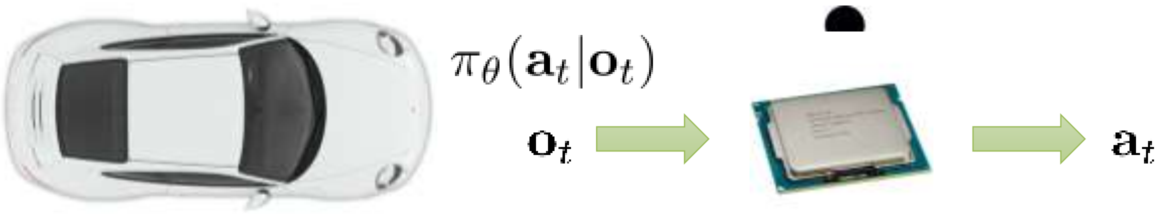
Imitation learning: what's the problem?

- 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

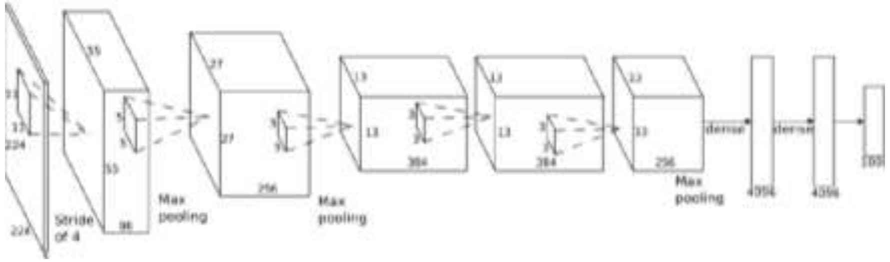


- Humans can learn autonomously; can our machines do the same?
  - Unlimited data from own experience
  - Continuous self-improvement

Next time: learning without humans



Terminology & notation



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

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

$c(\mathbf{s}_t, \mathbf{a}_t)$ – cost function
$r(\mathbf{s}_t, \mathbf{a}_t)$ – reward function

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

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

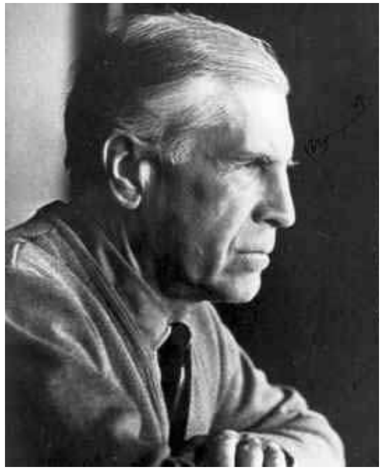
Aside: notation

$\mathbf{s}_t$ – state
$\mathbf{a}_t$ – action
$r(\mathbf{s}, \mathbf{a})$ – reward function

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

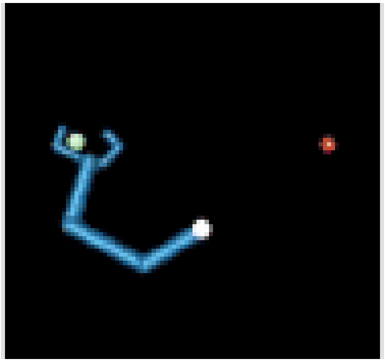
$$r(\mathbf{s}, \mathbf{a}) = -c(\mathbf{x}, \mathbf{u})$$

Richard Bellman



Lev Pontryagin

Cost/reward functions in theory and practice



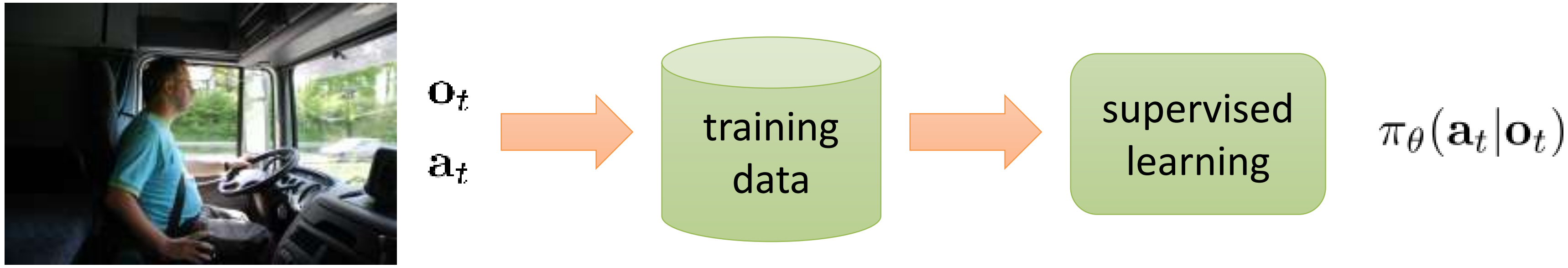
$$r(\mathbf{s}, \mathbf{a}) = \begin{cases} 1 \text{ if object at target} \\ 0 \text{ otherwise} \end{cases}$$

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

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

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

A cost function for imitation?



$$r(\mathbf{s},\mathbf{a}) = \log p(\mathbf{a} = \pi^\star(\mathbf{s})|\mathbf{s})$$

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

Ross et al. '11

The trouble with cost & reward functions

Mnih et al. '15

reinforcement learning agent

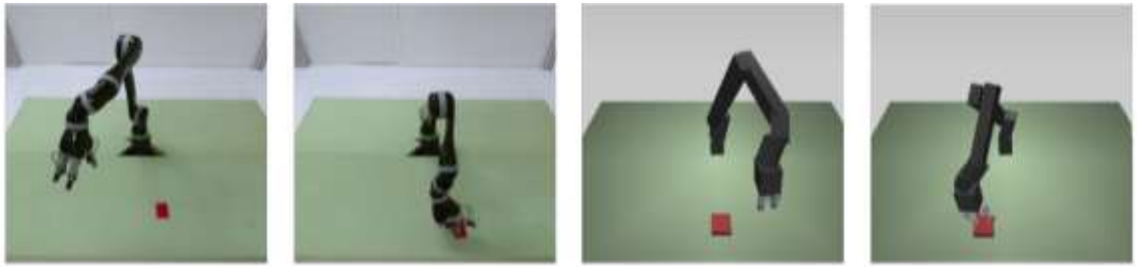
what is the reward?

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...

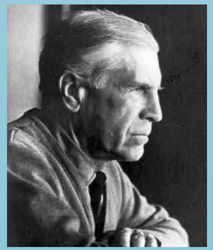
A note about terminology…

the "R" word

a bit of history…

reinforcement learning (the **problem** statement) $\max \sum_{t=1}^{T} E[r(\mathbf{s}_t, \mathbf{a}_t)]$ $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$

reinforcement learning (the **method**) without using the **model** $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$



Lev Pontryagin

Richard Bellman

Andrew Barto

Richard Sutton