Supervised Learning of Behaviors

CS 285: Deep Reinforcement Learning, Decision Making, and Control

Sergey Levine

Class Notes

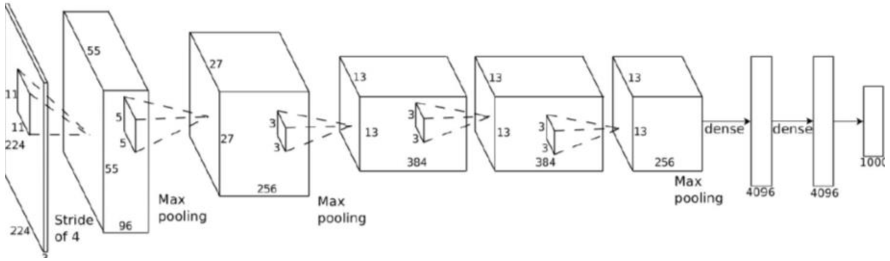
1. Homework 1 is out this evening
2. Remember to start forming final project groups
   - Final project assignment document is now out!
   - Proposal due Sep 25

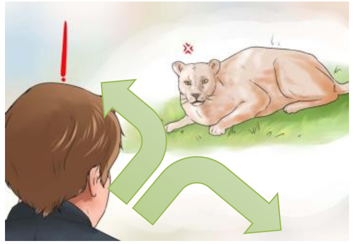
Today's Lecture

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. A little bit of theory
4. Case studies of recent work in (deep) imitation learning

- Goals:
    - Understand definitions & notation
    - Understand basic imitation learning algorithms
    - Understand tools for theoretical analysis

Terminology & notation





$\mathbf{o}_t$

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

$\mathbf{a}_t$

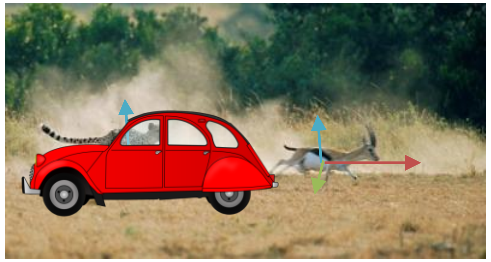
$\mathbf{s}_t$ – state

$\mathbf{o}_t$ – observation

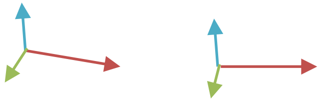
$\mathbf{a}_t$ – action

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

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

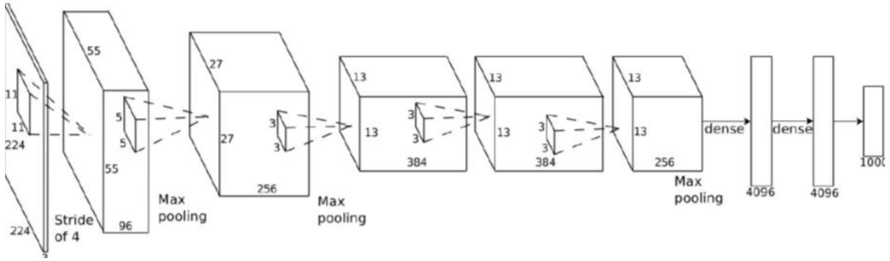


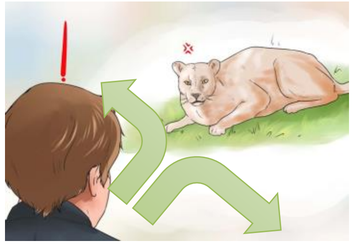
$\mathbf{o}_t$ – observation



$\mathbf{s}_t$ – state

Terminology & notation



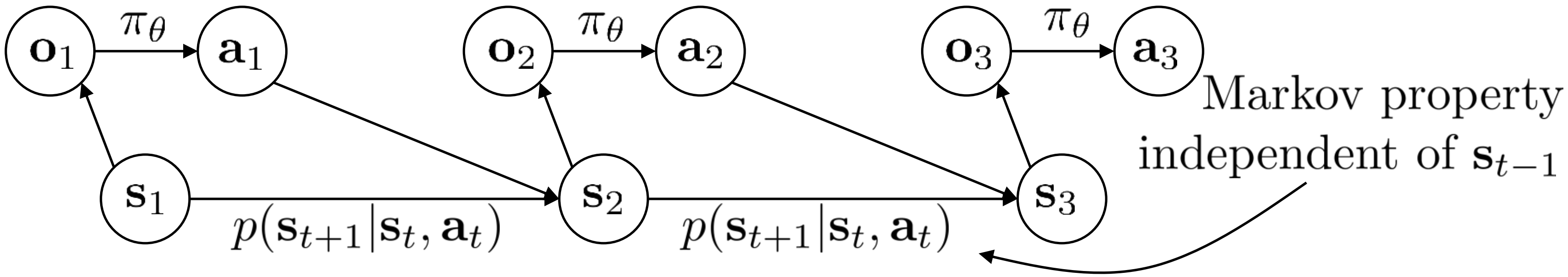


$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

$\mathbf{s}_t$ – 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)

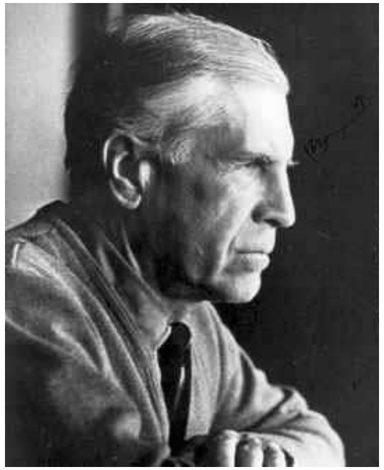


Aside: notation

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

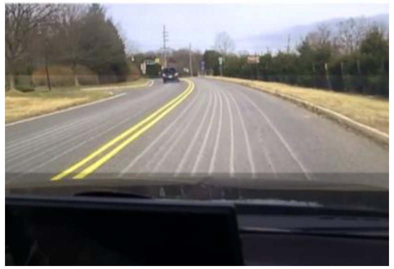
$\mathbf{x}_t$ – state
$\mathbf{u}_t$ – action   управление

Richard Bellman

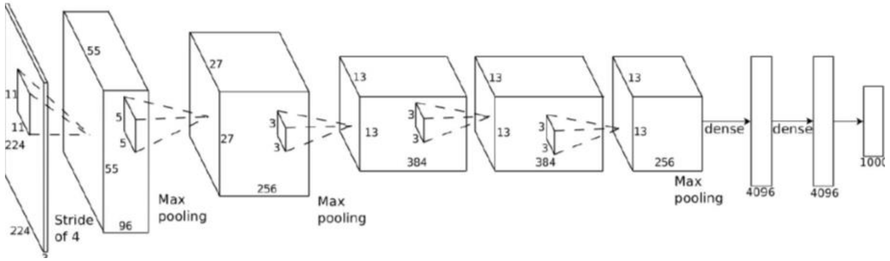


Lev Pontryagin

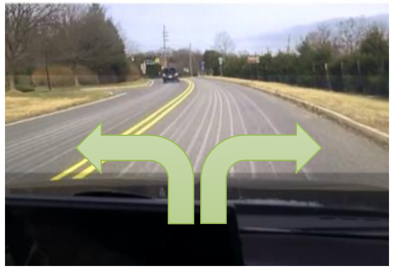
Imitation Learning



$\mathbf{o}_t$



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



$\mathbf{a}_t$

$\mathbf{o}_t$
$\mathbf{a}_t$

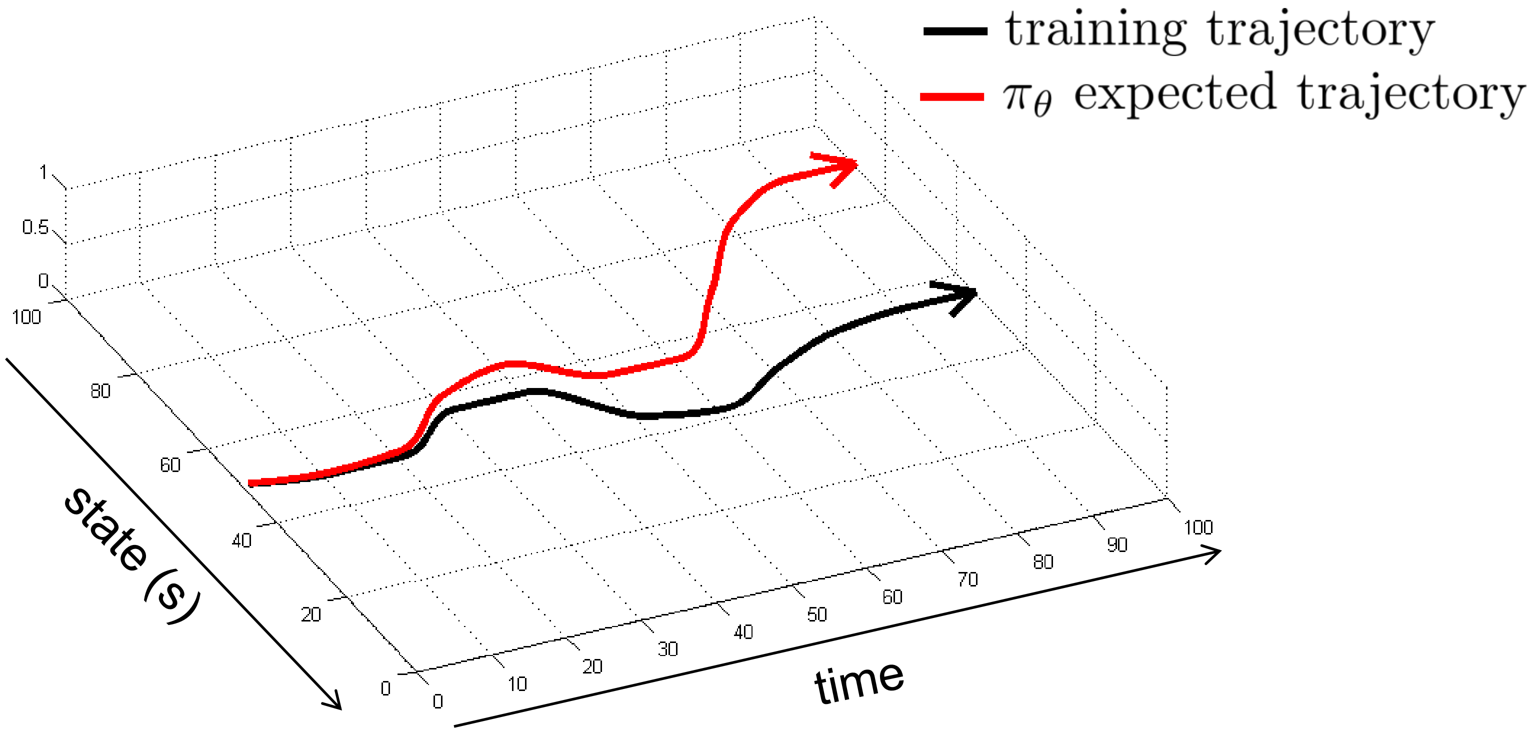
→ training data → supervised learning → $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

behavioral cloning

Images: Bojarski et al. '16, NVIDIA

Does it work?

No!

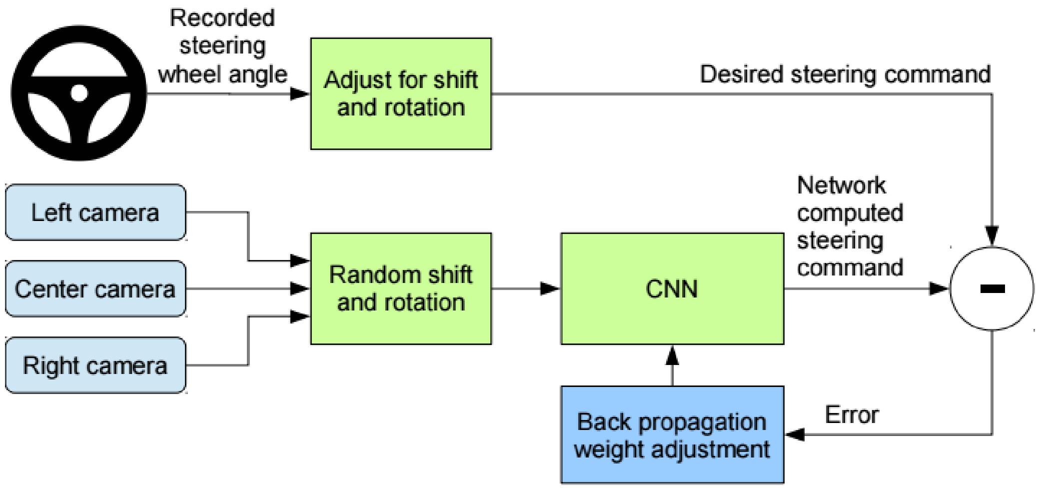


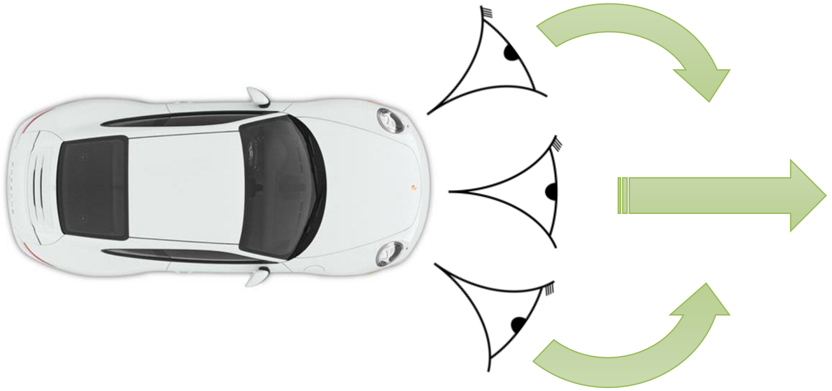
Does it work?

Yes!

Video: Bojarski et al. ‘16, NVIDIA

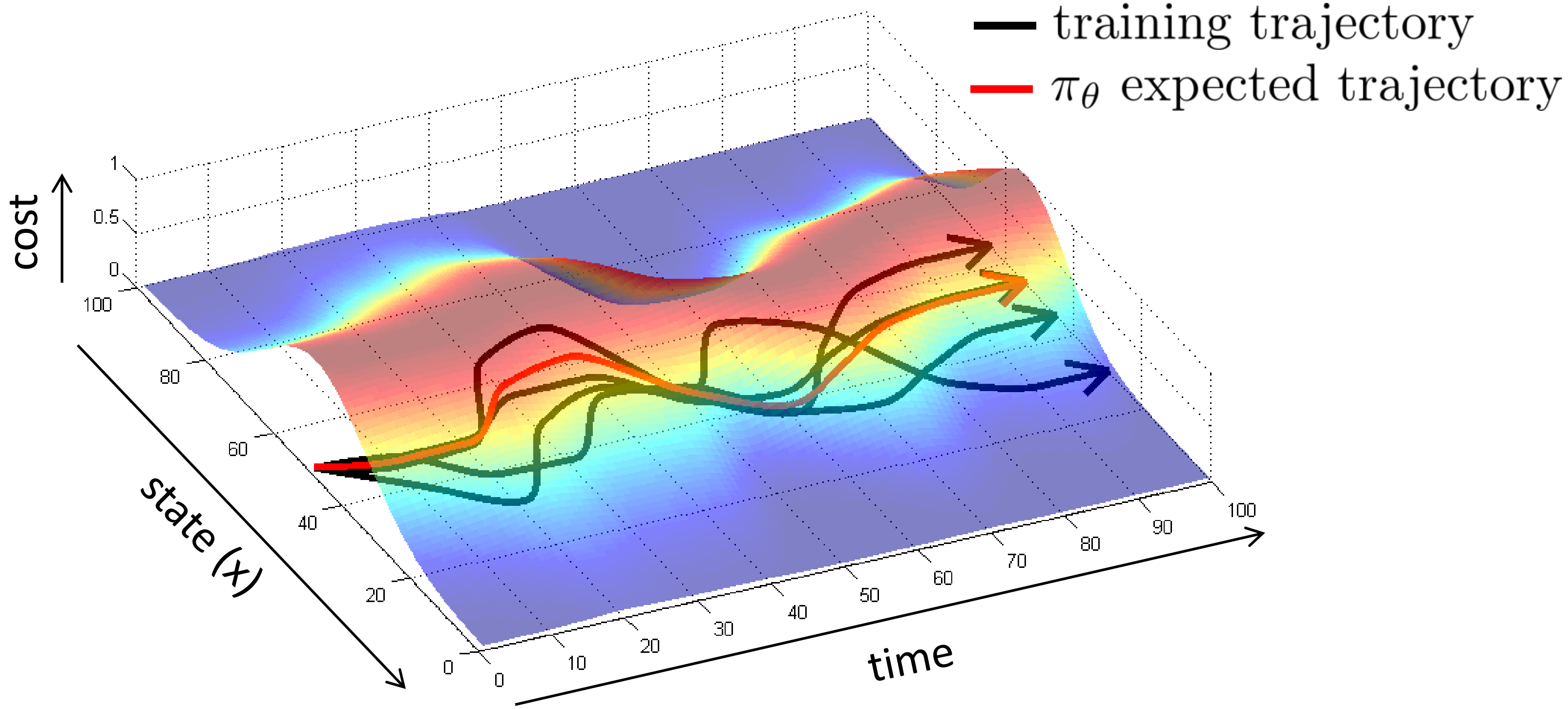
Why did that work?





Bojarski et al. ‘16, NVIDIA

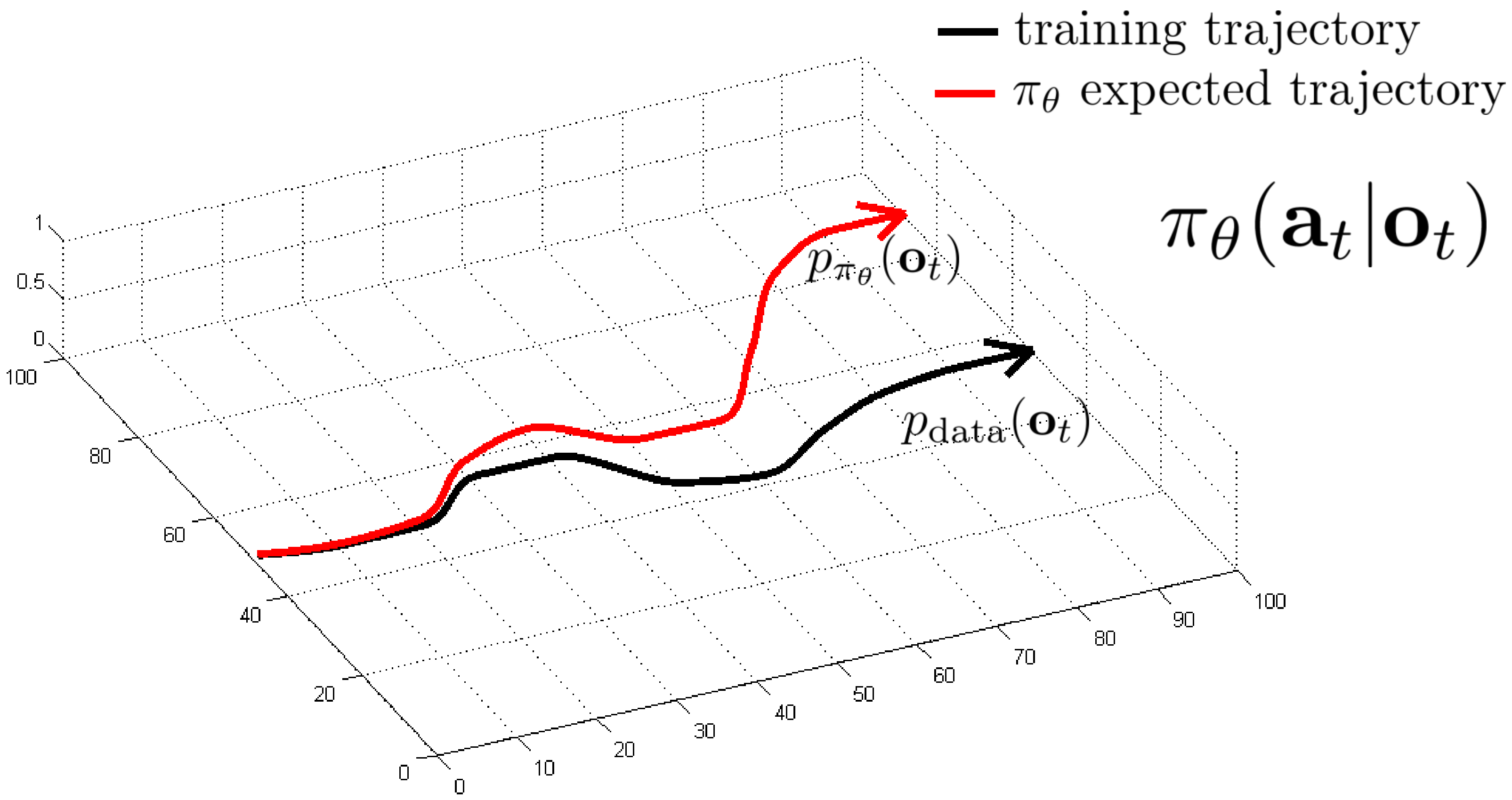
Can we make it work more often?



stability

(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 $p_{\text{data}}(\mathbf{o}_t)$

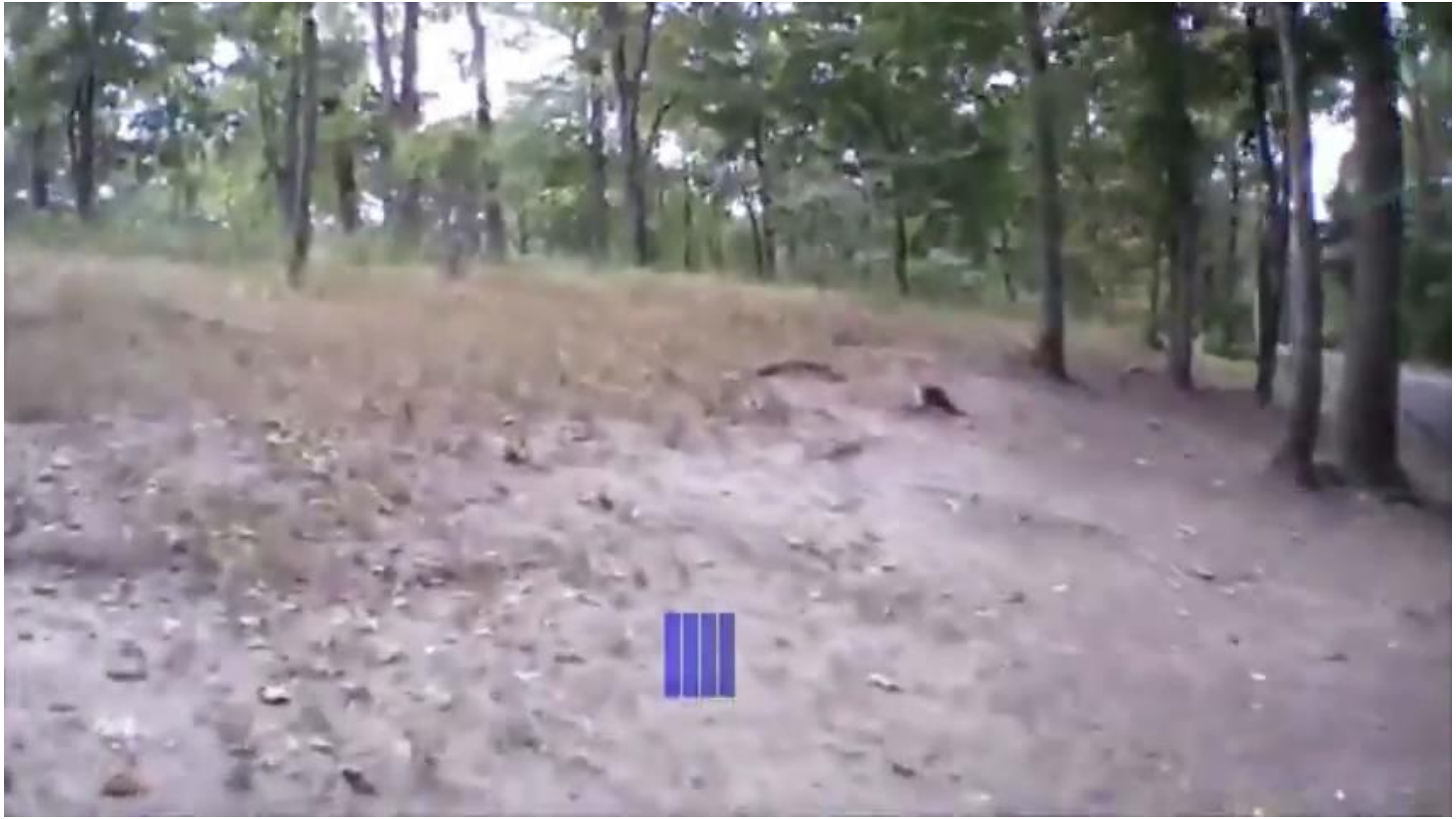
how? just run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

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

1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \dots, \mathbf{o}_N, \mathbf{a}_N\}$
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

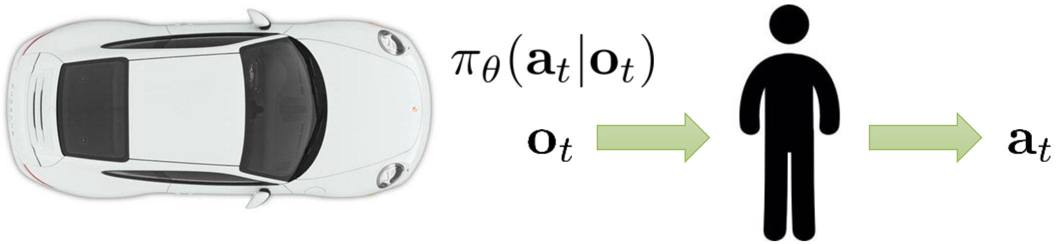
DAgger Example



Ross et al. '11

What's the problem?

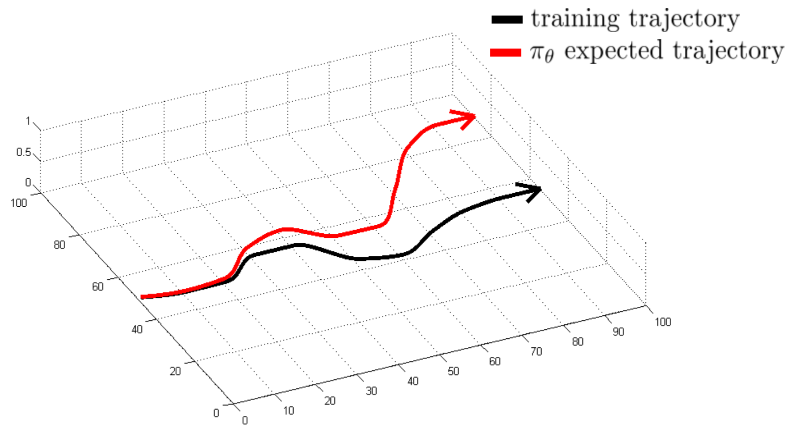
1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$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



Ross et al. '11

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

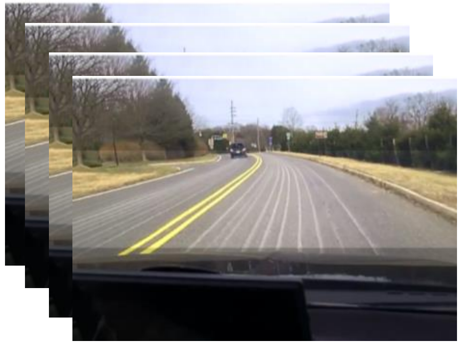
$$\pi_\theta(\mathbf{a}_t|\mathbf{o}_1, ..., \mathbf{o}_t)$$

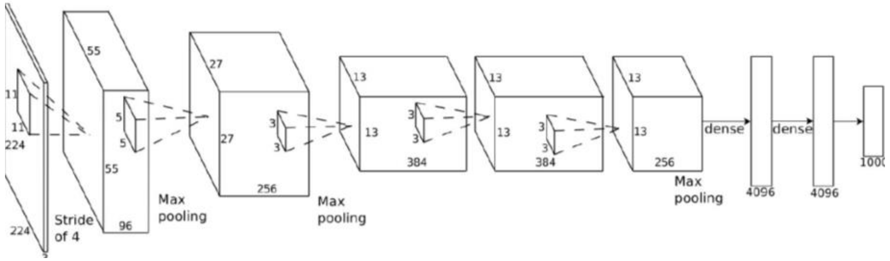
behavior depends on all past observations

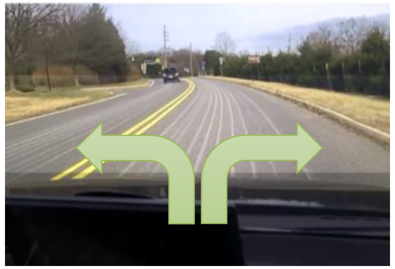
If we see the same thing twice, we do the same thing twice, regardless of what happened before

Often very unnatural for human demonstrators

How can we use the whole history?



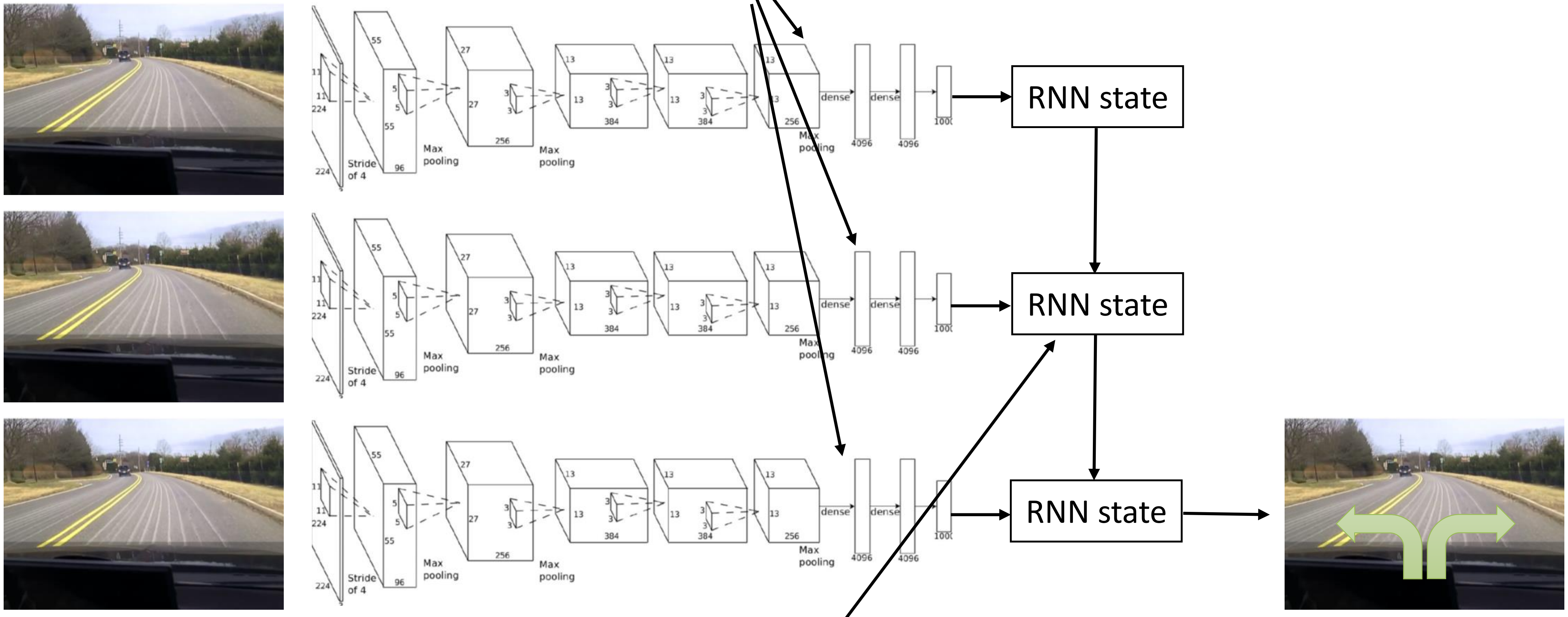




variable number of frames,
too many weights

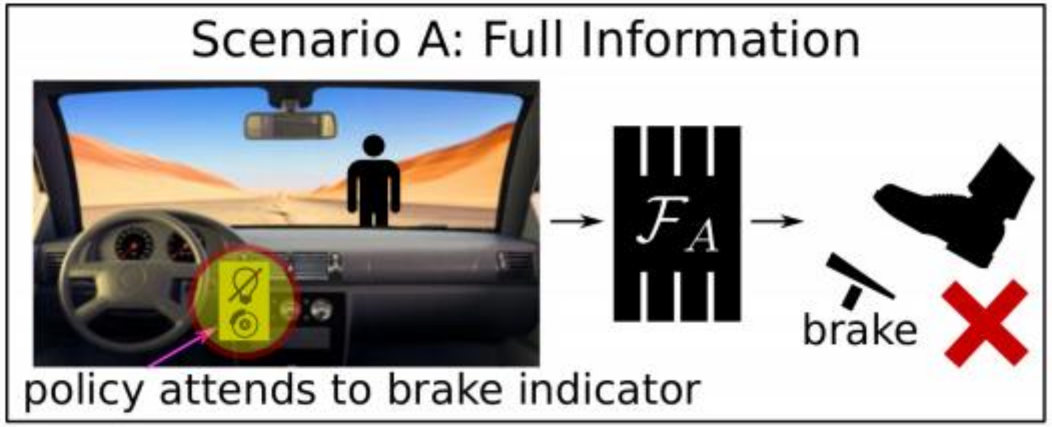
How can we use the whole history?

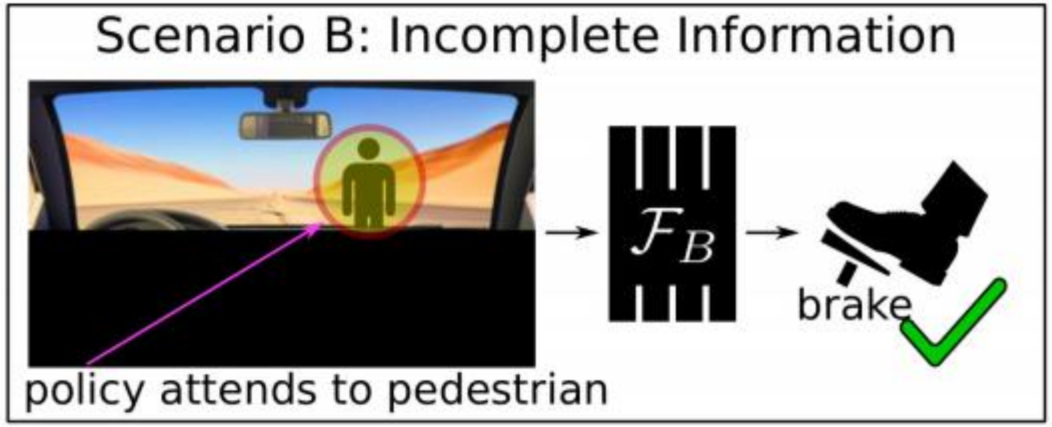
shared weights



Typically, LSTM cells work better here

Aside: why might this work **poorly**?





“causal confusion”

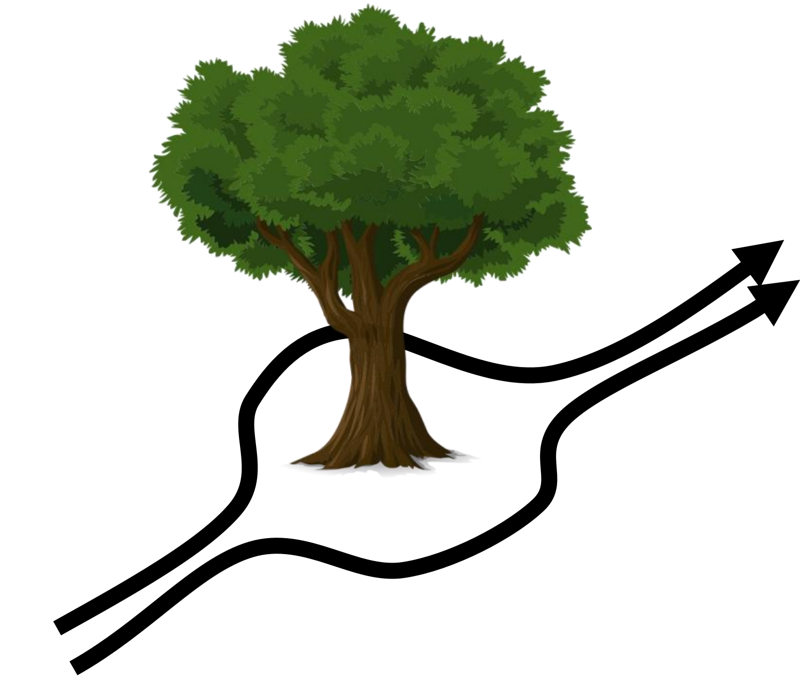
see: de Haan et al., “Causal Confusion in Imitation Learning”

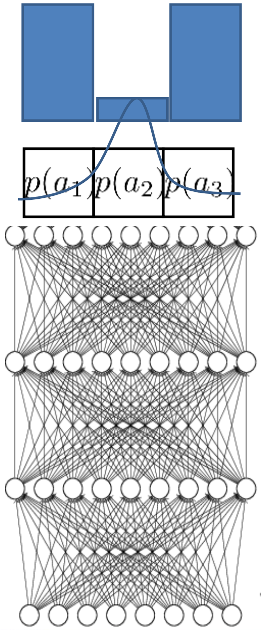
Question 1: Does including history exacerbate causal confusion?

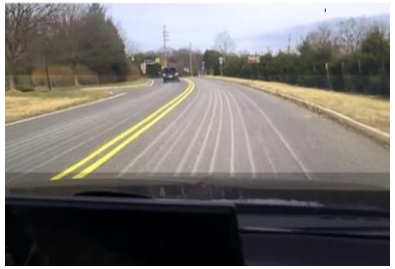
Question 2: Can DAgger mitigate causal confusion?

Why might we fail to fit the expert?

1. Non-Markovian behavior
2. Multimodal behavior





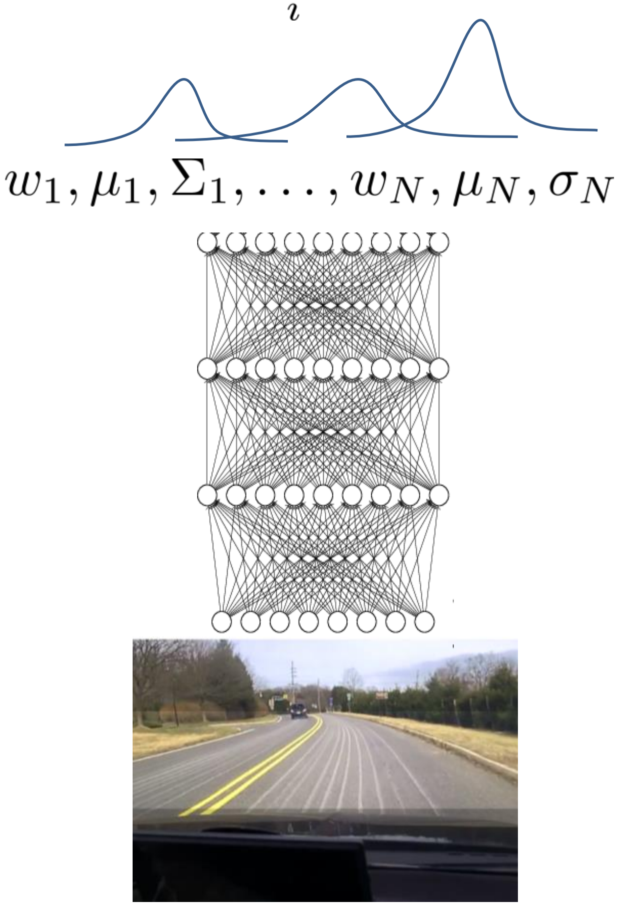


1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization

Why might we fail to fit the expert?

1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization

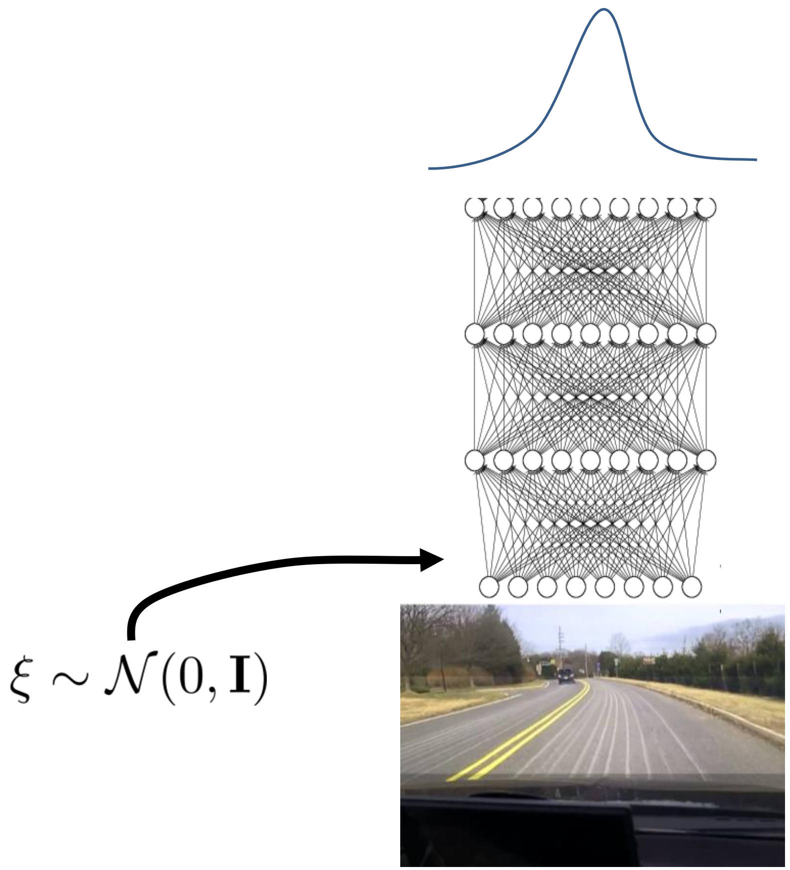
$$\pi(\mathbf{a}|\mathbf{o}) = \sum_i w_i \mathcal{N}(\mu_i, \Sigma_i)$$



Why might we fail to fit the expert?

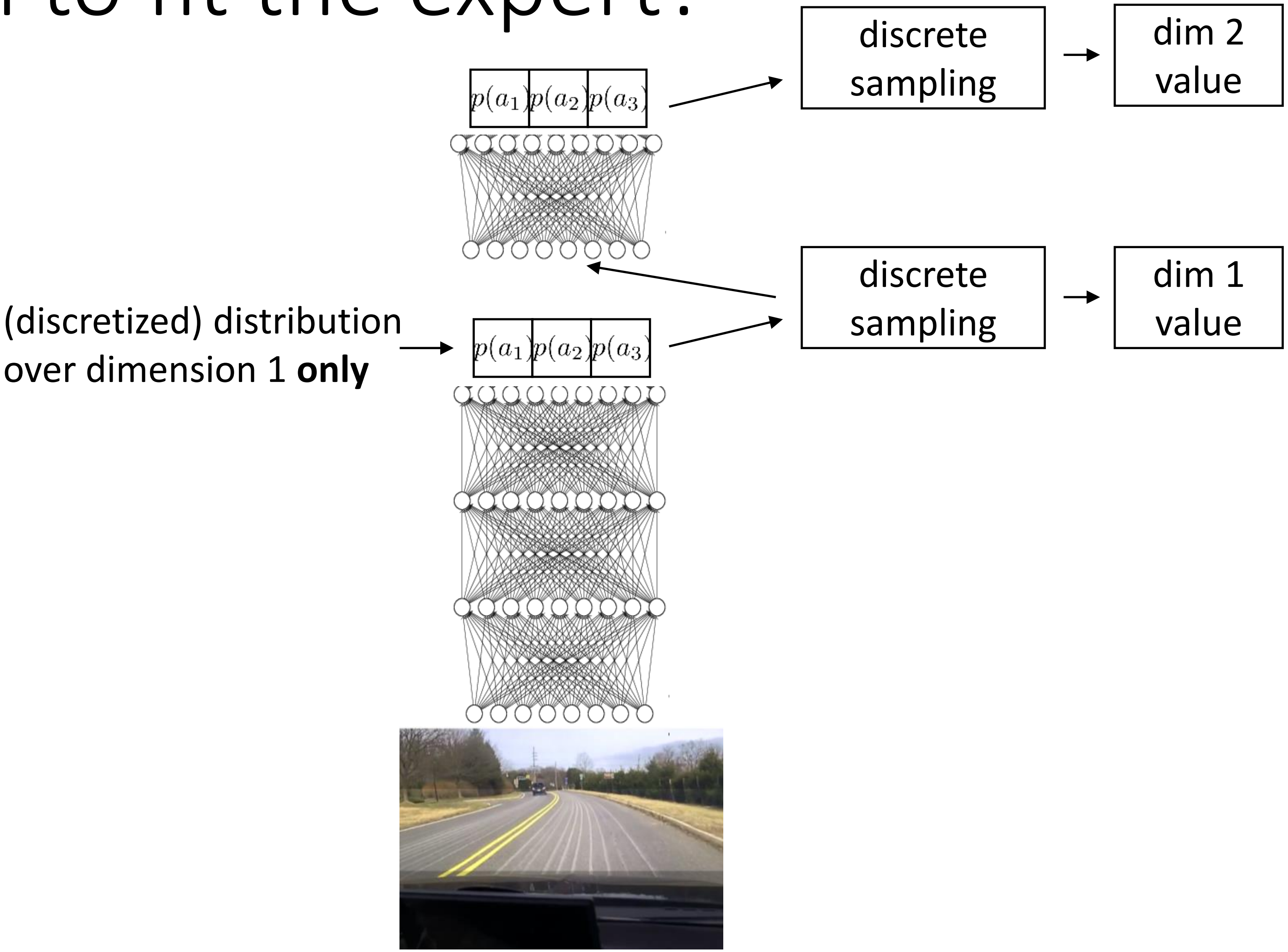
1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization

Look up some of these:
- Conditional variational autoencoder
- Normalizing flow/realNVP
- Stein variational gradient descent



Why might we fail to fit the expert?

1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization



Imitation learning: recap

$\mathbf{o}_t$

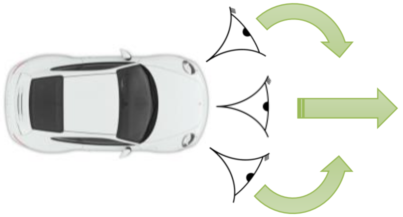
$\mathbf{a}_t$

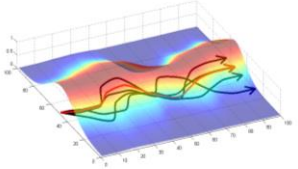
training data

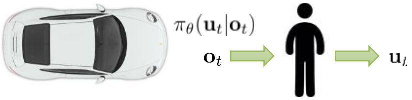
supervised learning

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

- 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





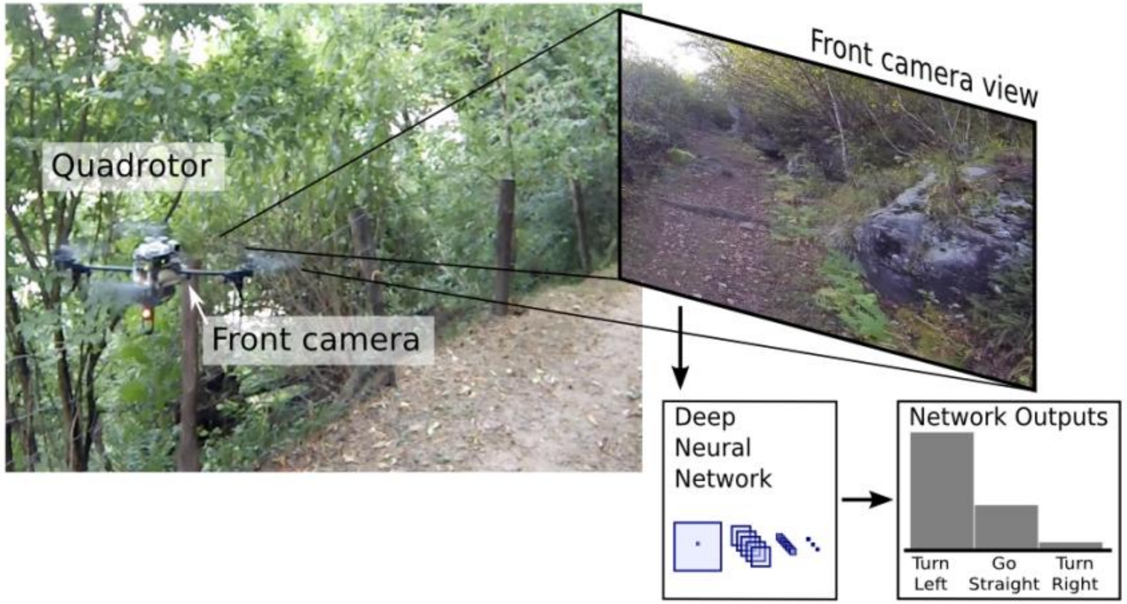


Break

Case study 1: trail following as classification

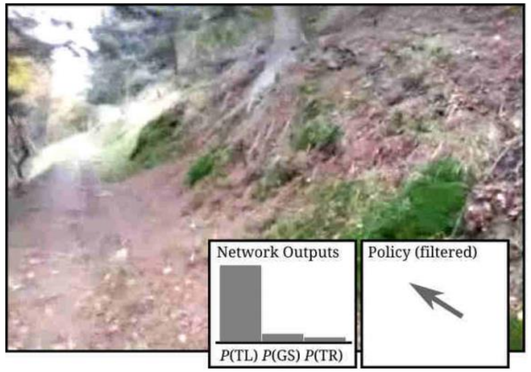
A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots

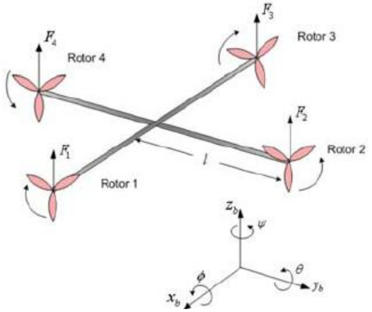
Alessandro Giusti[1], Jérôme Guzzi[1], Dan C. Cireşan[1], Fang-Lin He[1], Juan P. Rodríguez[1]
Flavio Fontana[2], Matthias Faessler[2], Christian Forster[2]
Jürgen Schmidhuber[1], Gianni Di Caro[1], Davide Scaramuzza[2], Luca M. Gambardella[1]



Imitation learning: what's the problem?

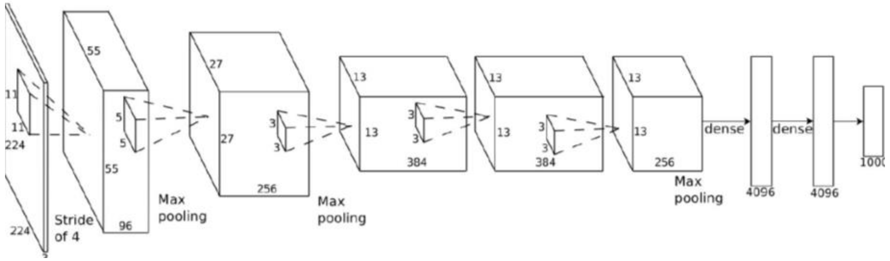
- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful
- Humans are not good at providing some kinds of actions





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

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

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

$$\min_{\boldsymbol{\theta}} E_{\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s}),\mathbf{s}'\sim p(\mathbf{s}'|\mathbf{s},\mathbf{a})}[\delta(\mathbf{s}' = \text{eaten by tiger})]$$

$$\min_{\boldsymbol{\theta}} E_{\mathbf{s}_{1:T},\mathbf{a}_{1:T}}\left[\sum_t \delta(\mathbf{s}_t = \text{eaten by tiger})\right]$$

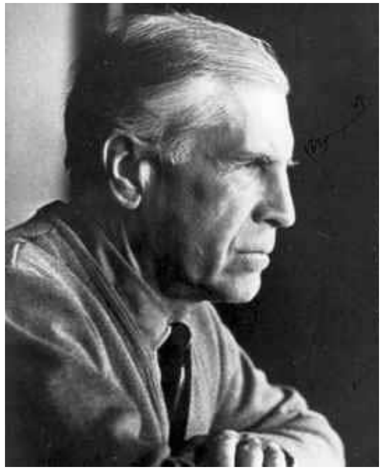
Aside: notation

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

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

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

Richard Bellman



Lev Pontryagin

A cost function for imitation?

$\mathbf{o}_t$

$\mathbf{a}_t$

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

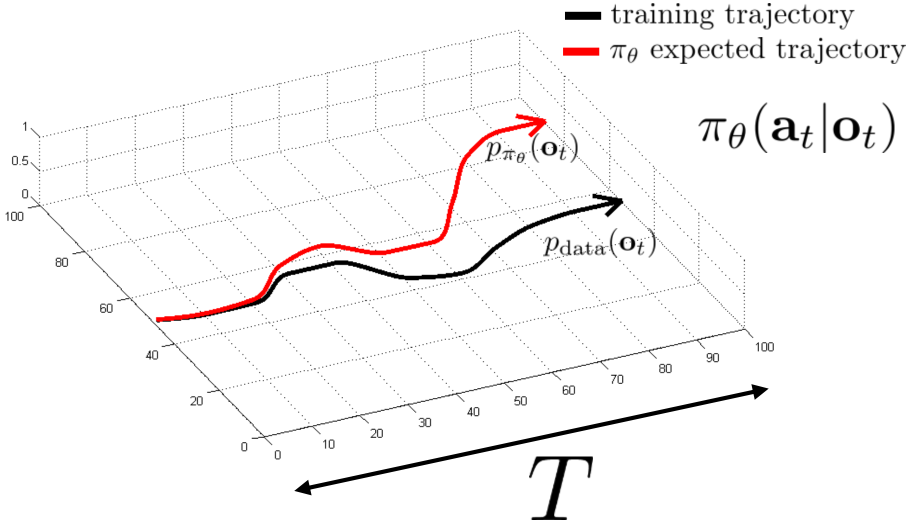
$$r(\mathbf{s},\mathbf{a}) = \log p(\mathbf{a} = \pi^\star(\mathbf{s})|\mathbf{s})$$

$$c(\mathbf{s},\mathbf{a}) = \begin{cases} 0 \text{ if } \mathbf{a} = \pi^\star(\mathbf{s}) \\ 1 \text{ otherwise} \end{cases}$$

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

Some analysis

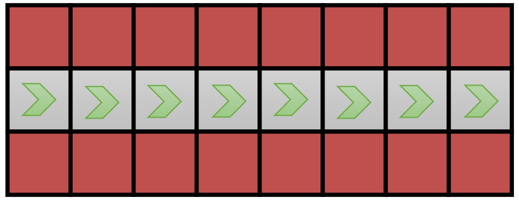


How bad is it?

$$c(\mathbf{s},\mathbf{a}) = \begin{cases} 0 \text{ if } \mathbf{a} = \pi^\star(\mathbf{s}) \\ 1 \text{ otherwise} \end{cases}$$

assume: $\pi_\theta(\mathbf{a} \neq \pi^\star(\mathbf{s})|\mathbf{s}) \leq \epsilon$

for all $\mathbf{s} \in \mathcal{D}_{\text{train}}$



$$\underbrace{E\left[\sum_t c(\mathbf{s}_t,\mathbf{a}_t)\right]}_{O(\epsilon T^2)} \leq \epsilon T + \underbrace{\qquad\qquad\qquad\qquad}_{T \text{ terms, each } O(\epsilon T)}$$

More general analysis

$$c(\mathbf{s}, \mathbf{a}) = \begin{cases} 0 \text{ if } \mathbf{a} = \pi^\star(\mathbf{s}) \\ 1 \text{ otherwise} \end{cases}$$

assume: $\pi_\theta(\mathbf{a} \neq \pi^\star(\mathbf{s})|\mathbf{s}) \leq \epsilon$

~~for all $\mathbf{s} \in \mathcal{D}_{\text{train}}$~~ for $\mathbf{s} \sim p_{\text{train}}(\mathbf{s})$

with DAgger, $p_{\text{train}}(\mathbf{s}) \rightarrow p_\theta(\mathbf{s})$

$$E\left[\sum_t c(\mathbf{s}_t, \mathbf{a}_t)\right] \leq \epsilon T$$

if $p_{\text{train}}(\mathbf{s}) \neq p_\theta(\mathbf{s})$:

$$p_\theta(\mathbf{s}_t) = \underbrace{(1-\epsilon)^t}_{\text{probability we made no mistakes}} p_{\text{train}}(\mathbf{s}_t) + (1 - (1-\epsilon)^t) \underbrace{p_{\text{mistake}}(\mathbf{s}_t)}_{\text{some } \textit{other} \text{ distribution}}$$

$$|p_\theta(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)| = (1 - (1-\epsilon)^t)|p_{\text{mistake}}(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)| \leq 2(1 - (1-\epsilon)^t)$$

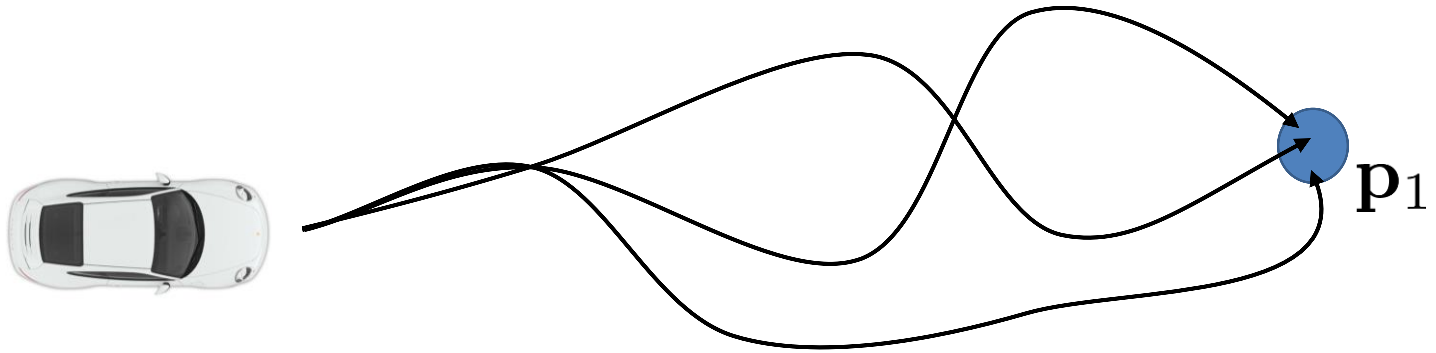
useful identity: $(1-\epsilon)^t \geq 1 - \epsilon t$ for $\epsilon \in [0,1]$ $\qquad \leq 2\epsilon t$

$$\sum_t E_{p_\theta(\mathbf{s}_t)}[c_t] = \sum_t \sum_{\mathbf{s}_t} p_\theta(\mathbf{s}_t) c_t(\mathbf{s}_t) \leq \sum_t \sum_{\mathbf{s}_t} p_{\text{train}}(\mathbf{s}_t) c_t(\mathbf{s}_t) + |p_\theta(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)| c_{\max}$$

$$\leq \sum_t \epsilon + 2\epsilon t \qquad O(\epsilon T^2)$$

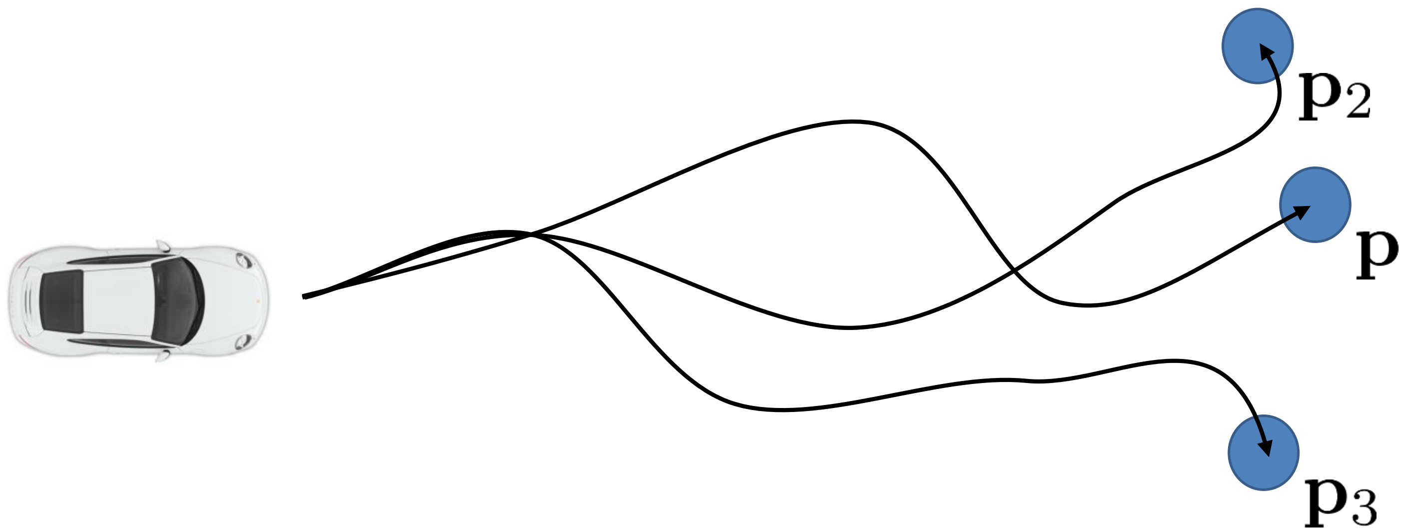
For more analysis, see Ross et al. "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning"

Another imitation idea



$\pi_\theta(\mathbf{a}|\mathbf{s})$

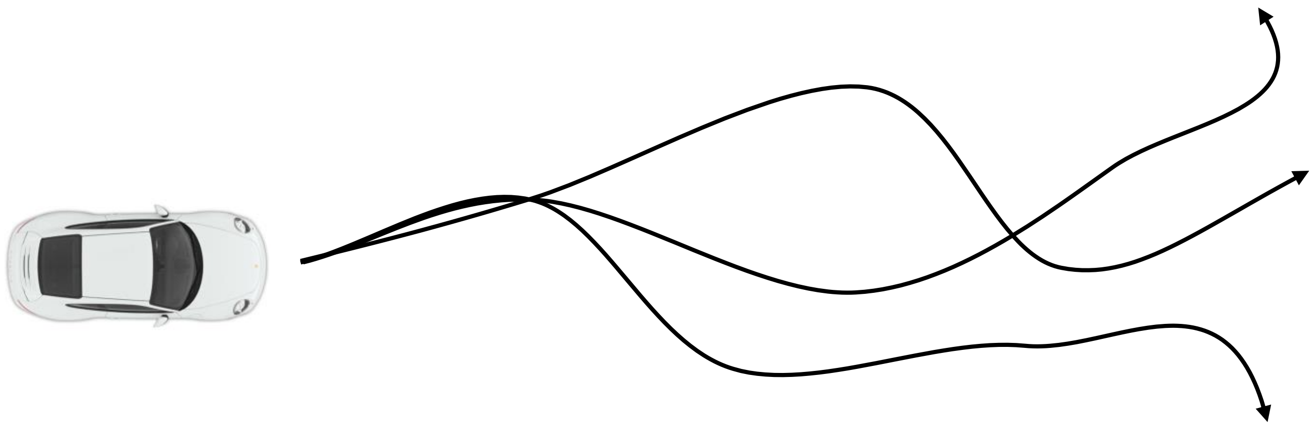
policy for reaching $\mathbf{p}_1$



$\pi_\theta(\mathbf{a}|\mathbf{s}, \mathbf{p})$

policy for reaching *any* $\mathbf{p}$

Goal-conditioned behavioral cloning



training time:

demo 1: $\{\mathbf{s}_1, \mathbf{a}_t, \ldots, \mathbf{s}_{T-1}, \mathbf{a}_{T-1}, \mathbf{s}_T\}$ ← successful demo for reaching $\mathbf{s}_T$

demo 2: $\{\mathbf{s}_1, \mathbf{a}_t, \ldots, \mathbf{s}_{T-1}, \mathbf{a}_{T-1}, \mathbf{s}_T\}$

demo 3: $\{\mathbf{s}_1, \mathbf{a}_t, \ldots, \mathbf{s}_{T-1}, \mathbf{a}_{T-1}, \mathbf{s}_T\}$

learn $\pi_\theta(\mathbf{a}|\mathbf{s}, \mathbf{g})$

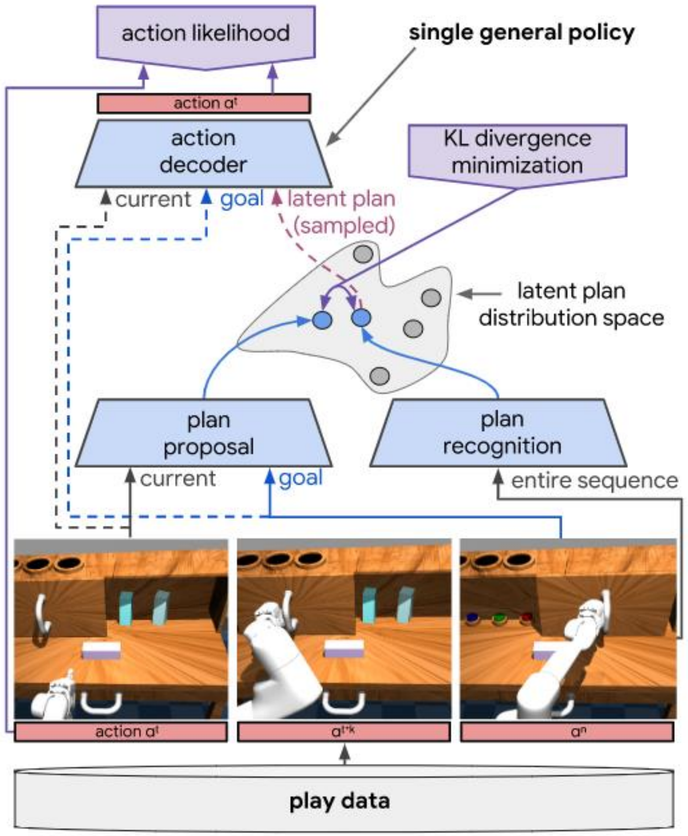
goal state (→ $\mathbf{g}$)

for each demo $\{\mathbf{s}_1^i, \mathbf{a}_1^i, \ldots, \mathbf{s}_{T-1}^i, \mathbf{a}_{T-1}^i, \mathbf{s}_T^i\}$

maximize $\log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i, \mathbf{g} = \mathbf{s}_T^i)$

Learning Latent Plans from Play

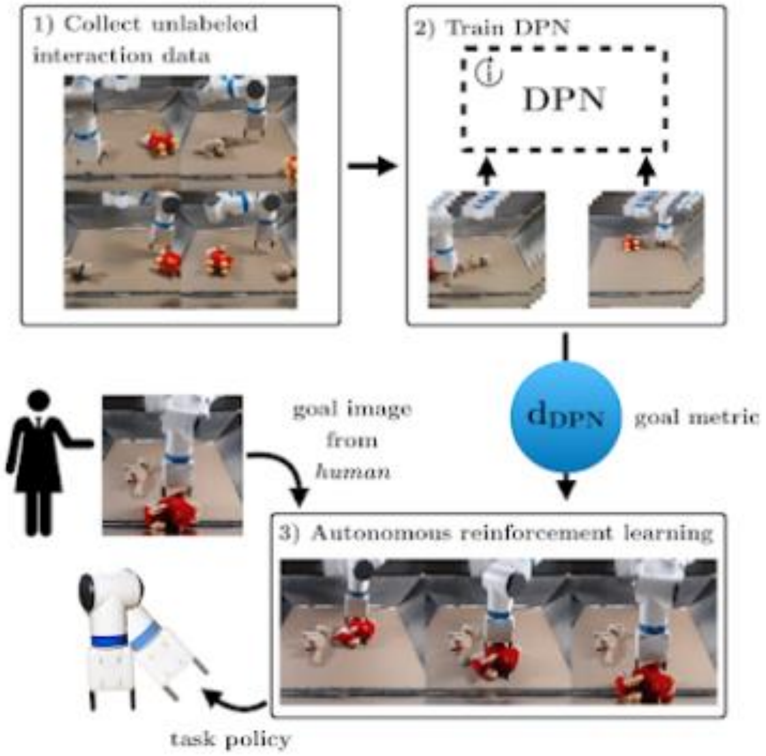
COREY LYNCH Google Brain
MOHI KHANSARI Google X
TED XIAO Google Brain
VIKASH KUMAR Google Brain
JONATHAN TOMPSON Google Brain
SERGEY LEVINE Google Brain
PIERRE SERMANET Google Brain



Unsupervised Visuomotor Control through Distributional Planning Networks

Tianhe Yu, Gleb Shevchuk, Dorsa Sadigh, Chelsea Finn

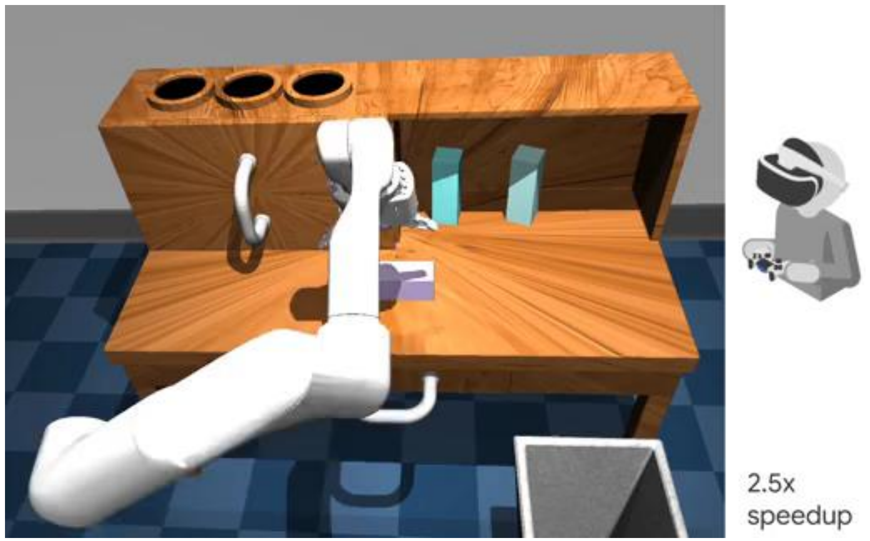
Stanford University



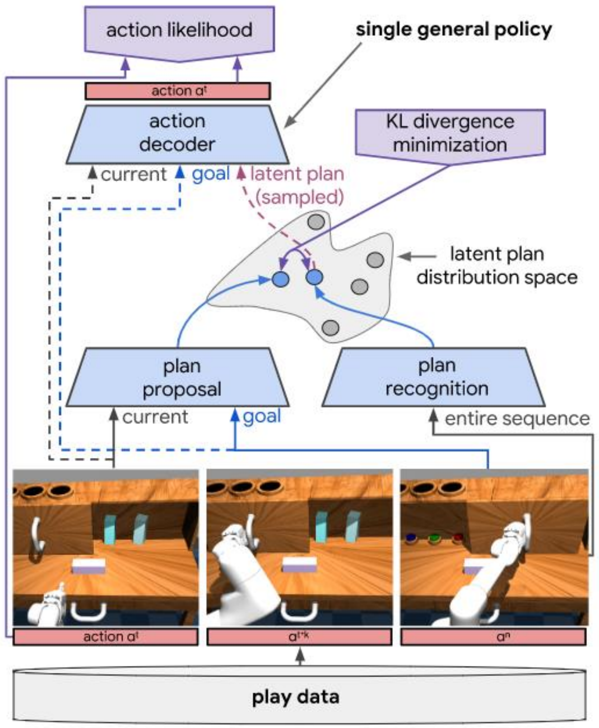
Learning Latent Plans from Play

COREY LYNCH Google Brain
MOHI KHANSARI Google X
TED XIAO Google Brain
VIKASH KUMAR Google Brain
JONATHAN TOMPSON Google Brain
SERGEY LEVINE Google Brain
PIERRE SERMANET Google Brain

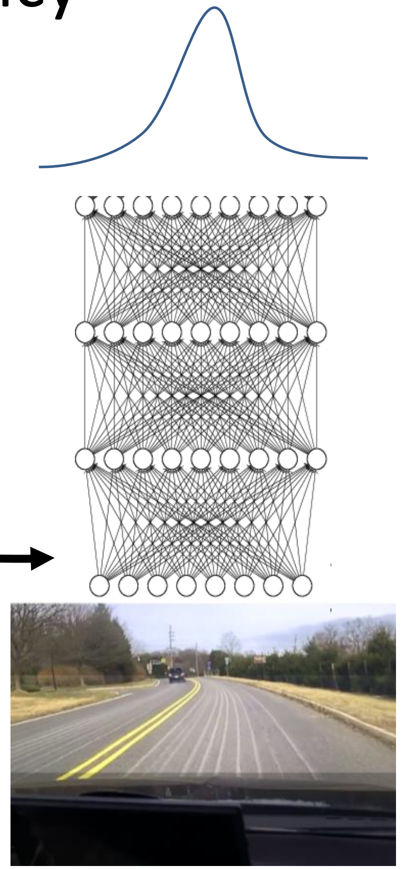
1. Collect **data**



2. Train **goal conditioned** policy



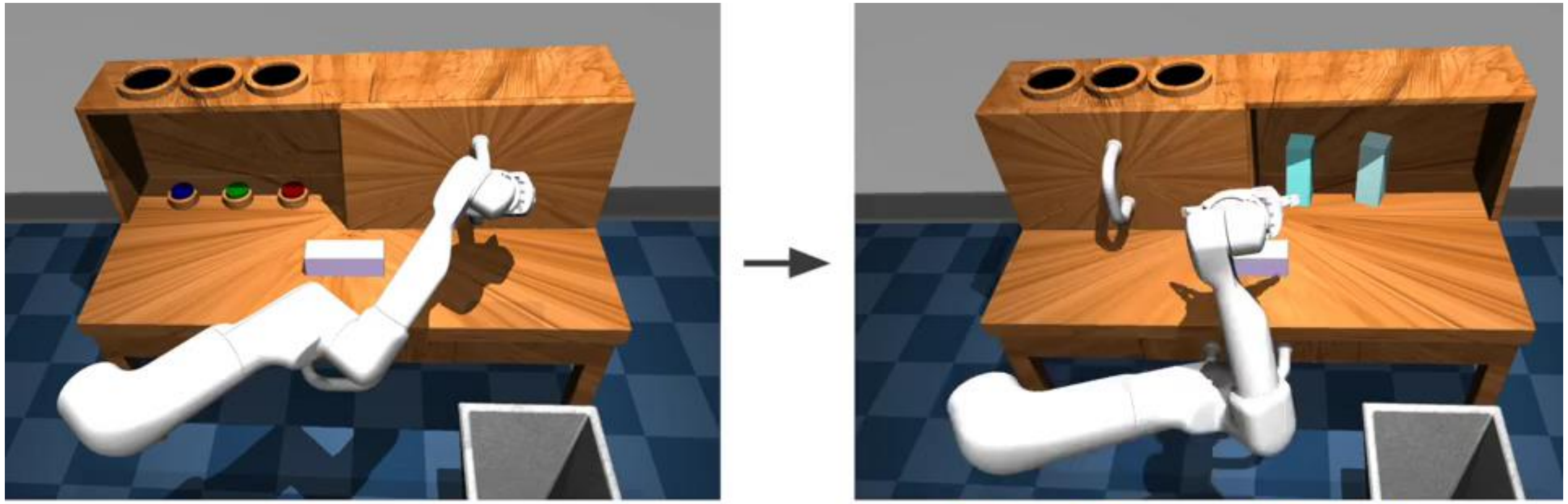
$$\xi \sim \mathcal{N}(0, \mathbf{I})$$



Learning Latent Plans from Play

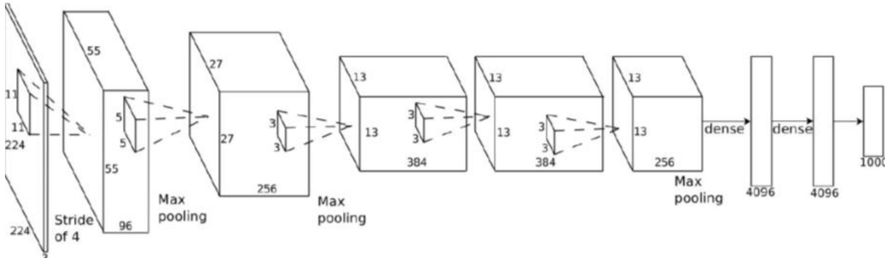
COREY LYNCH Google Brain
MOHI KHANSARI Google X
TED XIAO Google Brain
VIKASH KUMAR Google Brain
JONATHAN TOMPSON Google Brain
SERGEY LEVINE Google Brain
PIERRE SERMANET Google Brain

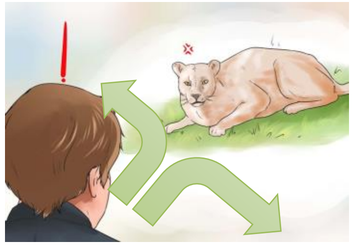
3. Reach goals



Goal → Single Play-LMP policy

Terminology & notation





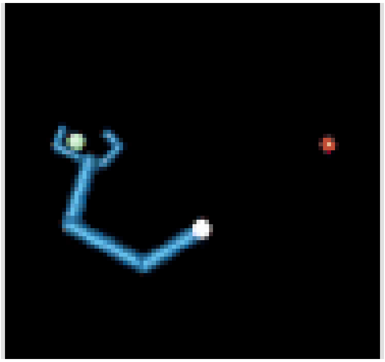
$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

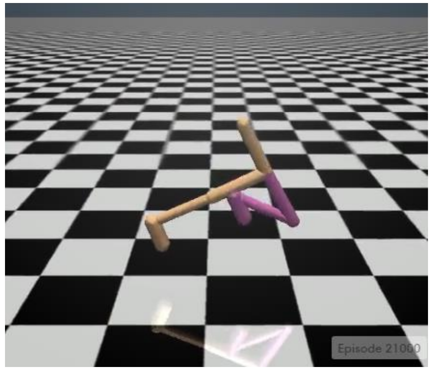
$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

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

$$\min_\theta E_{\mathbf{s}_{1:T}, \mathbf{a}_{1:T}} \left[\sum_t c(\mathbf{s}_t, \mathbf{a}_t) \right]$$

Cost/reward functions in theory and practice





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

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

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

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