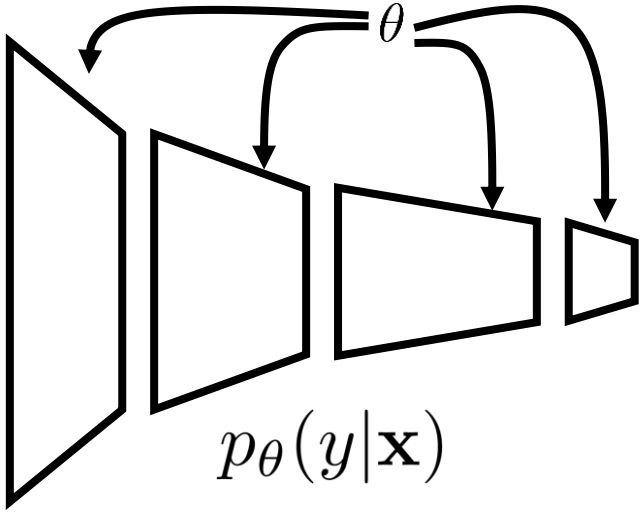
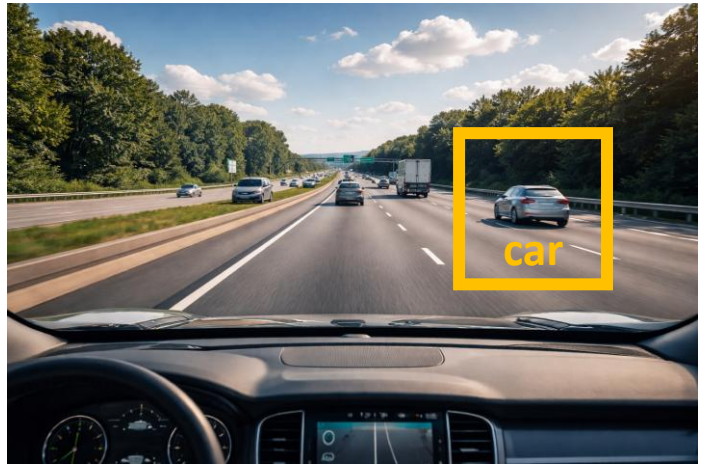
Supervised Learning of Behaviors

CS 185/285

Instructor: Sergey Levine
UC Berkeley

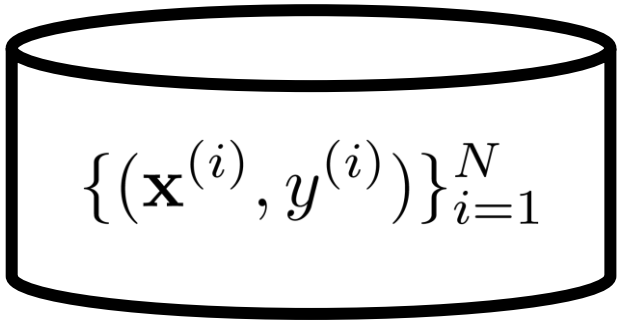
Part 1:
From supervised learning to imitation learning





$\mathbf{x}$

y

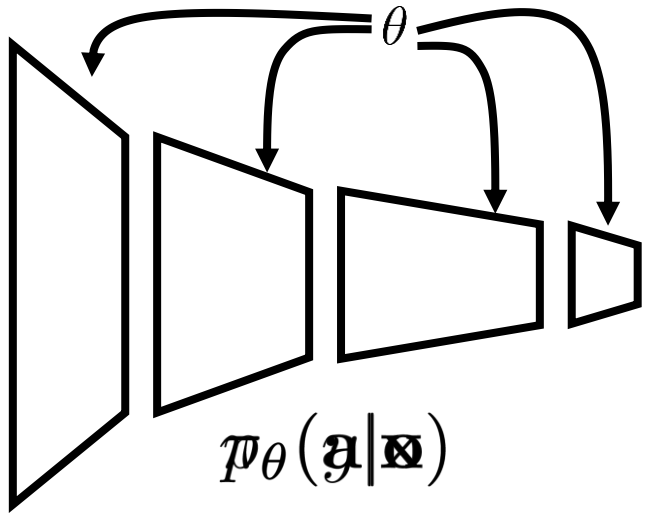


$$\arg\max_\theta \sum_{i=1}^{N} \log p_\theta(y|\mathbf{x})$$

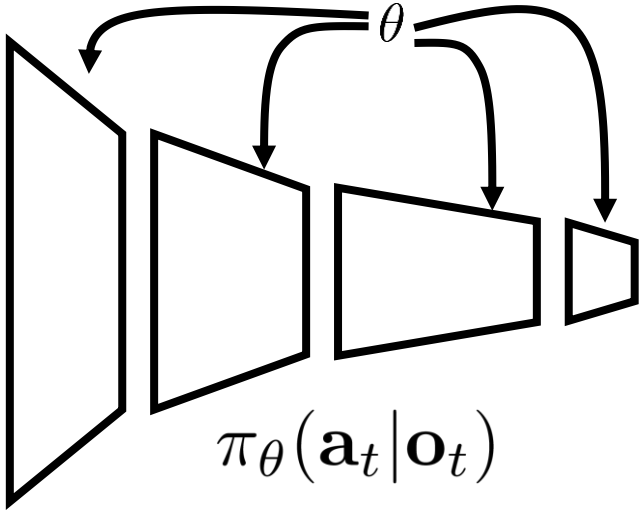
maximum likelihood estimation

a.k.a. supervised learning

$\mathbf{x}$



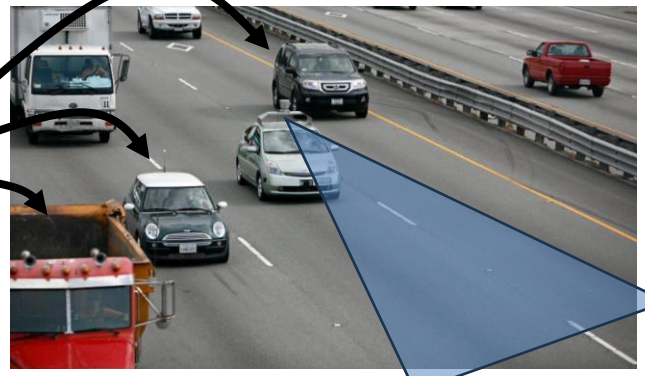
$\mathbf{y}$



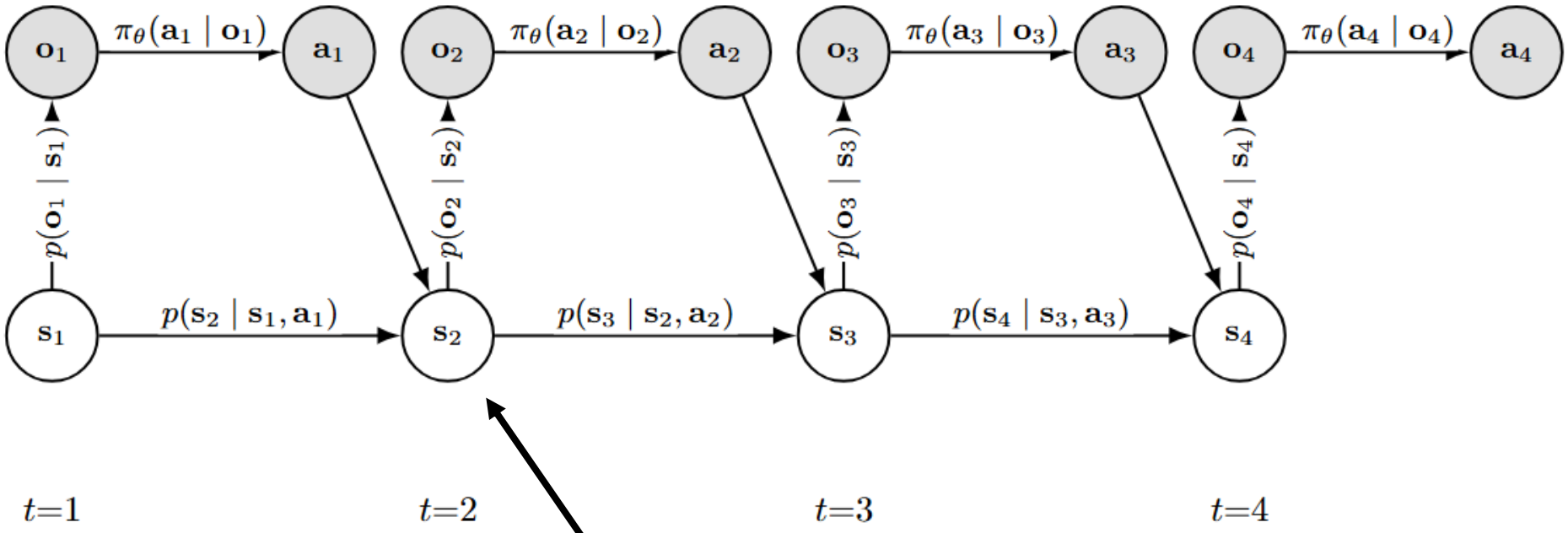
$\mathbf{o}_t$

$\mathbf{a}_t$

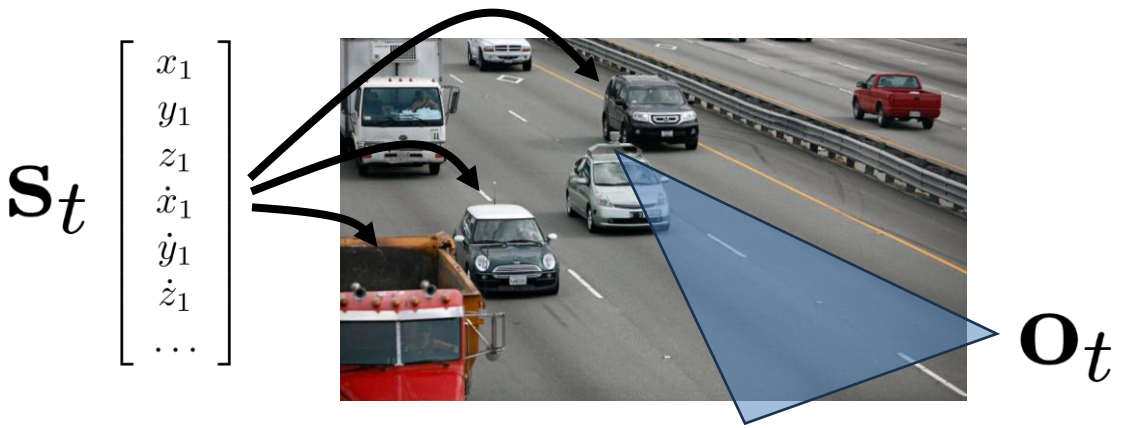
$$\mathbf{s}_t \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \dot{x}_1 \\ \dot{y}_1 \\ \dot{z}_1 \\ \dots \end{bmatrix}$$



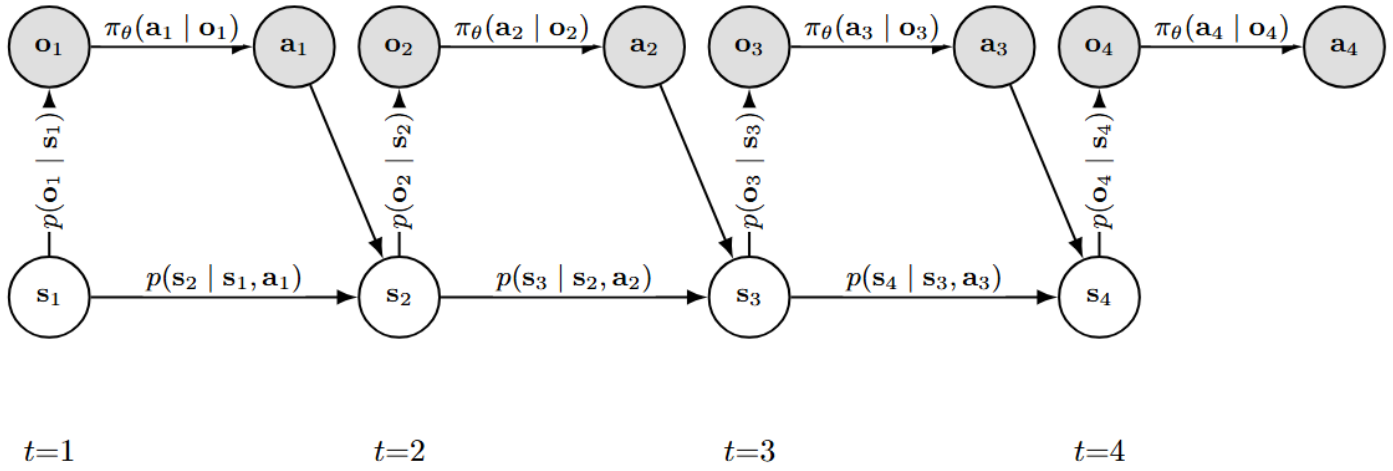
$\mathbf{o}_t$



Markov property: $\mathbf{s}_{t+1} \perp \mathbf{s}_{t-1} | \mathbf{s}_t$



Partially observed case



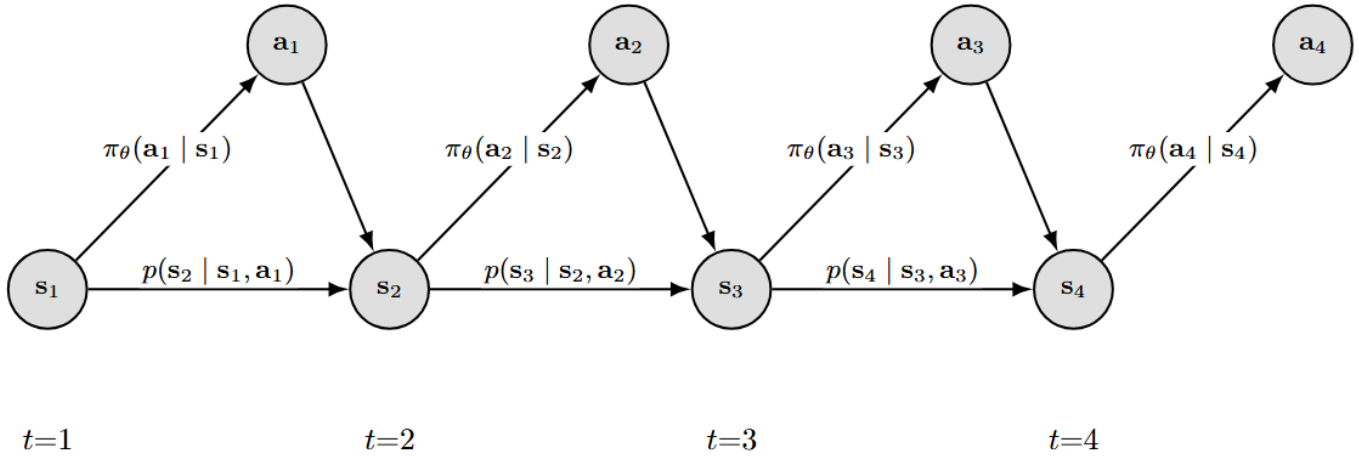
$$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$$

$$p(\mathbf{o}_t|\mathbf{s}_t)$$

$$p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$$

} typically don't need to know this

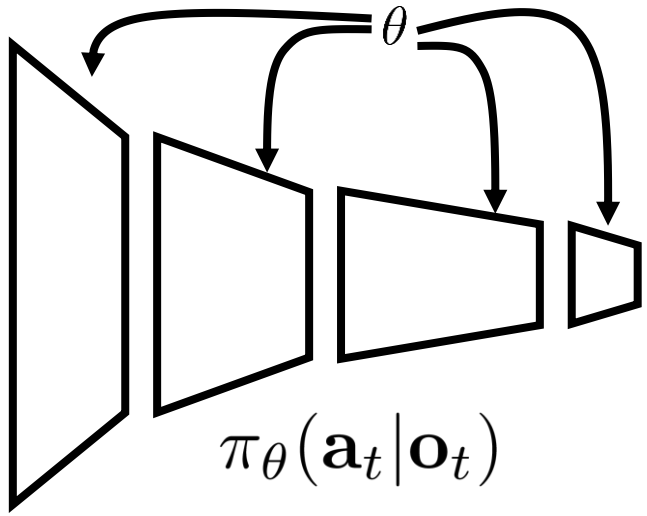
Fully observed case $\mathbf{o}_t = \mathbf{s}_t$



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

$$p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$$

$\mathbf{o}_t$



$\mathbf{a}_t$

Data? $\{(\mathbf{o}_1^{(i)}, \mathbf{a}_1^{(i)}, ..., \mathbf{o}_H^{(i)}, \mathbf{a}_H^{(i)})\}_{i=1}^N$

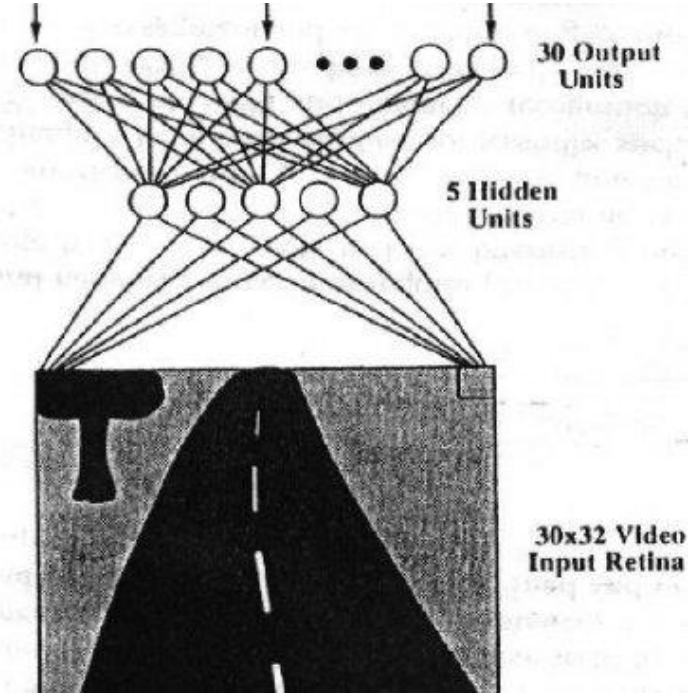
"demonstration trajectory"

$$\arg\max_\theta \sum_{i=1}^{N} \sum_{t=1}^{H} \log \pi_\theta(\mathbf{a}_t^{(i)}|\mathbf{o}_t^{(i)})$$

"behavioral cloning"

Part 2:
The behavioral cloning algorithm

ALVINN: **A**utonomous **L**and **V**ehicle **I**n a **N**eural **N**etwork 1989



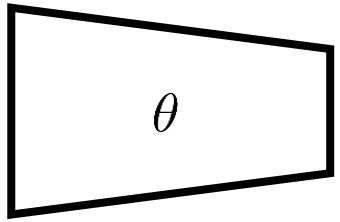
1. Collect data by asking a person to make demonstrations

$$\{(\mathbf{o}_1^{(i)}, \mathbf{a}_1^{(i)}, \ldots, \mathbf{o}_H^{(i)}, \mathbf{a}_H^{(i)})\}_{i=1}^N$$

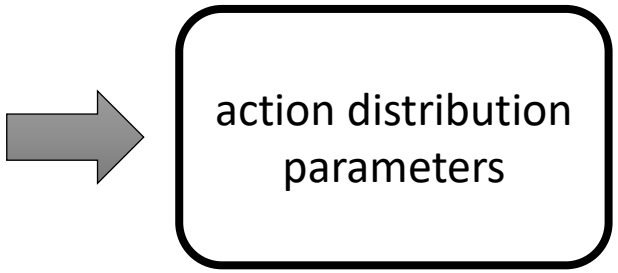
2. Run supervised learning (a.k.a. maximum likelihood estimation) on this data with your favorite neural network

$$\arg\max_\theta \sum_{i=1}^{N} \sum_{t=1}^{H} \log \pi_\theta(\mathbf{a}_t^{(i)} | \mathbf{o}_t^{(i)})$$

$\mathbf{o}_t$



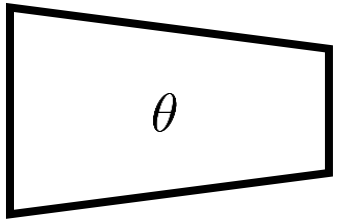
encoder
(e.g., ViT, ResNet)



sample

$\mathbf{a}_t$

$\mathbf{o}_t$



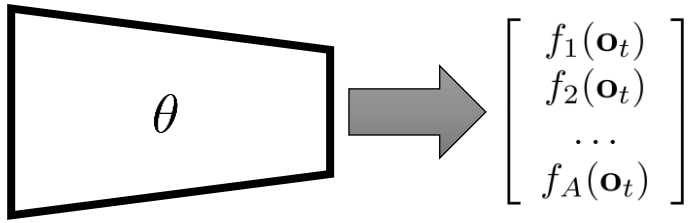
encoder
(e.g., ViT, ResNet)

action distribution parameters

sample

$\mathbf{a}_t$

Discrete actions
(e.g., tokens for LLM,
key presses for Atari games)

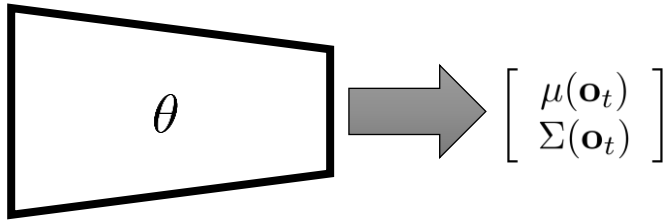


output = logits

$$p(a_t = 1|\mathbf{o}_t) = \frac{\exp(f_1(\mathbf{o}_t))}{\sum_{i=1}^{A} \exp(f_i(\mathbf{o}_t))}$$

We'll learn about other more advanced ways to represent complex action distributions later!

Continuous actions
(e.g., driving a car)



output = distribution parameters

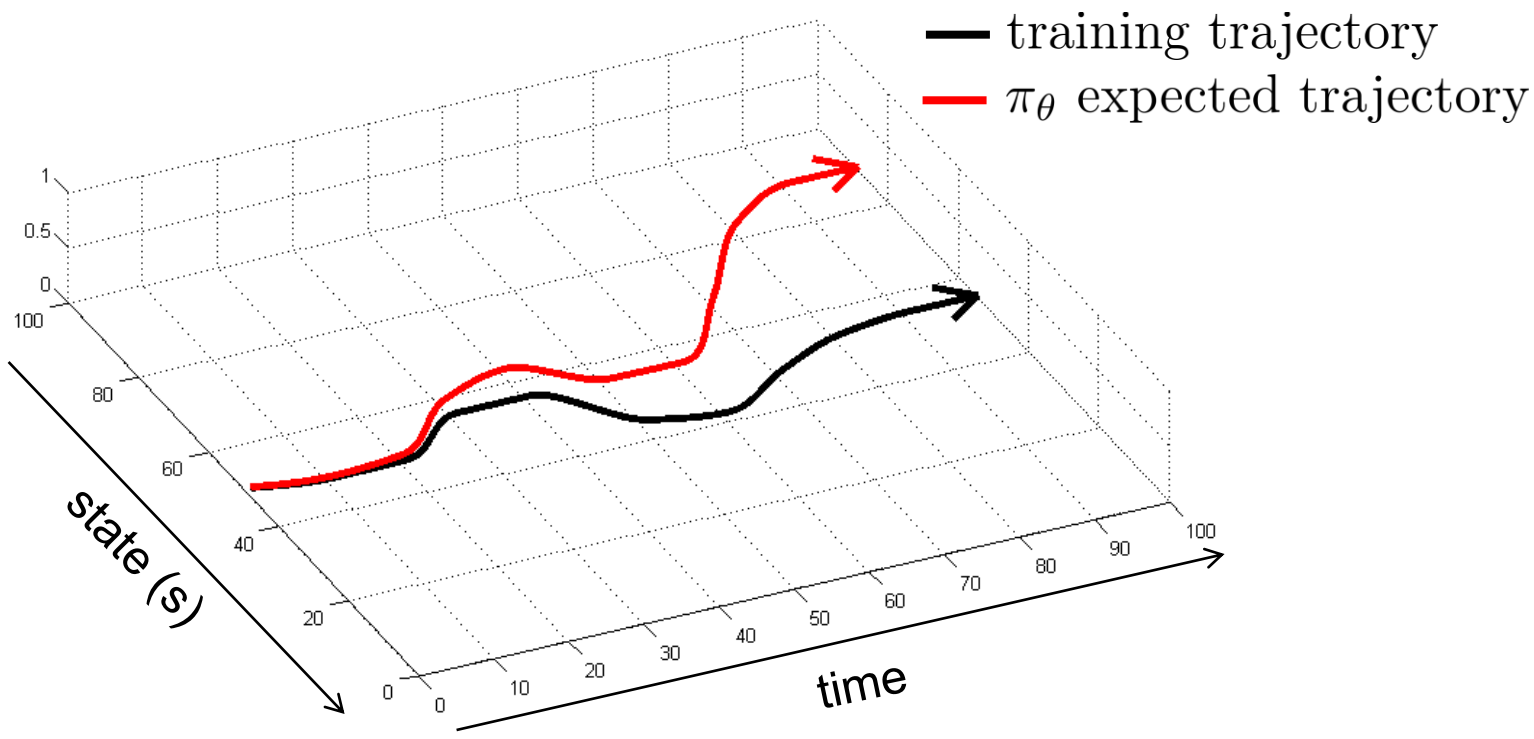
$$p(\mathbf{a}_t|\mathbf{o}_t) = \mathcal{N}(\mathbf{a}_t|\mu(\mathbf{o}_t), \Sigma(\mathbf{o}_t))$$

$$\log p(\mathbf{a}_t|\mathbf{o}_t) = \|\mathbf{a}_t - \mu(\mathbf{o}_t)\|^2$$
$$\text{in special case } \Sigma(\mathbf{o}_t) = \mathbf{I}$$

Part 3:
Does behavioral cloning work?

Does it work?

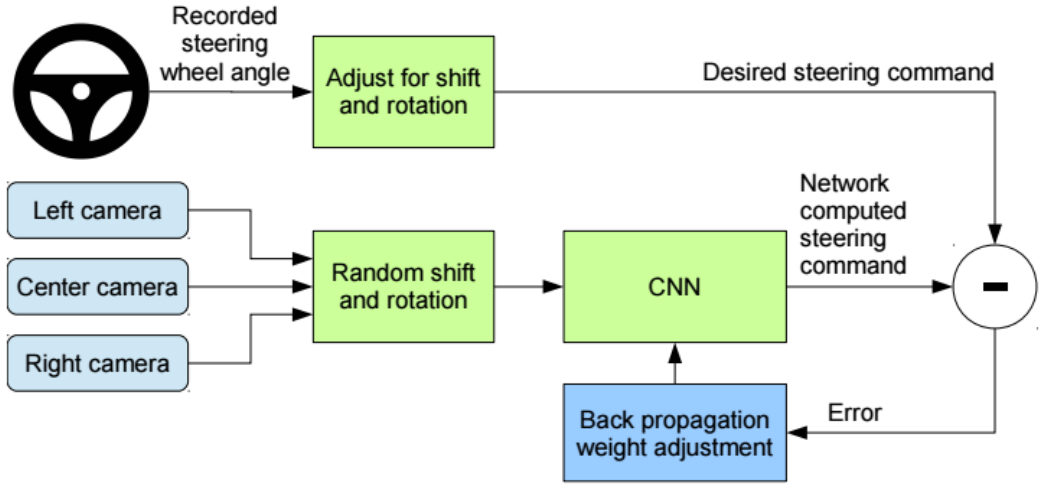
No!

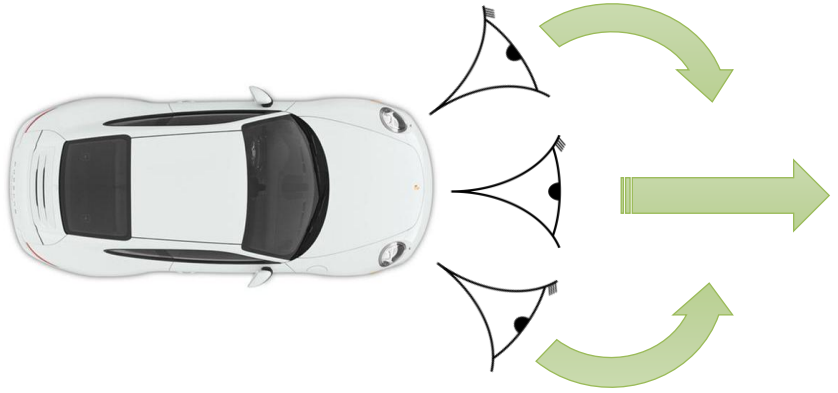


Does it work? Yes!

Video: Bojarski et al. ‘16, NVIDIA

Why did that work?

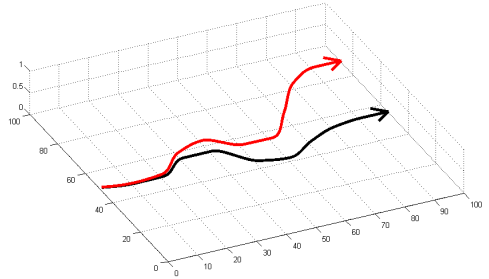


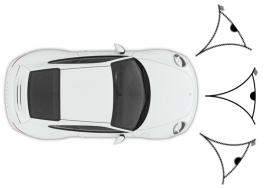


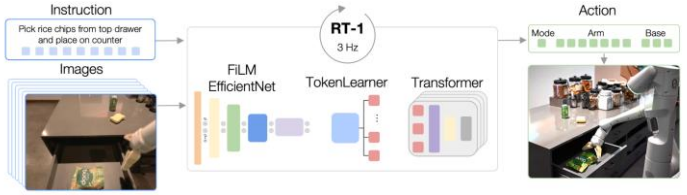
Bojarski et al. ‘16, NVIDIA

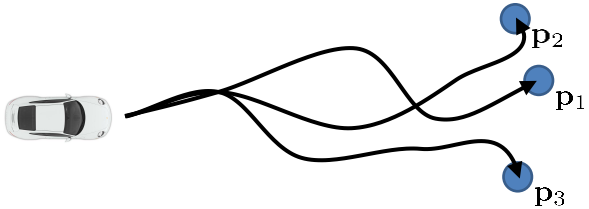
The moral of the story, and a list of ideas

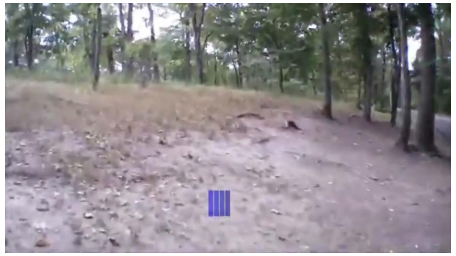
- Imitation learning via behavioral cloning is not guaranteed to work
  - This is **different** from supervised learning
  - The reason: i.i.d. assumption does not hold!
- We can formalize **why** this is and do a bit of theory
- We can address the problem in a few ways:
  - Change the algorithm (DAgger)
  - Use very powerful models that make very few mistakes
  - Be smart about how we collect (and augment) our data
  - Use multi-task learning



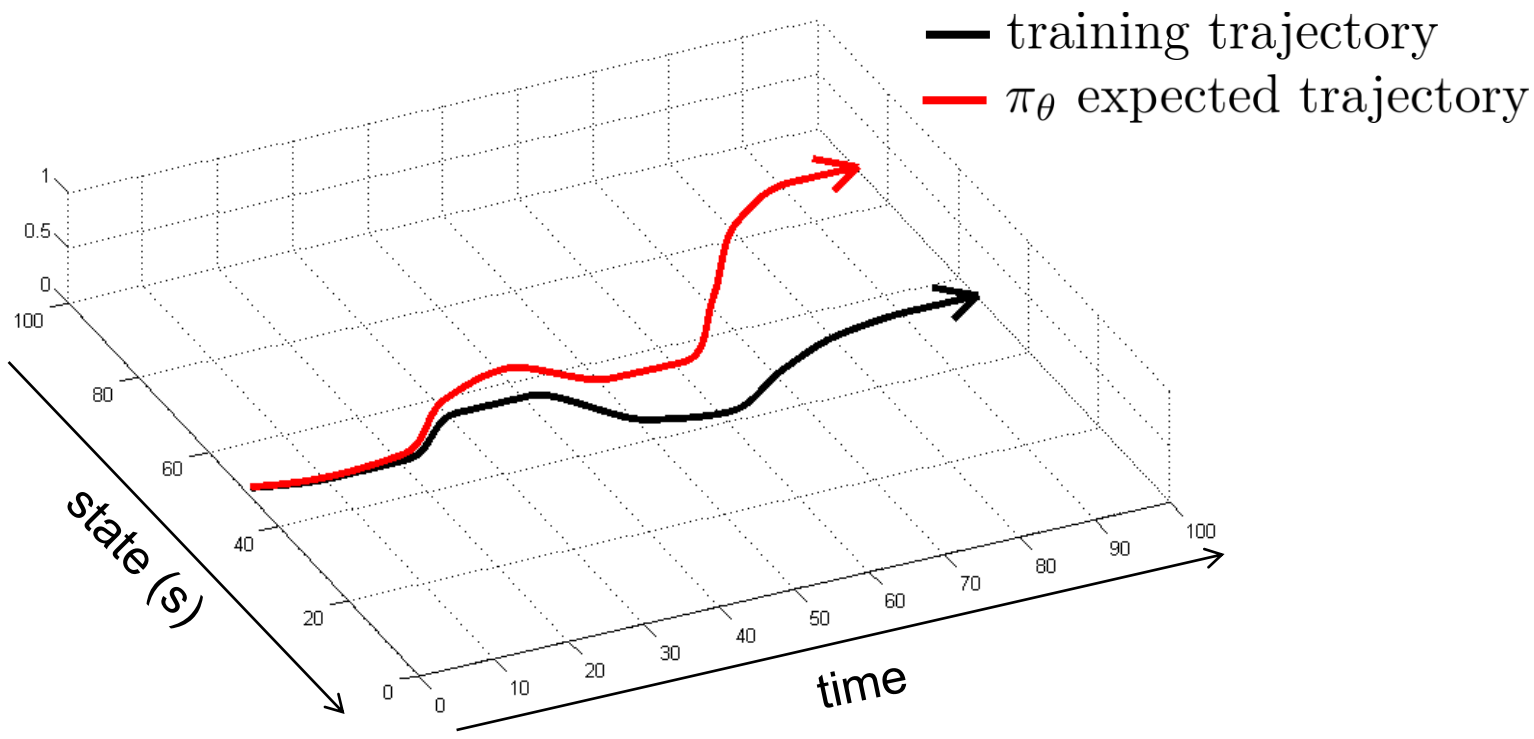








Back to this…


training trajectory
π_θ expected trajectory
state (s)
time
1
0.5
0
100
80
60
40
20
0
0
10
20
30
40
50
60
70
80
90
100

Distributional shift

Definition: we train $p_\theta(y|\mathbf{x})$ on $\mathbf{x} \sim p_{\text{train}}(\mathbf{x})$

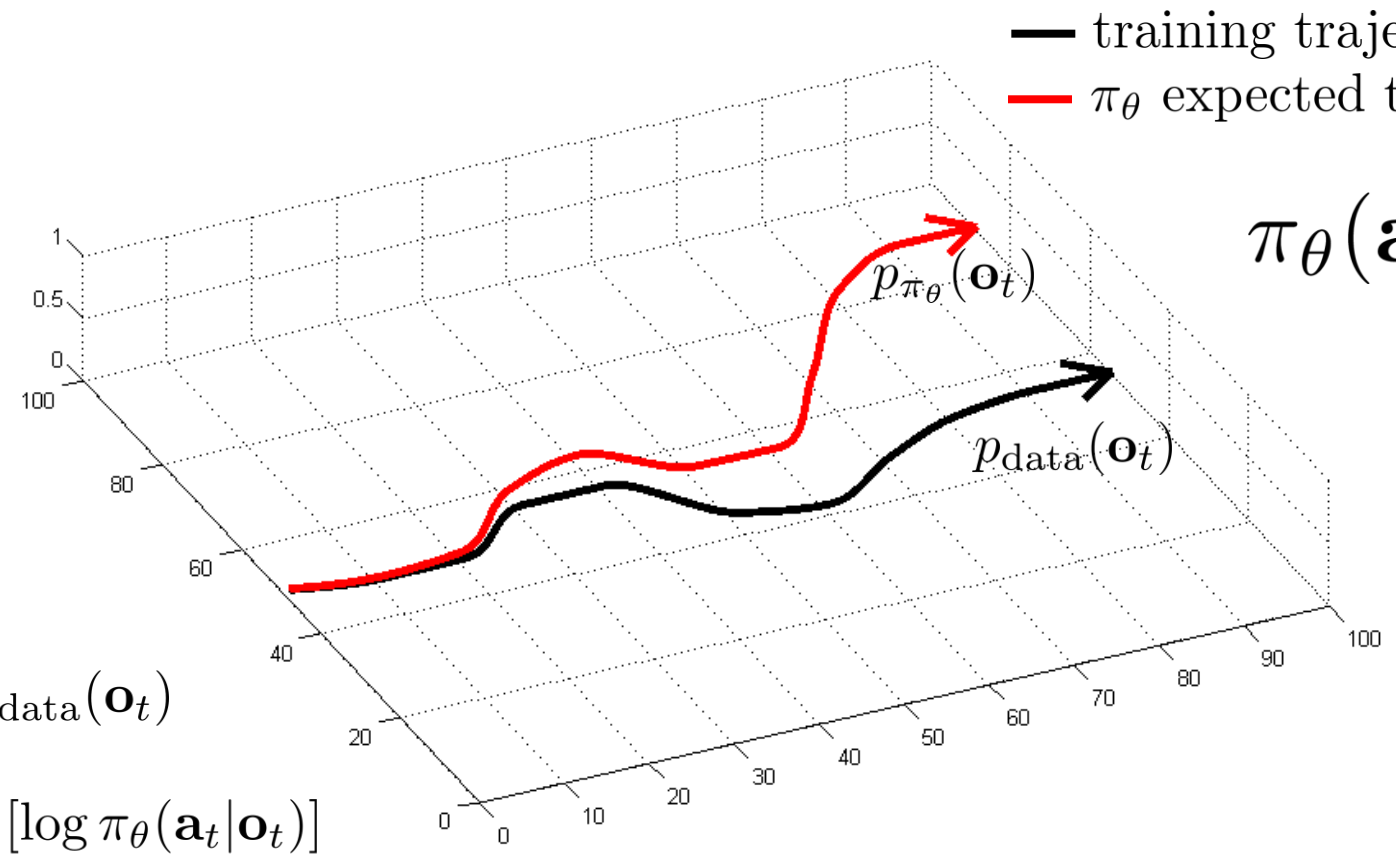
we test $p_\theta(y|\mathbf{x})$ on $\mathbf{x} \sim p_{\text{test}}(\mathbf{x})$

$$p_{\text{test}}(\mathbf{x}) \neq p_{\text{train}}(\mathbf{x})$$

How bad could it be?

Imagine you prepare for a math exam, and get an exam on ancient Greek literature

Distributional shift with BC



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

we test under $p_{\pi_\theta}(\mathbf{o}_t)$

$\max_\theta E_{\mathbf{o}_t \sim p_{\text{data}}(\mathbf{o}_t)}[\log \pi_\theta(\mathbf{a}_t|\mathbf{o}_t)]$

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

How bad could it be?

Intermission

Part 4:
Just how bad could it be, really?

Some things we might like to know

Is it worse than supervised learning?

Do the problems vanish with more data?

Does it get worse for some control problems?

Theory can help us answer these questions

What theory **gives** us:

intuition for tradeoffs (data, horizon, etc.)

“how bad can it be” (worst case)

What theory **doesn’t** give us:

a practical “proof” that it works (or doesn’t)

a way to understand *typical* behavior

Quantifying how bad it could be

Assume the data is produced by a good policy $\pi^\star(\mathbf{s}_t)$ (fully observed case for simplicity)

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

incur a cost of 1 if you make a "mistake"

$$\mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)] = \sum_{\mathbf{a}_t} \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) c(\mathbf{s}_t, \mathbf{a}_t) = \pi_\theta(\mathbf{a}_t \neq \pi^\star(\mathbf{s}_t)|\mathbf{s}_t)$$

"probability of a mistake"

expected value w.r.t. actions sampled from π_θ

Quantifying how bad it could be

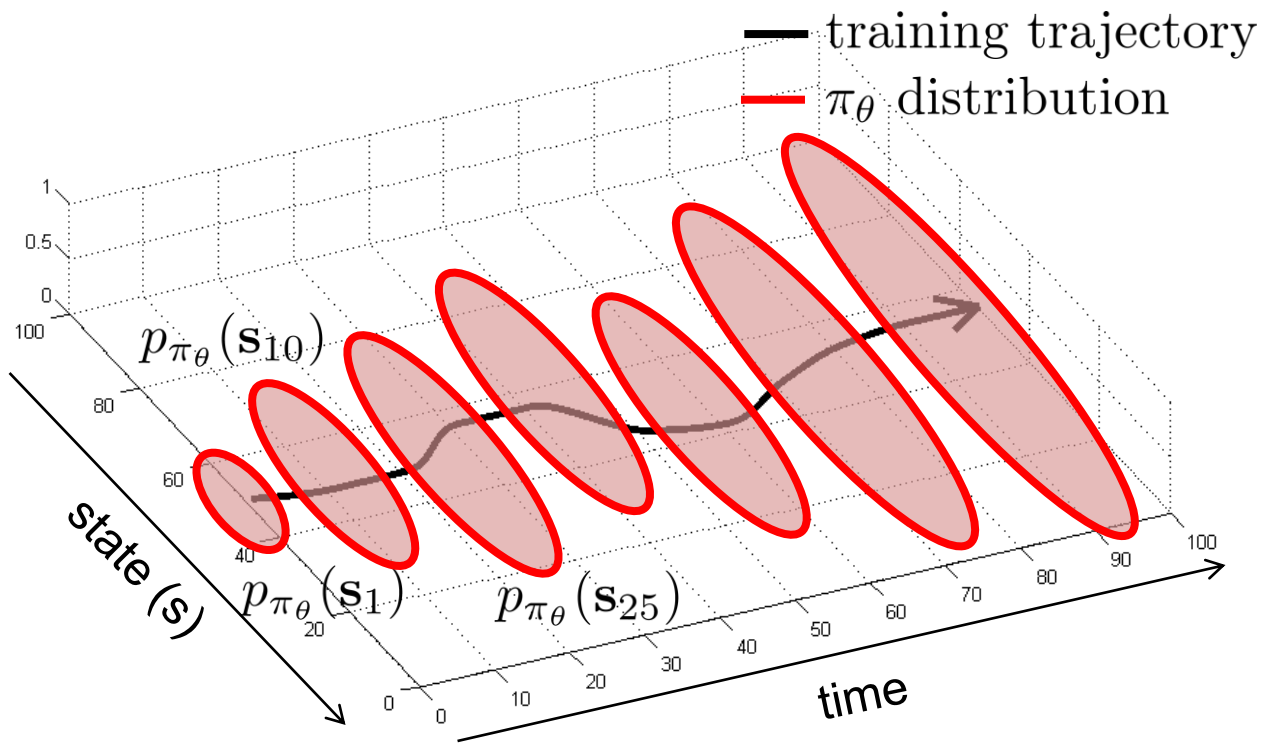
$$\sum_{t=1}^{H} \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t), \mathbf{s}_t \sim p_{\pi_\theta}(\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)]$$

note that this depends on past actions

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

This sum corresponds to the *expected total number of mistakes*

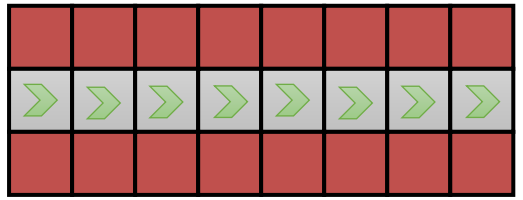
Question: how does the total number of mistakes scale with the length of the trajectory (the "horizon")



A worst-case situation

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

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



$$\sum_{t=1}^{H} \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t), \mathbf{s}_t \sim p_{\pi_\theta}(\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)] \leq \underbrace{\epsilon H}_{H \text{ terms, each } O(\epsilon H)}$$

$O(\epsilon H^2)$

A more general analysis

assume: $\mathbb{E}_{\mathbf{s}_t \sim p_{\text{train}}(\mathbf{s}_t)}[\pi_\theta(\mathbf{a}_t \neq \pi^\star(\mathbf{s}_t)|\mathbf{s}_t)] \leq \epsilon$

"if we sample a state from the training distribution, we are unlikely to make a mistake"

A useful property:

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

A more general analysis

Total variation divergence $D_{\mathrm{TV}}(p, q) = \frac{1}{2} \sum_x |p(x) - q(x)| \leq 1$

$$p_{\pi_\theta}(\mathbf{s}_t) = (1-\epsilon)^t p_{\mathrm{train}}(\mathbf{s}_t) + (1 - (1-\epsilon)^t) p_{\mathrm{mistake}}(\mathbf{s}_t)$$

$$D_{\mathrm{TV}}(p_{\mathrm{train}}, p_{\pi_\theta}) = \frac{1}{2} \sum_{\mathbf{s}_t} |p_{\mathrm{train}}(\mathbf{s}_t) - p_{\pi_\theta}(\mathbf{s}_t)| = \frac{1}{2} \sum_{\mathbf{s}_t} |p_{\mathrm{train}}(\mathbf{s}_t) - (1-\epsilon)^t p_{\mathrm{train}}(\mathbf{s}_t) - (1 - (1-\epsilon)^t) p_{\mathrm{mistake}}(\mathbf{s}_t)|$$

$$(1-\epsilon)^t p_{\mathrm{train}}(\mathbf{s}_t) + (1 - (1-\epsilon)^t) p_{\mathrm{train}}(\mathbf{s}_t)$$

$$= (1 - (1-\epsilon)^t) \frac{1}{2} \sum_{\mathbf{s}_t} |p_{\mathrm{train}}(\mathbf{s}_t) - p_{\mathrm{mistake}}(\mathbf{s}_t)|$$

$$\leq (1 - (1-\epsilon)^t) \leq \epsilon t$$

useful identity:

$(1-\epsilon)^t \geq 1 - \epsilon t$ for $\epsilon \in [0, 1]$

A more general analysis

$$\sum_{t=1}^{H} \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t), \mathbf{s}_t \sim p_{\pi_\theta}(\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)]$$

$$= \sum_{t=1}^{H} \sum_{\mathbf{s}_t} p_{\pi_\theta}(\mathbf{s}_t) \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)]$$

$$= \sum_{t=1}^{H} \sum_{\mathbf{s}_t} (p_{\text{train}}(\mathbf{s}_t) + p_{\pi_\theta}(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)) \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)]$$

$$\leq \sum_{t=1}^{H} \sum_{\mathbf{s}_t} (p_{\text{train}}(\mathbf{s}_t) + |p_{\pi_\theta}(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)|) \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)]$$

$$= \sum_{t=1}^{H} \left[\sum_{\mathbf{s}_t} p_{\text{train}}(\mathbf{s}_t) \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)] \right] + \left[\sum_{\mathbf{s}_t} |p_{\pi_\theta}(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)| \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)] \right]$$

A more general analysis

$$\sum_{t=1}^{H} \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t), \mathbf{s}_t \sim p_{\pi_\theta}(\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)]$$

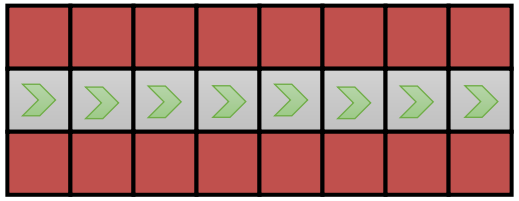
$$= \sum_{t=1}^{H} \Bigg[\underbrace{\sum_{\mathbf{s}_t} p_{\text{train}}(\mathbf{s}_t) \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)]}_{\mathbb{E}_{\mathbf{s}_t \sim p_{\text{train}}(\mathbf{s}_t)}[\pi_\theta(\mathbf{a}_t \neq \pi^\star(\mathbf{s}_t)|\mathbf{s}_t)] \leq \epsilon} \Bigg] + \Bigg[\underbrace{\sum_{\mathbf{s}_t} |p_{\pi_\theta}(\mathbf{s}_t) - p_{\text{train}}(\mathbf{s}_t)|}_{= 2D_{\text{TV}}(p_{\pi_\theta}, p_{\text{train}}) \leq 2\epsilon t} \underbrace{\mathbb{E}_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}[c(\mathbf{s}_t, \mathbf{a}_t)]}_{\leq 1} \Bigg]$$

$$\leq \sum_{t=1}^{H} \epsilon + 2\epsilon t \qquad O(\epsilon H^2)$$

This is bad because error increases *quadratically* with horizon

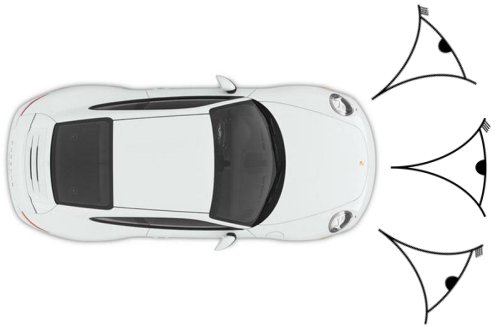
For more analysis, see Ross et al. “A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning”

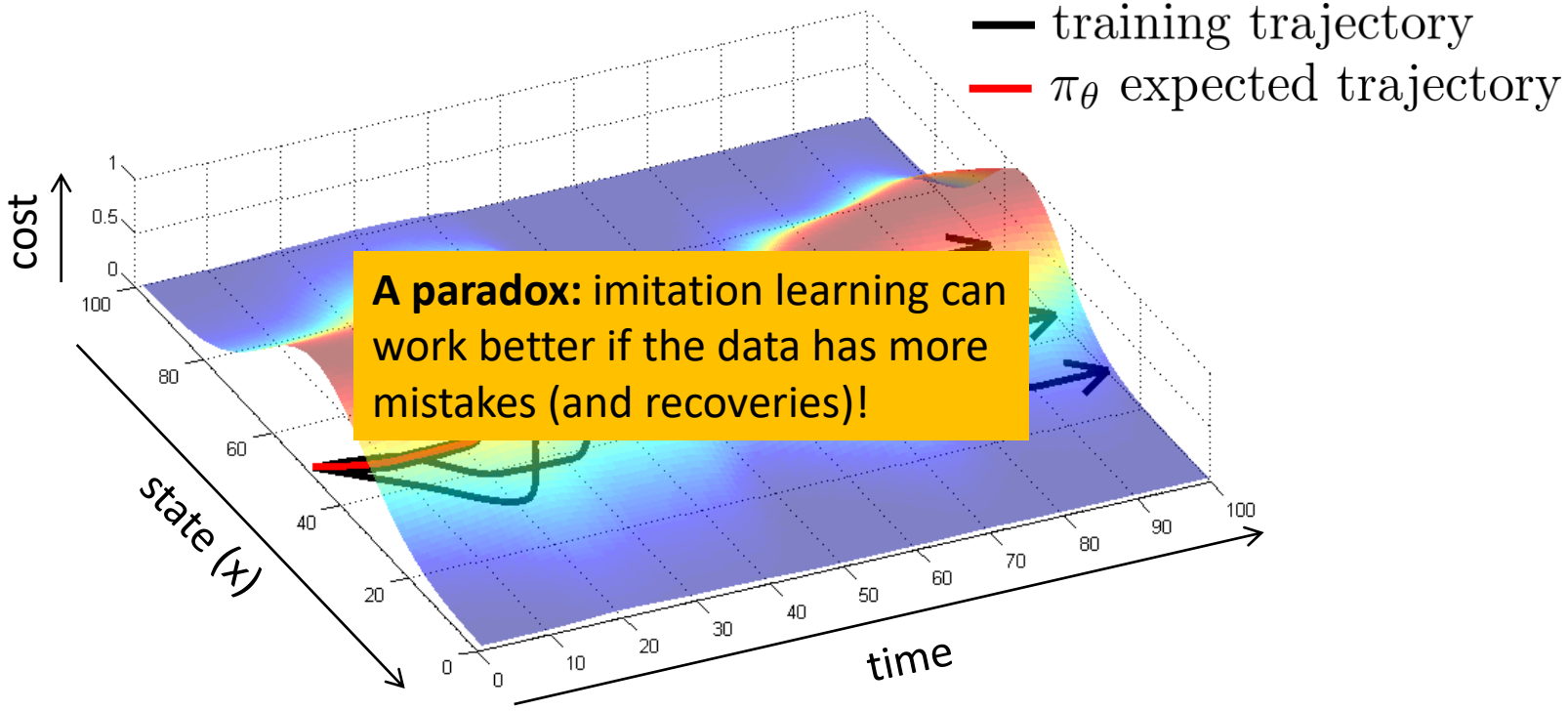
Why is this rather **pessimistic**?



In reality, we can often **recover** from mistakes

But that doesn't mean that **imitation learning** will allow us to learn how to do that!

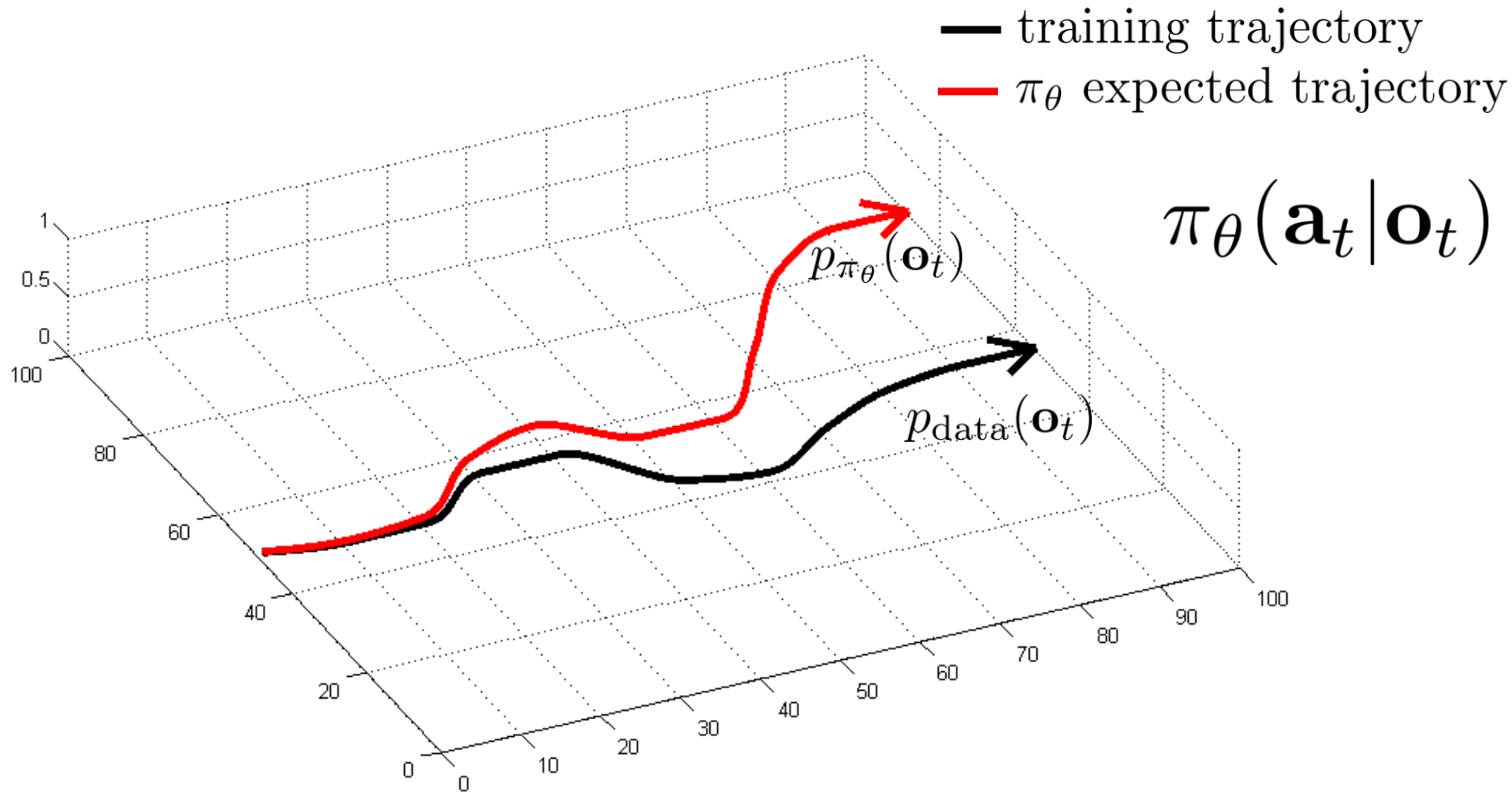




A paradox: imitation learning can work better if the data has more mistakes (and recoveries)!

Part 5:
Can we do *better* than behavioral cloning?

What can we do about it?



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

Fixing distributional shift

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

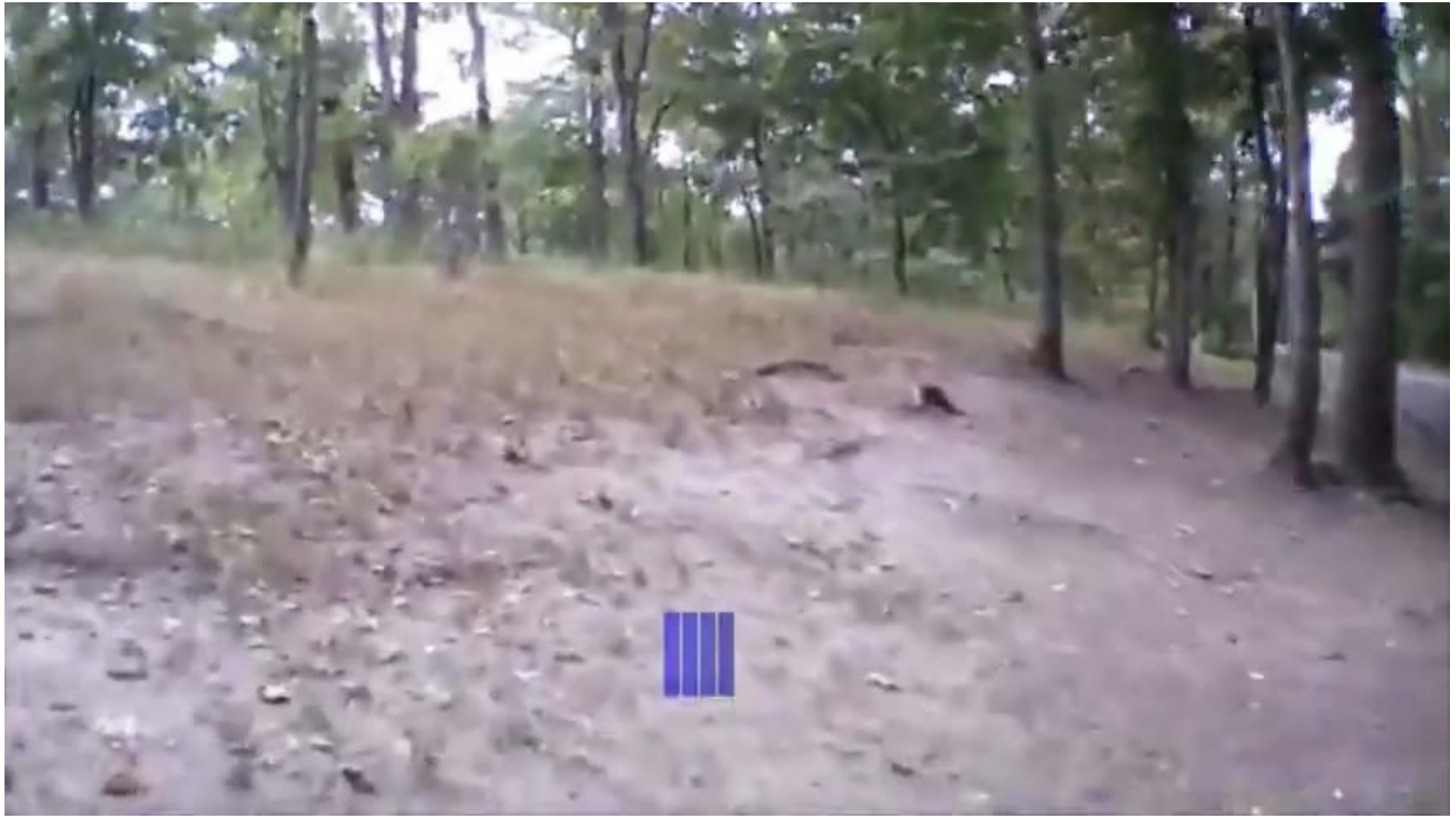
how? just run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

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

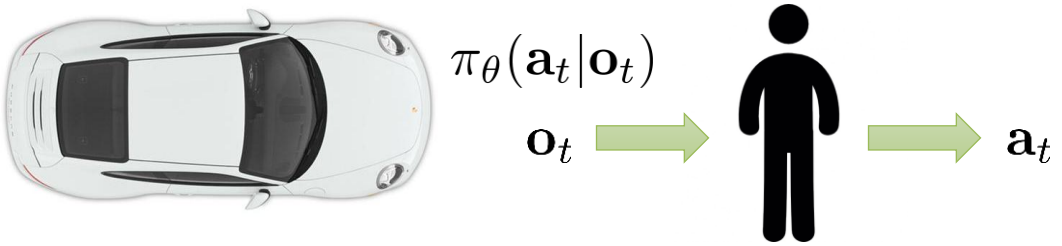
DAgger Example



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. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{a}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

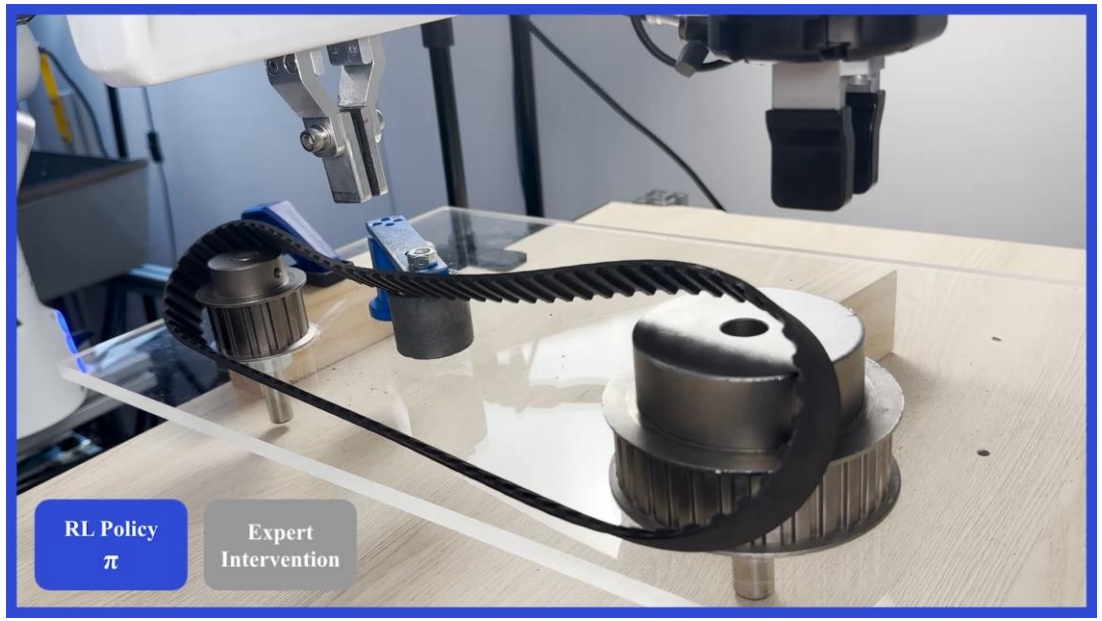


Ross et al. '11

A common variant of DAgger

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 $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$
3. ask human to *take over* at some time step t
4. store all $(\mathbf{o}_t, \mathbf{u}_t)$ examples from human intervention

Is this still guaranteed to fix distributional shift?



Recap

- Imitation learning via behavioral cloning is not guaranteed to work
  - This is **different** from supervised learning
  - The reason: i.i.d. assumption does not hold!
- We can formalize **why** this is and do a bit of theory
- We can address the problem in a few ways:
  - Change the algorithm (DAgger)
  - Use very powerful models that make very few mistakes
  - Be smart about how we collect (and augment) our data
  - Use multi-task learning