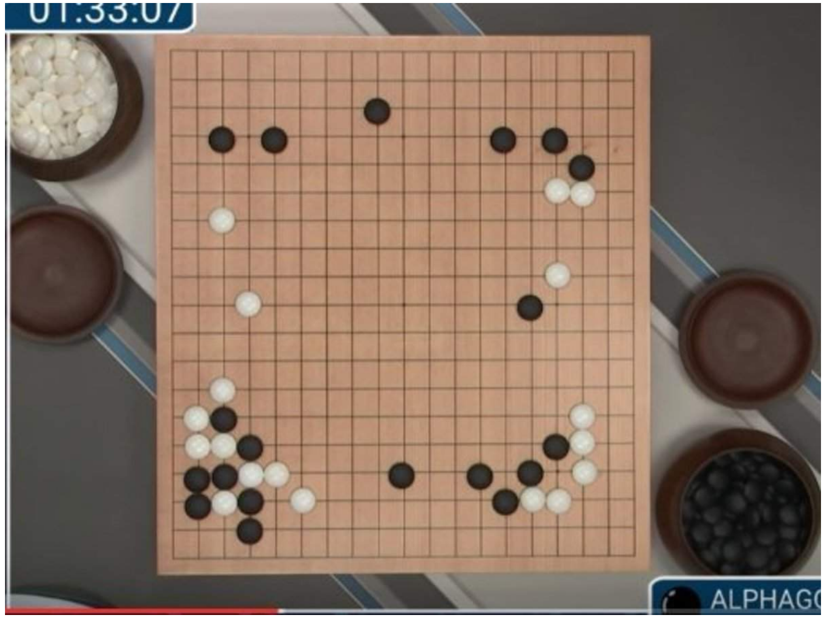
Reinforcement Learning
Autumn 2024

Abhishek Gupta

TA: Jacob Berg



Lecture outline

Recap: Imitation Learning + Why it is hard

↓

Multimodality and Underfitting in Imitation

↓

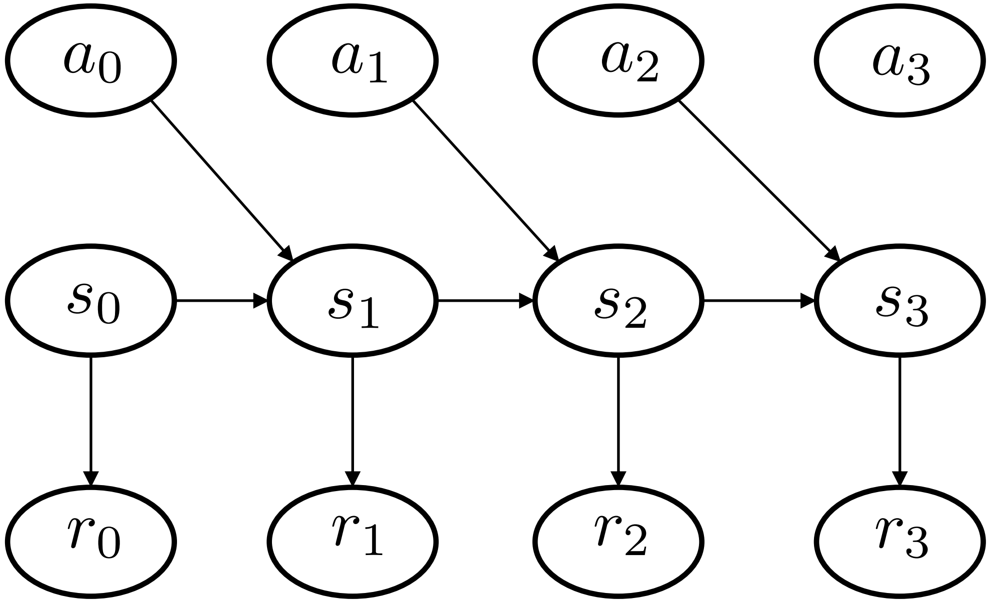
Compounding Error in Imitation

↓

Frontiers in Imitation

Framework for RL - Markov Decision Process

Augment Markov chain with rewards and actions



States: $\mathcal{S}$

Initial state dist: $\rho_0(s)$

Actions: $\mathcal{A}$

Discount: γ

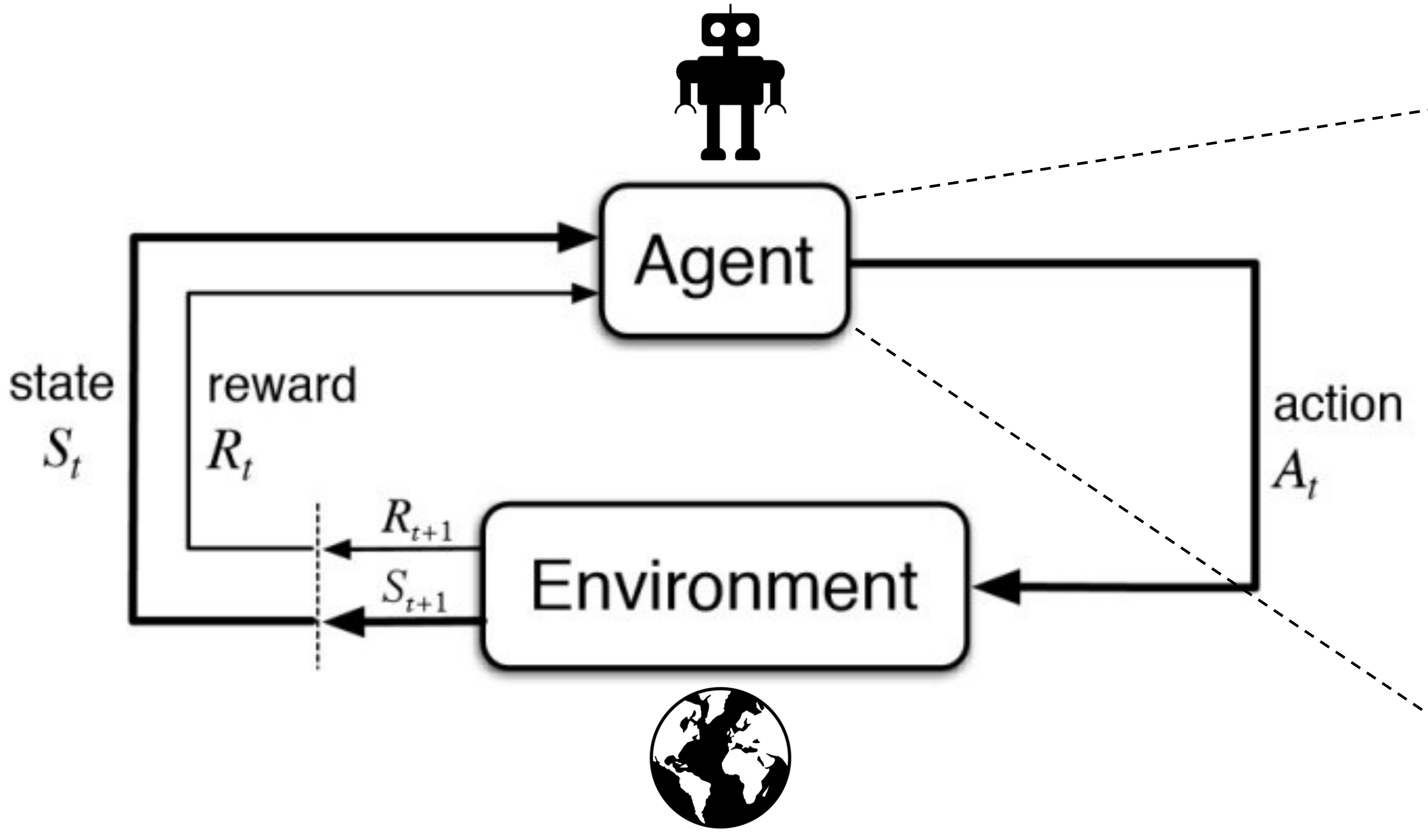
Rewards: $\mathcal{R}$

Transition Dynamics - $p(s_{t+1}|s_t, a_t)$

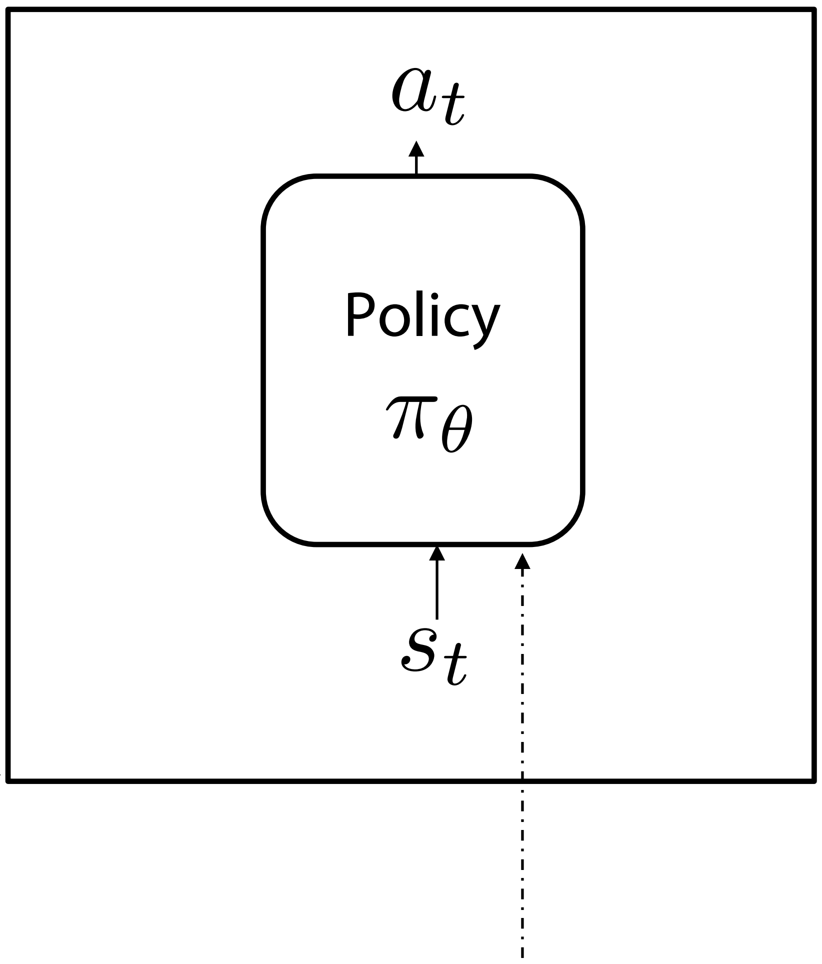
Markov property $p(s_0, s_1, s_2, a_0, a_1, a_2) = p(s_0)p(a_0|s_0)p(s_1|s_0, a_0)p(a_1|s_1)p(s_2|s_1, a_1)p(a_2|s_2)$

Trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_T, a_T, r_T)$

Reinforcement Learning Formalism



Rules for choosing actions



Needs to be learned

$$\max_{\theta} \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} r(s_t, a_t) \right]$$

Trajectory sampled using policy

Idea 1: Imitation Learning via Supervised Learning

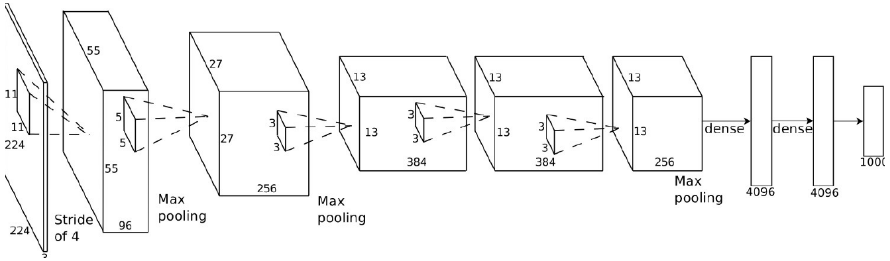
Given: Demonstrations of optimal behavior

$$\arg\max_{\theta} \mathbb{E}_{(s^*,a^*)\sim\mathcal{D}} \left[\log \pi_\theta(a^*|s^*)\right]$$

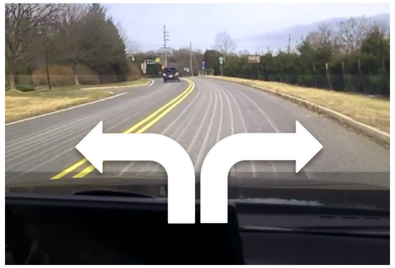
Goal: Train a policy to mimic the demonstrator

Idea: Treat imitation learning as a supervised learning problem! → Behavior Cloning

$\mathbf{o}_t$



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



$\mathbf{a}_t$

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

Idea 1: Imitation Learning via Supervised Learning

Given: Demonstrations of optimal behavior

Goal: Train a policy to mimic the demonstrator

$$\arg\max_{\theta} \mathbb{E}_{(s^*, a^*) \sim \mathcal{D}} \left[\log \pi_\theta(a^* | s^*)\right]$$

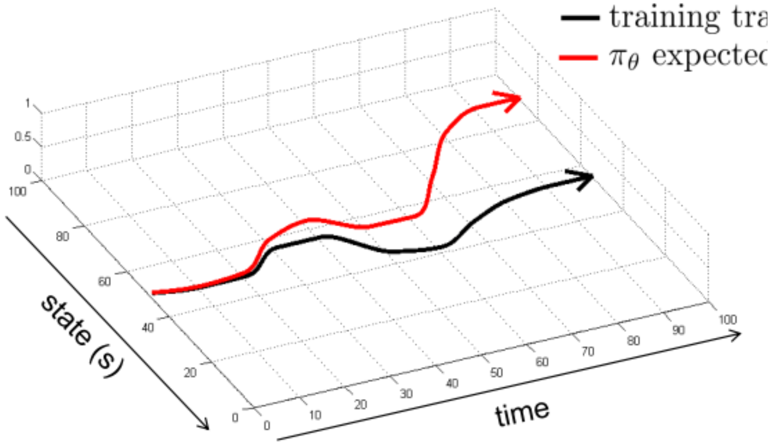
Discrete vs continuous

Maximum likelihood

```
if isinstance(env.action_space, gym.spaces.Box):
    criterion = nn.MSELoss()
else:
    criterion = nn.CrossEntropyLoss()
# Extract initial policy
model = student.policy.to(device)
def train(model, device, train_loader, optimizer):
   model.train()
   for batch_idx, (data, target) in enumerate(train_loader):
      data, target = data.to(device), target.to(device)
      optimizer.zero_grad()
      if isinstance(env.action_space, gym.spaces.Box):
         if isinstance(student, (A2C, PPO)):
            action, _, _ = model(data)
         else:
            action = model(data)
         action_prediction = action.double()
      else:
         dist = model.get_distribution(data)
         action_prediction = dist.distribution.logits
         target = target.long()
      loss = criterion(action_prediction, target)
      loss.backward()
      optimizer.step()
```

So does behavior cloning really work?

- Imitation Learning $\neq$ Supervised Learning



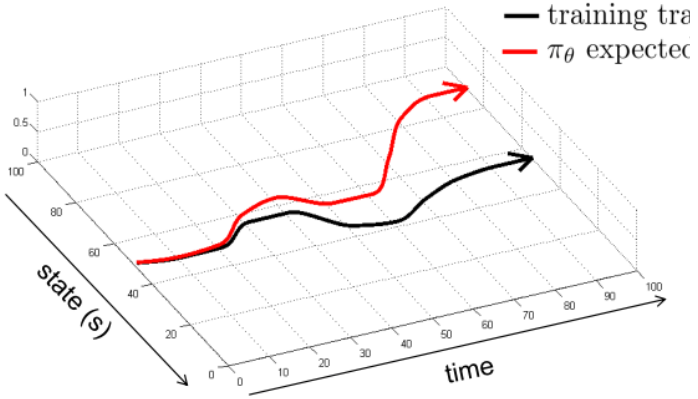
Compounding error!

$$\arg\max_\theta \mathbb{E}_{(s^*,a^*)\sim\mathcal{D}}\left[\log \pi_\theta(a^*|s^*)\right] \qquad \mathbb{E}_{(s,a)\sim\rho(\pi)}\left[1(a=a^*)\right]$$

Not the same!

How well does BC do?: Intuition

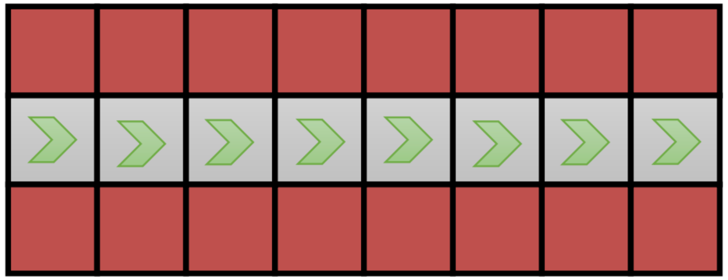
Behavior cloning has quadratically compounding error



$$\pi_\theta(a \neq \pi^*(s_t)|s_t) \leq \epsilon$$

Horizon H

If you fall off, assume the worst



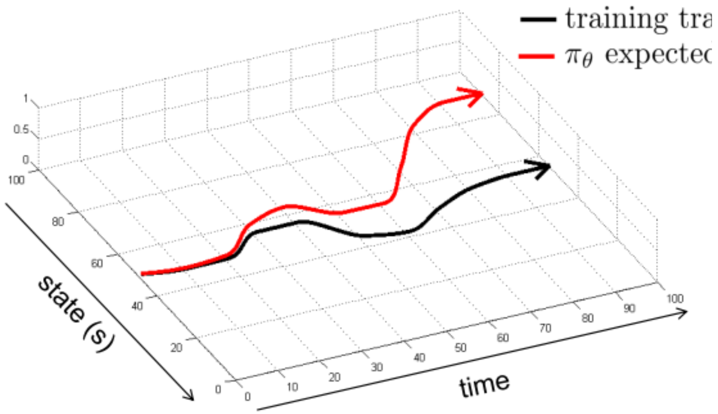
$$\underbrace{\mathbb{E}\left[\sum_t c(s_t, a_t)\right]}_{O(\epsilon H^2)} \leq \epsilon H + \dots + \dots$$

Union bound

Let’s try and understand where the problem lies?

Behavior cloning has challenges in both theory and practice

$$\sum_t \mathbb{E}_{(s_t,a_t)\sim p_{\pi_\theta}(s_t,a_t)}\left[c(s_t,a_t)\right] \leq O(\epsilon H^2)$$



Underfitting

$$\pi_\theta(a \neq \pi^*(s_t)|s_t) \leq \epsilon$$

Compounding error

$$\leq O(\epsilon H^2)$$

Lecture outline

Recap: Imitation Learning + Why it is hard

↓

Multimodality and Underfitting in Imitation

↓

Compounding Error in Imitation

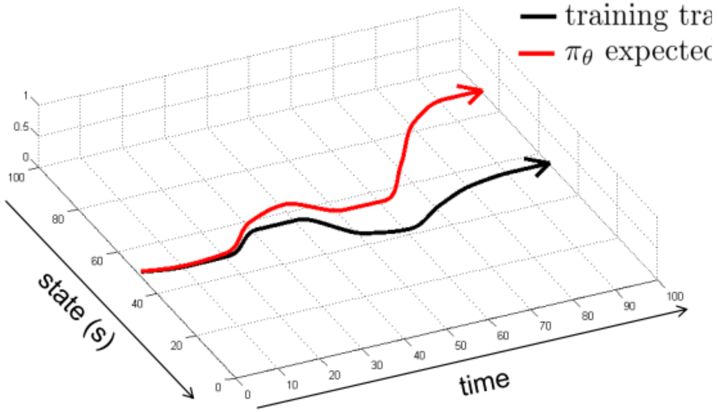
↓

Frontiers in Imitation

Let’s try and understand where the problem lies?

Behavior cloning has challenges in both theory and practice

$$\sum_t \mathbb{E}_{(s_t,a_t)\sim p_{\pi_\theta}(s_t,a_t)}\left[c(s_t,a_t)\right] \leq O(\epsilon H^2)$$



Underfitting

$$\pi_\theta(a \neq \pi^*(s_t)|s_t) \leq \epsilon$$

Compounding error

$$\leq O(\epsilon H^2)$$

But won't a bigger neural net just solve this?

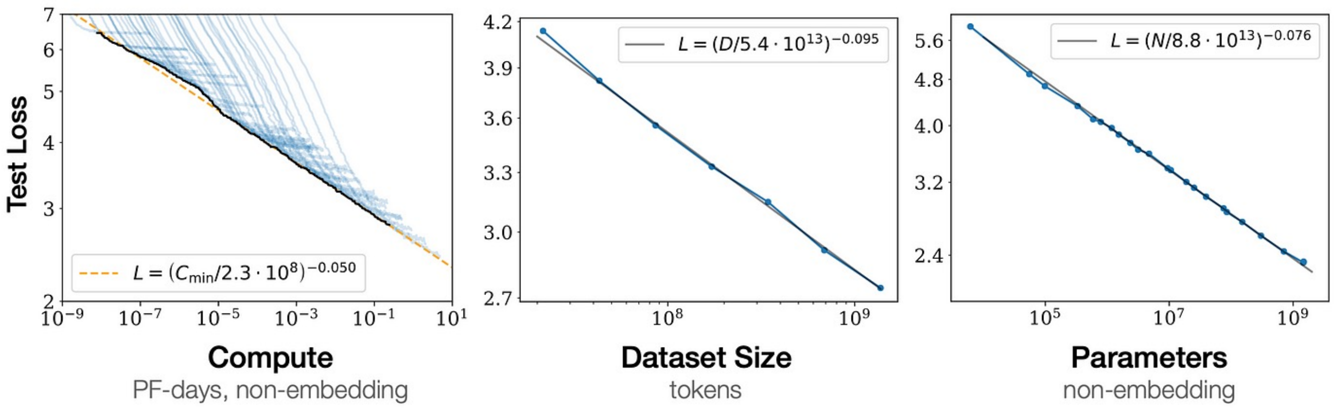
- Behavior cloning can underfit the data

$$\sum_t \mathbb{E}_{(s_t,a_t)\sim p_{\pi_\theta}(s_t,a_t)}\left[c(s_t,a_t)\right] \leq O(\epsilon H^2)$$

$$\pi_\theta(a \neq \pi^*(s_t)|s_t) \leq \epsilon$$
$$\text{for } s_t \sim p_{\text{train}}(s_t)$$

May not be able to satisfy this

Q: won't a bigger model just solve the problem?

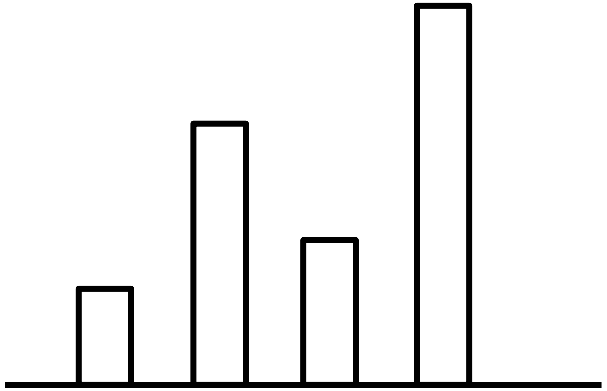


Kind of, but there's a fundamental problem!

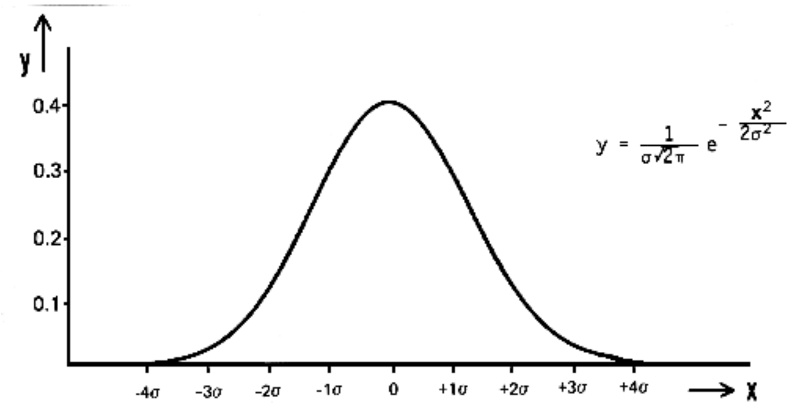
Distributional Expressivity

- Policy expressivity is a combination of expressivity of the function approximator and of the distribution family

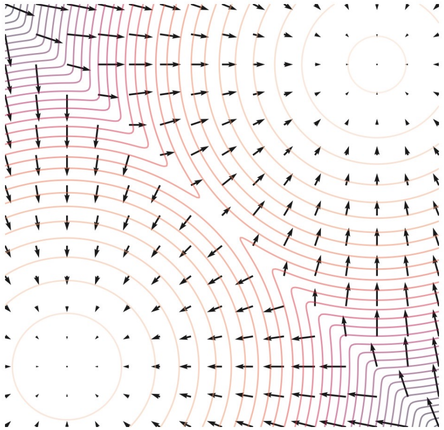
<u>Categorical</u>



<u>Gaussian</u>



<u>Diffusion policy</u>

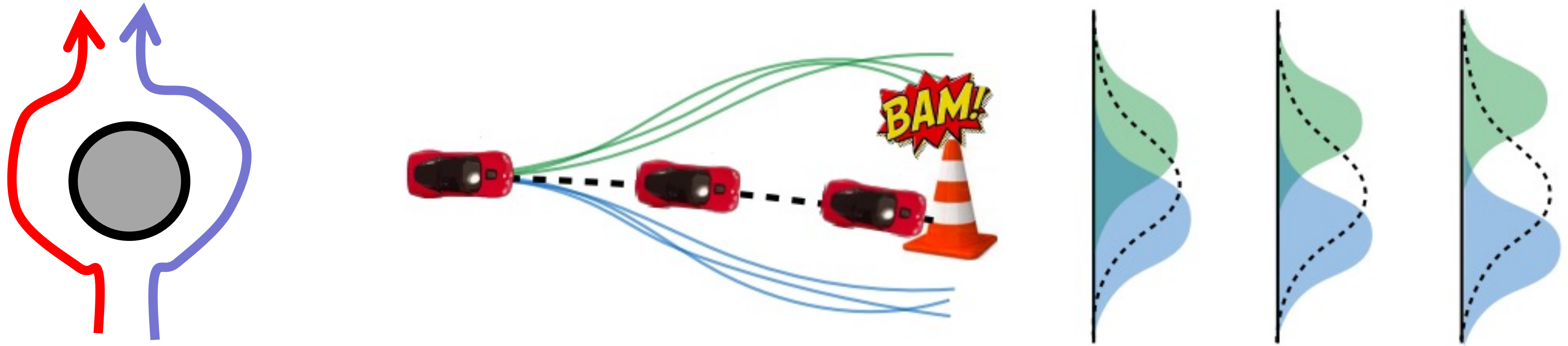


Tradeoff between expressivity and tractability

How does this reflect on imitation learning?

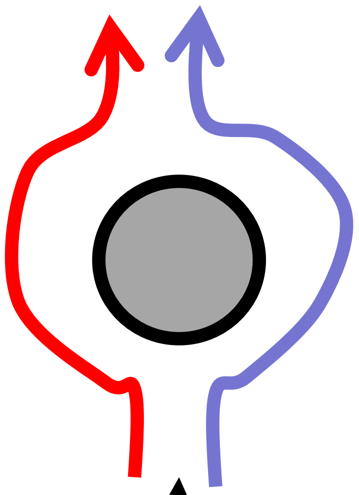
Let us consider a case with Gaussian policy

$$\arg\max_{\theta} \mathbb{E}_{(s^*,a^*)\sim\mathcal{D}} \left[\log \pi_\theta(a^*|s^*)\right]$$



A combination of **distributional expressivity** and **objective** lead to mode averaging

Let's take a closer look at the objective



$$\arg\max_{\theta} \mathbb{E}_{(s^*,a^*)\sim\mathcal{D}} [\log \pi_\theta(a^*|s^*)]$$

$$\max_{\theta} \mathbb{E}_{s^*\sim p_{\pi_e}(.)} \left[\mathbb{E}_{a^*\sim\pi_e(.|s^*)} [\log \pi_\theta(a^*|s^*) - \log \pi_e(a^*|s^*)]\right]$$

$$\min_{\theta} \mathbb{E}_{s^*\sim p_{\pi_e}(.)} \left[\mathbb{E}_{a^*\sim\pi_e(.|s^*)} \left[\log \frac{\pi_e(a^*|s^*)}{\pi_\theta(a^*|s^*)}\right]\right] = \mathbb{E}_{s^*\sim p_{\pi_e}(.)} [D_{\mathrm{KL}}(\pi_e(.|s^*)||\pi_\theta(.|s^*))]$$

Leads to mode averaging

Forward KL divergence

One instance of a broader class of divergences – f divergences $D_f(p(x), q(x)) = \mathbb{E}_{q(x)}\left[f\left(\frac{p(x)}{q(x)}\right)\right]$

https://timvieira.github.io/blog/post/2014/10/06/kl-divergence-as-an-objective-function/

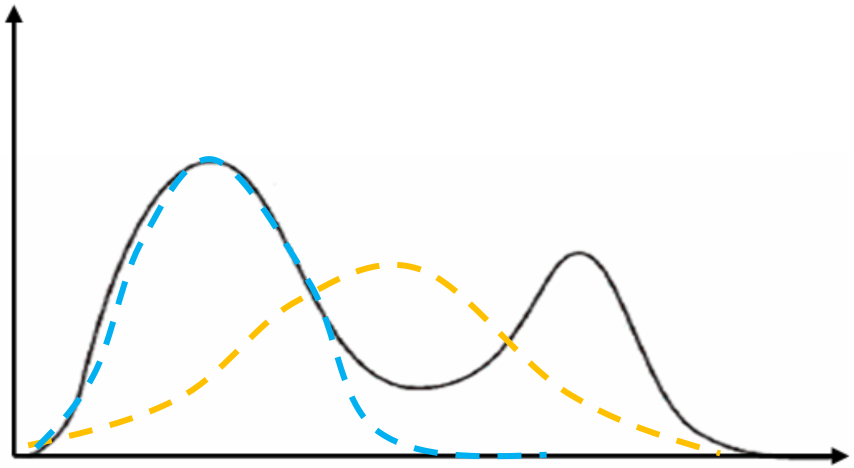
Effects of choice of f-divergence on behavior

Different divergences lead to different properties

$$\mathbb{E}_{s^* \sim p_{\pi_e}(.)}\left[D_{\mathrm{KL}}(\pi_e(.|s^*)||\pi_\theta(.|s^*))\right] \longrightarrow \mathbb{E}_{s^* \sim p_{\pi_e}(.)}\left[D_f(\pi_e(.|s^*), \pi_\theta(.|s^*))\right]$$

Forward KL (behavior cloning)

More general class of divergences



$$D_f(p(x), q(x)) = \mathbb{E}_{q(x)}\left[f\left(\frac{p(x)}{q(x)}\right)\right]$$

Forward KL (mode covering) $\quad f(x) = x\log(x)$

Reverse KL (mode seeking) $\quad f(x) = -\log(x)$

So how do we fix BC?

Use a different f-divergence!
(Change f)

or

Use a richer distribution class!
(Change π_θ)

Using alternative f-divergences: Reverse KL

- Reverse KL helps, is mode seeking $D_{\mathrm{RKL}}(\pi_e(.|s^*), \pi^\theta(.|s^*)) = \mathbb{E}_{\pi^\theta(.|s^*)}\left[\log\left(\frac{\pi^\theta(.|s^*)}{\pi_e(.|s^*)}\right)\right]$
- Challenge – requires known expert likelihood
- We need a sample based estimate!

Imitation Learning as f-Divergence Minimization

Liyiming Ke[1], Sanjiban Choudhury[1], Matt Barnes[1], Wen Sun[2], Gilwoo Lee[1], and Siddhartha Srinivasa[1]

Go read this!

$$\min_\theta \mathbb{E}_{\pi^\theta(.|s^*)}\left[\log\left(\frac{\pi^\theta(.|s^*)}{\pi_e(.|s^*)}\right)\right] \longleftrightarrow \min_\theta \max_\phi \mathbb{E}_{a\sim\pi^\theta(.|s^*)}\left[\phi(a)\right] - \mathbb{E}_{a\sim\pi_e(.|s^*)}\left[f^*(\phi(a))\right]$$

(Intractable) (Tractable – GAN style optimization)

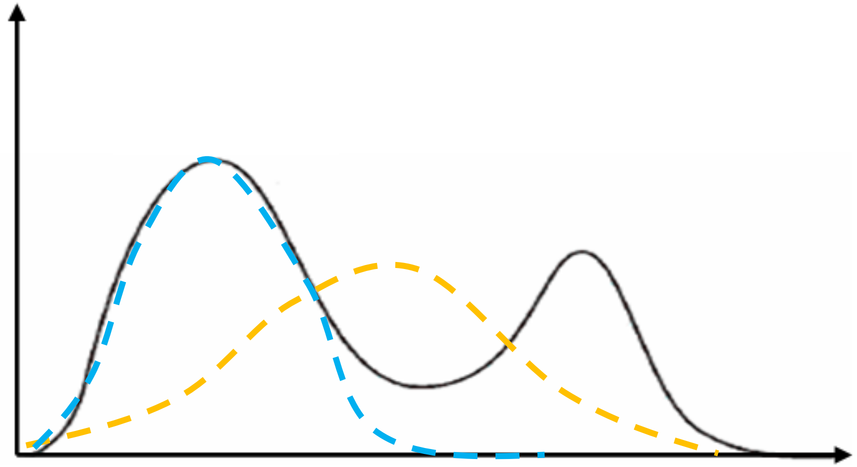
Effects of choice of f-divergence on behavior

Different divergences lead to different properties

$$\mathbb{E}_{s^* \sim p_{\pi_e}(.)}[D_{\mathrm{KL}}(\pi_e(.|s^*)||\pi_\theta(.|s^*))] \longrightarrow \mathbb{E}_{s^* \sim p_{\pi_e}(.)}[D_f(\pi_e(.|s^*), \pi_\theta(.|s^*))]$$

Forward KL (behavior cloning)

More general class of divergences



$$D_f(p(x), q(x)) = \mathbb{E}_{q(x)}\left[f\left(\frac{p(x)}{q(x)}\right)\right]$$

Forward KL (mode covering) $f(x) = x\log(x)$

Reverse KL (mode seeking) $f(x) = -\log(x)$

So how do we fix BC?

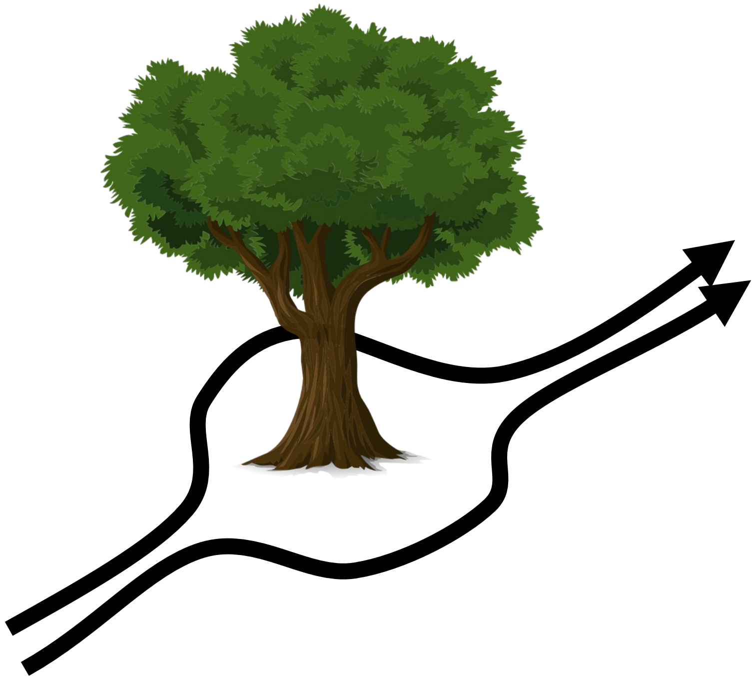
Use a different f-divergence!
(Change f)

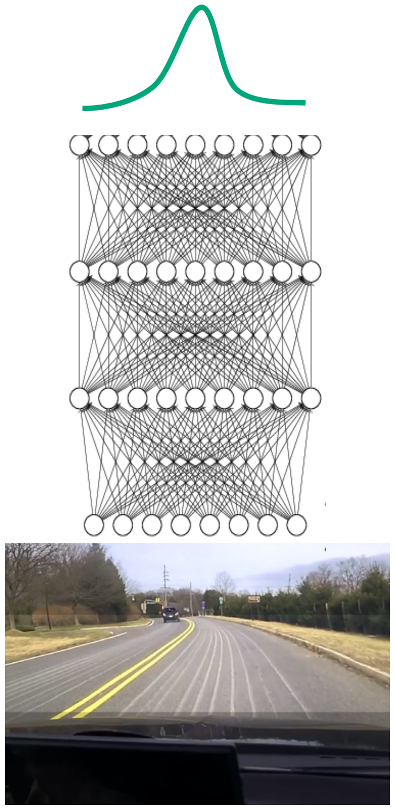
or

$\boxed{\text{Use a richer distribution class! (Change } \pi_\theta\text{)}}$

Using Richer Policy Distribution Classes

Multimodal behavior → use more **expressive** probability distributions, no mode averaging issues



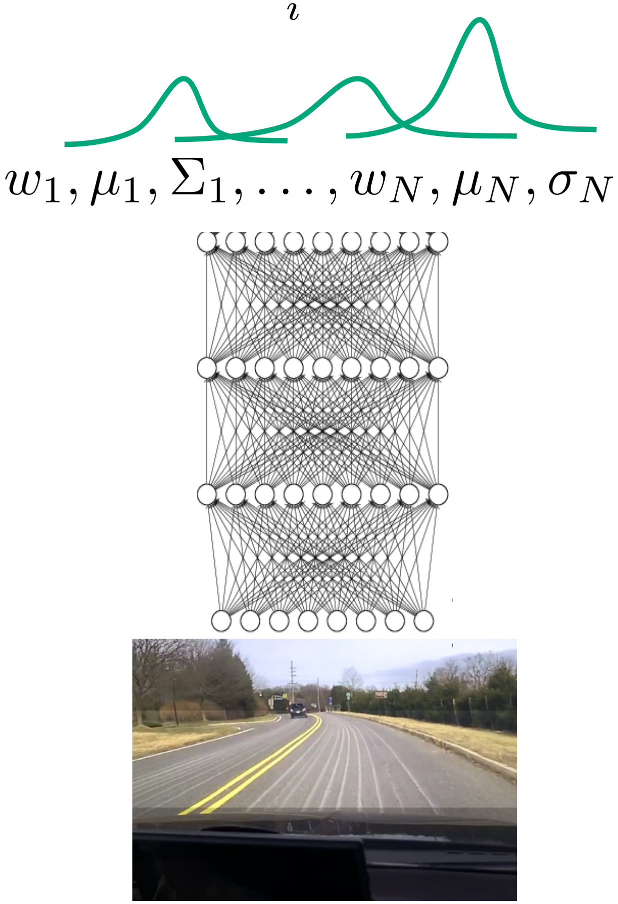


1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. …

Why might we fail to fit the expert?

1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. …

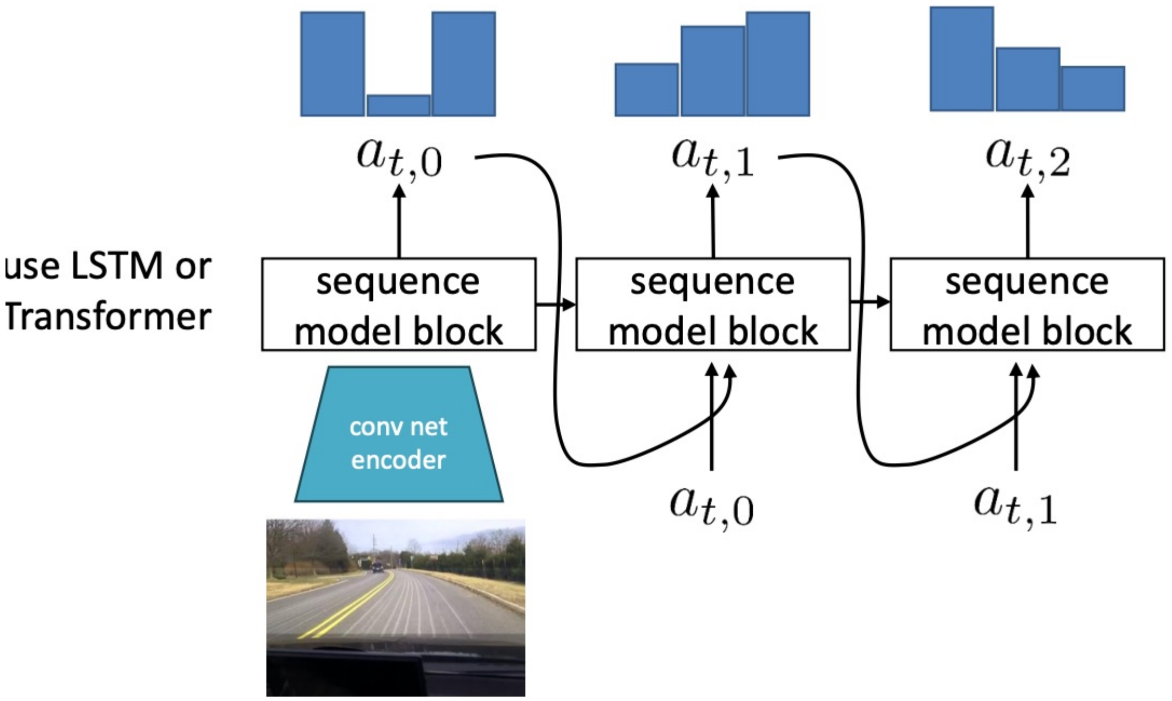
$$\pi(\mathbf{a}|\mathbf{o}) = \sum_i w_i \mathcal{N}(\mu_i, \Sigma_i)$$



Why might we fail to fit the expert?

1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. …

$$\mathbf{a}_t = \begin{pmatrix} 0.1 \\ 1.2 \\ -0.3 \end{pmatrix} \begin{array}{l} a_{t,0} \\ a_{t,1} \\ a_{t,2} \end{array}$$



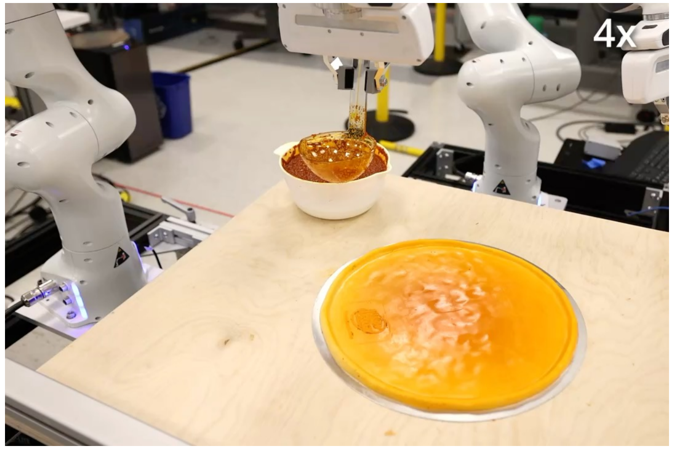
Why does this work?

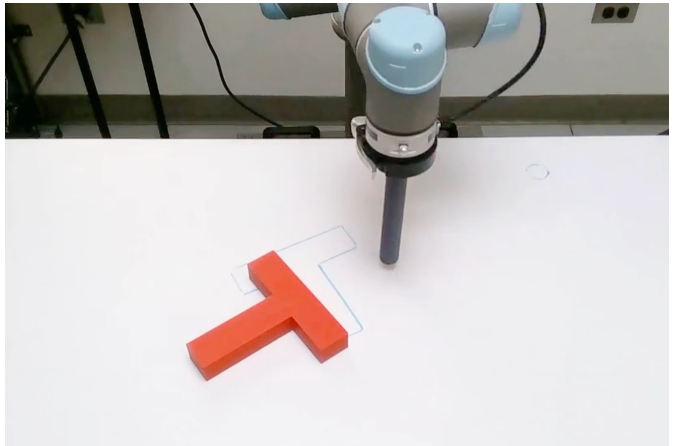
$$\text{first step: } p(a_{t,0}|\mathbf{s}_t)$$
$$\text{second step: } p(a_{t,1}|\mathbf{s}_t, a_{t,0})$$
$$\text{third step: } p(a_{t,2}|\mathbf{s}_t, a_{t,0}, a_{t,1})$$

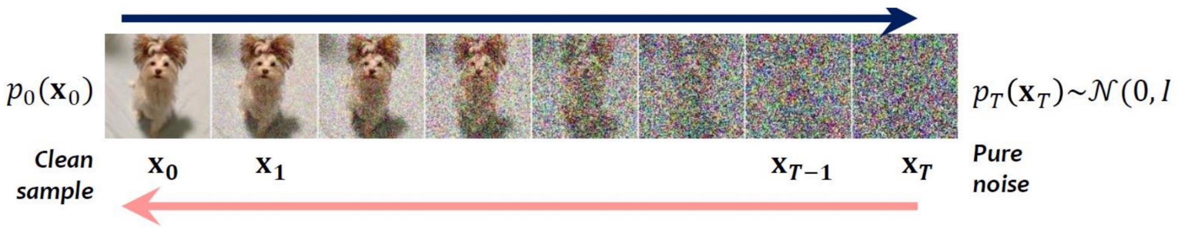
$$\begin{aligned} & p(a_{t,2}|\mathbf{s}_t, a_{t,0}, a_{t,1})p(a_{t,1}|\mathbf{s}_t, a_{t,0})p(a_{t,0}|\mathbf{s}_t) \\ &= p(a_{t,0}, a_{t,1}, a_{t,2}|\mathbf{s}_t) \\ &= p(\mathbf{a}_t|\mathbf{s}_t) \end{aligned}$$

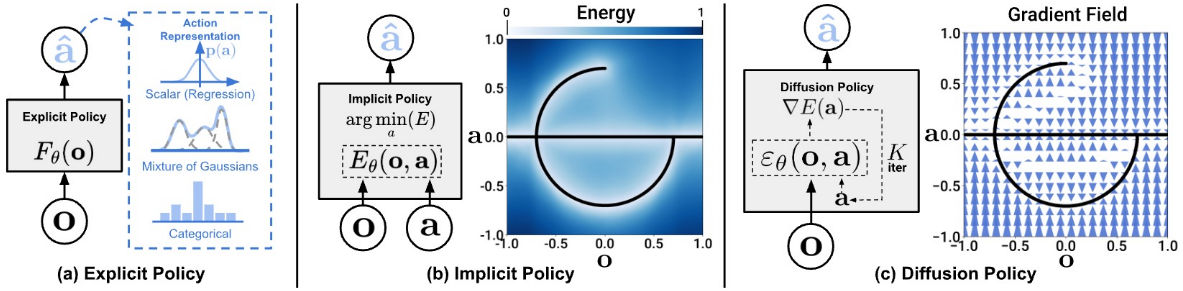
Why might we fail to fit the expert?

1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. …





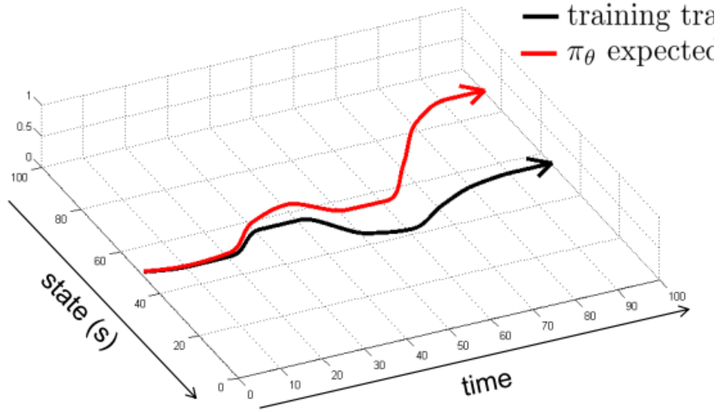




Let's try and understand where the problem lies?

Behavior cloning has challenges in both theory and practice

$$\sum_t \mathbb{E}_{(s_t,a_t)\sim p_{\pi_\theta}(s_t,a_t)}\left[c(s_t,a_t)\right] \leq O(\epsilon H^2)$$



Underfitting

$$\pi_\theta(a \neq \pi^*(s_t)|s_t) \leq \epsilon$$

Compounding error

$$\leq O(\epsilon H^2)$$

Richer policy class

Alternative Divergence

Lecture outline

Recap: Imitation Learning + Why it is hard

↓

Multimodality and Underfitting in Imitation

↓

Compounding Error in Imitation

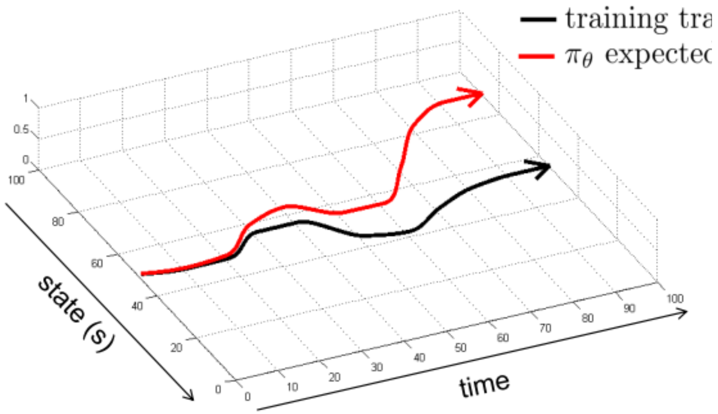
↓

Frontiers in Imitation

Let's try and understand where the problem lies?

Behavior cloning has challenges in both theory and practice

$$\sum_t \mathbb{E}_{(s_t,a_t)\sim p_{\pi_\theta}(s_t,a_t)}\left[c(s_t,a_t)\right] \leq O(\epsilon H^2)$$



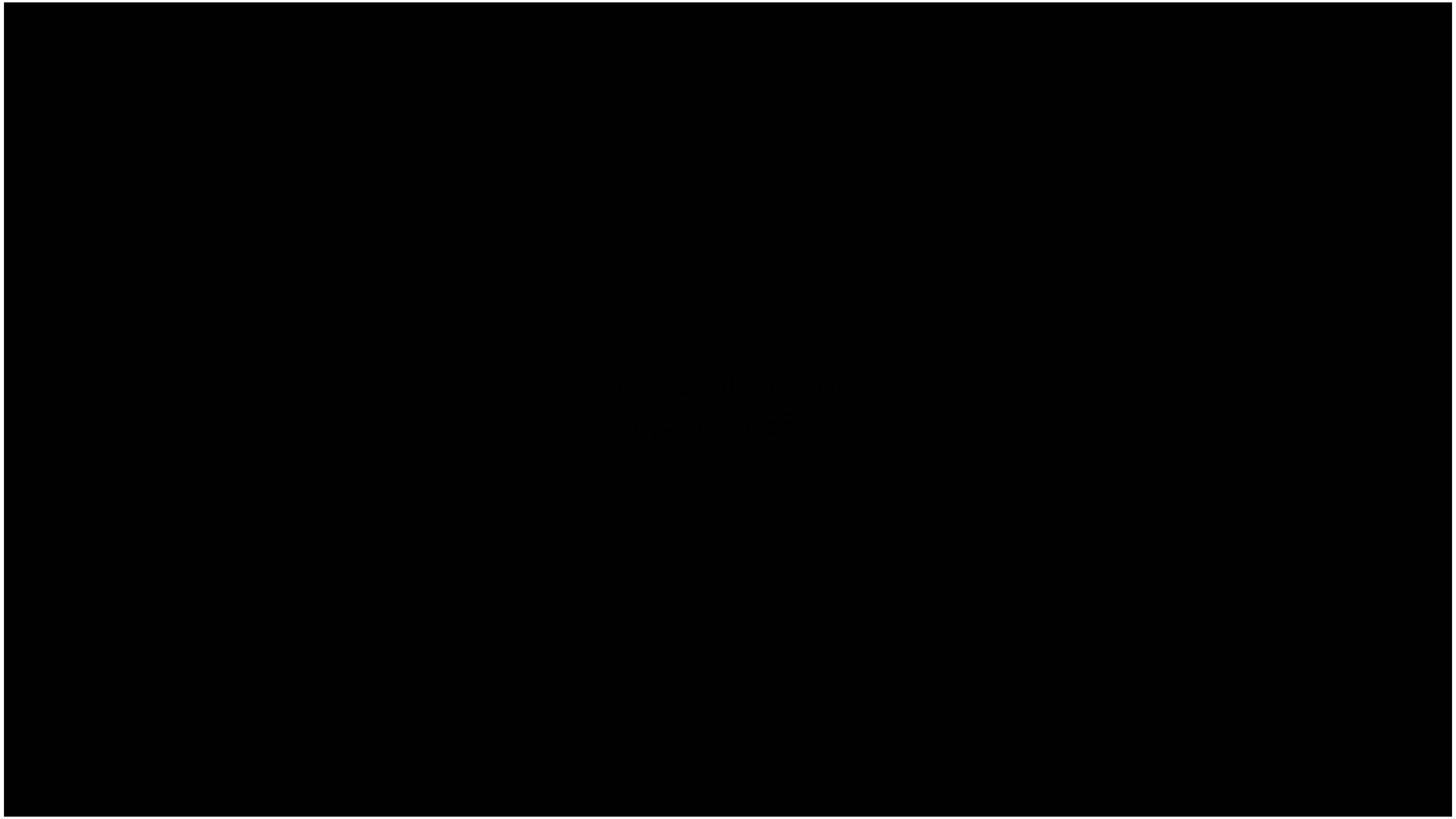
Underfitting

$$\pi_\theta(a \neq \pi^*(s_t)|s_t) \leq \epsilon$$

Compounding error

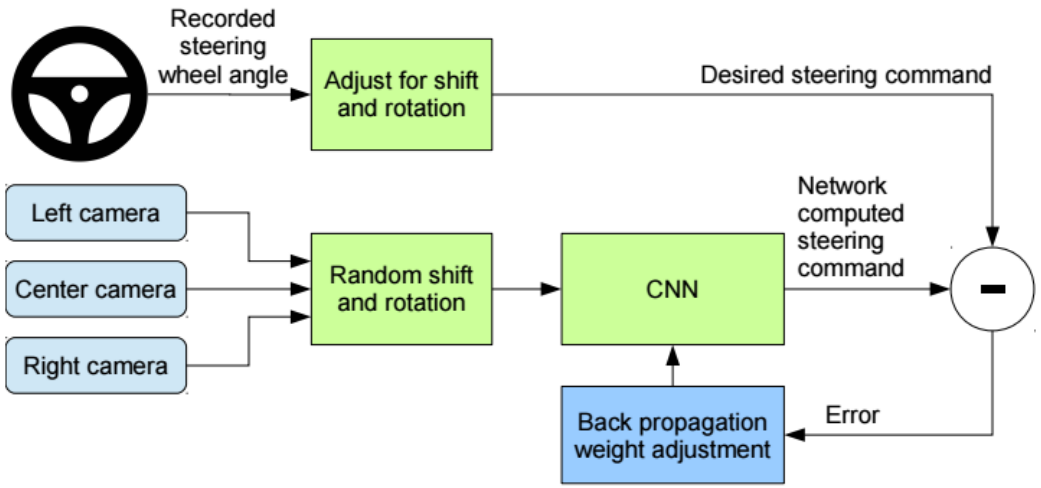
$$\leq O(\epsilon H^2)$$

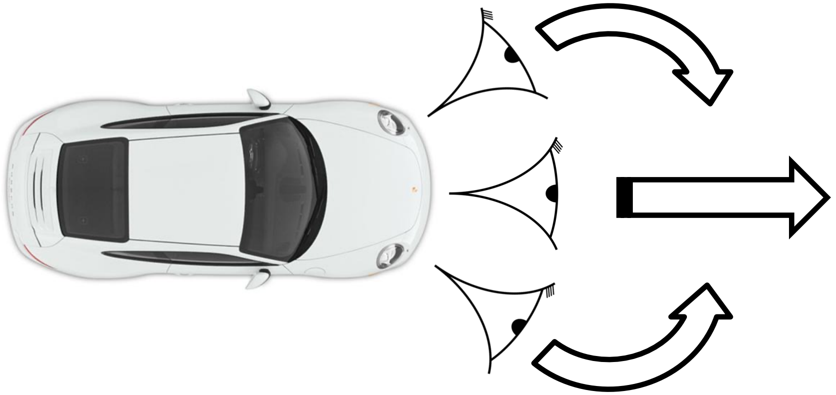
Can we avoid compounding error in special cases?



Video: Bojarski et al. ‘16, NVIDIA

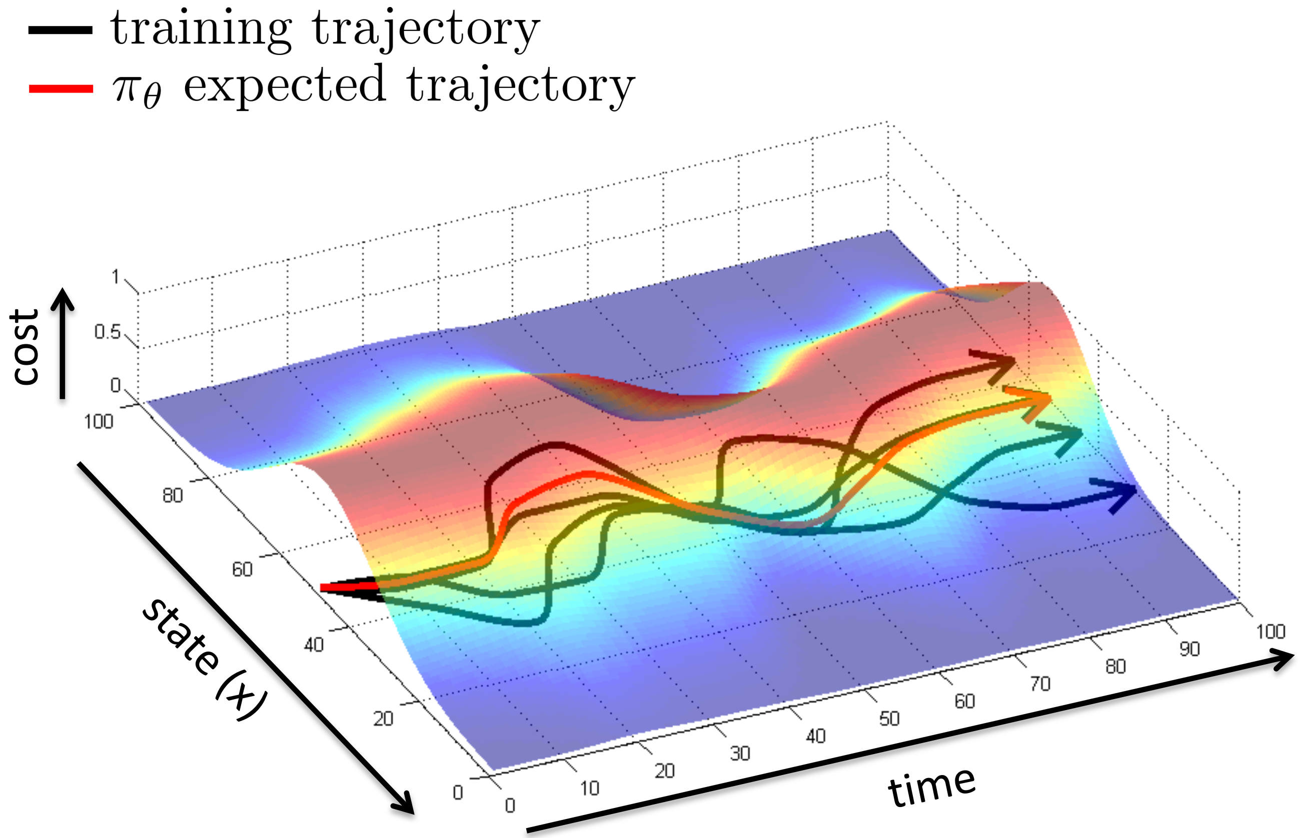
Why did that work?





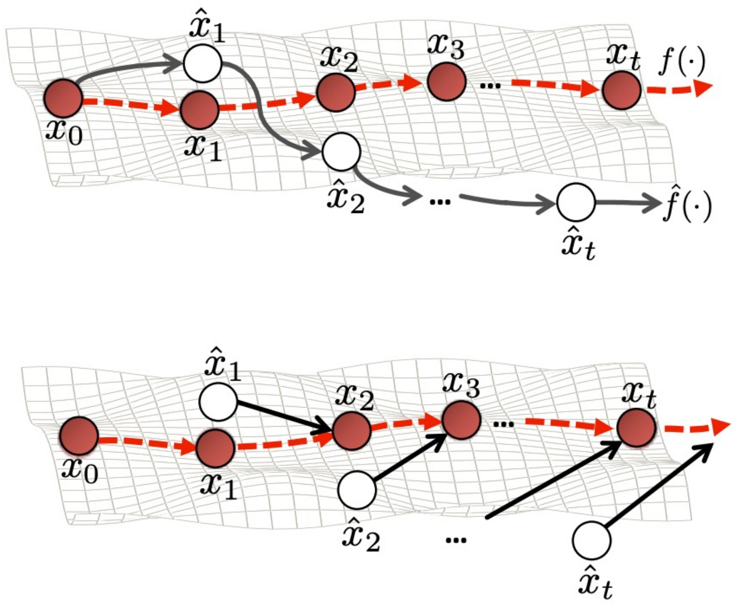
Bojarski et al. ‘16, NVIDIA

What is the general principle?

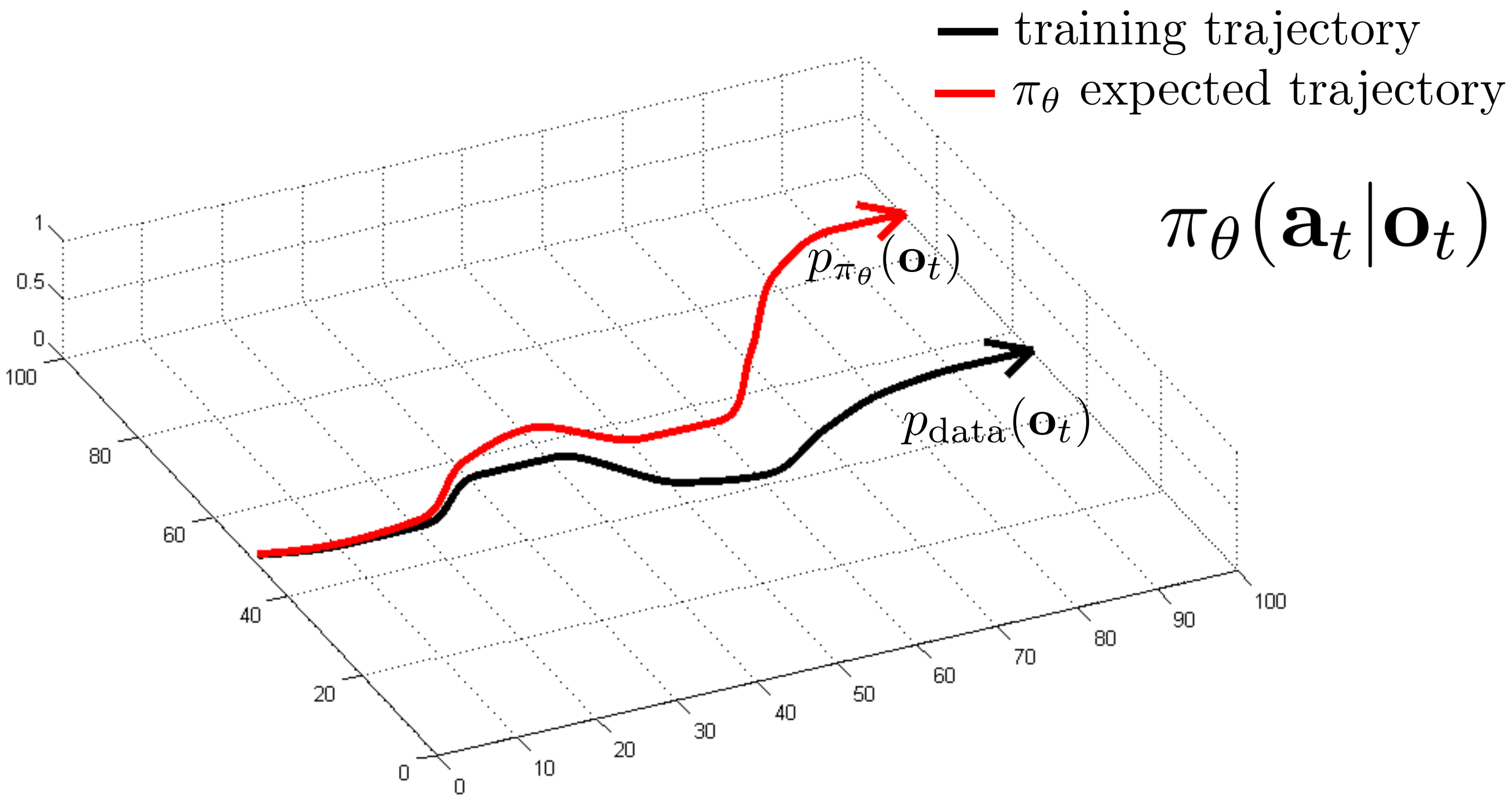


stability

Corrective labels that bring you back to the data



What might this mean mathematically?



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

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

Concrete Instantation: DAgger

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 **A**ggregation

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: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

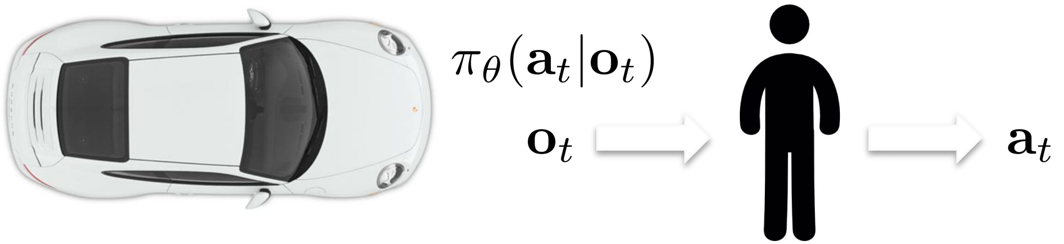
Ross et al. ‘11

DAgger Example

Ross et al. '13

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: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

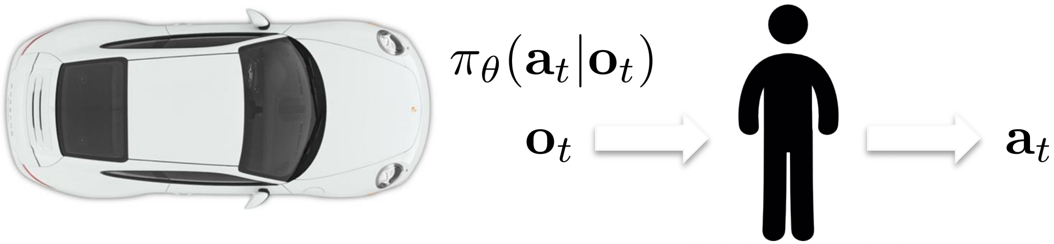


How might we fix this?

"Generate" corrective labels automatically

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: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

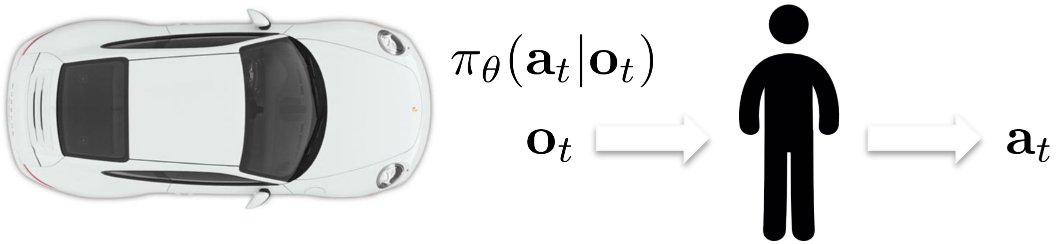
Do at data collection time



How might we fix this?

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: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

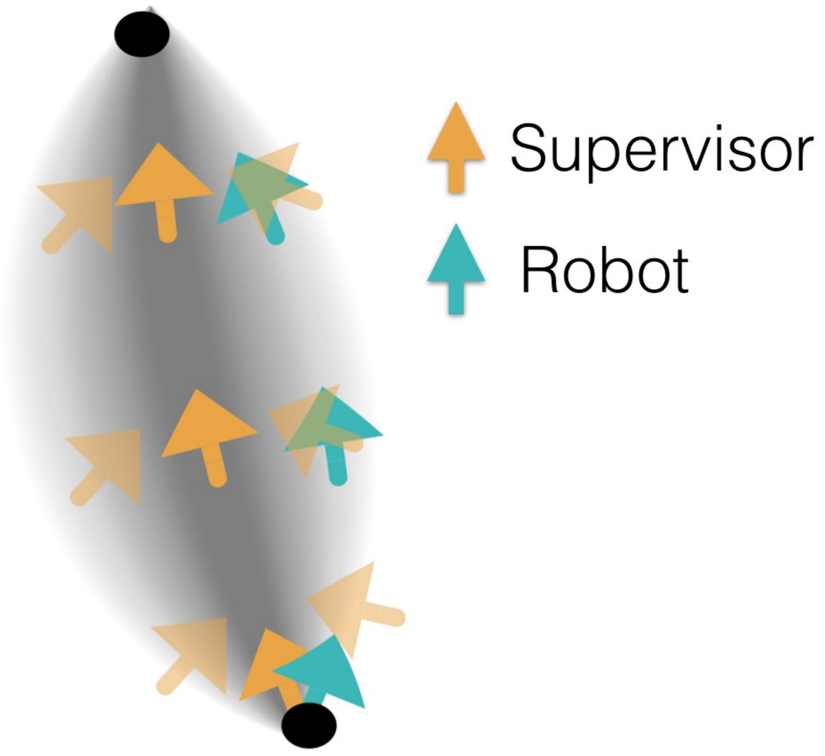
Do at data collection time



Ross et al. '11

Noising the Data Collection Process

Key idea: force the human to correct for noise **during** training

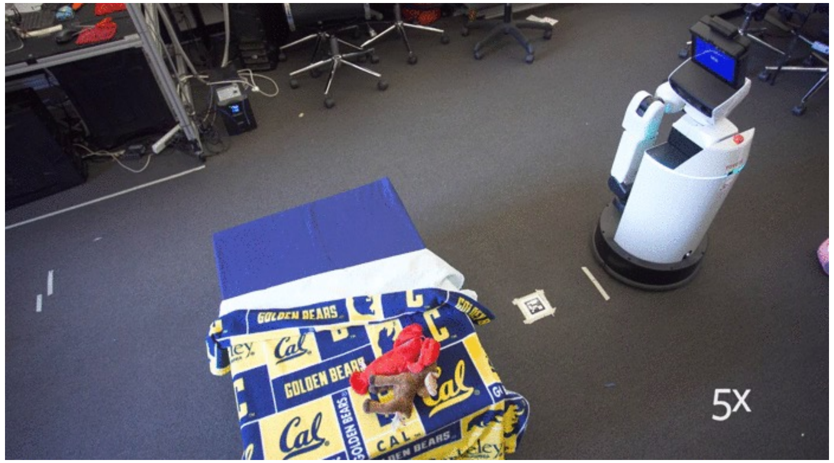


Noise Injection

Under noise during data collection

Maximize likelihood

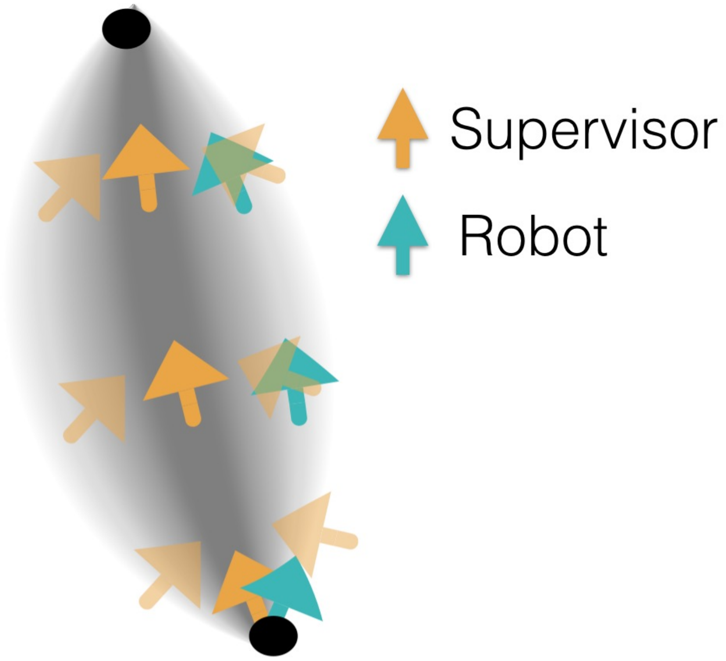
$$\hat{\psi}_{k+1} = \underset{\psi}{\text{argmin}}\ E_{p(\xi|\pi_{\theta^*},\psi_k)} - \sum_{t=0}^{T-1} \log\left[\pi_{\theta^*}(\pi_{\hat{\theta}}(\mathbf{x_t})|\mathbf{x_t},\psi)\right]$$



DART: Noise Injection for Robust Imitation Learning, Laskey et al CoRL '17

Why might this not be enough?

Key idea: force the human to correct for noise **<u>during</u>** training



Noise Injection

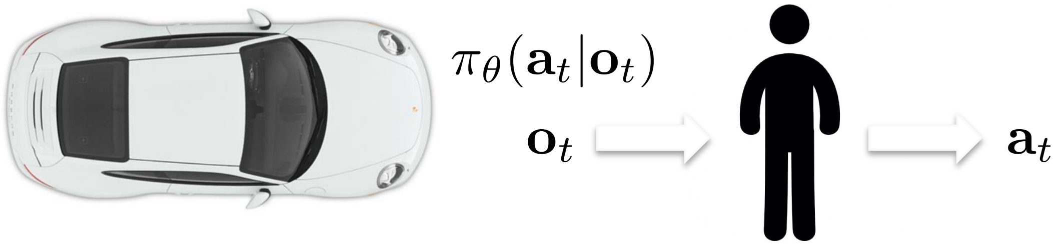
Assumes that the expert **<u>can</u>** actually perform behaviors under noise
→ Not always possible!

DART: Noise Injection for Robust Imitation Learning, Laskey et al CoRL ‘17

How might we fix this?

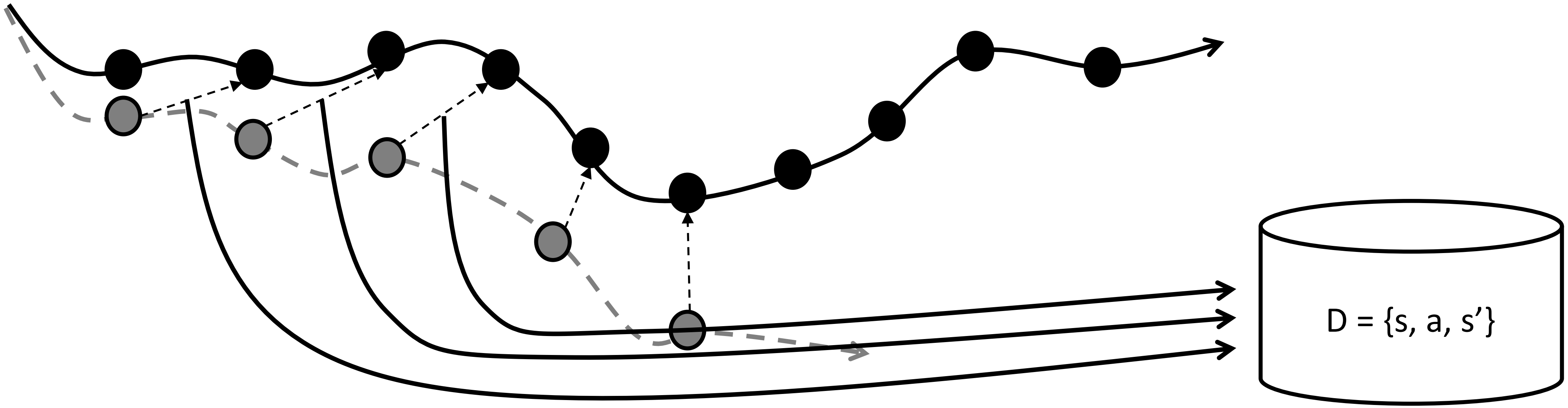
"Generate" corrective labels automatically

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 avoid expensive online data collection/labeling?



Generate corrective labels to dataset for imitation

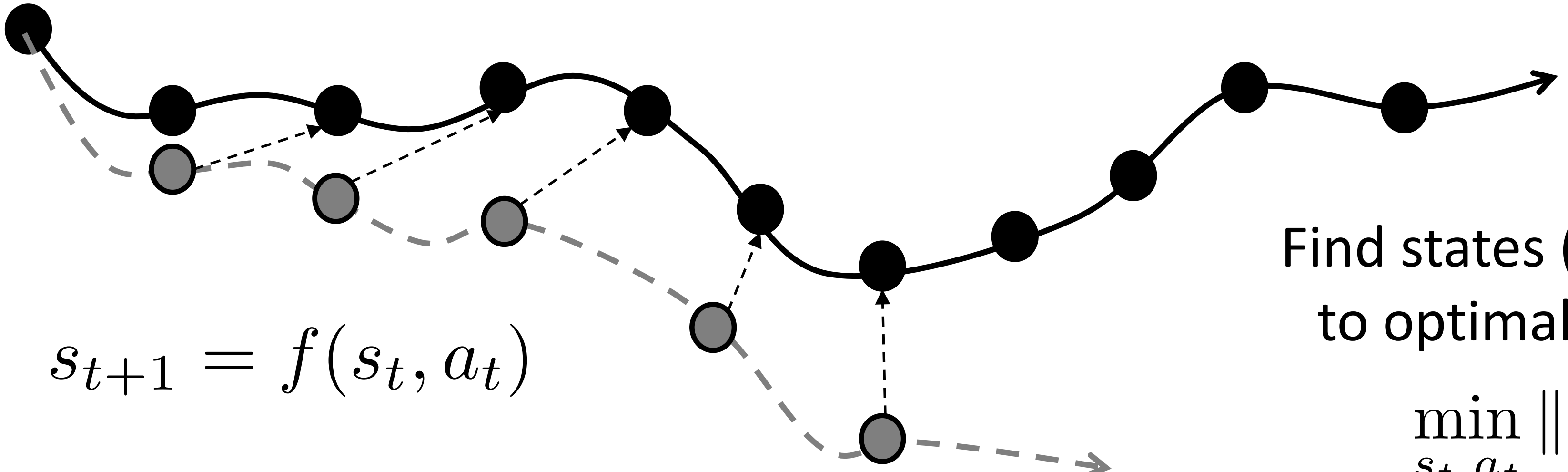

Abhay Deshpande


Yunchu Zhang


Liyiming Ke

How can we find corrective labels without an expensive human in the loop and online data collection?

Generating Corrective Labels From True Dynamics



$$s_{t+1} = f(s_t, a_t)$$

Find states (s_t), actions (a_t) that lead back to optimal states under true dynamics

$$\min_{s_t, a_t} \|s^*_{t+1} - f(s_t, a_t)\| \leq \epsilon$$

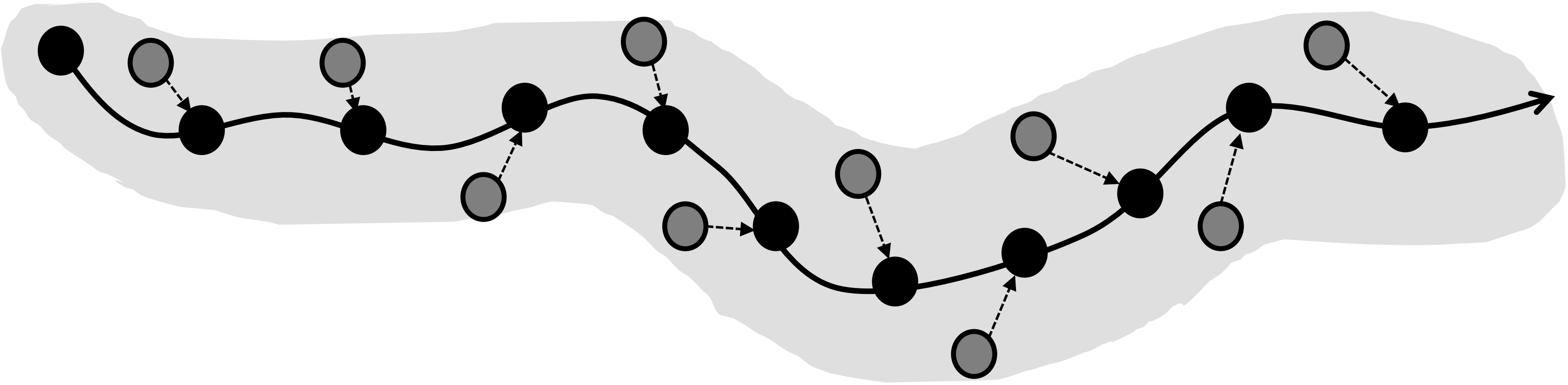
Easy with known dynamics

Intuition: find labels to bring OOD states back in distribution

But models are unknown! ☹

CCIL: Continuity-based Data Augmentation for Corrective Imitation Learning, Ke et al '24
Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels, Deshpande et al '24

Generating Corrective Labels with **Learned** Dynamics



Ok models are unknown, let's learn them!

$$\min_{\hat{f}} \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}} \left[\|\hat{f}(s_t, a_t) - s_{t+1}\|_2 \right]$$

$$\|s^*_{t+1} - \hat{f}_\phi(s_t, a_t)\| \leq \epsilon$$

But learned dynamics $\hat{f}_\phi$ are not globally accurate?

↓

Under approximately Lipschitz smooth models, trust models around training data

Find states (s_t), actions (a_t) that lead back to optimal states under ~~true~~ learned dynamics, **where learned dynamics can be trusted**

$$\min_{s_t, a_t} \|s^*_{t+1} - \hat{f}_\phi(s_t, a_t)\| \leq \epsilon \longleftarrow \text{Corrective label}$$

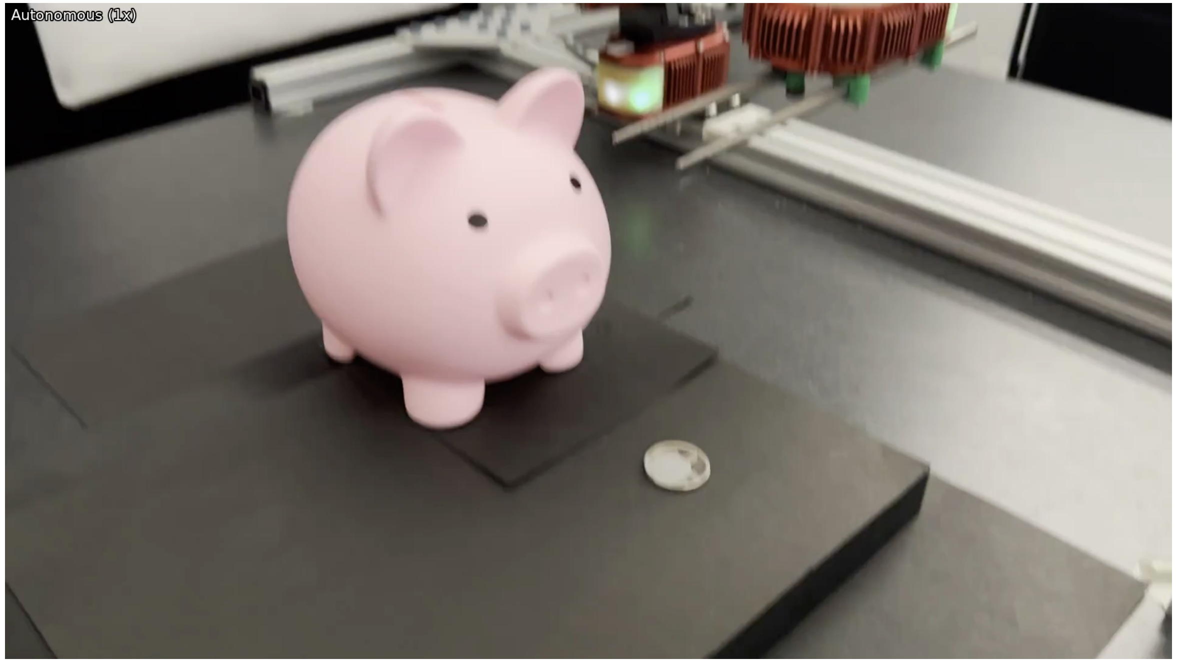
$$\text{s.t} \quad \|s^*_t - s_t\| \leq \epsilon_1, \|a^*_t - a_t\| \leq \epsilon_2 \longleftarrow \text{Close to data}$$

CCIL: Continuity-based Data Augmentation for Corrective Imitation Learning, Ke et al '24
Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels, Deshpande et al '24

How well does generating corrective labels work?

With corrective labels

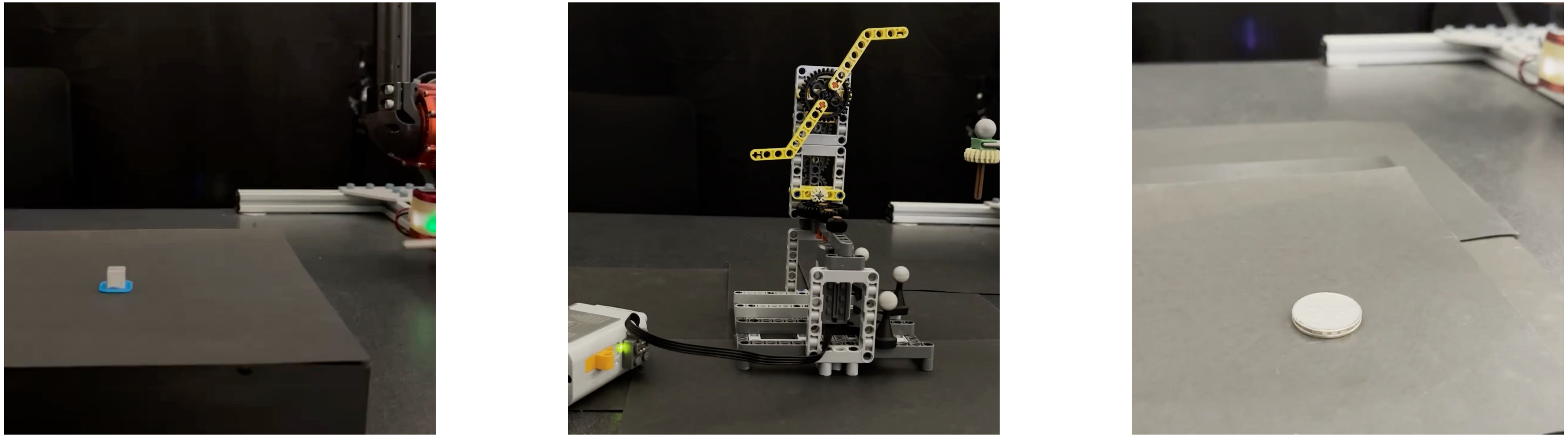
Without corrective labels



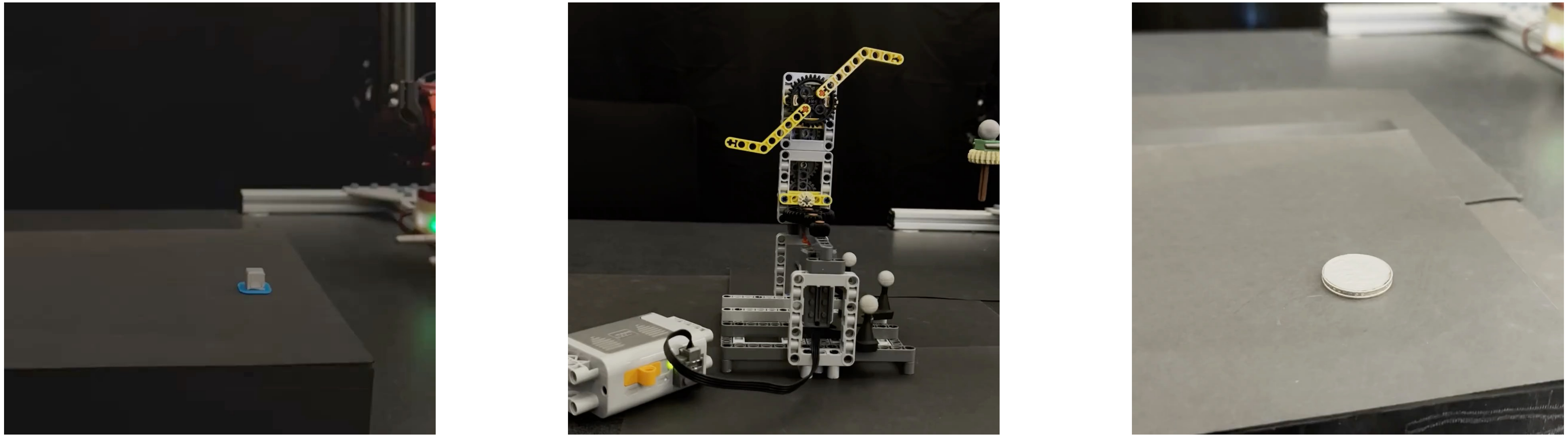
CCIL: Continuity-based Data Augmentation for Corrective Imitation Learning, Ke et al '24
Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels, Deshpande et al '24

How well does generating corrective labels work?

With corrective labels



Without corrective labels



Lecture outline

Recap: Imitation Learning + Why it is hard

↓

Multimodality and Underfitting in Imitation

↓

Compounding Error in Imitation

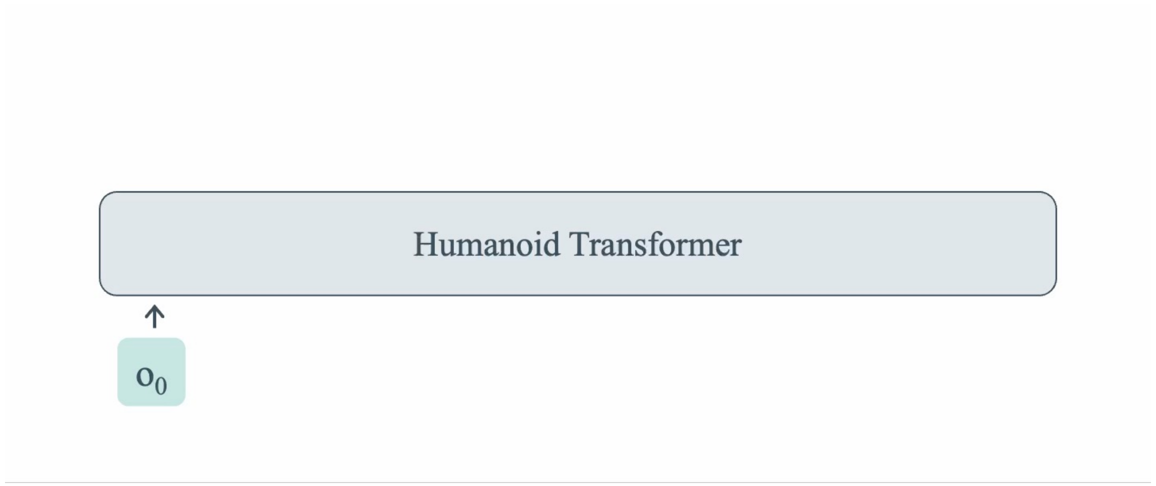
↓

Frontiers in Imitation

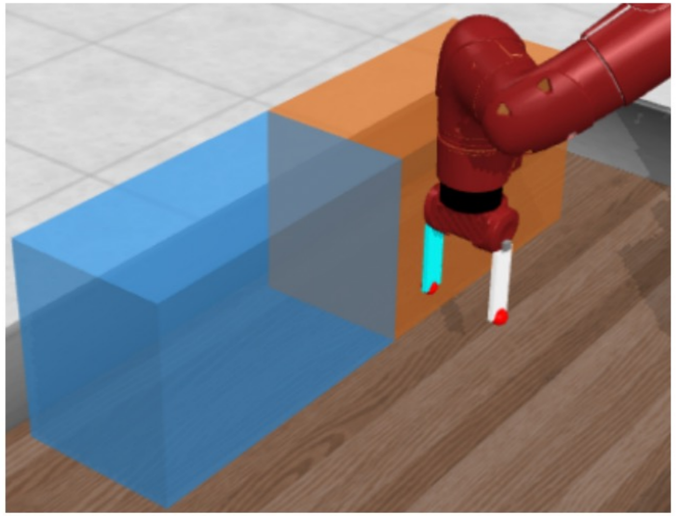
So does this solve all the issues in imitation?

Frontiers in Imitation Learning

Non-Markovian Demonstrators



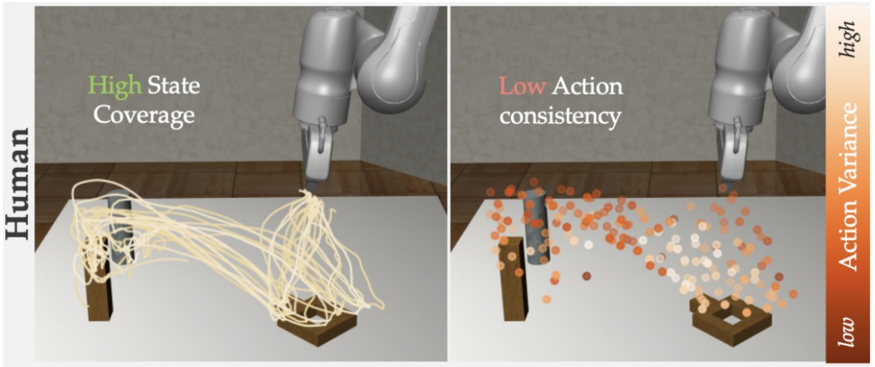
Characterizing generalization



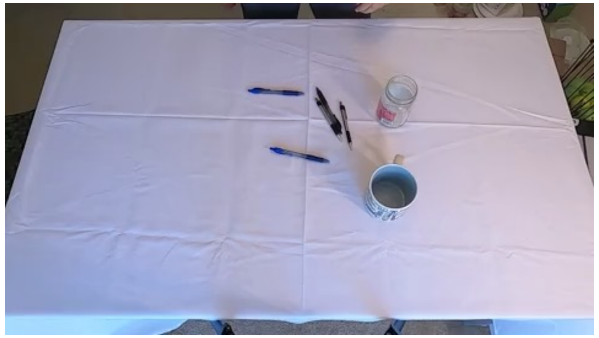
Action-Free Data

Frontiers in Imitation Learning

Data Curation and Quality



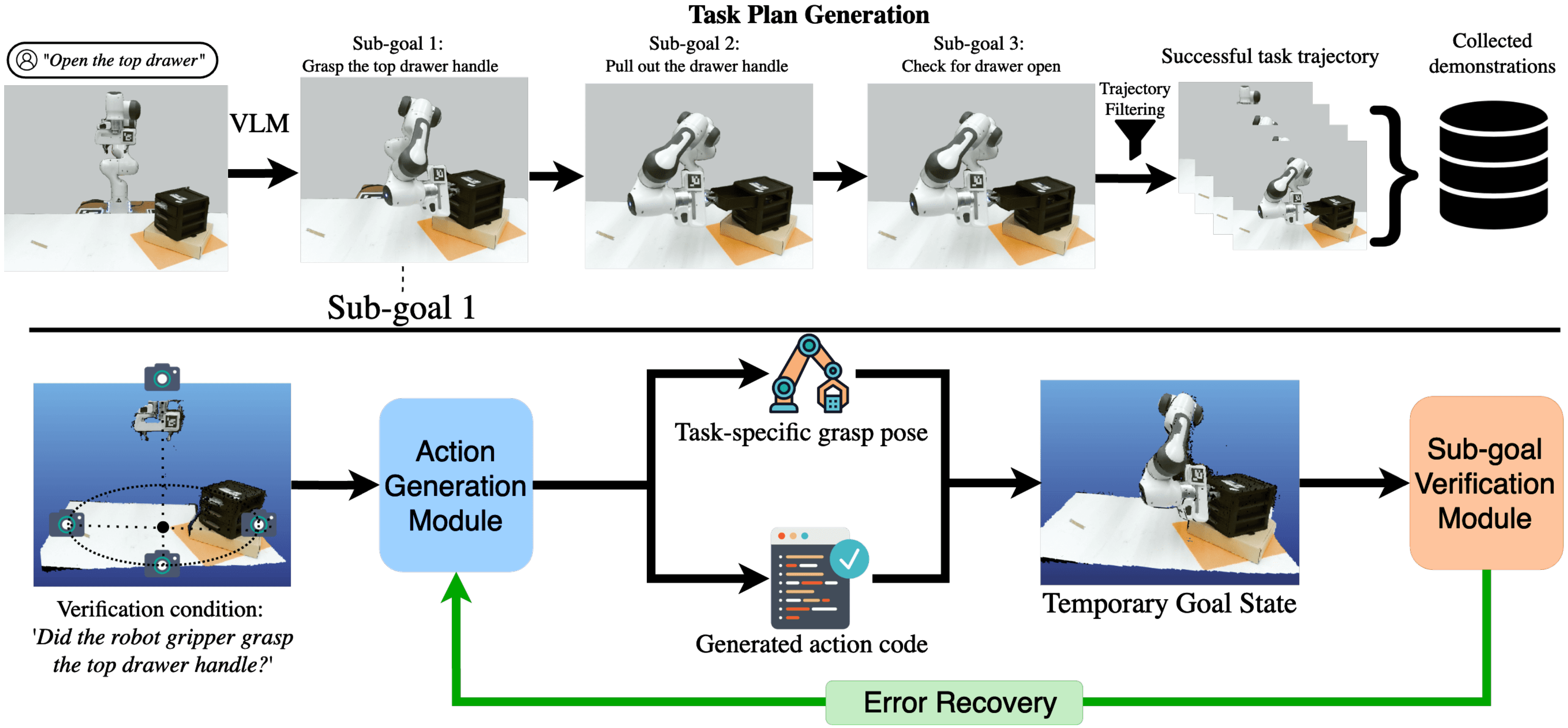
Embodiment Shift



Teleoperation Interfaces

Frontiers in Imitation Learning

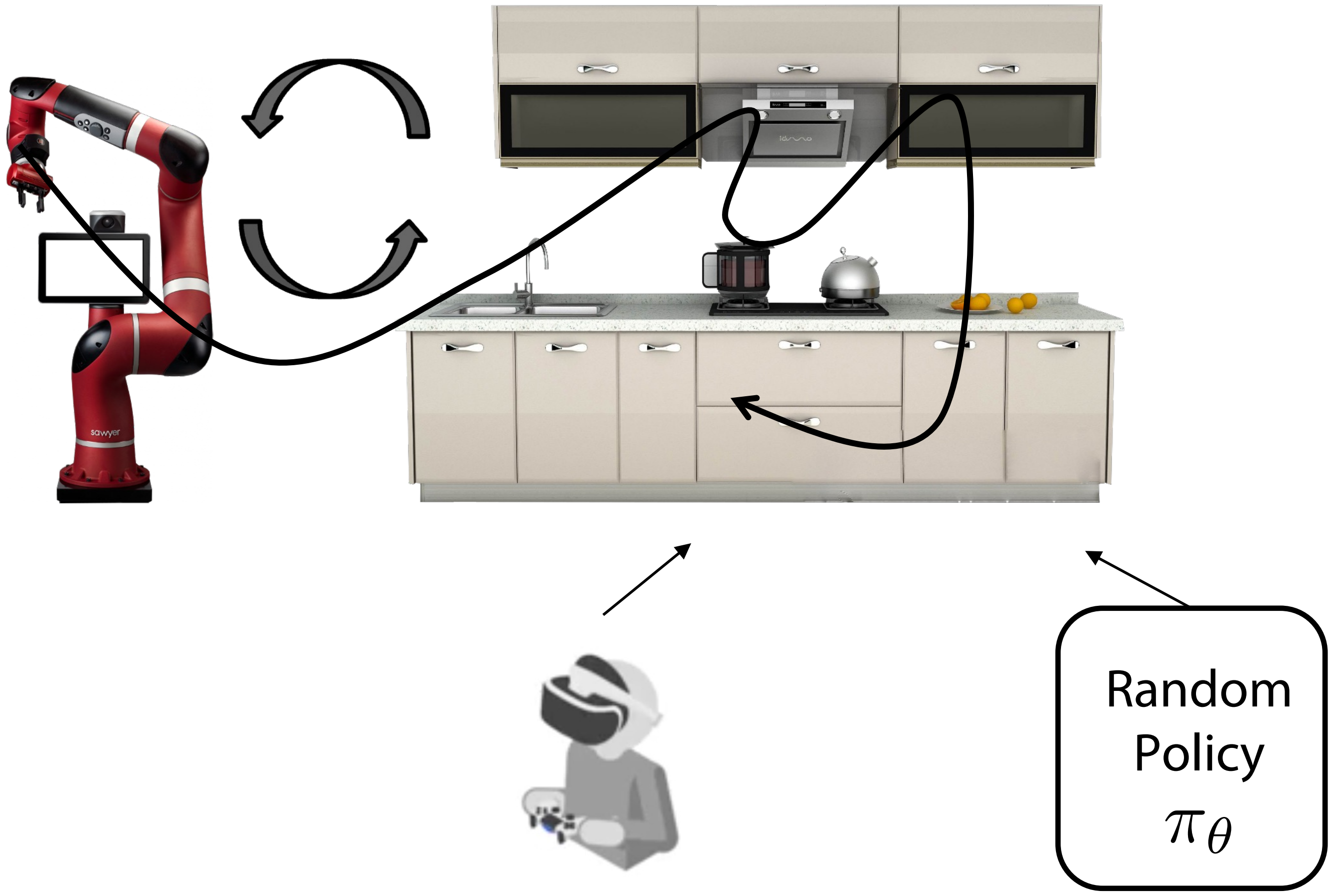
Learning how to retry and improve

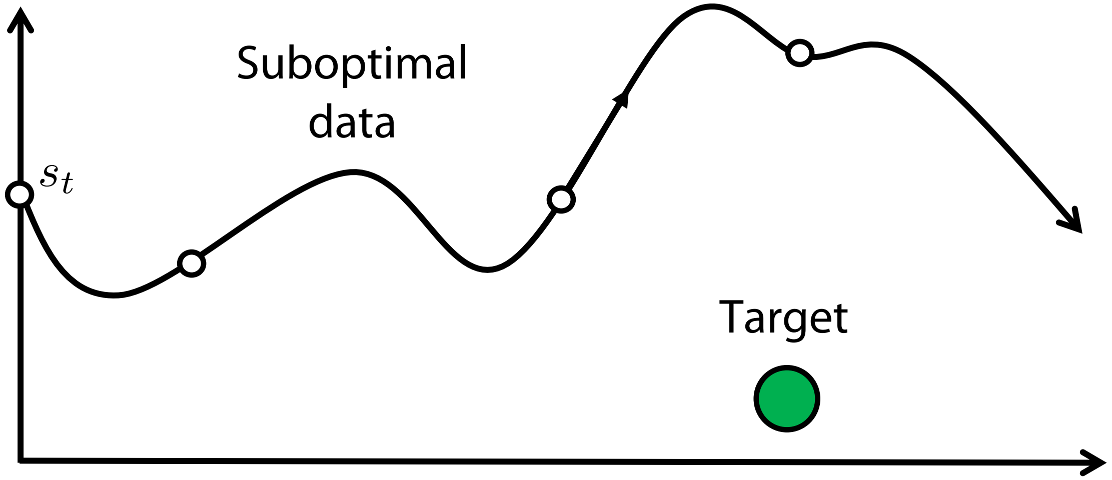


Duan et al

Let’s dive into a few

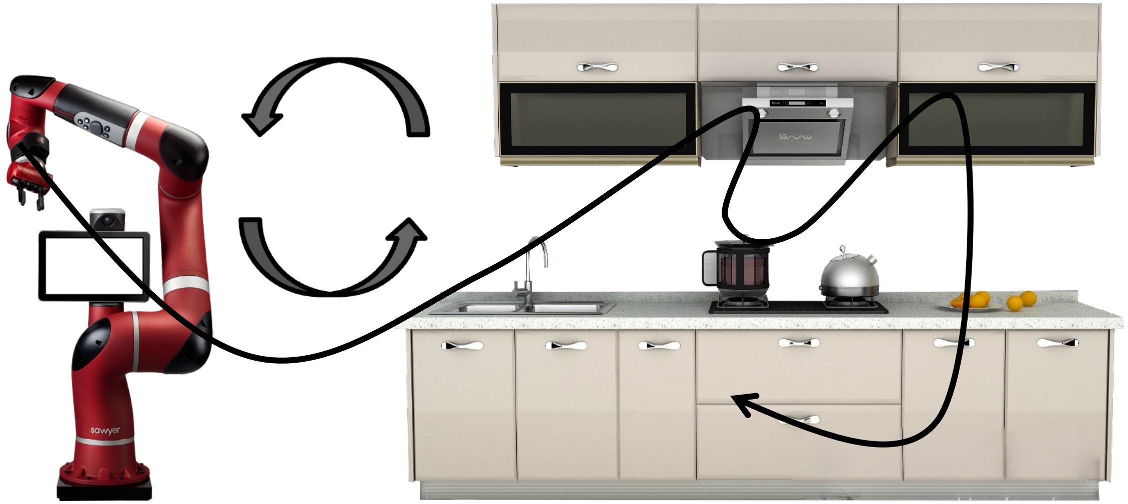
Accounting for Suboptimal Data

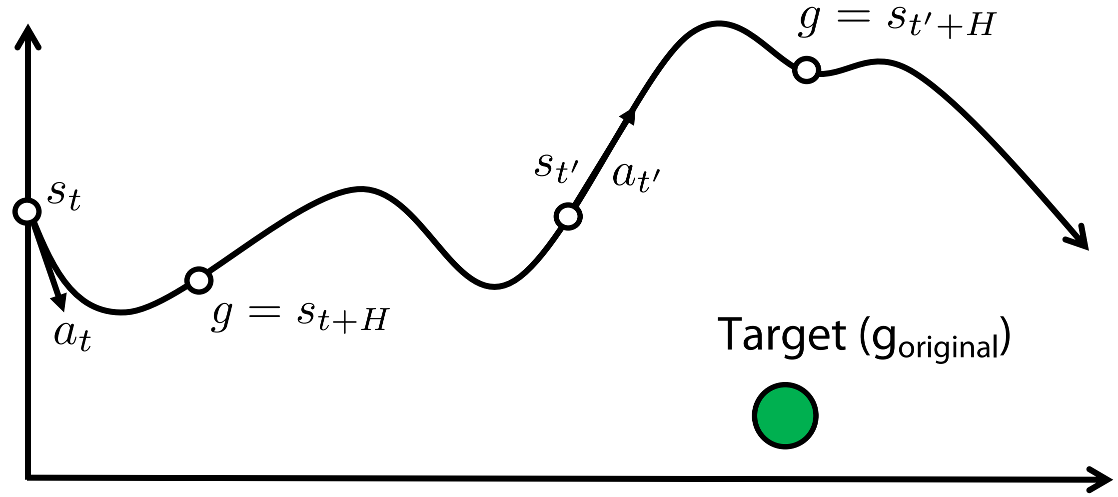




How can we use this suboptimal data,
despite not reaching the target?

Hindsight relabeling for Imitation Learning





Key insight: maybe the data is not bad, it's just been labeled for the wrong problem!

↓

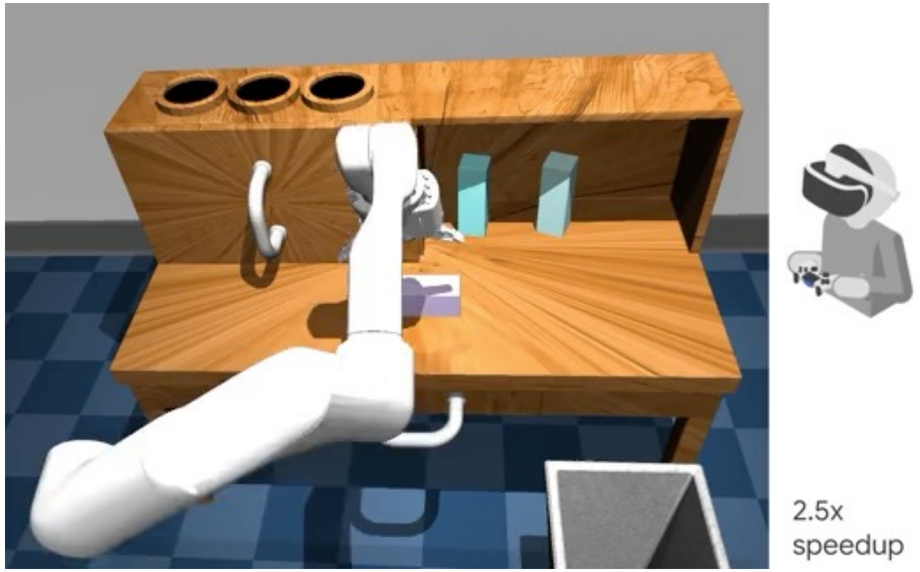
Relabel the right goal in "hindsight"

Learn a multi-goal policy $\pi_\theta(a|s,g)$

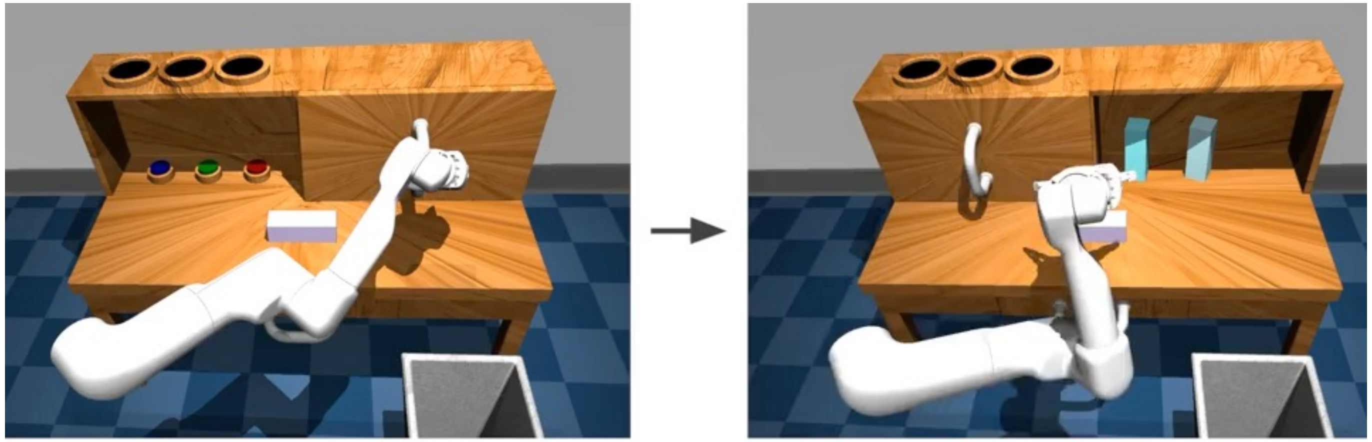
↓

Treat reached states as **optimal** goals

What does this result in?



Undirected play data



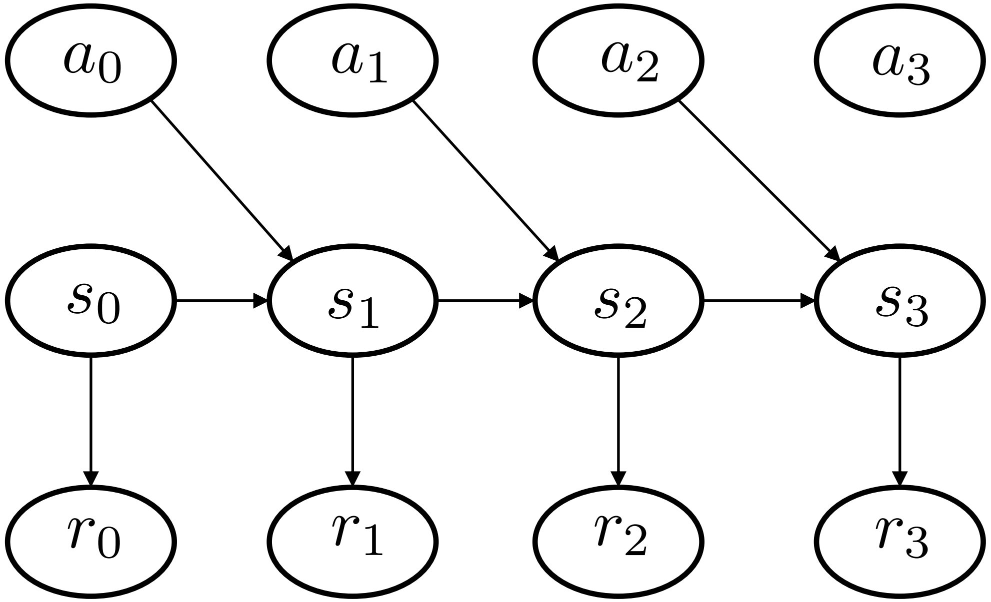
Goal

Single Play-LMP policy

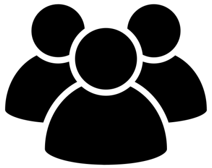
Goal-directed behavior

Dealing with non-Markovian demonstrators

Markov property $p(s_0, s_1, s_2, a_0, a_1, a_2) = p(s_0)p(a_0|s_0)p(s_1|s_0, a_0)p(a_1|s_1)p(s_2|s_1, a_1)p(a_2|s_2)$



Are human demonstrators Markovian?



$\pi_\theta(a|s)$

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

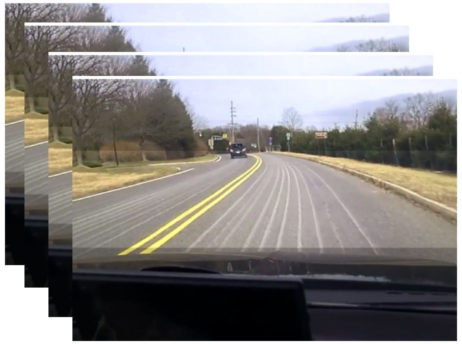
Not necessarily!

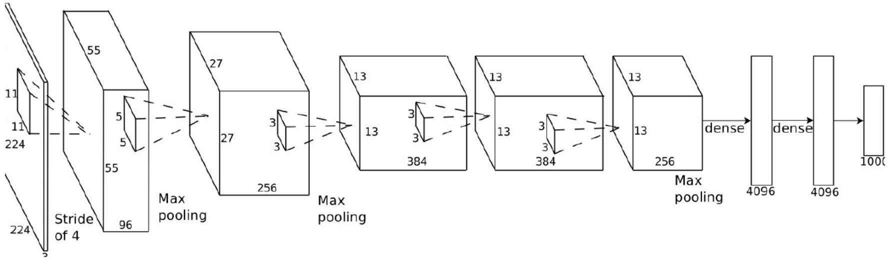
Humans often rely on history

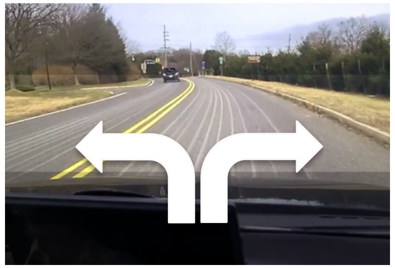
Mixtures of Markovian humans may not be Markovian

How can we deal with non-Markovian demonstrators?

Learn $\pi_\theta(a_t | s_t, s_{t-1}, \ldots, s_0)$







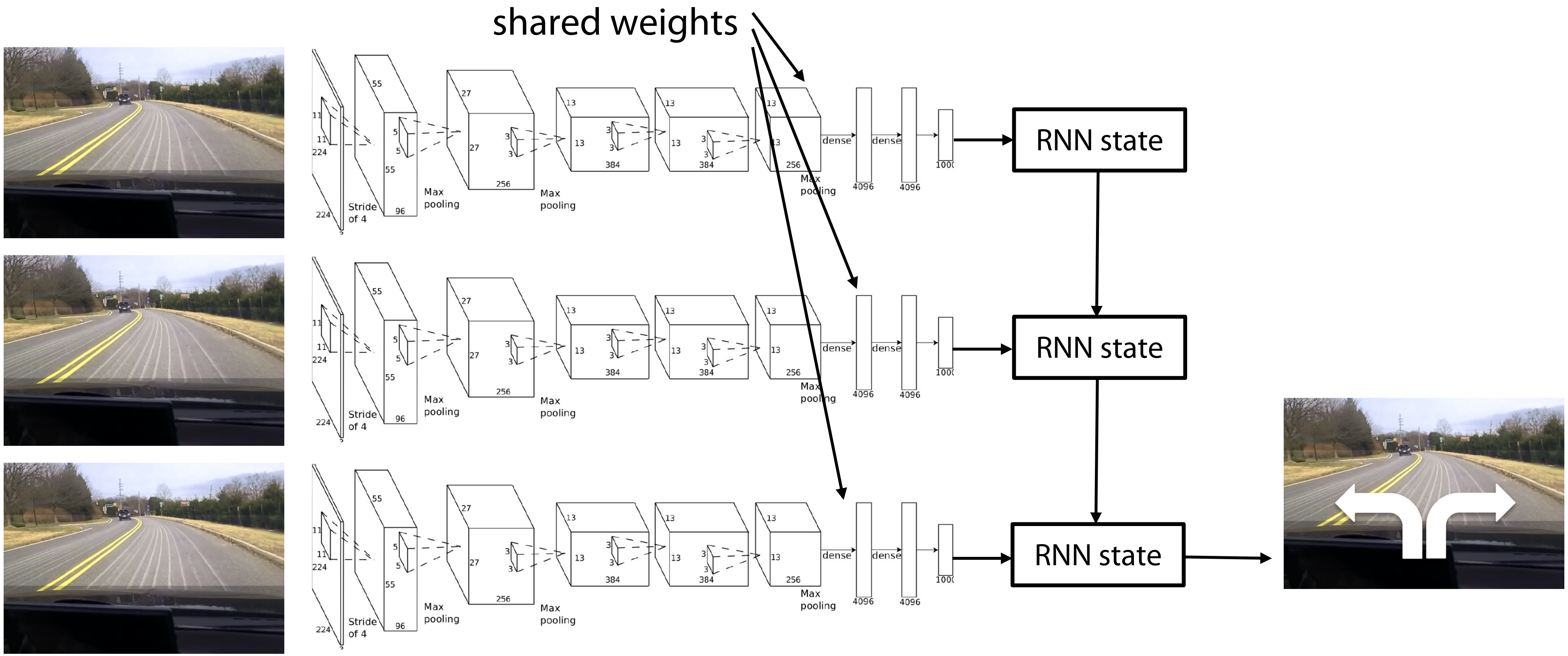
variable number of frames,
too many weights

Option 1: Stack all the past frames into a feedforward NN

Credit: Sergey Levine

How can we deal with non-Markovian demonstrators?

Learn $\pi_\theta(a_t | s_t, s_{t-1}, \ldots, s_0)$



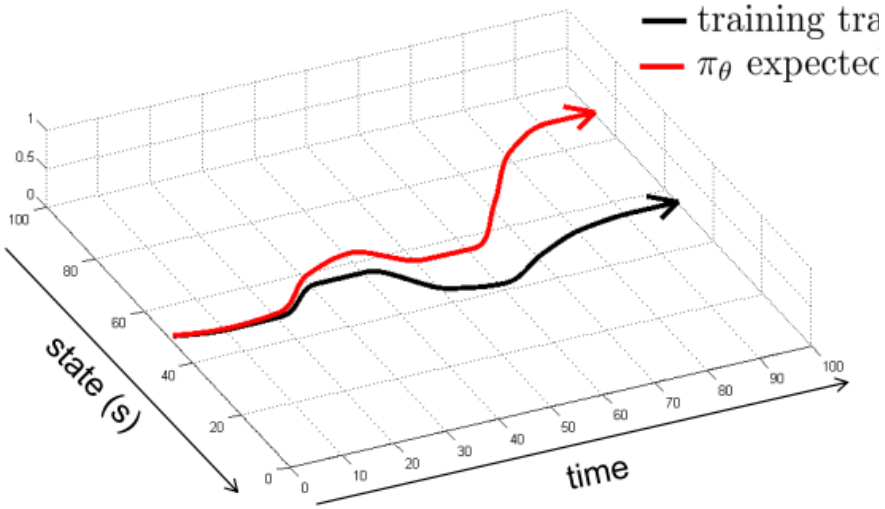
Option 2: Use a recurrent model (LSTM/transformer/RNN)

Credit: Sergey Levine

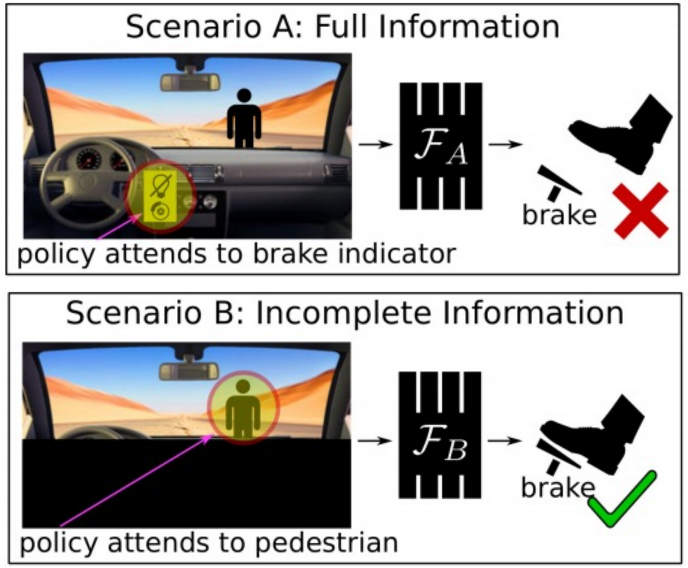
Why might this be challenging?

Learn $\pi_\theta(a_t | s_t, s_{t-1}, \ldots, s_0)$

Easier to go OOD



Learns spurious shortcut behaviors



Some cool imitation videos

1x and tesla humanoid robots

ALOHA and CherryBot Fine Manipulation

TRI Diffusion Policies

Perspectives on Imitation

$\mathbf{o}_t$

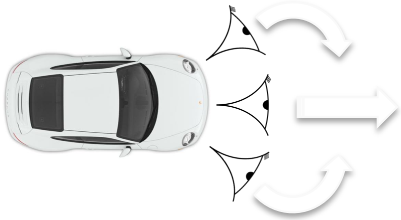
$\mathbf{a}_t$

training data

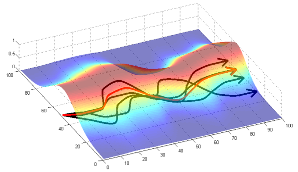
supervised learning

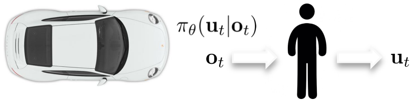
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

- Pros:
  - Easy to use, no additional infra
  - Can sometimes be unreasonably effective



- Cons:
  - Challenges of compounding error, multimodality
  - Doesn’t really generalize
  - Very expensive in terms of data collection!





Lecture outline

Recap: Imitation Learning + Why it is hard

↓

Multimodality and Underfitting in Imitation

↓

Compounding Error in Imitation

↓

Frontiers in Imitation