



Reinforcement Learning

Spring 2024

Abhishek Gupta

TAs: Patrick Yin, Qiuyu Chen



Logistics

- Paper reading teams should start getting together from next week
- Start finding project teams for final projects

Lecture outline

Recap: MDP formalism + why should we care?



Imitation learning: preliminaries and behavior cloning



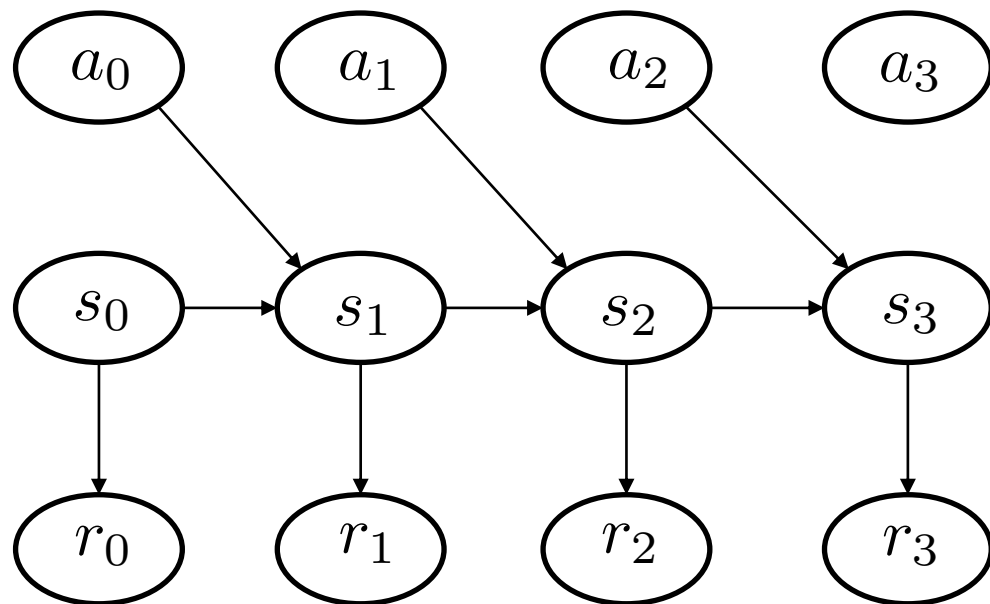
Multimodality and Underfitting in Imitation



Compounding Error 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

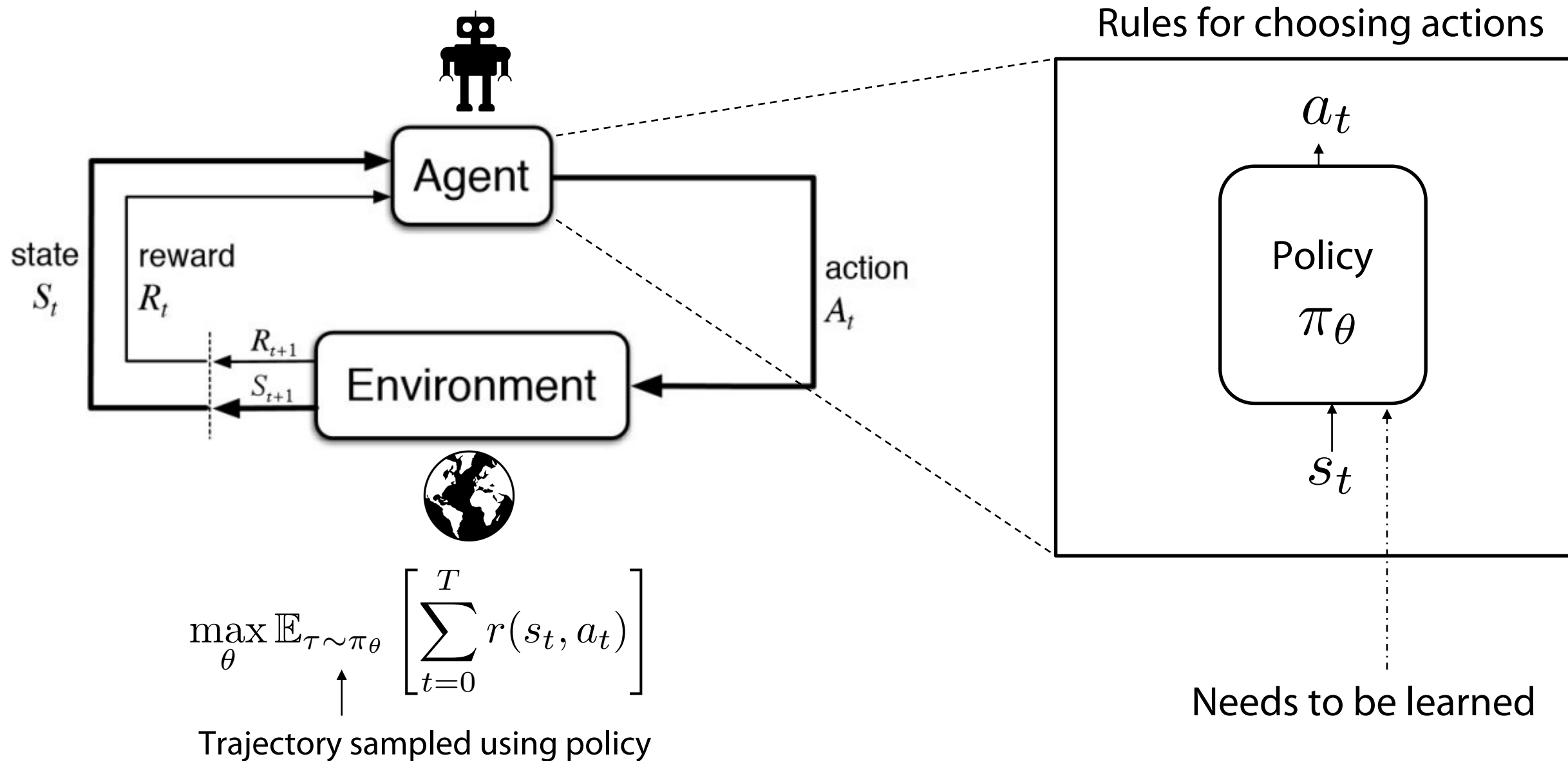
Rewards: \mathcal{R}

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

Markov property $p(s_1, s_2, s_3) = p(s_3|s_2)p(s_2|s_1)p(s_1)$

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

Reinforcement Learning Formalism



Unpacking the Expectation

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

Trajectory View - Ancestral sampling along MDP

Initial state	$\mathbb{E}_{\substack{s_0 \sim \rho_0(s) \\ a_0 \sim \pi_{\theta}(\cdot s_0) \\ s_1 \sim p(\cdot s_0, a_0) \\ a_1 \sim \pi_{\theta}(\cdot s_1) \\ s_2 \sim p(\cdot s_1, a_1) \\ \dots}} \left[\sum_{t=0}^T r(s_t, a_t) \right]$
Policy	
Dynamics	
Policy	
Dynamics	

Compact representation

$$\mathbb{E}_{\substack{s_0 \sim \rho_0(s) \\ a_t \sim \pi_{\theta}(\cdot | s_t) \\ s_{t+1} \sim p(s_{t+1} | s_t, a_t)}} \left[\sum_{t=0}^T r(s_t, a_t) \right]$$

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

Stationary View – sampling from stationary dist

$$d_t^{\pi}(s, a) = \mathbb{P}(s_t = s, a_t = a \mid s_0 \sim \rho_0, \forall i < t, a_i \sim \pi_{\theta}(\cdot | s_i), s_{i+1} \sim p(\cdot | s_i, a_i))$$

(Likelihood of being at state s , action a at time step t)

$$\mu_{\pi}^{\gamma}(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t d_t^{\pi}(s, a)$$

(Likelihood of being at state s , action a across **all** steps)

γ subsumed into E

$$\mathbb{E}_{(s,a) \sim \mu_{\gamma}^{\pi}(s,a)} \left[r(s, a) \right]$$

No sequential sampling

No sum over rewards

Some notation: Q-functions and V-functions

Estimate of how “good” a policy is – estimate of future returns under a policy π

Q-function

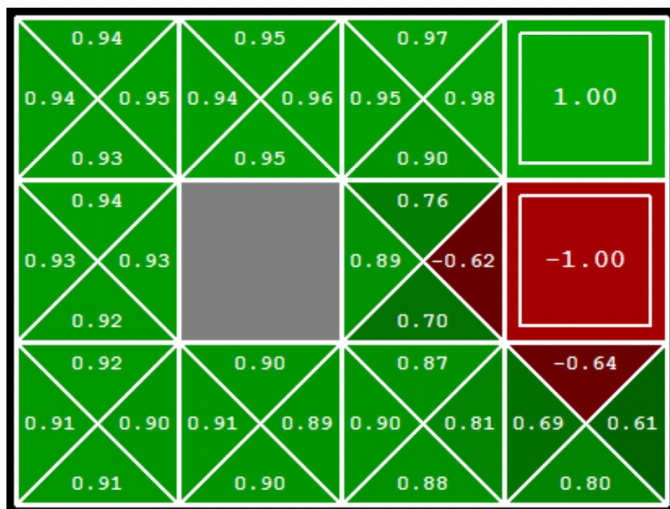
Take one action and then follow policy from s

$$Q^\pi(s, a) = \mathbb{E}_{\pi, p} \left[\sum_t r(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

V-function

Follow policy from s

$$V^\pi(s, a) = \mathbb{E}_{\pi, p} \left[\sum_t r(s_t, a_t) \mid s_0 = s \right]$$



$$V^\pi(s, a) = \mathbb{E}_{a \sim \pi(\cdot | s)} [Q^\pi(s, a)]$$

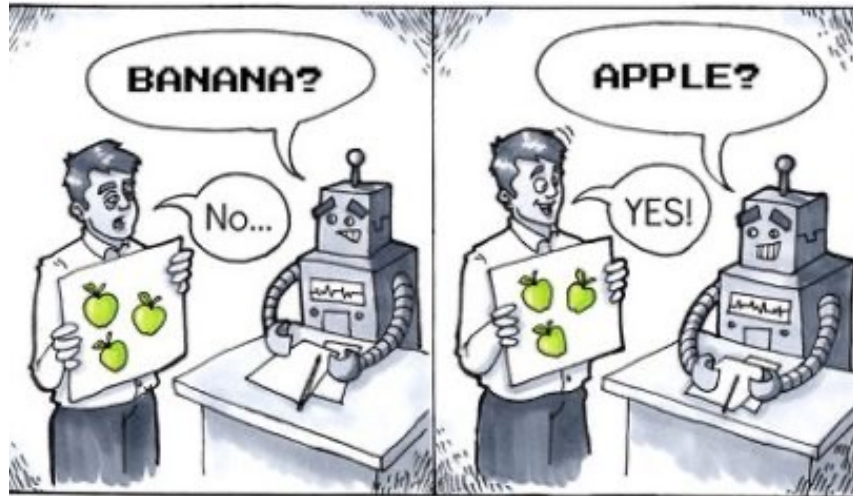
Will be useful soon!

$$J(\pi) = \mathbb{E}_{s \sim \rho_0(s)} [V^\pi(s)]$$

Average value over initial states

Ok so is this just supervised learning?

Supervised learning aims to maximize likelihood of observed data under the model



Supervised Learning

$$\max_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} [\log \hat{p}_{\theta}(y|x)]$$

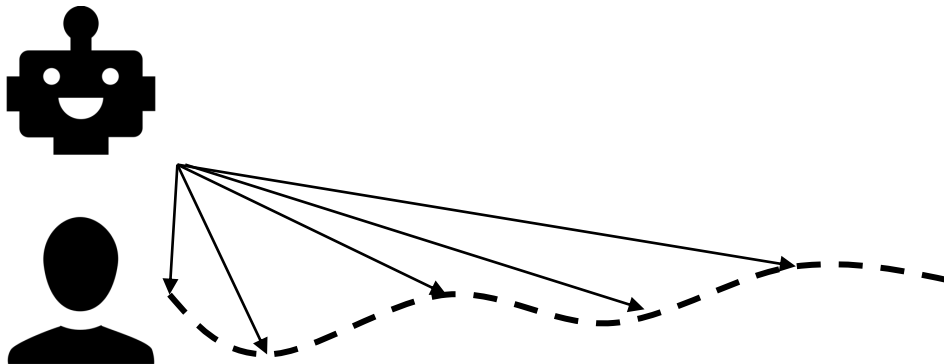
Why is this not just supervised learning?

Supervised Learning

$$\max_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} [\log \hat{p}_{\theta}(y|x)]$$

Sampling from expert

$$D_{\text{KL}}(p^* || p_{\theta}) \quad \text{IID}$$

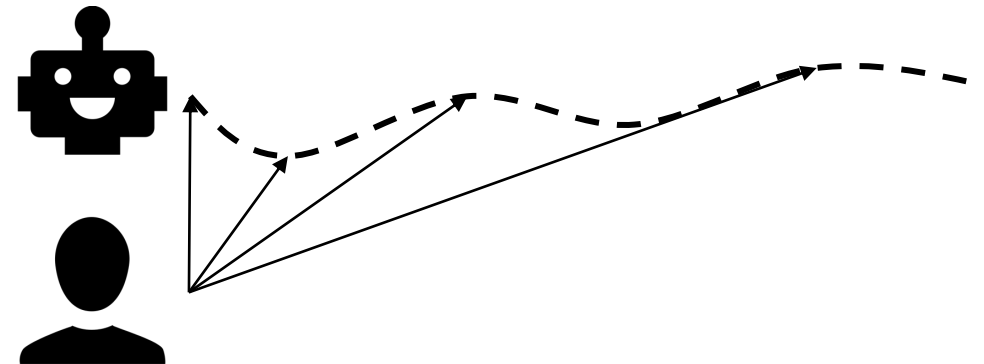


Reinforcement Learning

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

Sampling from policy

$$D_{\text{KL}}(p_{\theta} || p^*) \quad \text{Non-IID}$$



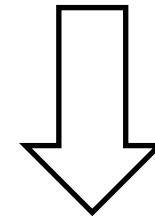
Why is this not just supervised learning?

Supervised Learning

$$\max_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} [\log \hat{p}_{\theta}(y|x)]$$

The resulting paradigms are different in many ways:

1. Optimization and learning dynamics
2. Balancing exploration and exploitation



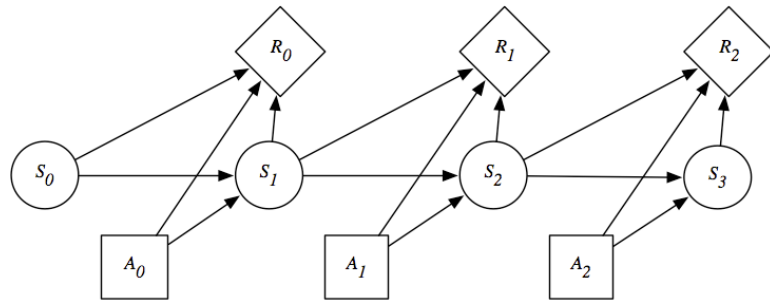
Reinforcement Learning

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

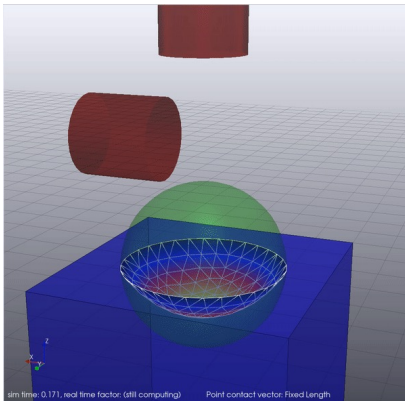
But many overlapping tools! In fact often we try to convert RL into a supervised problem

Ok so why should we care about RL?

Solves sequential decision making problems



Has black-box assumptions



Enables continual improvement

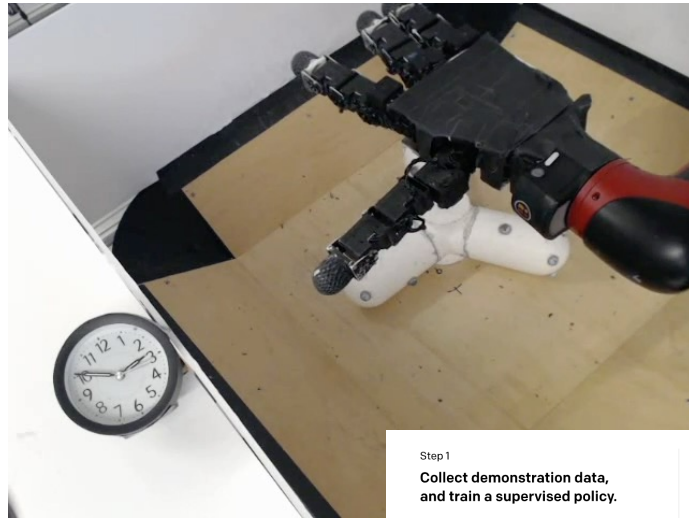


Reduces burden of human data collection



Applications of RL: Robotics/LLMs/Science/Games

RL can enable robotic learning of hard to specify/script behaviors in the presence of contact



Step 1
Collect demonstration data,
and train a supervised policy.

A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



This data is used
to fine-tune GPT-3
with supervised
learning.

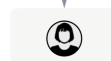


Step 2
Collect comparison data
and train a reward model

A prompt and
several model
outputs are
sampled.



A labeler ranks
the outputs from
best to worst.



This data is used
to train our
reward model.



A new prompt is
sampled from
the dataset.

The policy
generates
an output.

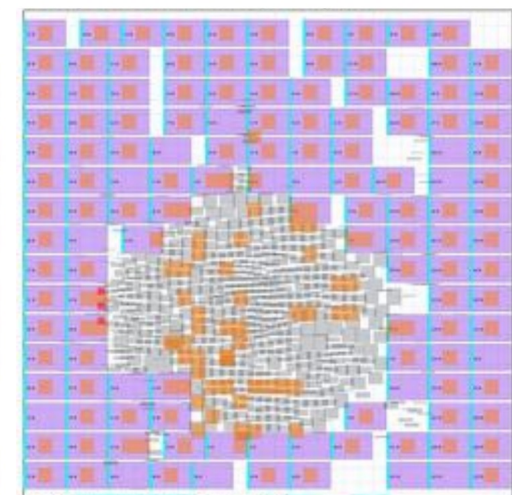
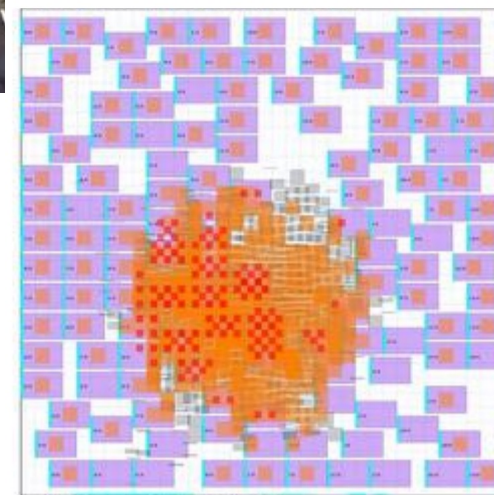
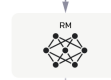
The reward model
calculates a
reward for
the output.

The reward is
used to update
the policy
using PPO.

Write a story
about frogs

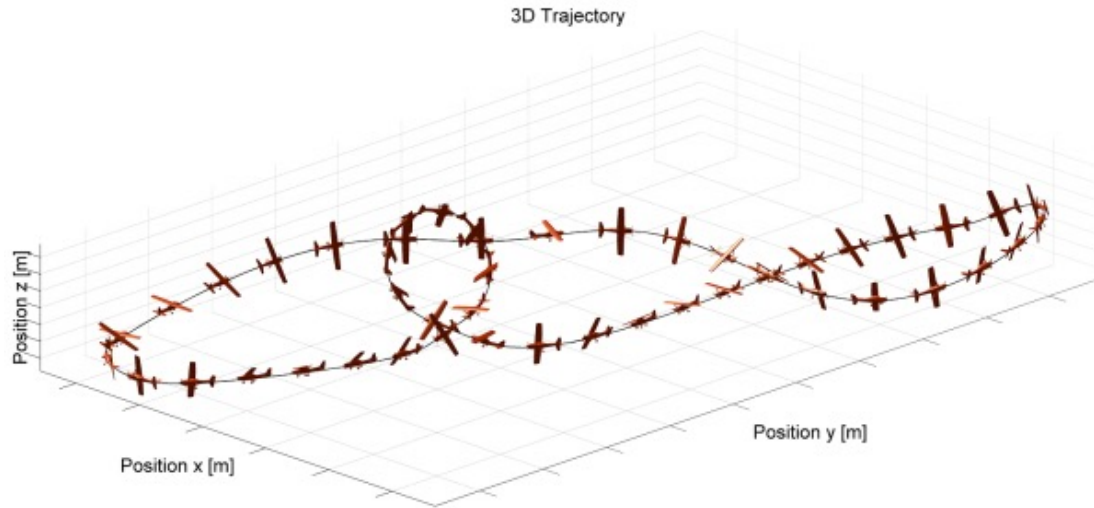


Once upon a time...

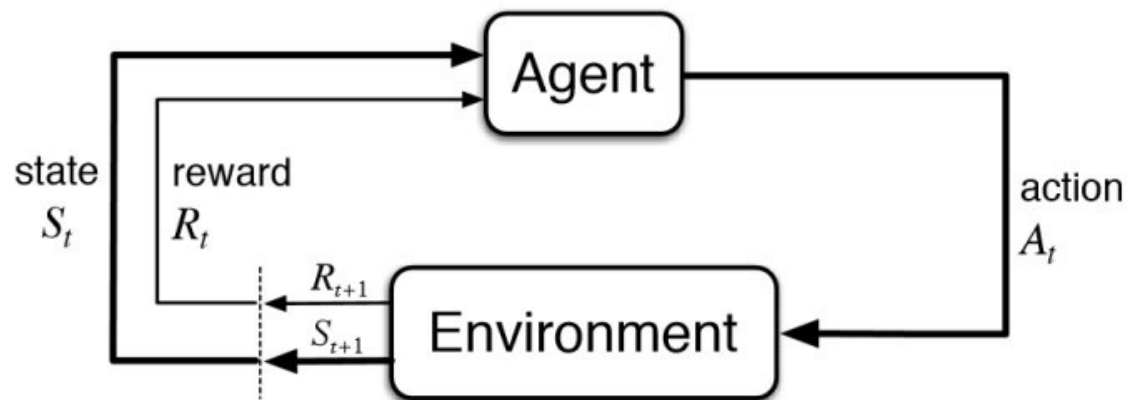


Where is Reinforcement Learning not useful?

Not the right call for very safety-critical, repetitive applications



So is sequential decision making = RL?

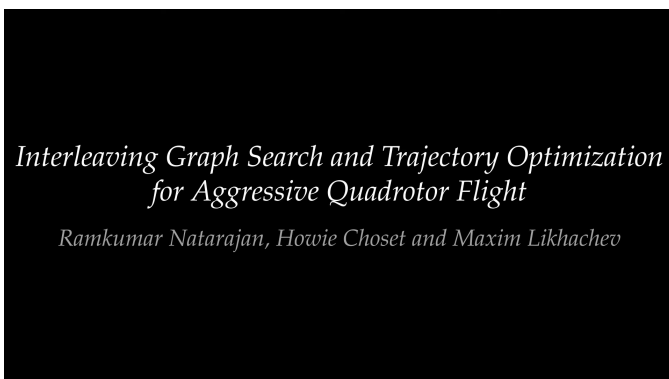


We conflated sequential decision making and RL!

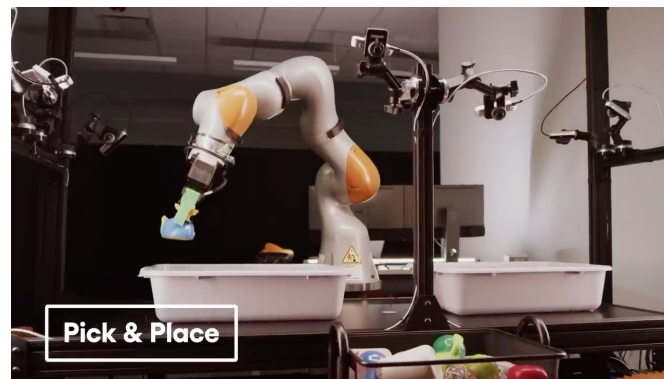
RL is sequential decision making under a particular set of assumptions:

1. Sampling access to the environment
2. Access to reward
3. Goal-directed behavior

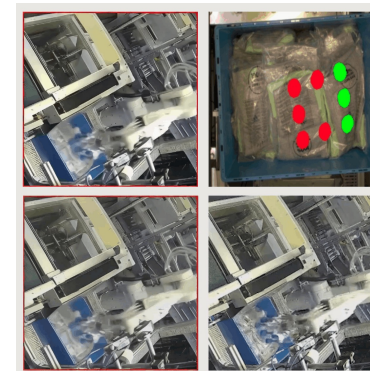
Trajectory optimization/planning



Imitation Learning



Unsupervised Decision Making



Trajectory Optimization

Sequential decision making with "known" models

*Interleaving Graph Search and Trajectory Optimization
for Aggressive Quadrotor Flight*

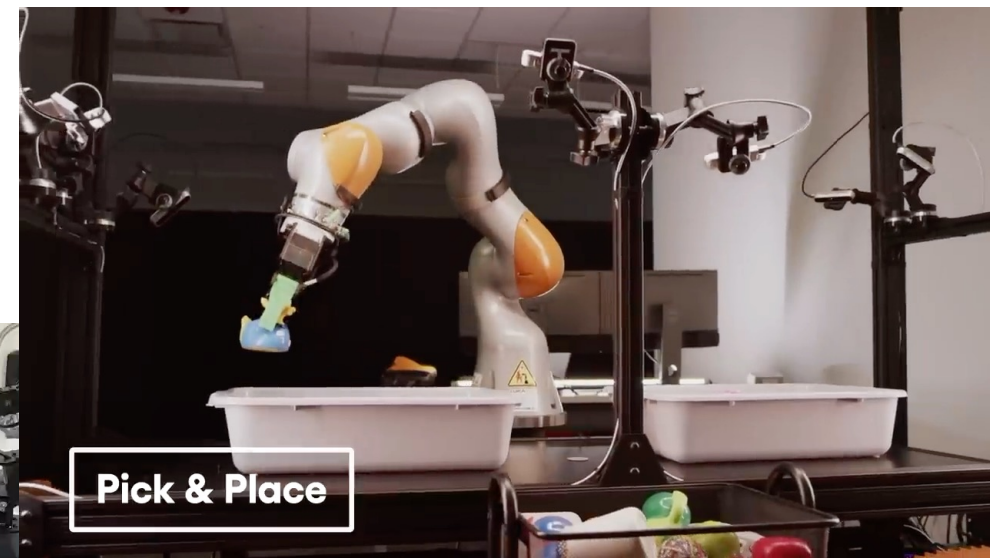
Ramkumar Natarajan, Howie Choset and Maxim Likhachev

We combine RRT and local smoothing of contact dynamics to generate complex contact-rich manipulation plans.

May be hard to construct perfect, known models

Imitation Learning

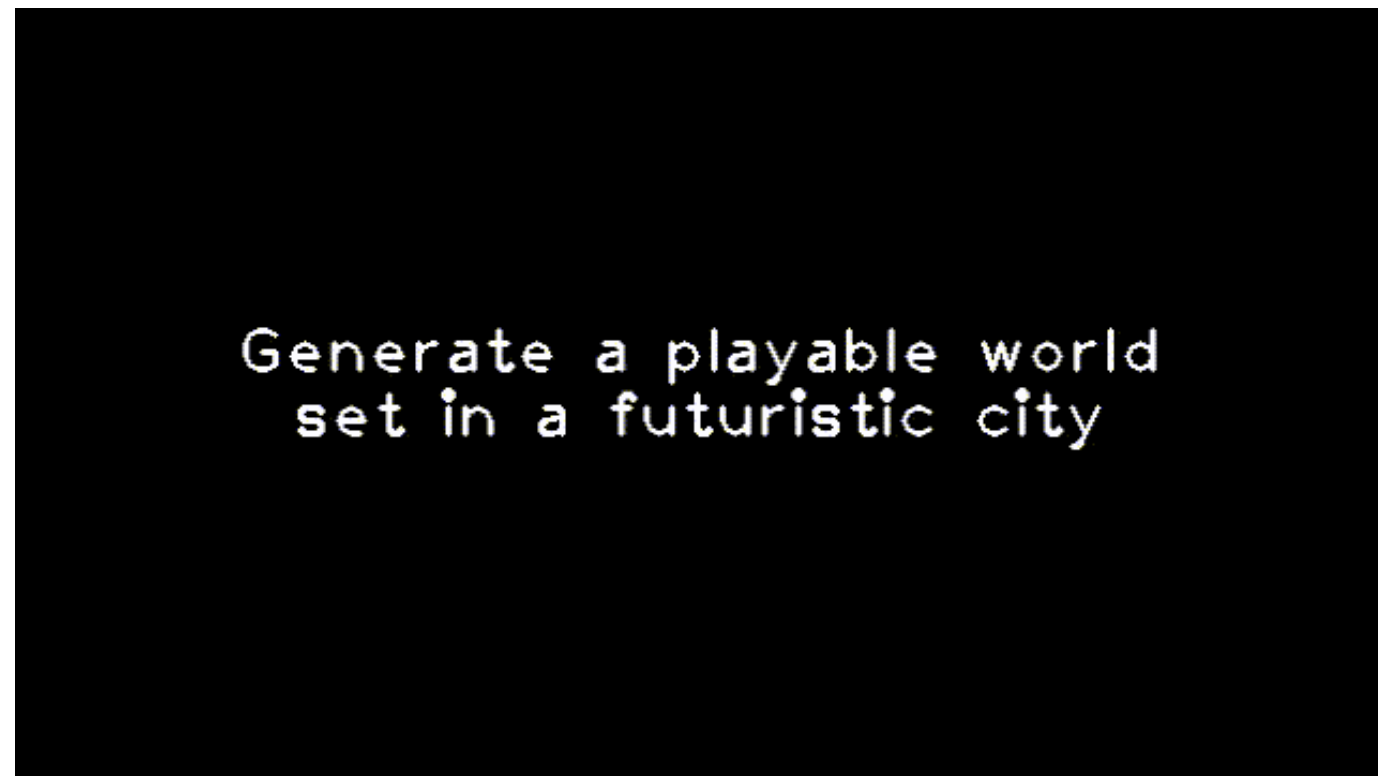
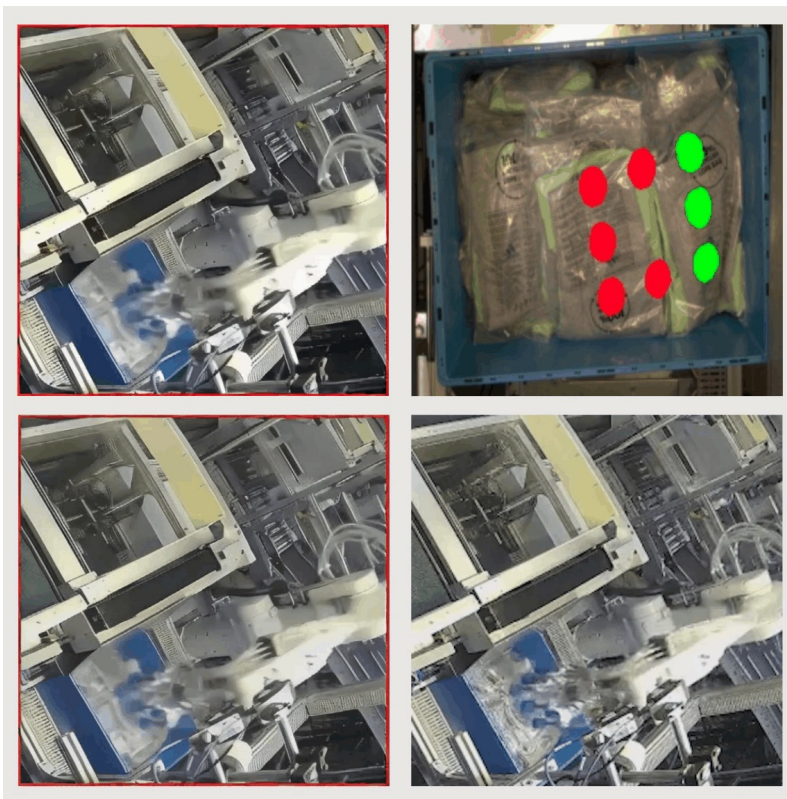
Sequential decision making provided expert data



Often called learning from demonstrations

Self-Supervised Prediction of the World

Sequential decision making without reward – self-supervised prediction



Often called model-based RL

How should we think about designing effective RL algorithms?

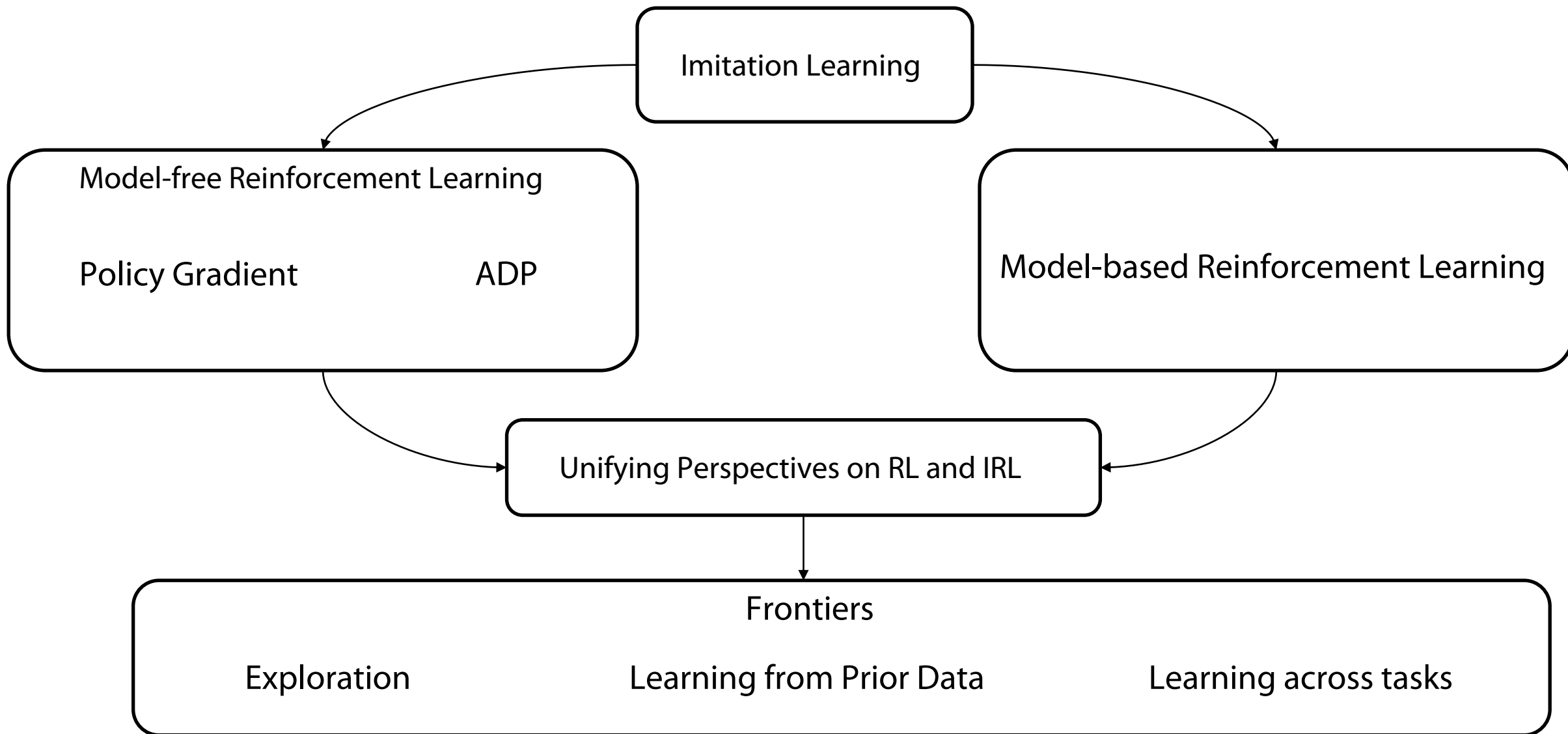


Easy to specify
objectives

Stable performant
optimization algorithms

Efficient **data** collection

Class Structure



Lecture outline

Recap: MDP formalism + why should we care?



Imitation learning: preliminaries and behavior cloning

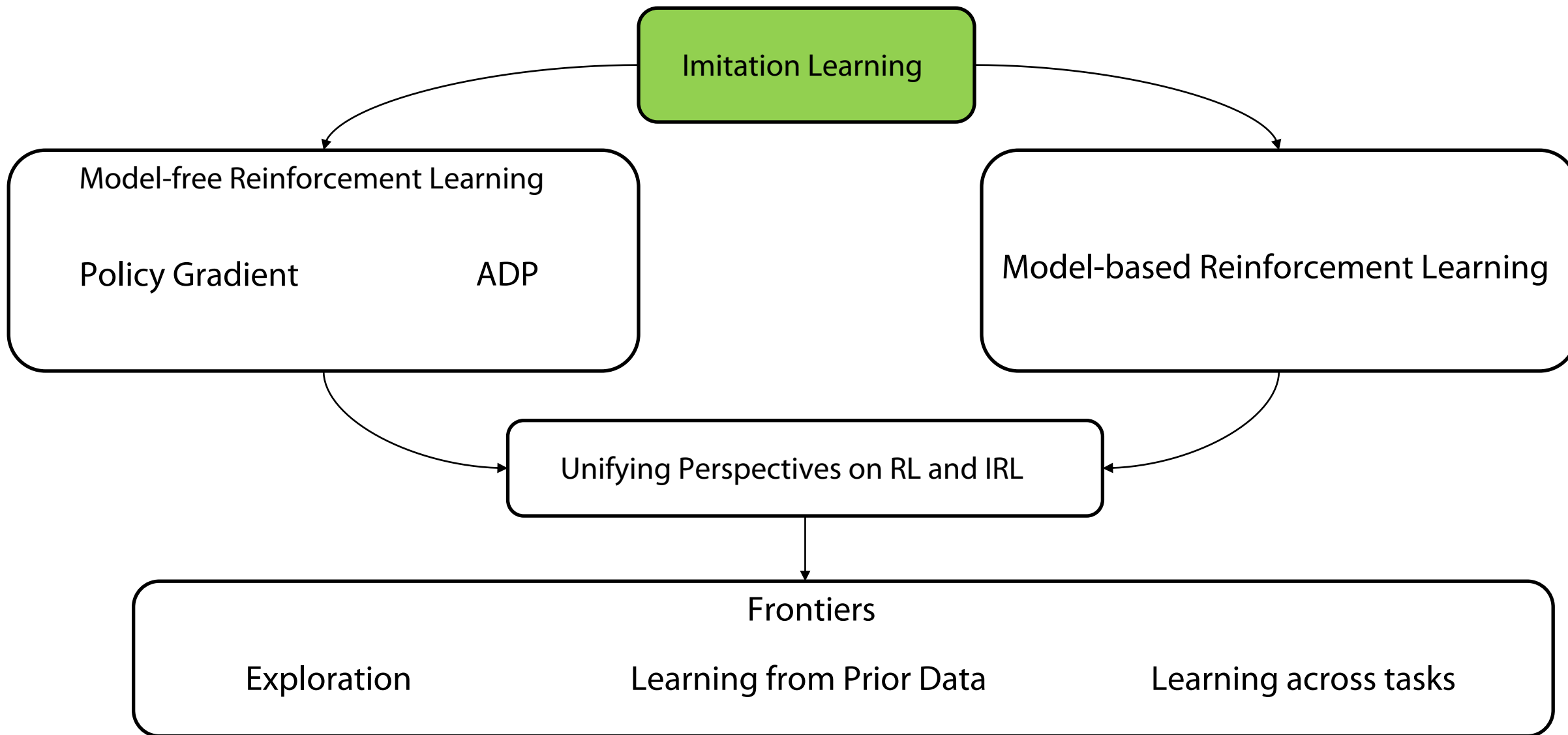


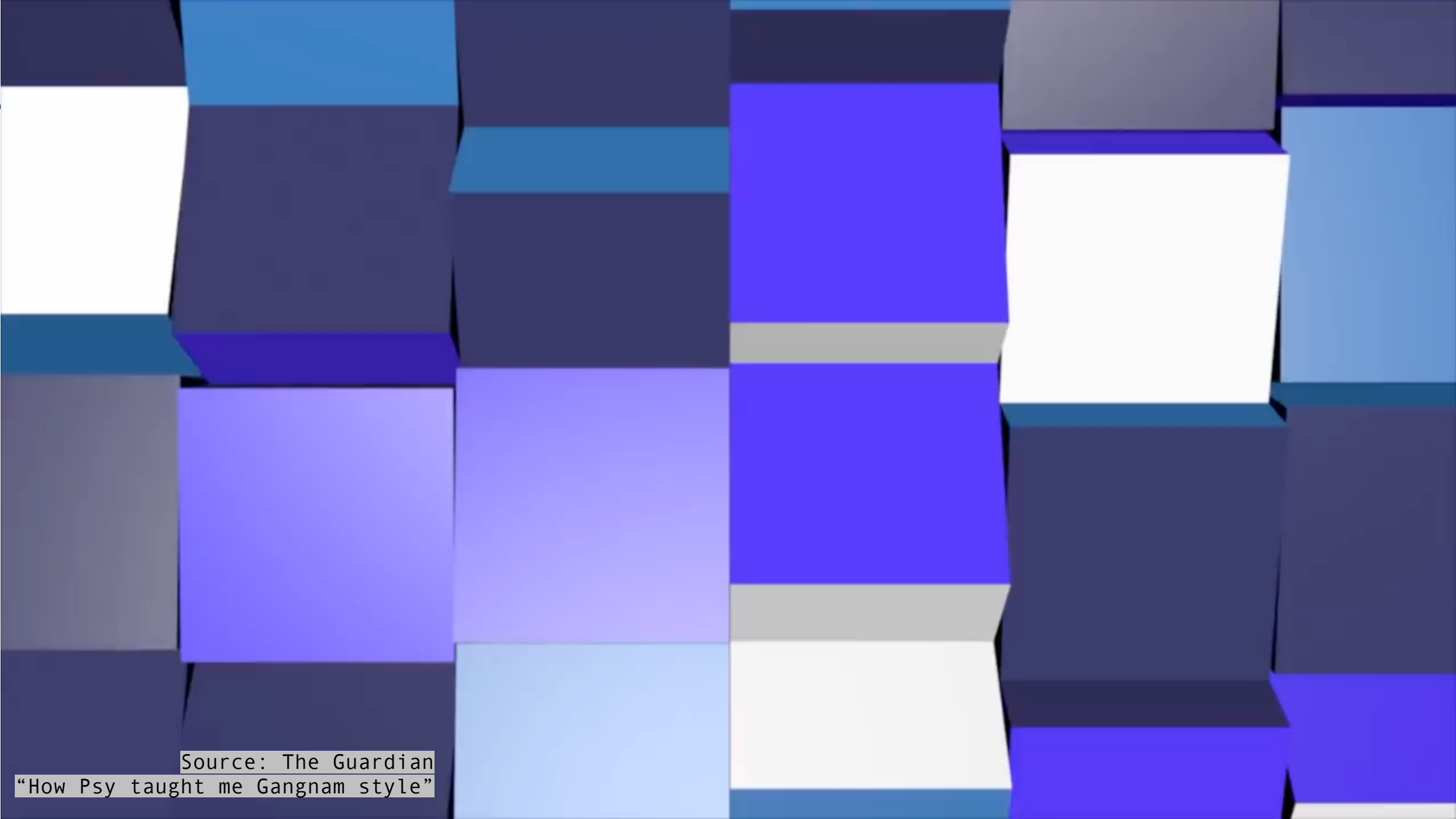
Multimodality and Underfitting in Imitation



Compounding Error in Imitation

Class Structure





Source: The Guardian

“How Psy taught me Gangnam style”

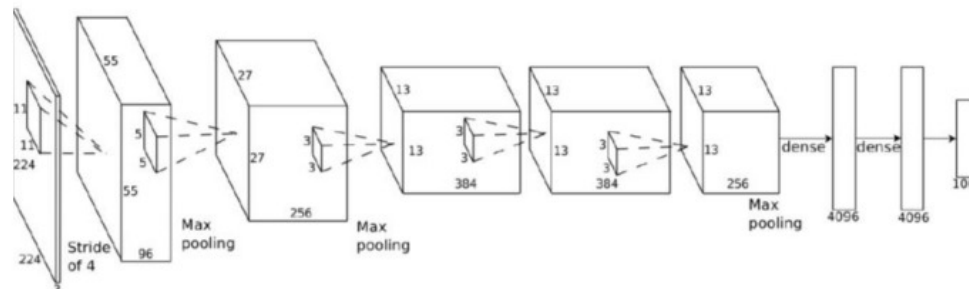
Imitation Learning: Intuition

Given: Demonstrations of optimal behavior

$$\mathcal{D} = \{s_0^i, a_0^i, s_1^i, a_1^i, \dots, s_T^i, a_T^i\}_{i=1}^N$$

Goal: Train a policy to mimic the demonstrator

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

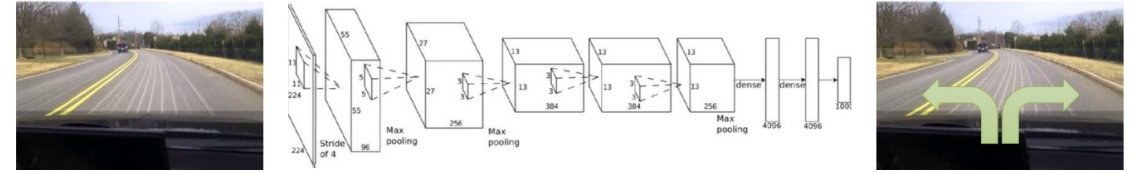


Pros: No rewards, online experience needed (?)

Why would we do this?

Given: Demonstrations of optimal behavior

Goal: Train a policy to mimic the demonstrator



Pros:

- ⊕ Avoids need for rewards, exploration
- ⊕ Natural way to do task specification
- ⊕ Can work well in practice

Cons:

- ⊖ Requires expert data, can be expensive
- ⊖ Cannot get better on deployment
- ⊖ Struggles on long horizon tasks

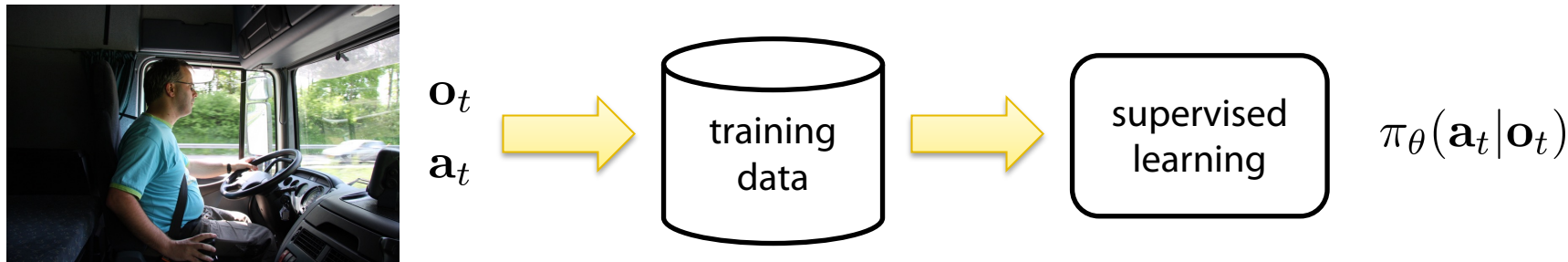
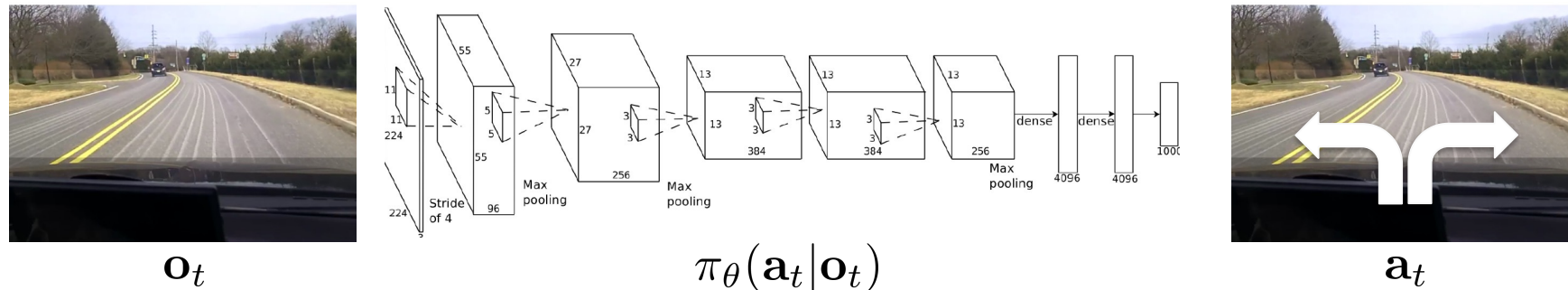
Idea 1: Imitation Learning via Supervised Learning

Given: Demonstrations of optimal behavior

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

Goal: Train a policy to mimic the demonstrator

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



Idea 1: Imitation Learning via Supervised Learning

Given: Demonstrations of optimal behavior

Goal: Train a policy to mimic the demonstrator

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

Discrete vs continuous

```
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()
```

Maximum likelihood

Idea 1: Imitation Learning via Supervised Learning

Given: Demonstrations of optimal behavior

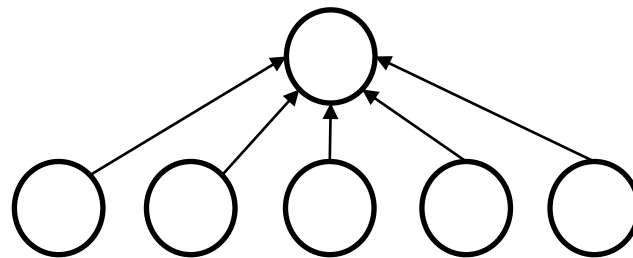
Goal: Train a policy to mimic the demonstrator

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

Tabular

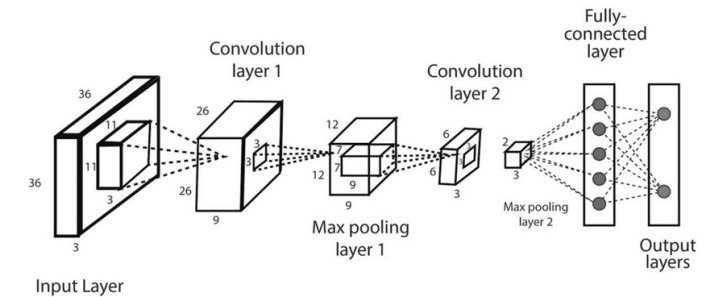
8.67	8.93	9.11	9.30	9.42
8.49		9.09	9.42	9.68
8.33		1.00		10.00
7.13	5.04	3.15	5.68	8.45
-10.00	-10.00	-10.00	-10.00	-10.00

Linear



$$\pi(a|s) = \langle \phi(s, a), w \rangle$$

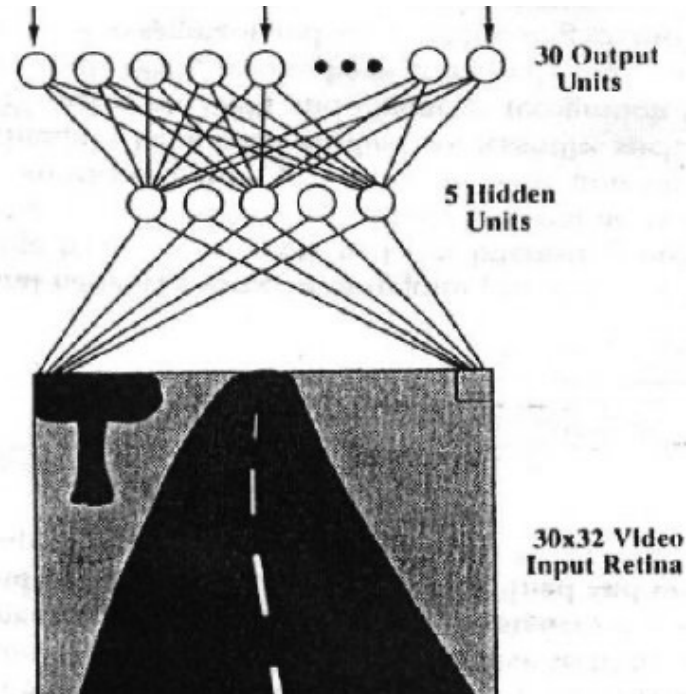
Arbitrary function approx



In practice, amounts to simple gradient based training with backpropagation

The original deep imitation learning system

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

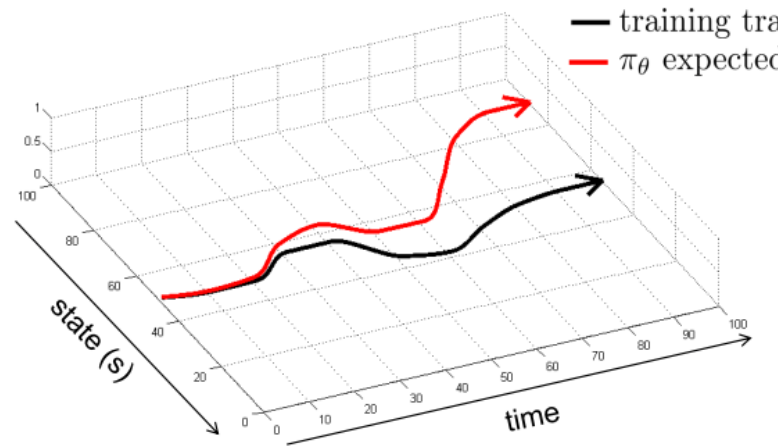


Where we are in 2024?



So does behavior cloning really work?

- Imitation Learning \neq Supervised Learning



Compounding error!

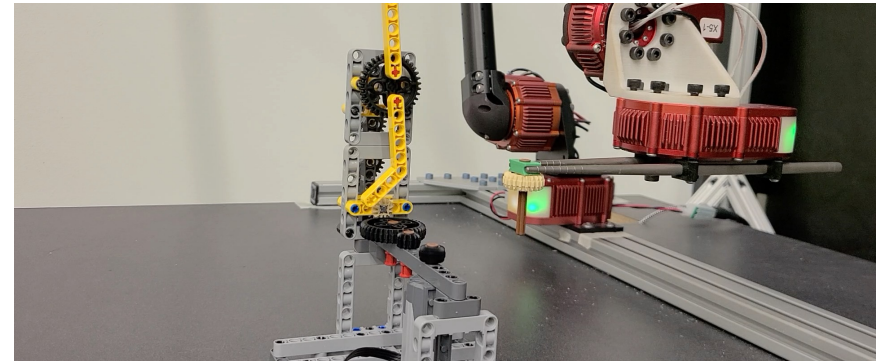
$$\arg \max_{\theta} \mathbb{E}_{(s^*, a^*) \sim \mathcal{D}} [\log \pi_{\theta}(a^* | s^*)] \qquad \mathbb{E}_{(s, a) \sim \rho(\pi)} [\mathbf{1}(a = a^*)]$$



Not the same!

So does behavior cloning really work?

- Fails in practice as well!



What do we actually want?

- Imitation Learning can be formalized as matching the expert

(cost for generating an action
different than the expert)

$$c(s_t, a_t) = \begin{cases} 0, & \text{if } a_t = \pi^*(s_t), \\ 1, & \text{otherwise} \end{cases}$$

Measure deviation from expert
actions when the policy is rolled out

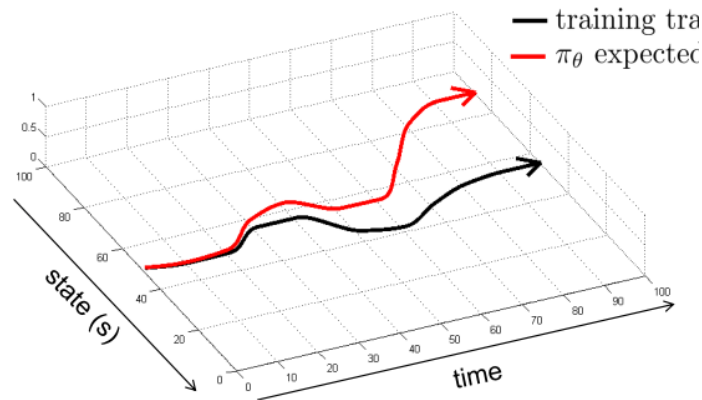
$$\mathbb{E}_{(s_t, a_t) \sim p_{\pi_\theta}(s_t, a_t)} [c(s_t, a_t)]$$

How bad is behavior cloning?

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

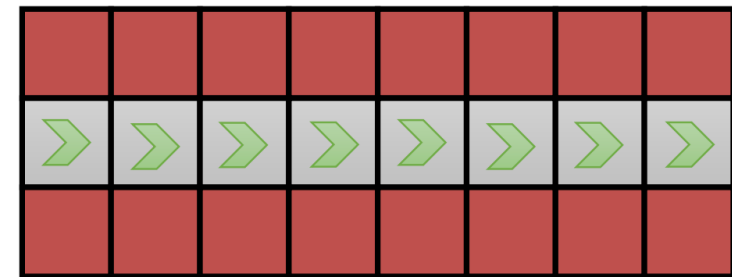
How well does BC do?: Intuition

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



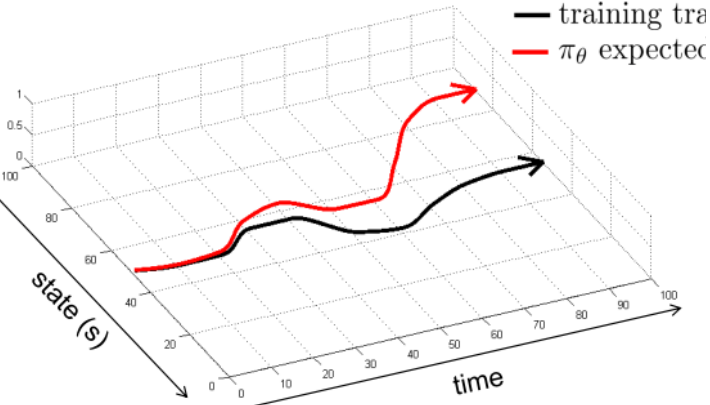
$$\underbrace{\mathbb{E} \left[\sum_t c(s_t, a_t) \right]}_{O(\epsilon H^2)} \leq \epsilon H + \dots + \dots$$

Union bound

Let's try and understand where the problem lies?

Behavior cloning has challenges in both theory and practice

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



Underfitting

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

Compounding error

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

Lecture outline

Recap: MDP formalism + why should we care?



Imitation learning: preliminaries and behavior cloning



Multimodality and Underfitting in Imitation

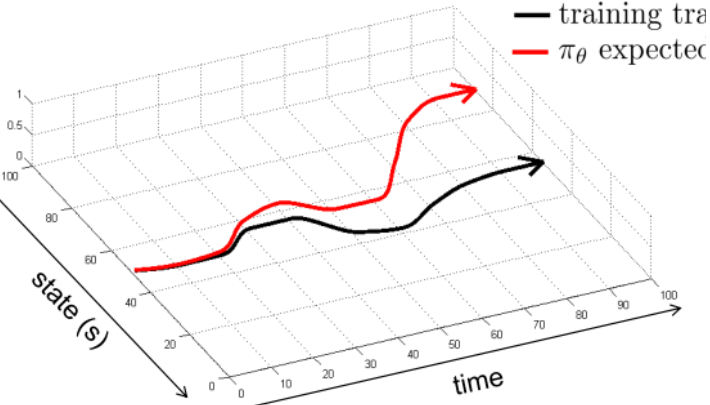


Compounding Error in Imitation

Let's try and understand where the problem lies?

Behavior cloning has challenges in both theory and practice

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



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

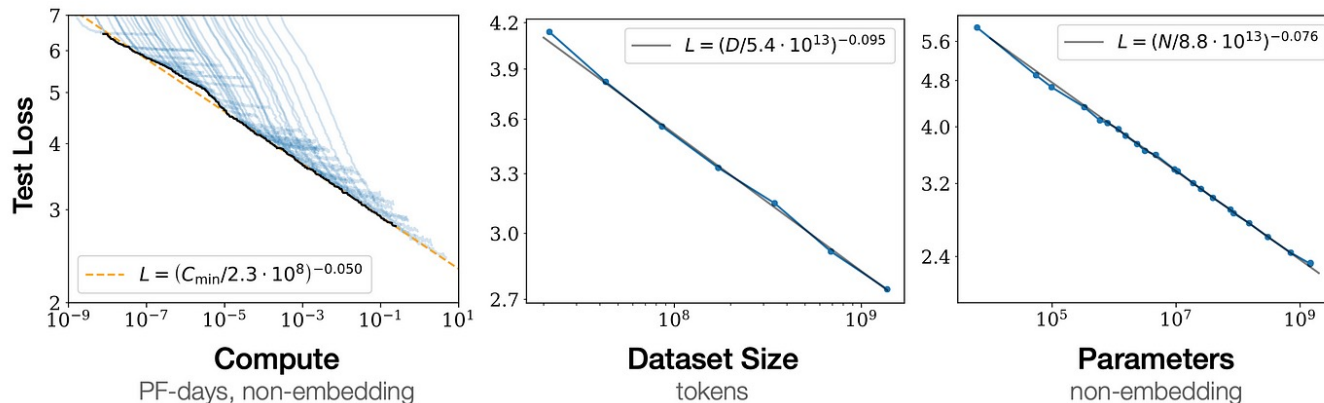
$$\sum_t \mathbb{E}_{(s_t, a_t) \sim p_{\pi_\theta}(s_t, a_t)} [c(s_t, a_t)] \leq O(\epsilon H^2)$$

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

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?

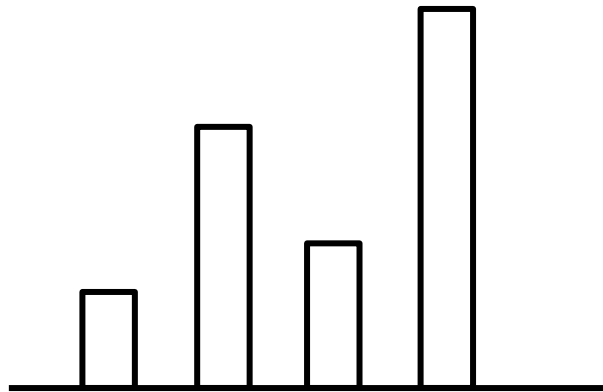


Kind of, but there's a fundamental problem!

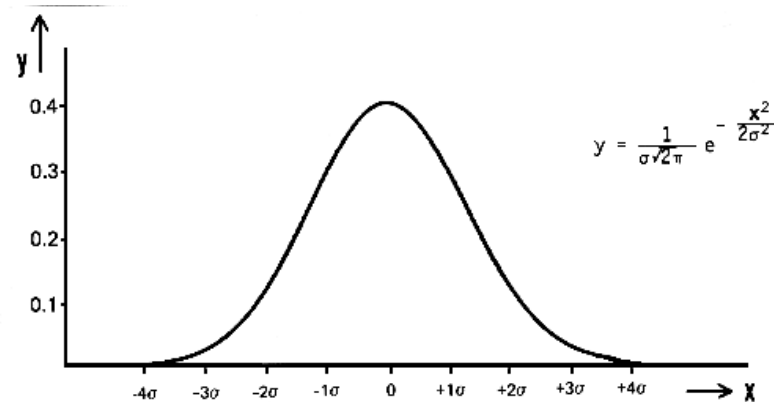
Distributional Expressivity

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

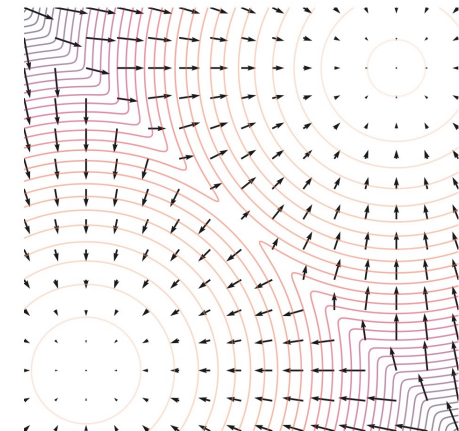
Categorical



Gaussian



Diffusion policy

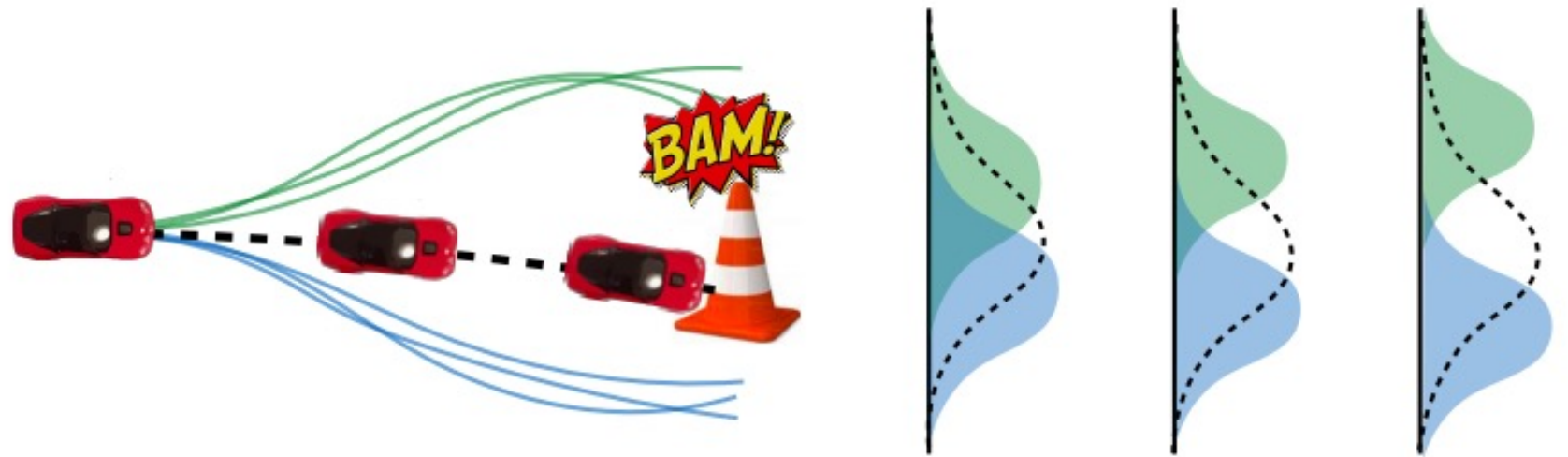
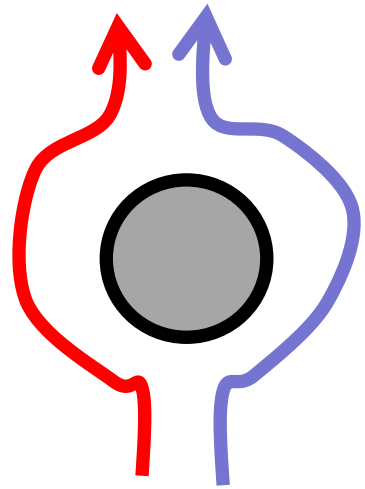


Tradeoff between expressivity and tractability

How does this reflect on imitation learning?

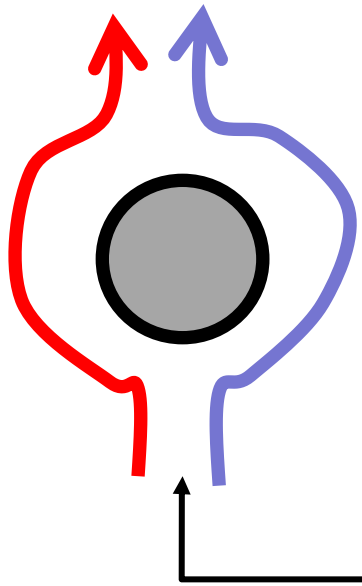
Let us consider a case with Gaussian policy

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



A combination of distributional expressivity and objective lead to mode averaging

Let's take a closer look at the objective



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

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

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

Leads to mode averaging

Forward KL divergence

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

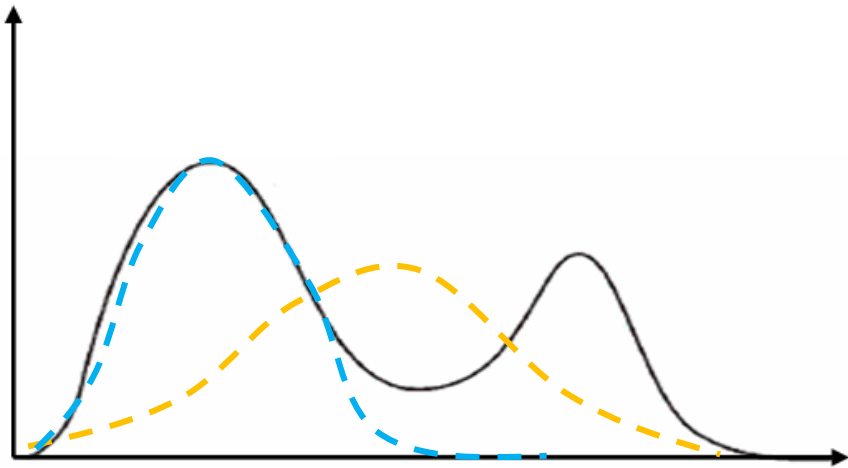
Effects of choice of f-divergence on behavior

Different divergences lead to different properties

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

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 Use a richer distribution class!
(Change π_θ)

Using alternative f-divergences: Reverse KL

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

Imitation Learning as f-Divergence Minimization

Liyiming Ke¹, Sanjiban Choudhury¹, Matt Barnes¹, Wen Sun², Gilwoo Lee¹,
and Siddhartha Srinivasa¹

Go read this!

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

(Intractable) (Tractable – GAN style optimization)

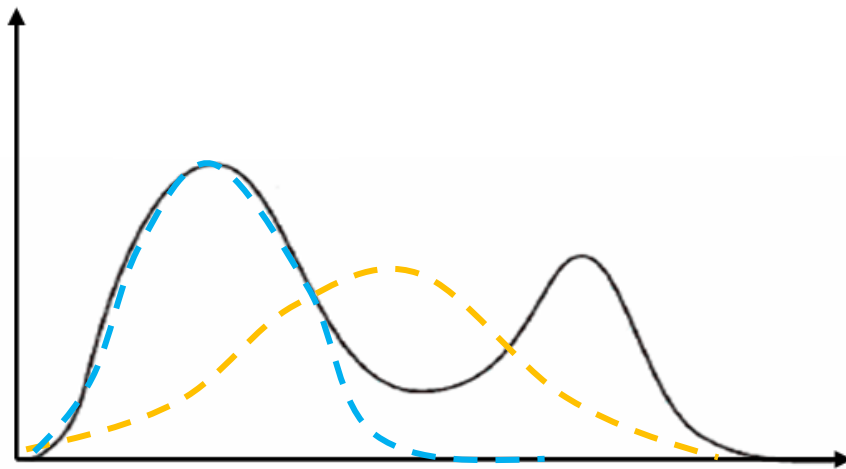
Effects of choice of f-divergence on behavior

Different divergences lead to different properties

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

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?

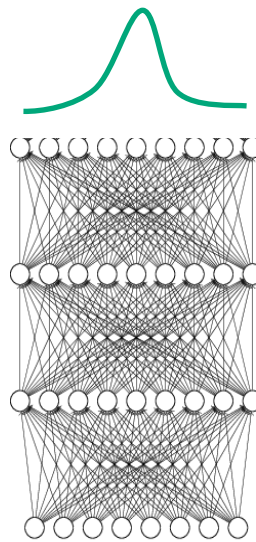
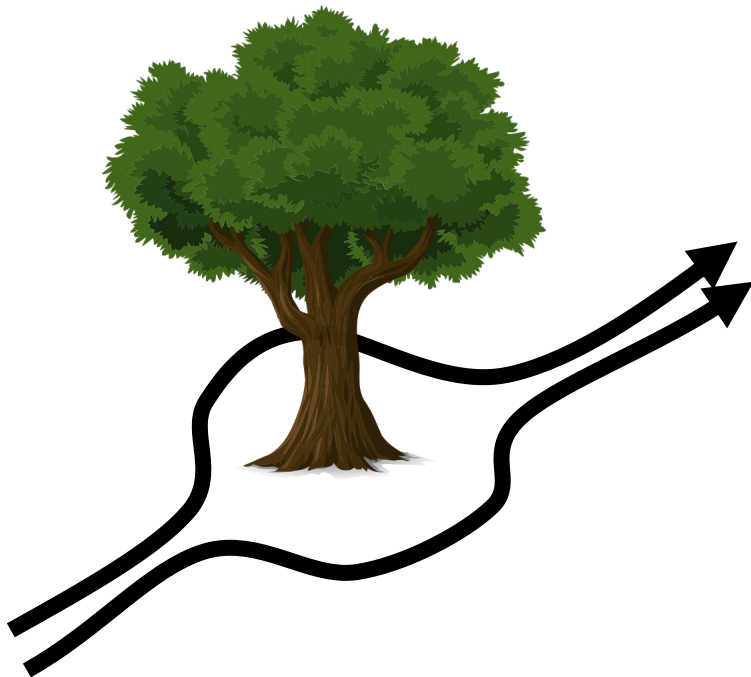
Use a different f-divergence!
(Change f)

or

Use a richer distribution class!
(Change π_θ)

Using Richer Policy Distribution Classes

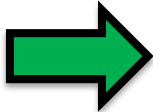
Multimodal behavior → use more **expressive** probability distributions, no mode averaging issues



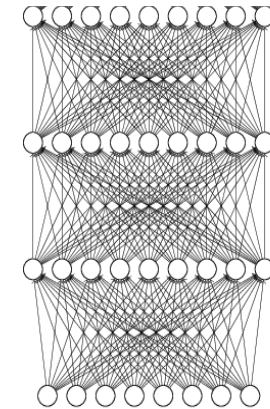
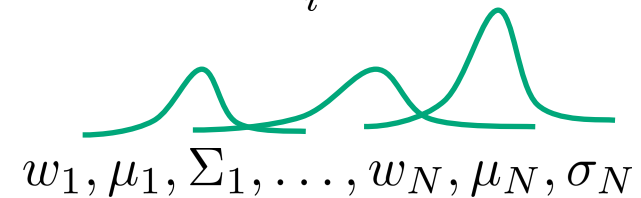
1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. ...



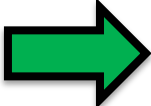
Why might we fail to fit the expert?

- 
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. ...

Why does this work?

first step: $p(a_{t,0}|\mathbf{s}_t)$

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

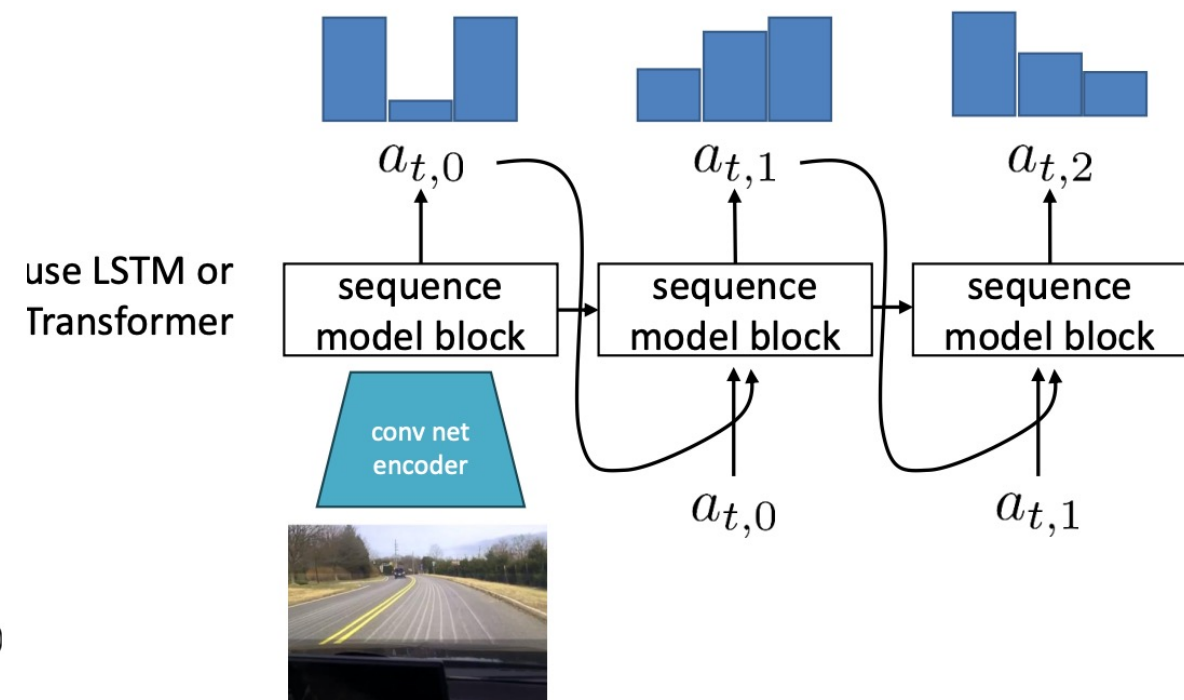
third step: $p(a_{t,2}|\mathbf{s}_t, a_{t,0}, a_{t,1})$

$$p(a_{t,2}|\mathbf{s}_t, a_{t,0}, a_{t,1})p(a_{t,1}|\mathbf{s}_t, a_{t,0})p(a_{t,0}|\mathbf{s}_t)$$

$$= p(a_{t,0}, a_{t,1}, a_{t,2}|\mathbf{s}_t)$$

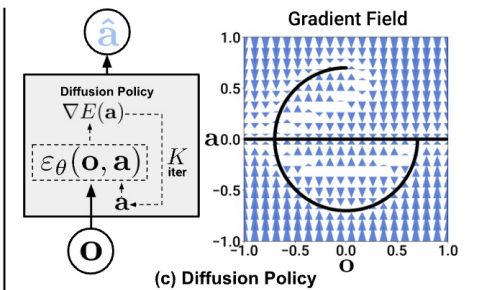
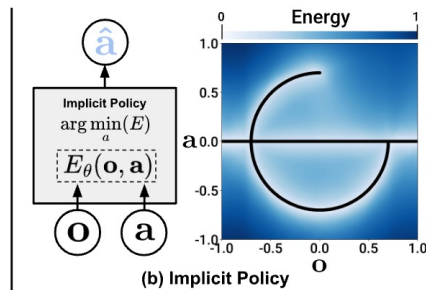
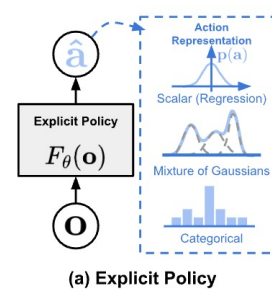
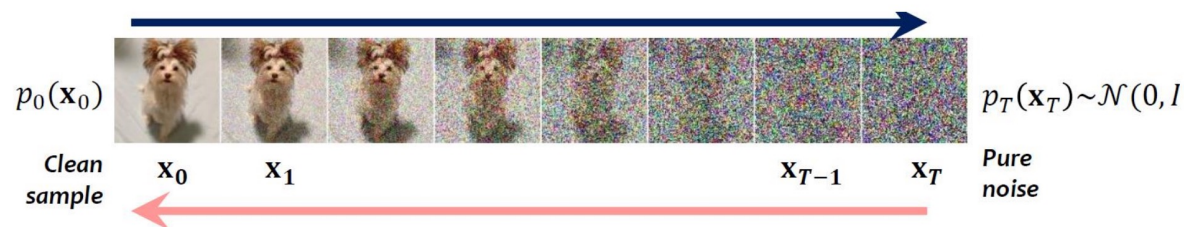
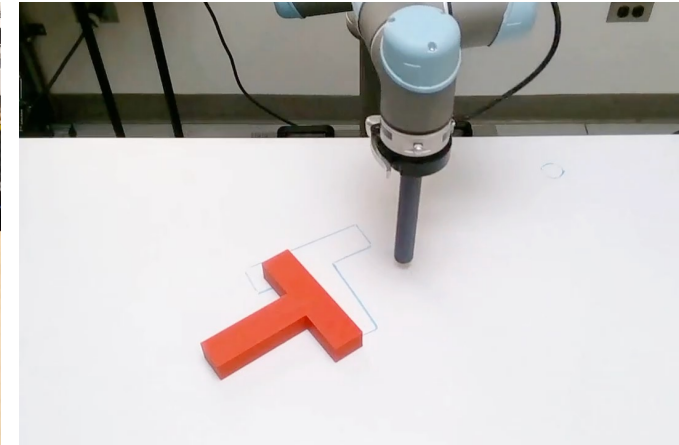
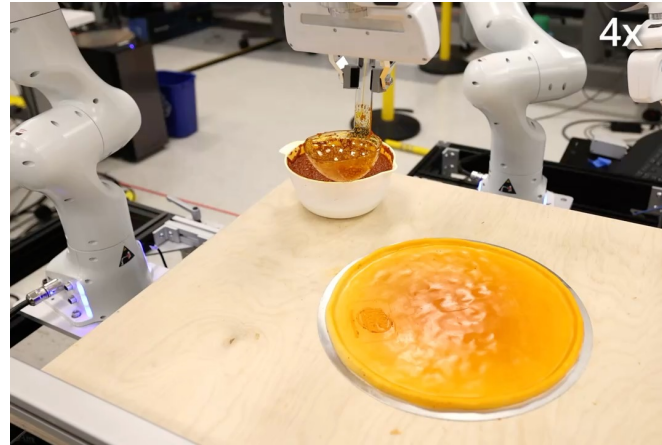
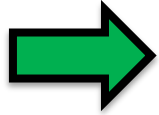
$$= p(\mathbf{a}_t|\mathbf{s}_t)$$

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



Why might we fail to fit the expert?

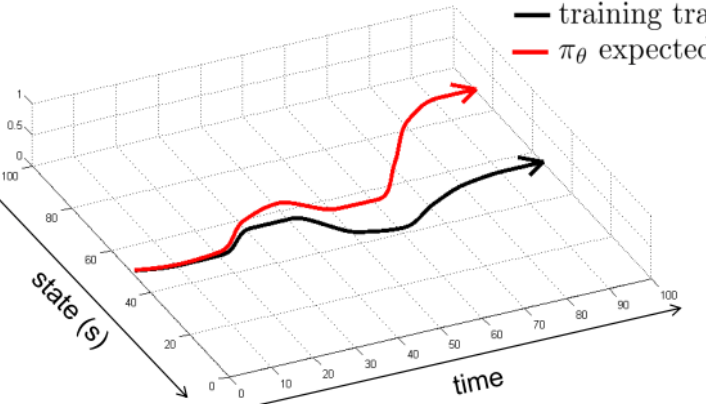
1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. ...



Let's try and understand where the problem lies?

Behavior cloning has challenges in both theory and practice

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



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: MDP formalism + why should we care?



Imitation learning: preliminaries and behavior cloning



Multimodality and Underfitting in Imitation

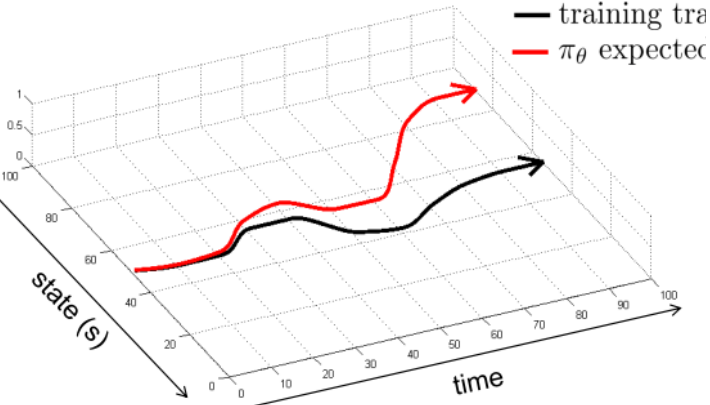


Compounding Error in Imitation

Let's try and understand where the problem lies?

Behavior cloning has challenges in both theory and practice

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



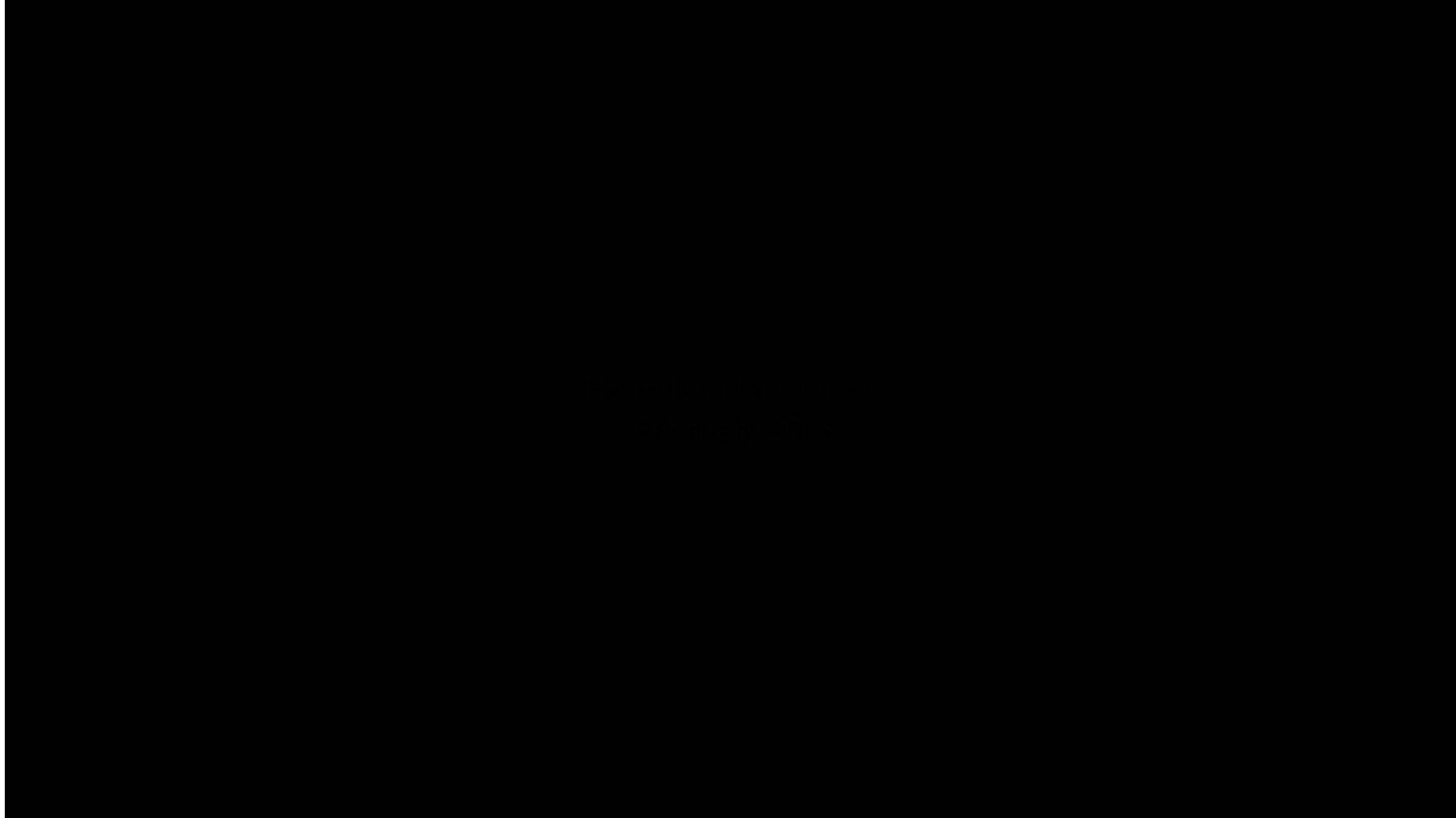
Underfitting

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

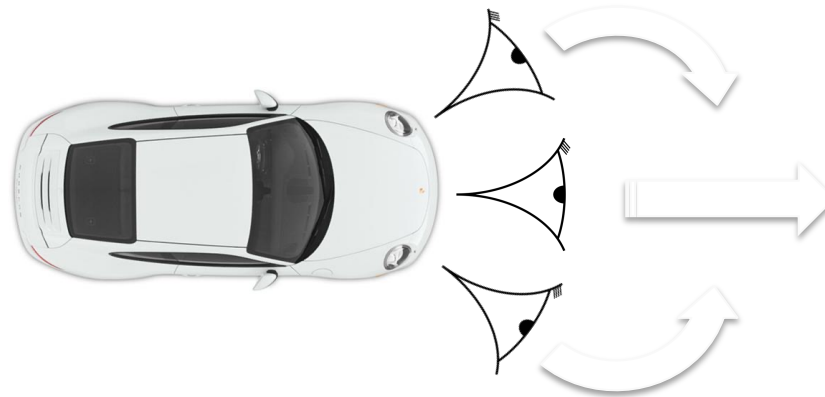
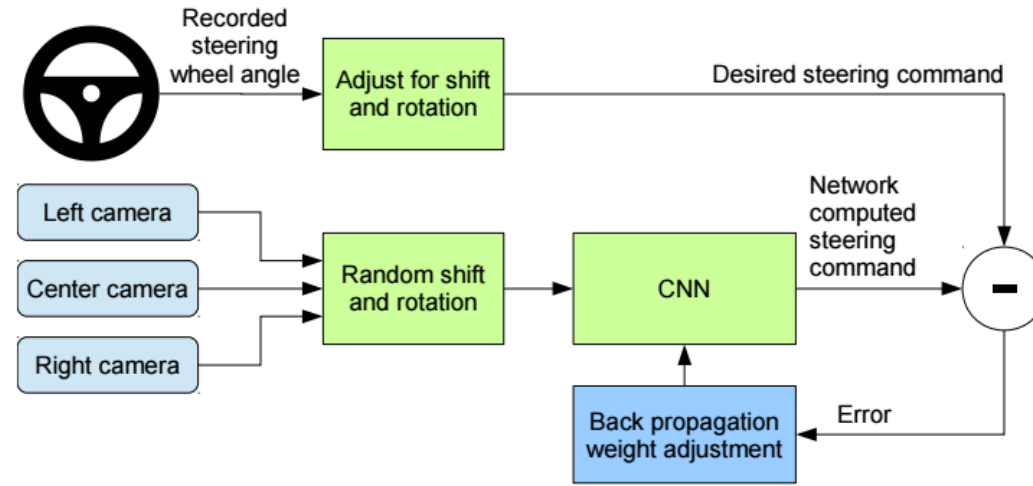
Compounding error

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

Can we avoid compounding error in special cases?

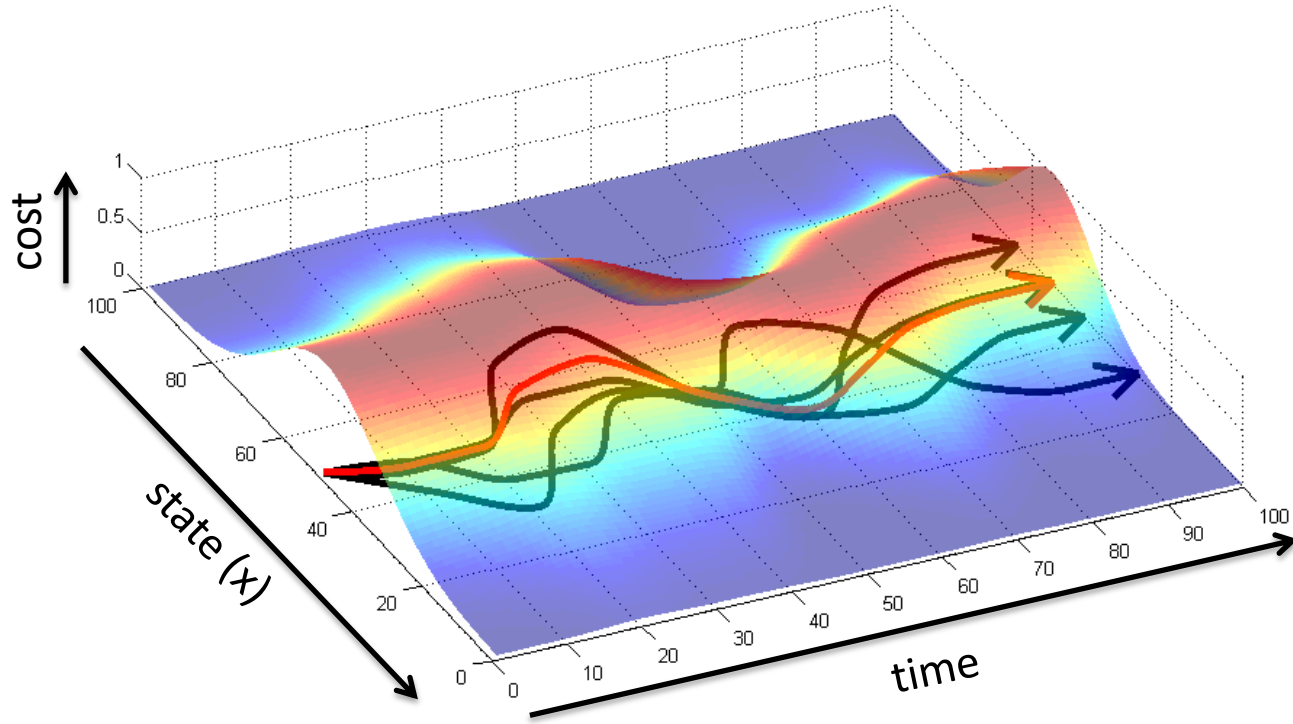


Why did that work?



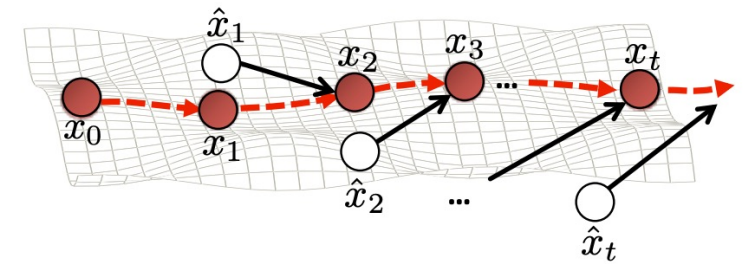
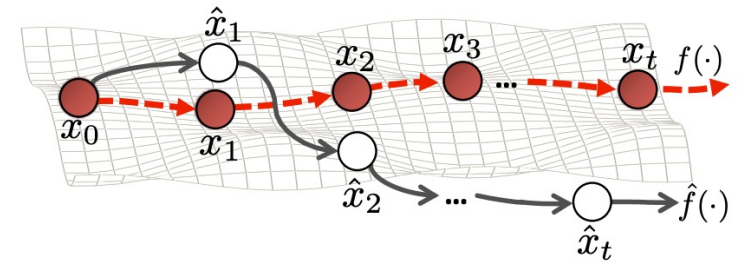
What is the general principle?

- training trajectory
- π_θ expected trajectory

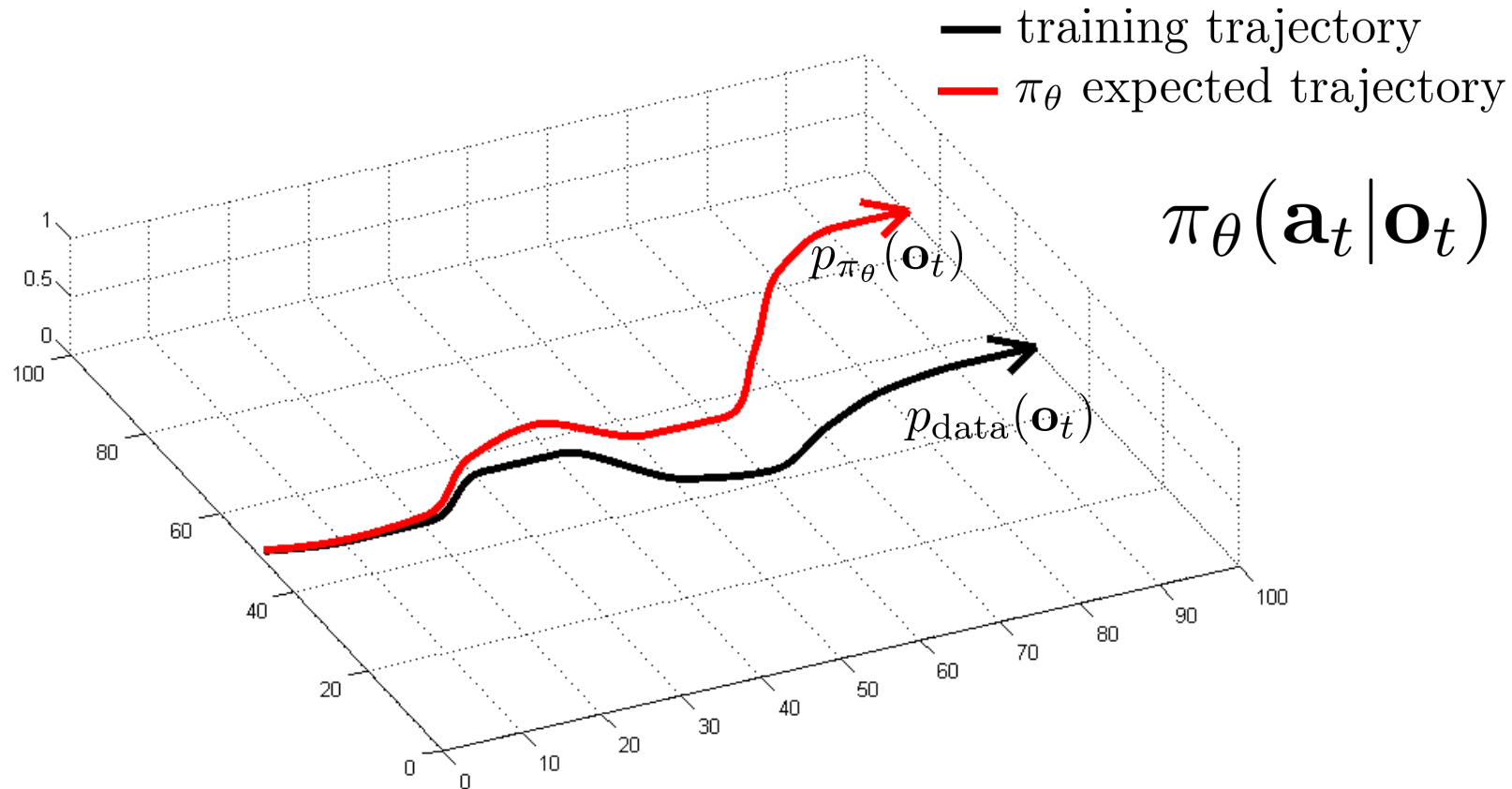


stability

Corrective labels that bring you back to the data



What might this mean mathematically?



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

Concrete Instantiation: 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: Dataset Aggregation

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 !

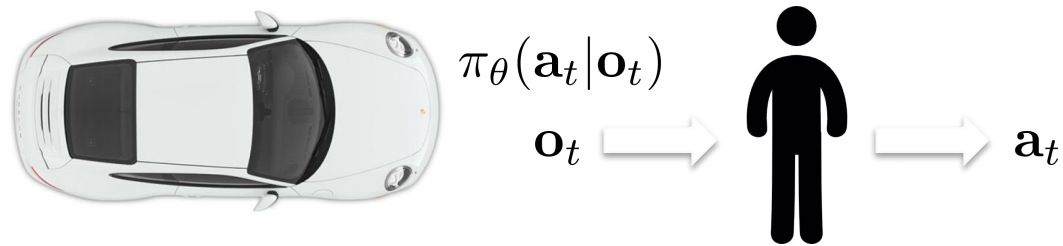
- 
1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \dots, \mathbf{o}_N, \mathbf{a}_N\}$
 2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \dots, \mathbf{o}_M\}$
 3. Ask human to label \mathcal{D}_π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

Dagger Example



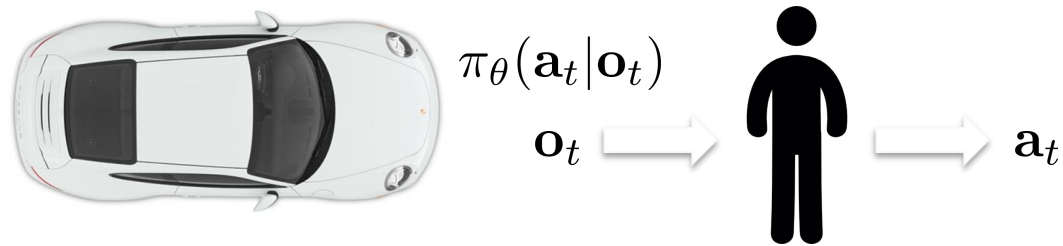
What's the problem?

1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \dots, \mathbf{o}_N, \mathbf{a}_N\}$
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}_π with actions \mathbf{a}_t
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$



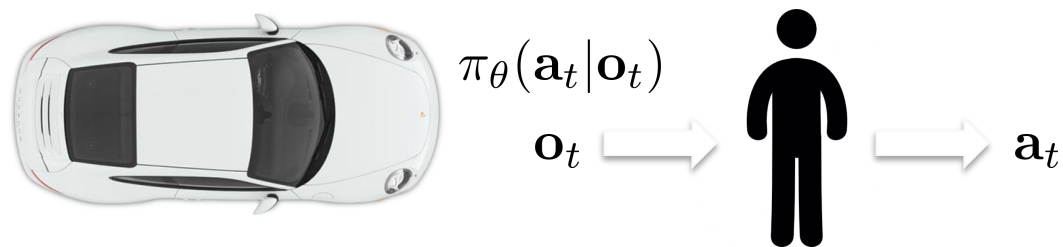
How might we fix this?

- "Generate" corrective labels automatically
1. train $\pi_{\theta}(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \dots, \mathbf{o}_N, \mathbf{a}_N\}$
 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}_{π} 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 $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \dots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \dots, \mathbf{o}_M\}$ ← Do at data collection time
3. Ask human to label \mathcal{D}_π with actions \mathbf{a}_t
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$



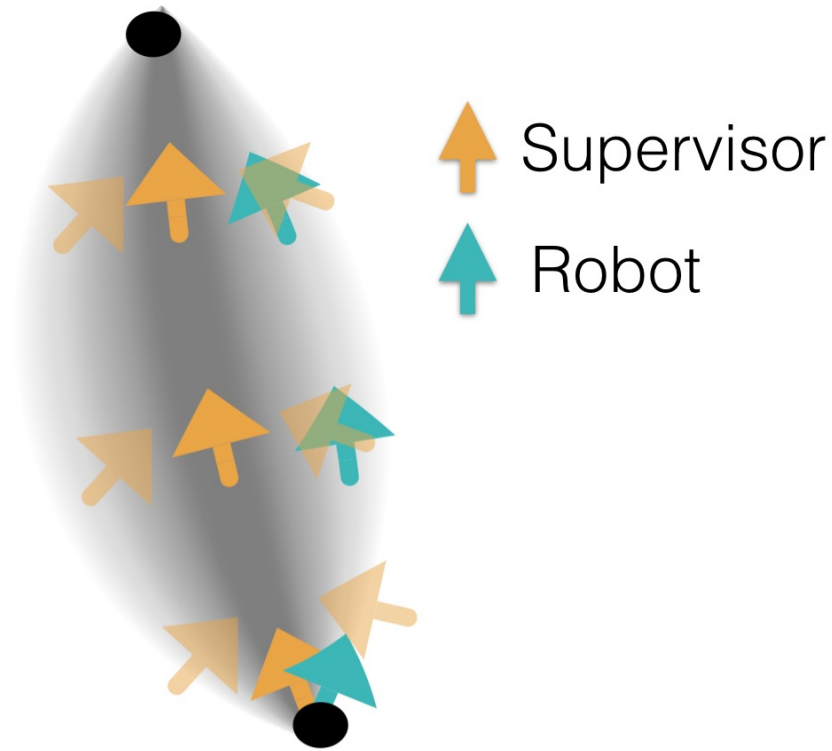
Noising the Data Collection Process

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

Under noise during data collection

Maximize likelihood

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

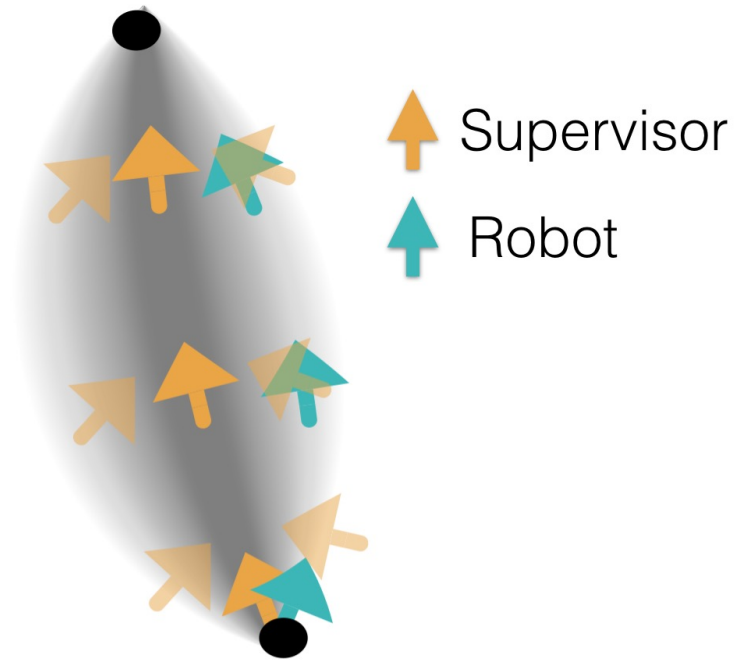


Noise Injection



Why might this not be enough?

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



Noise Injection

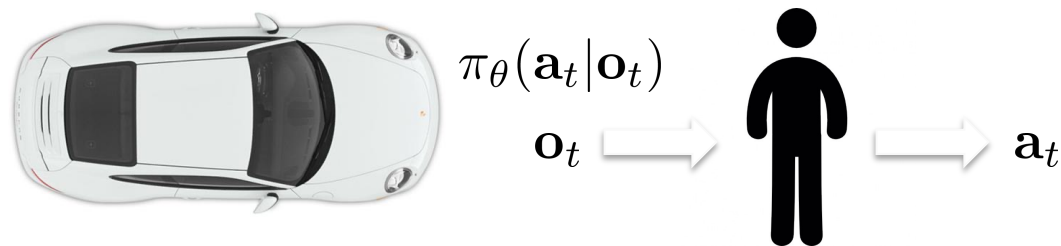


Assumes that the expert can actually perform behaviors under noise
→ Not always possible!

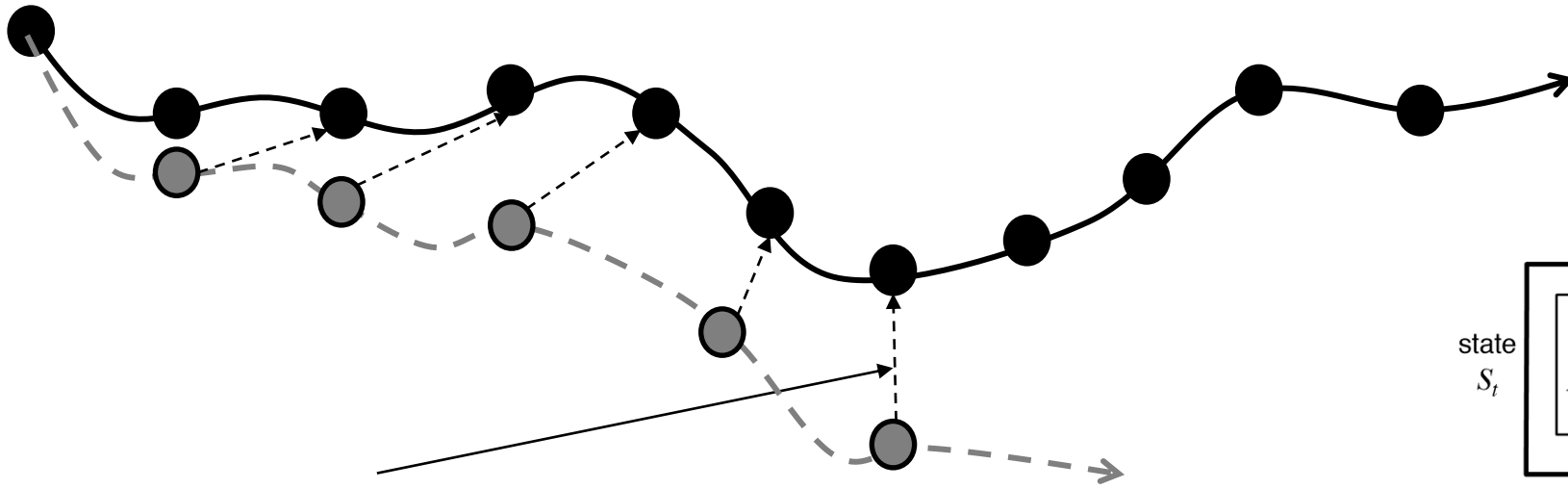
How might we fix this?

"Generate"
corrective labels
automatically

1. train $\pi_{\theta}(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \dots, \mathbf{o}_N, \mathbf{a}_N\}$
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}_{π} with actions \mathbf{a}_t
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\pi}$

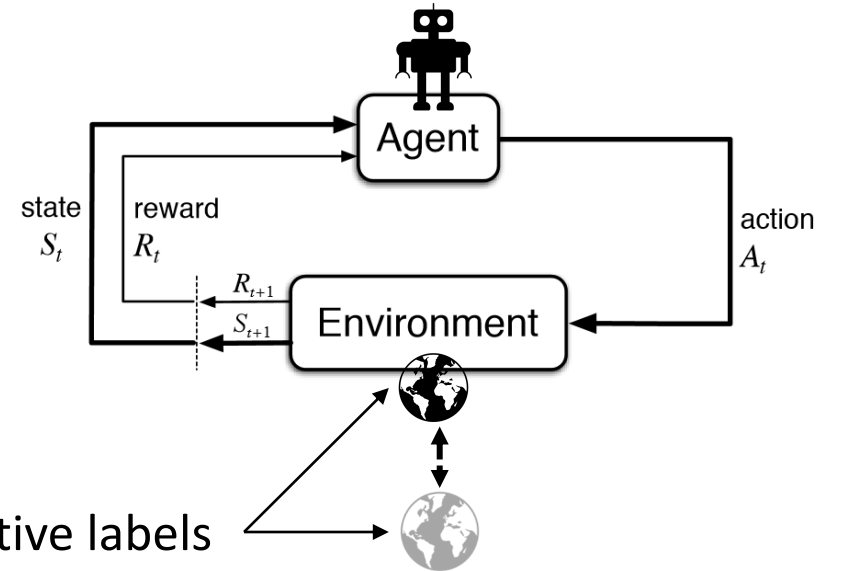


How can we find corrective labels?



How might we obtain these corrections?

(Learned) Dynamics model can help find corrective labels



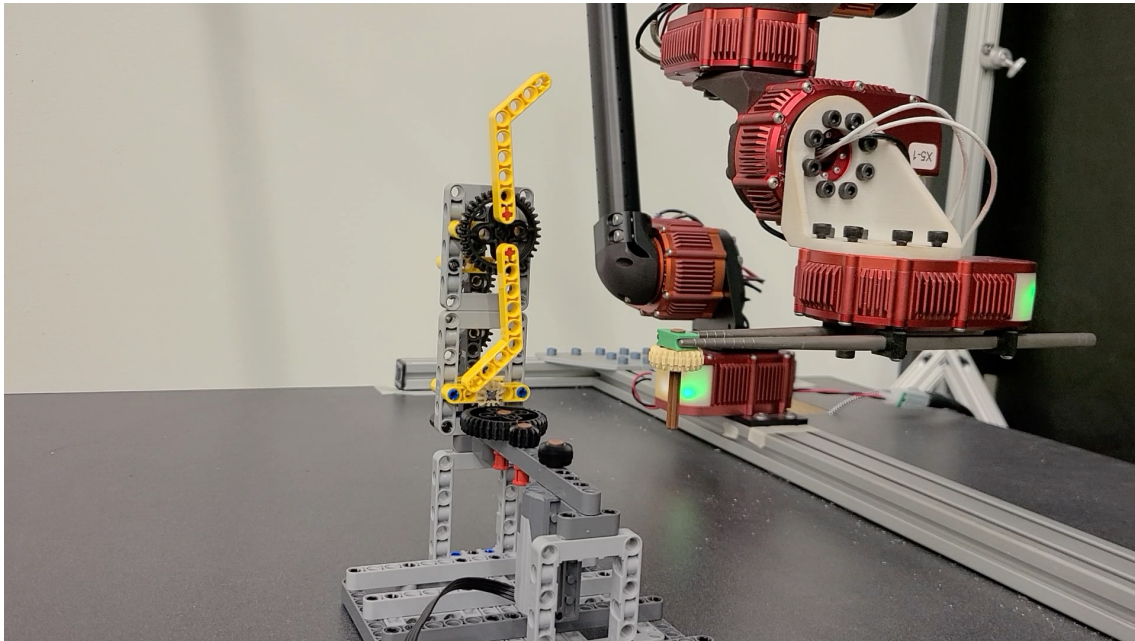
Augment D with states (s_t), actions (a_t) that lead back to optimal states under dynamics

$$\|s_{t+1}^* - f(s_t, a_t)\| \leq \epsilon \quad s_{t+1} = f(s_t, a_t)$$

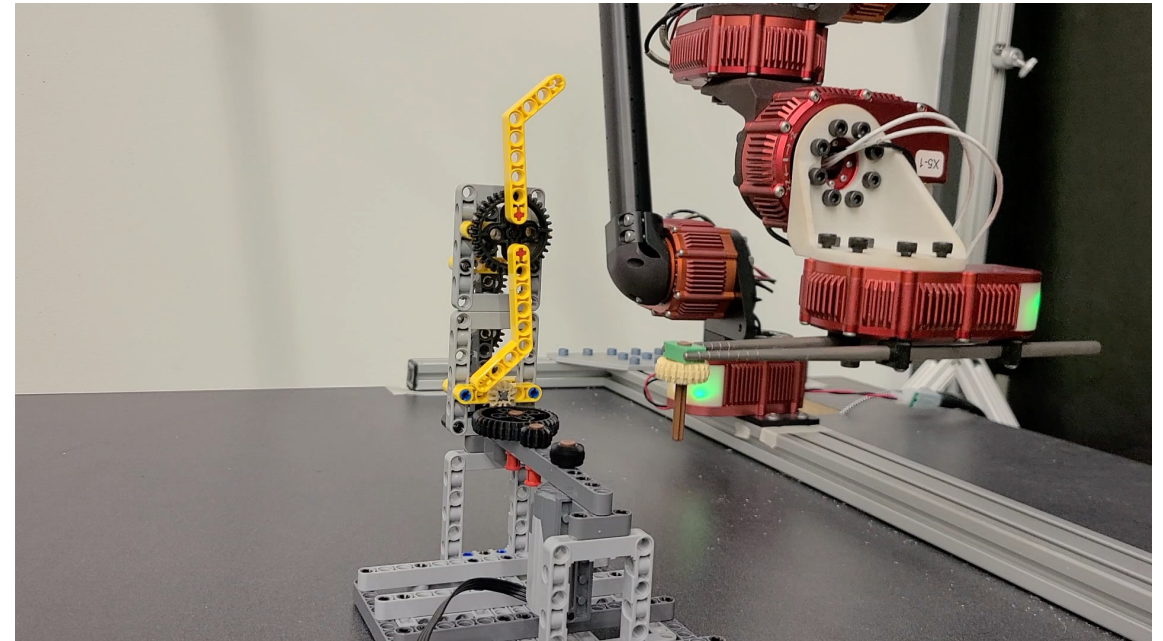
*with caveats

Overall Learning Pipeline with Corrective Labels

Standard behavior cloning



Corrective labels



Lecture outline

Recap: MDP formalism + why should we care?



Imitation learning: preliminaries and behavior cloning



Multimodality and Underfitting in Imitation

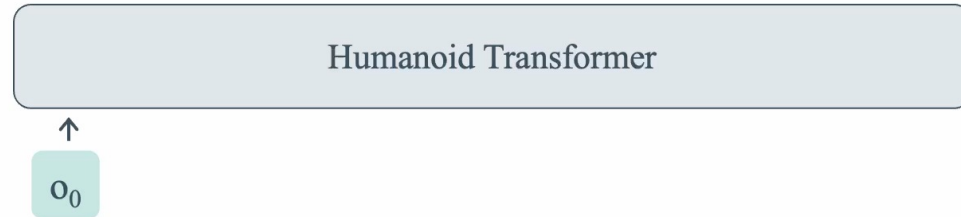


Compounding Error in Imitation

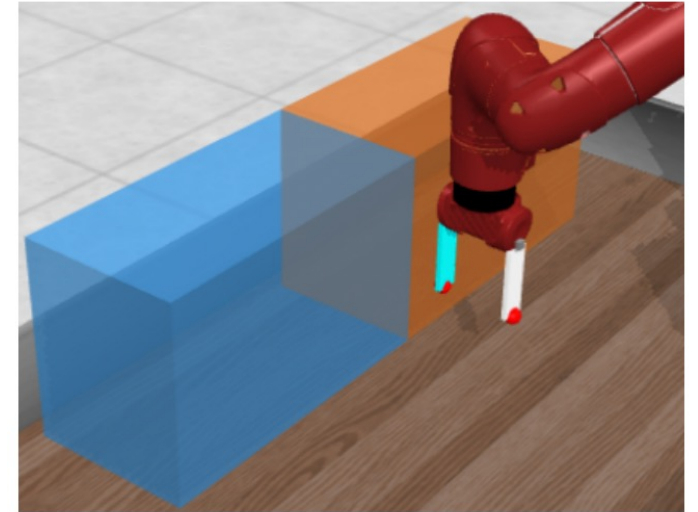
So does this solve all the issues in imitation?

Frontiers in Imitation Learning

Non-Markovian Demonstrators



Characterizing generalization

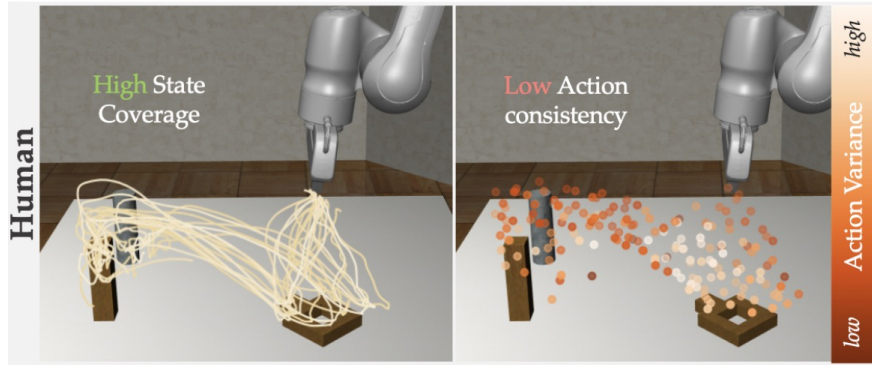


Action-Free Data

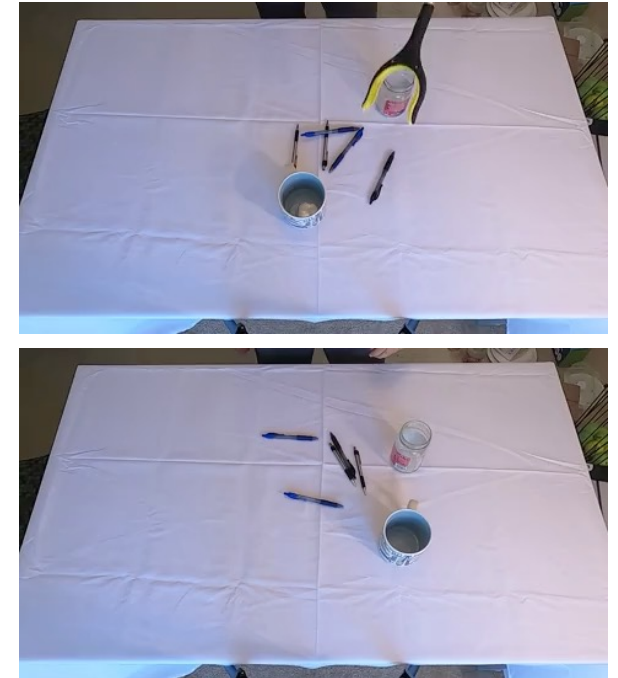


Frontiers in Imitation Learning

Data Curation and Quality



Embodiment Shift



Teleoperation Interfaces



Some cool imitation videos

1x and tesla humanoid robots



- 1X END-TO-END AUTONOMY
UPDATE, JAN 2024

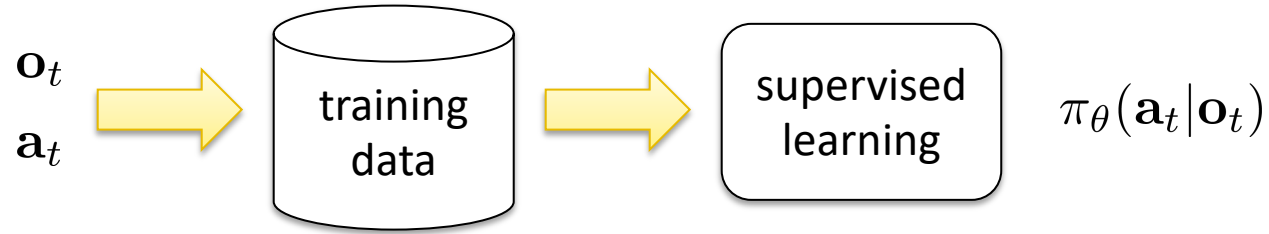
ALOHA and CherryBot Fine Manipulation



TRI Diffusion Policies

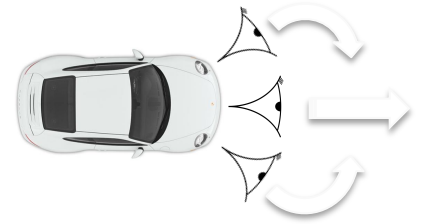


Perspectives on Imitation – don't believe everything you see online



■ 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!

