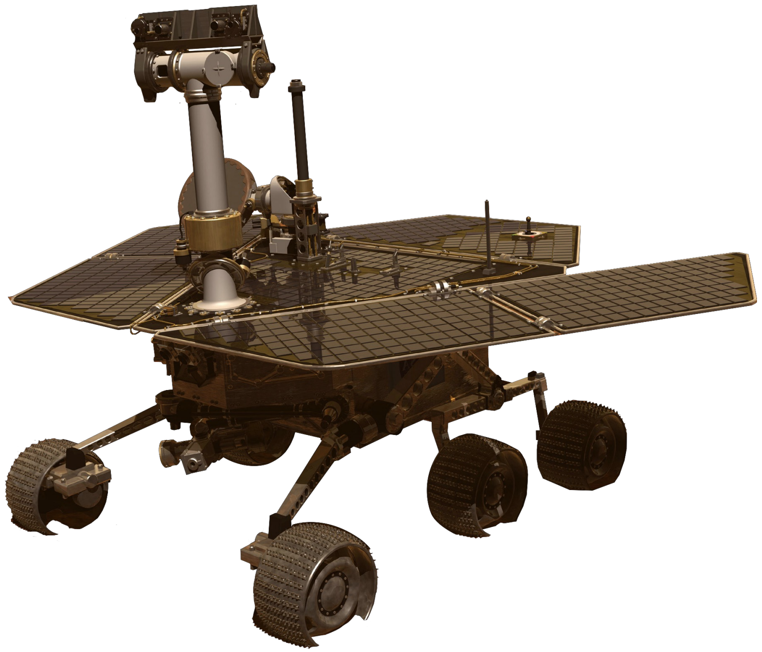
Autonomous Robotics

Winter 2025

Abhishek Gupta

TAs: Carolina Higuera, Entong Su, Bernie Zhu



Slides borrowed from many sources – Sidd Srinivasa, Sanjiban Choudhury, M Likhachev

Class Outline

State Estimation

- Robotic System Design
- Filtering
- Localization
- SLAM

Control

- Feedback Control
- PID Control
- MPC
- LQR

Planning

- Search
- Heuristic Search
- Motion Planning
- Lazy Search

Learning

- Imitation Learning
- Policy Gradient
- Actor-Critic
- Model-Based RL

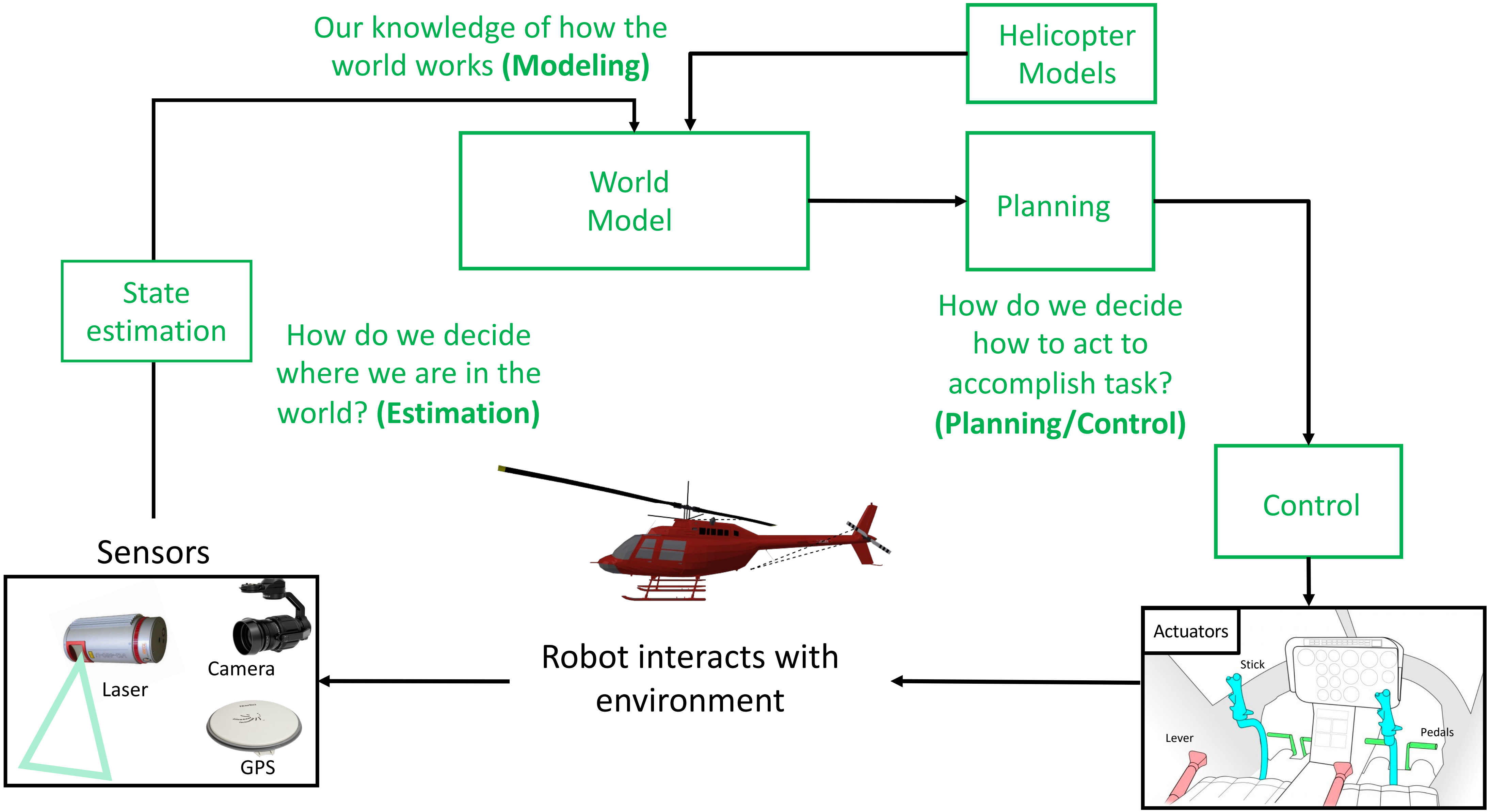
Why is RRT* preferable to RRT?

0

Provable guarantees	0%
Speed	0%
Path optimality	0%
None of the above	0%

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

We have built a model-based control system!

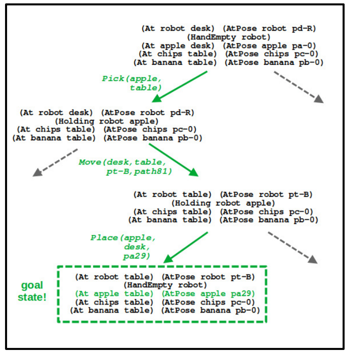


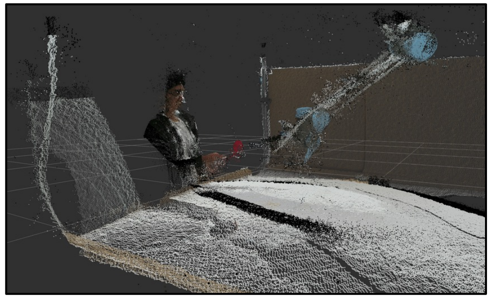
Are we done?

Model-based Control for Robotics

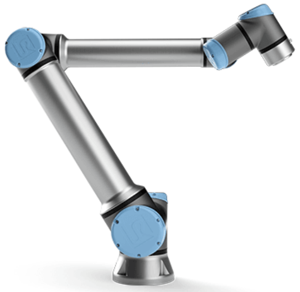
Focus on addressing all problems at once

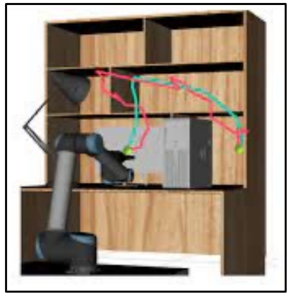
$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \tau_g(\mathbf{q}) + \mathbf{B}\mathbf{u},$$

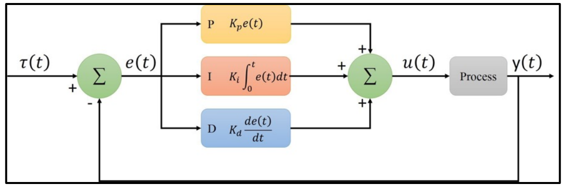




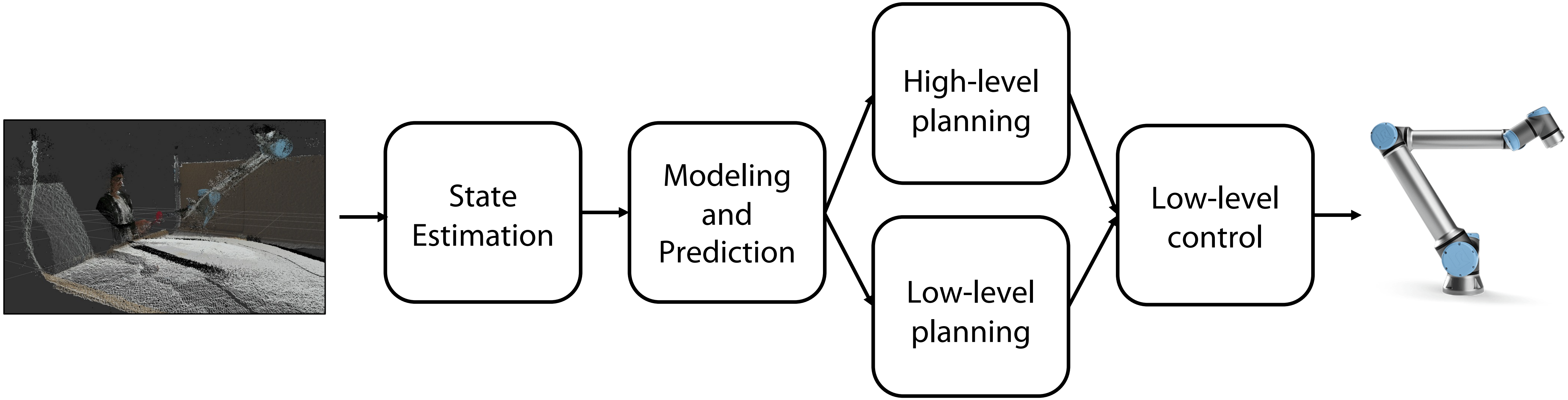
State Estimation → Modeling and Prediction → Planning → Low-level control →



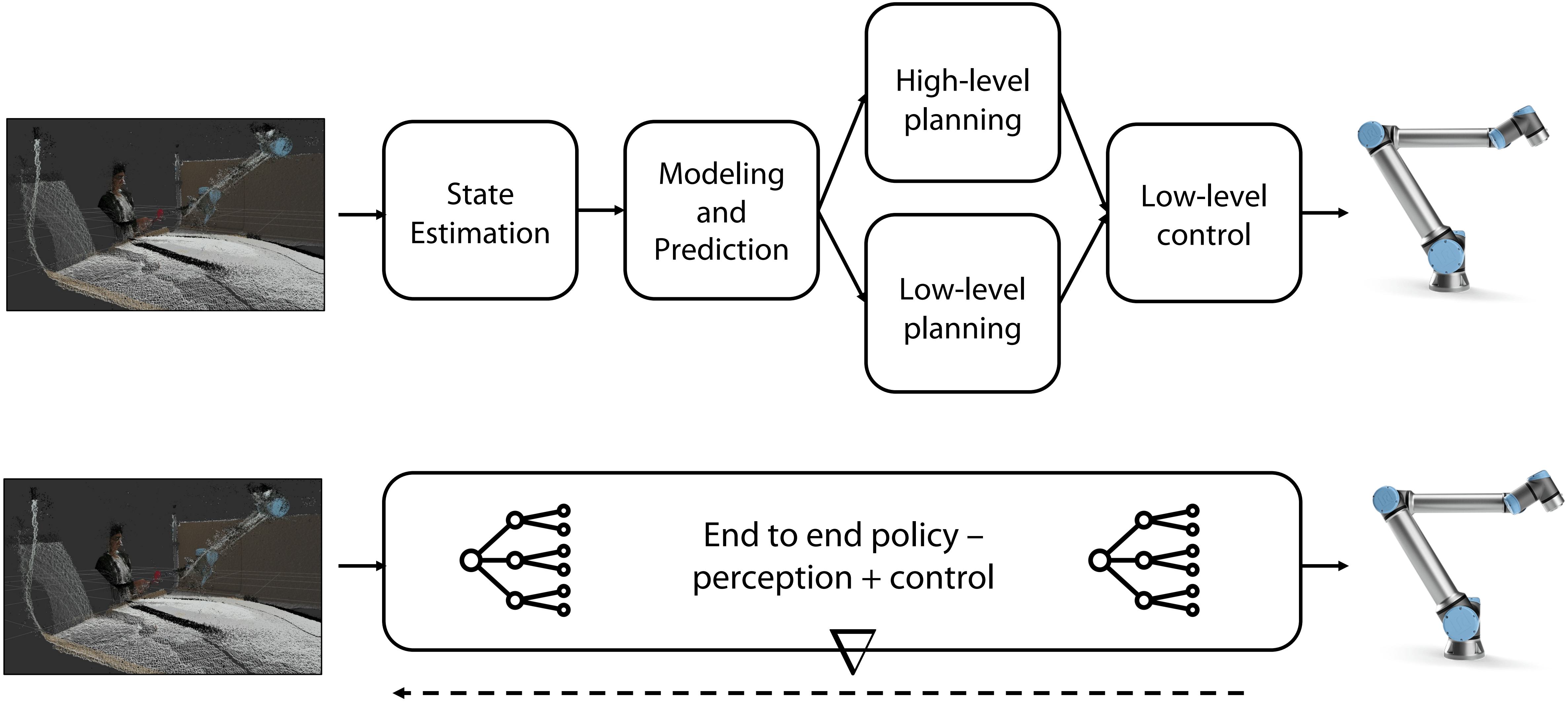




What does a typical model based look like?



End-to-End Learning Based Control for Robots



Why might we want/not want to do this?

Modules compensate for each other

Avoids hand-designing and supervising interfaces

Often more performant/less biased

Lack of Interpretability

Lack of Reusability

Often data inefficient

Lecture Outline

A Formalism for Sequential Decision Making

↓

Imitation Learning: Behavior Cloning

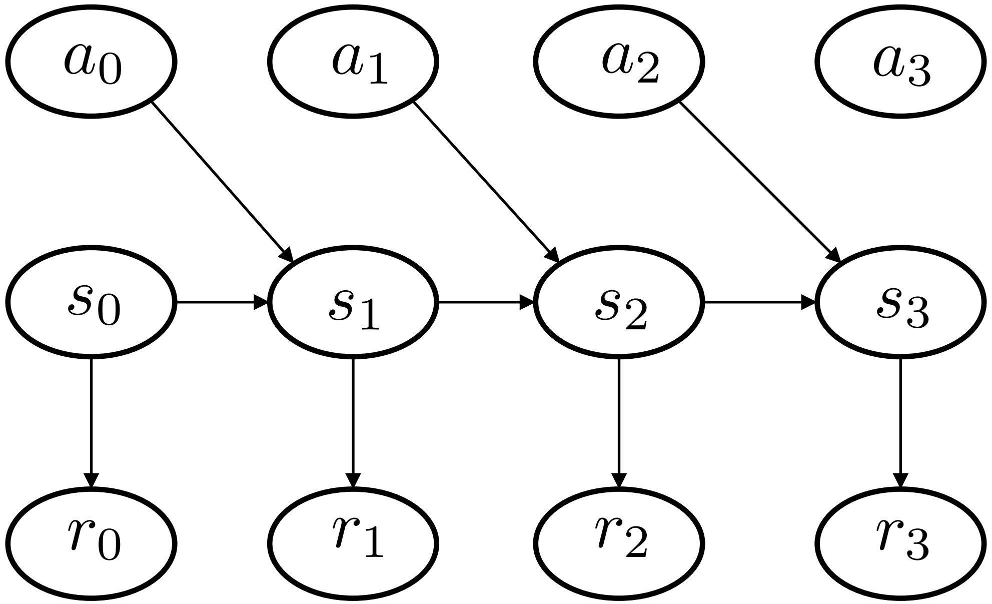
↓

Imitation Learning: Improvements – Compounding Error

↓

Imitation Learning: Improvements – Multimodality

Framework for Sequential Decision Making - Markov Decision Process



States: $\mathcal{S}$

Actions: $\mathcal{A}$

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

Key: MDPs obey the Markov property
Past is independent of the future conditioned on the present

Mapping MDPs to the Real World

Task: Place kettle in sink

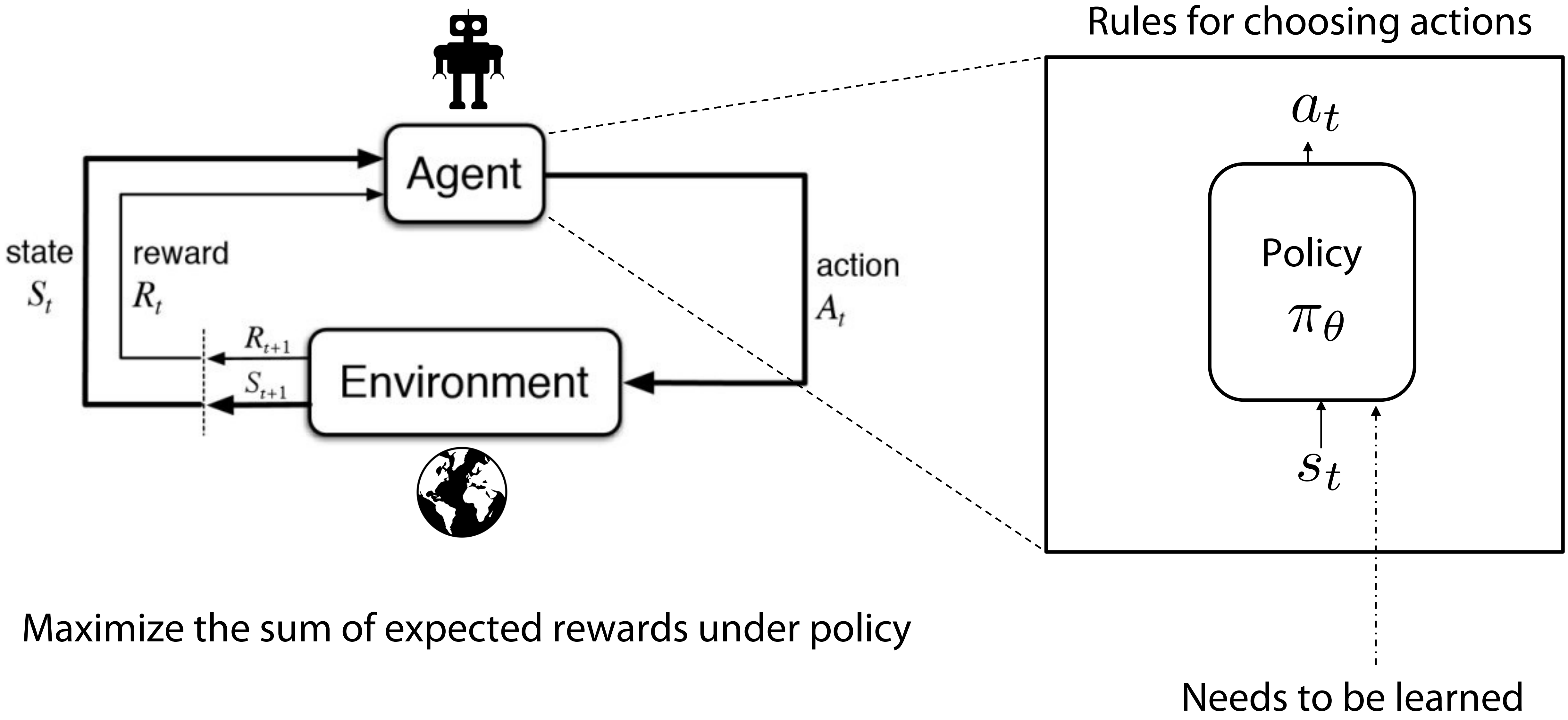
State: Camera Images / Joint Encoders

Action: Joint torques/velocities

Reward: Distance from kettle to sink

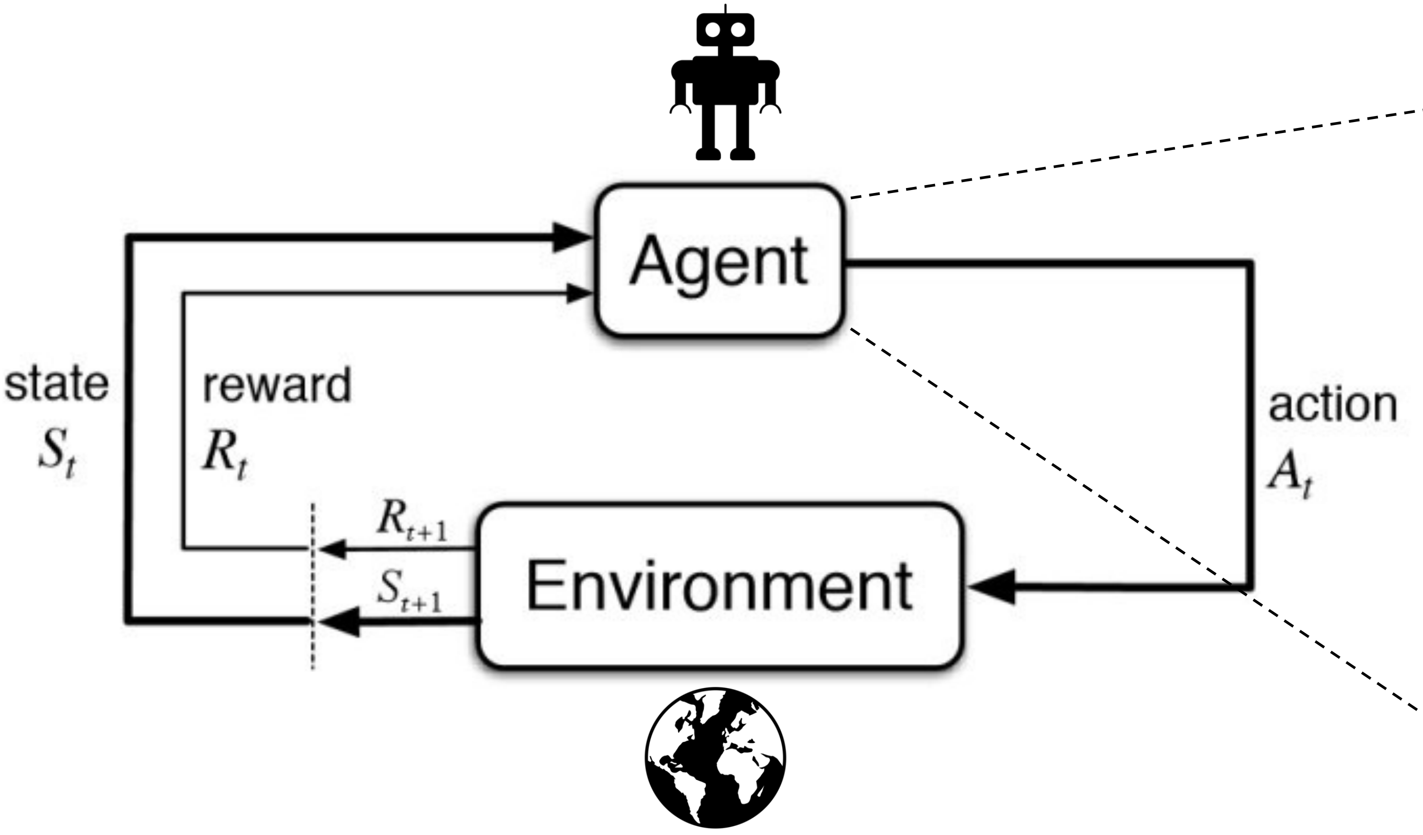
Transition: World physics

Reinforcement Learning Formalism



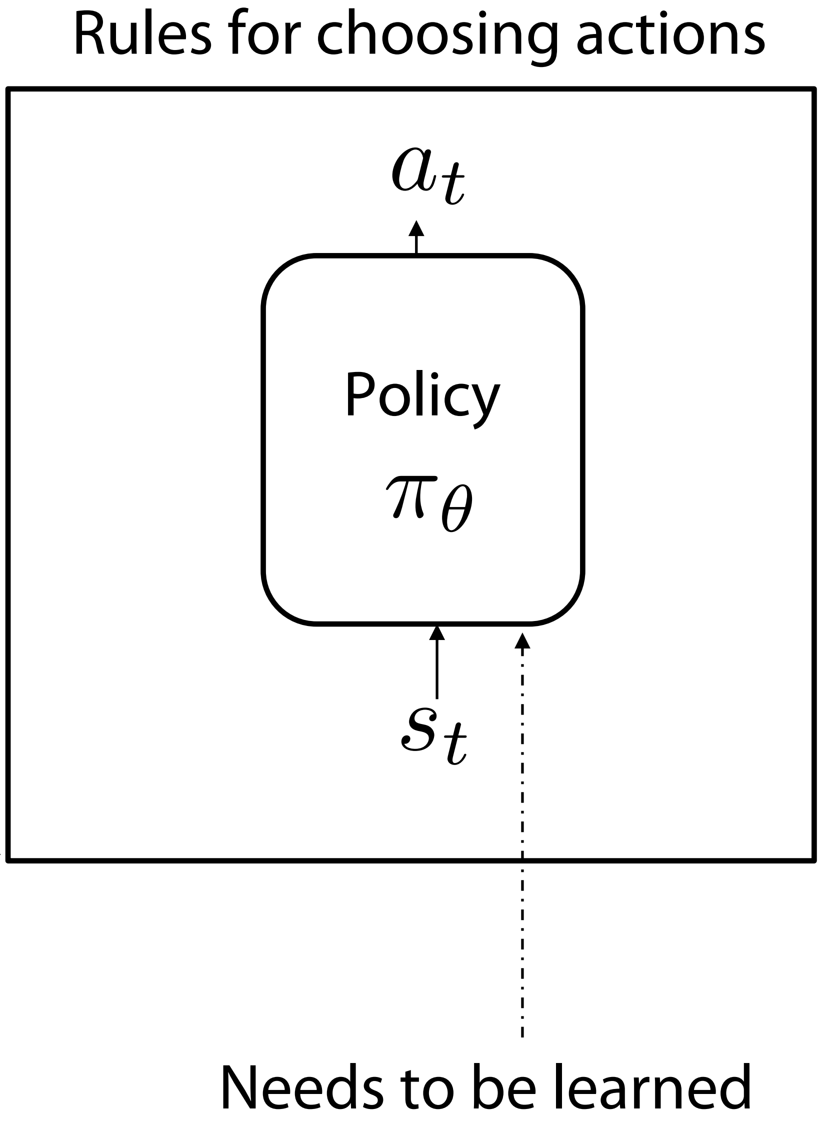
Maximize the sum of expected rewards under policy

Reinforcement Learning Formalism



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

Trajectory sampled using policy



Why isn't this just optimal control?

<u>Optimal control</u>

$$\min_{u_{1:T}} \sum_{t=1}^{T} c(x_t, u_t)$$

$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t)$$

<u>Reinforcement Learning</u>

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

Cosmetic differences:

- Costs vs rewards
- Often discrete vs continuous time

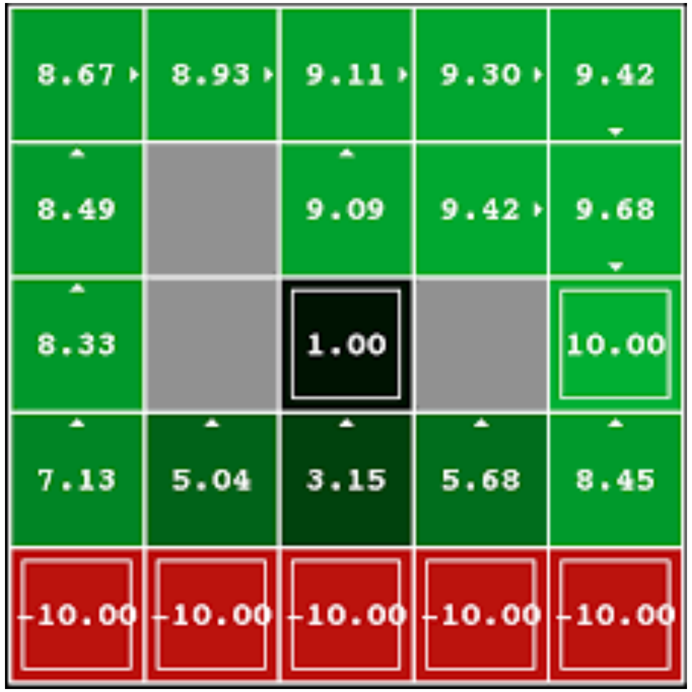
Real differences:

- Known model vs sample-able model

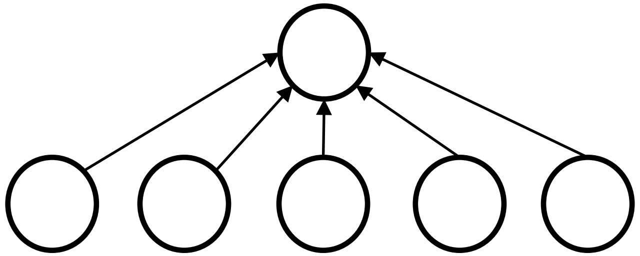
Main thing to learn - Policies

Policies are mappings from states to optimal actions

Tabular

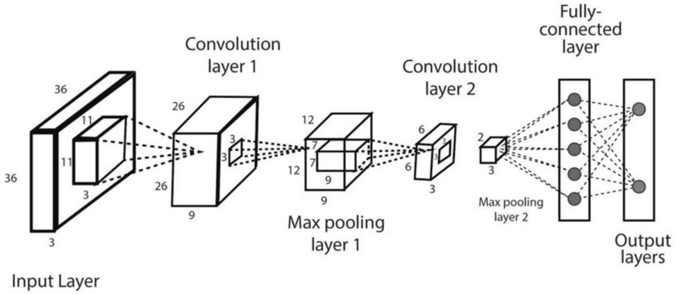


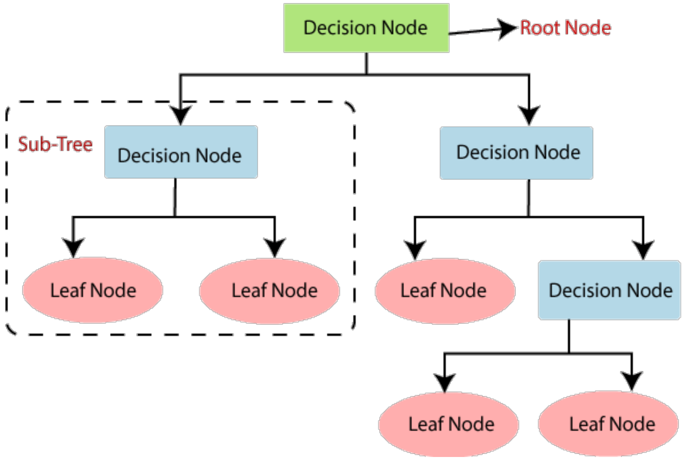
Linear



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

Arbitrary function approx





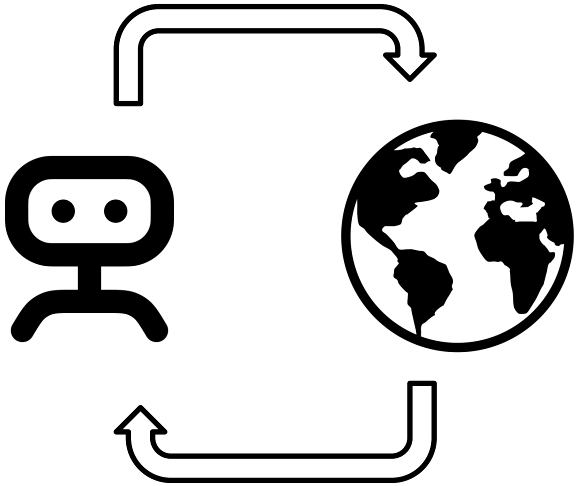
Ok so how can we learn policies?

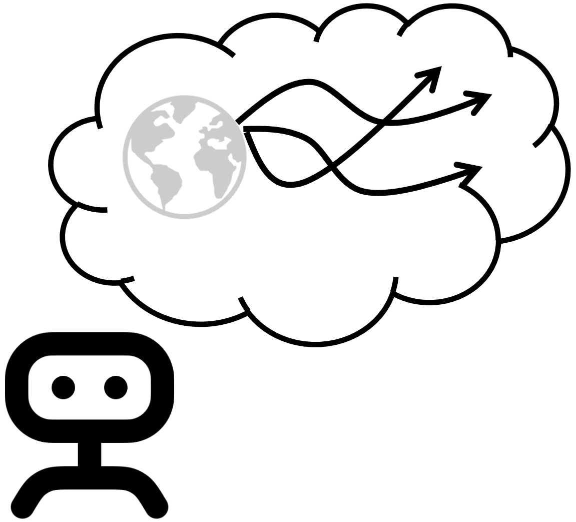
$$\max_{\theta} \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} r(s_t, a_t) \right]$$

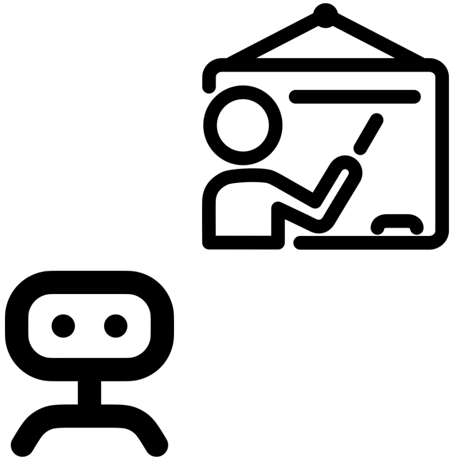
<u>Model-free RL</u>

<u>Model-based RL</u>

<u>Imitation Learning</u>



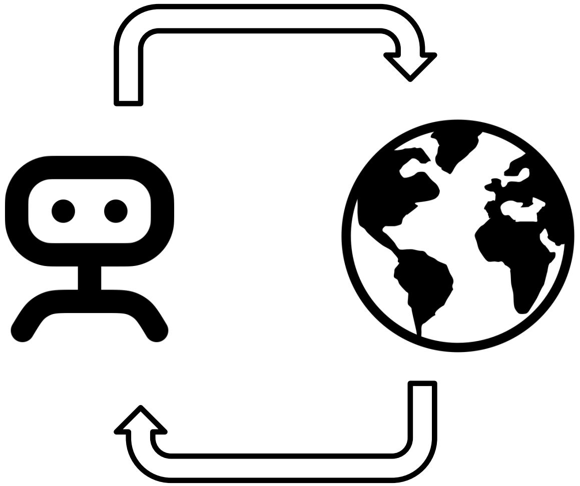




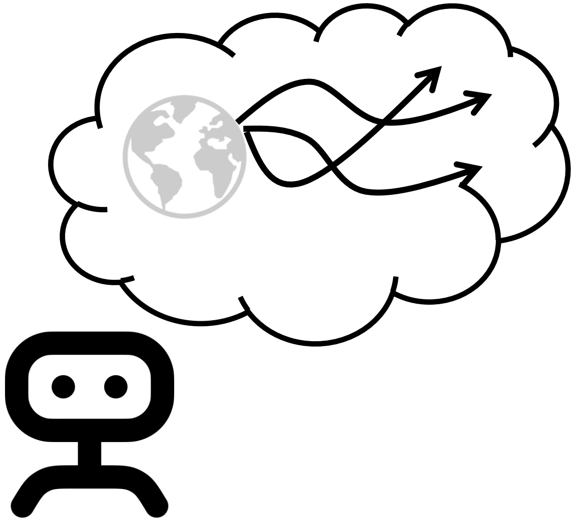
Ok so how can we learn policies?

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

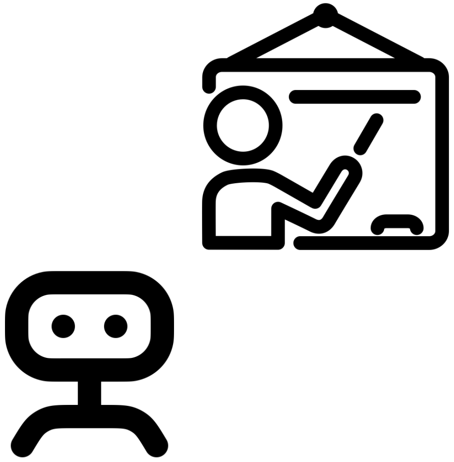
Model-free RL



Model-based RL



Imitation Learning



Lecture Outline

A Formalism for Sequential Decision Making

↓

Imitation Learning: Behavior Cloning

↓

Imitation Learning: Improvements – Compounding Error

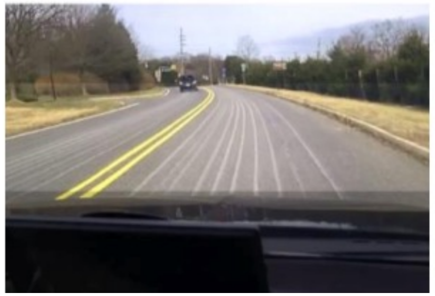
↓

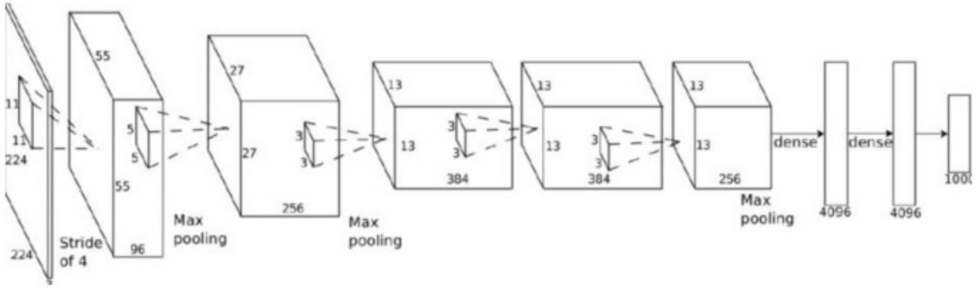
Imitation Learning: Improvements – Multimodality

Imitation Learning: Intuition

Given: Demonstrations of optimal behavior

Goal: Train a policy to mimic the demonstrator





Pros: No rewards, online experience needed (?)

Idea 1: Imitation Learning via Behavior Cloning

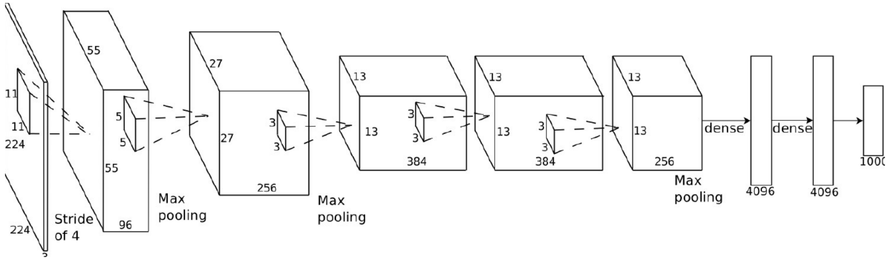
Given: Demonstrations of optimal behavior

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

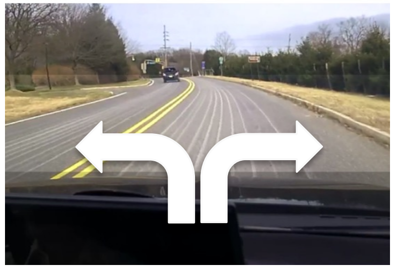
Goal: Train a policy to mimic the demonstrator

Idea: Treat imitation learning as a supervised learning problem!

$\mathbf{o}_t$



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



$\mathbf{a}_t$

$\mathbf{o}_t$

$\mathbf{a}_t$

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

Idea 1: Imitation Learning via Behavior Cloning

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

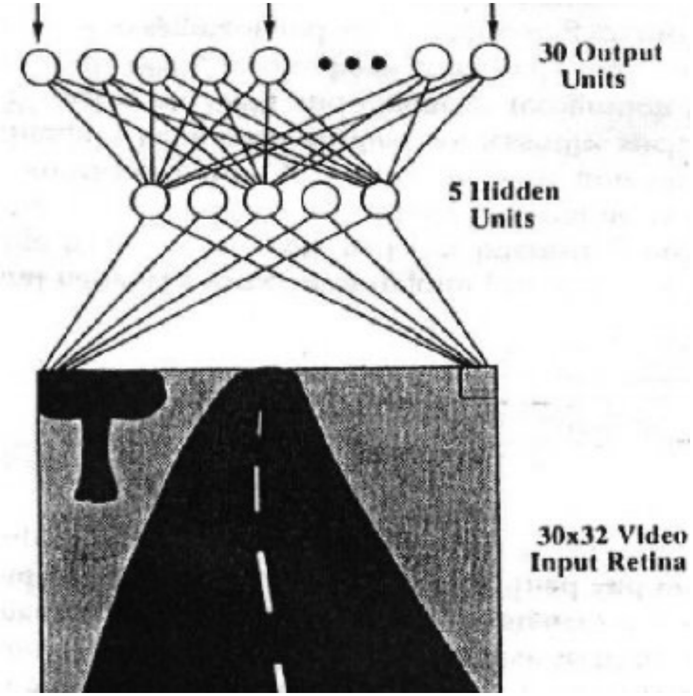
Discrete vs continuous

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()
```

The original deep imitation learning system

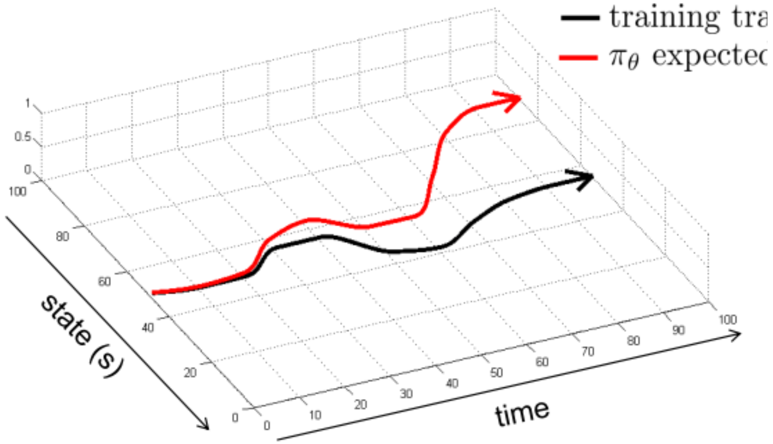
ALVINN: **A**utonomous **L**and **V**ehicle **I**n a **N**eural **N**etwork
1989



Where we are in 2025?

So does behavior cloning really work?

- Imitation Learning $\neq$ Supervised Learning



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

Not the same!

So does behavior cloning really work?

- Fails in practice as well!

So is all hope lost?

Lecture Outline

A Formalism for Sequential Decision Making

↓

Imitation Learning: Behavior Cloning

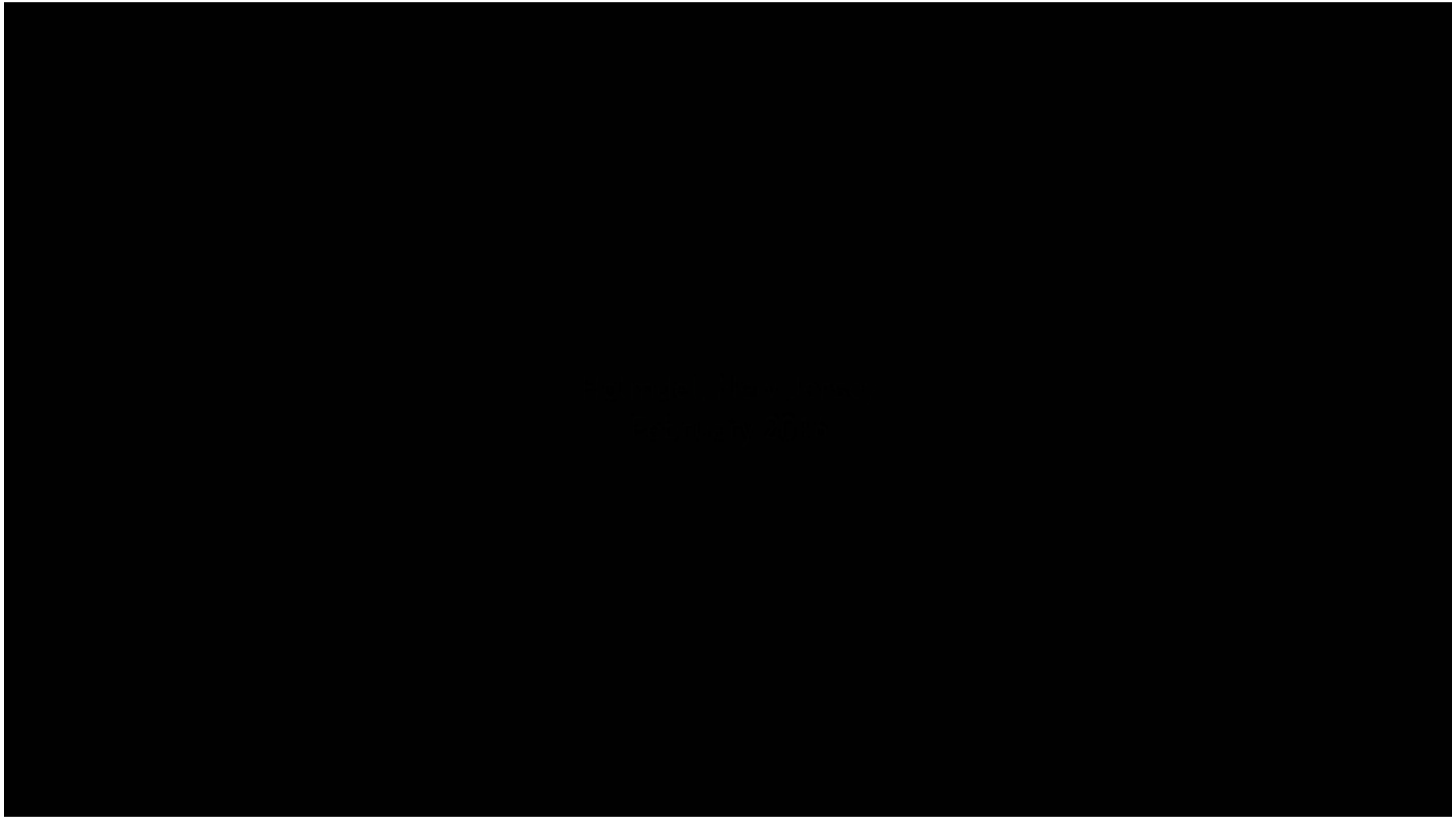
↓

Imitation Learning: Improvements – Compounding Error

↓

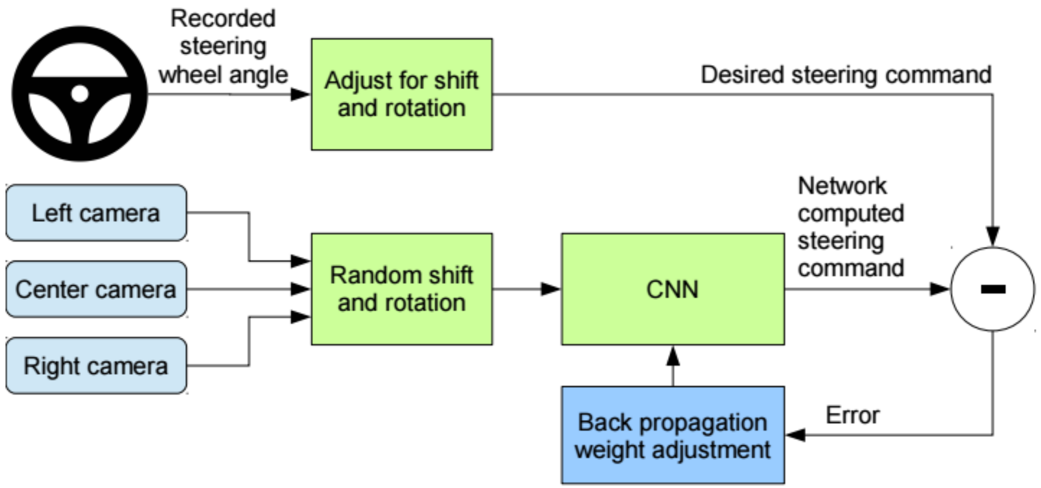
Imitation Learning: Improvements – Multimodality

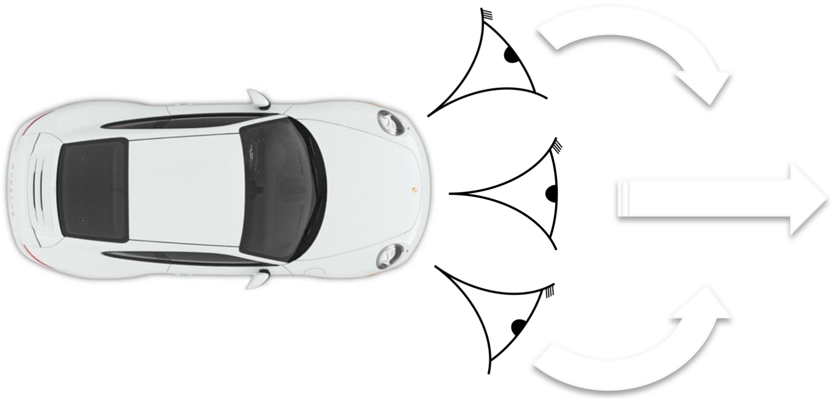
Can it work in special cases?



Video: Bojarski et al. ‘16, NVIDIA

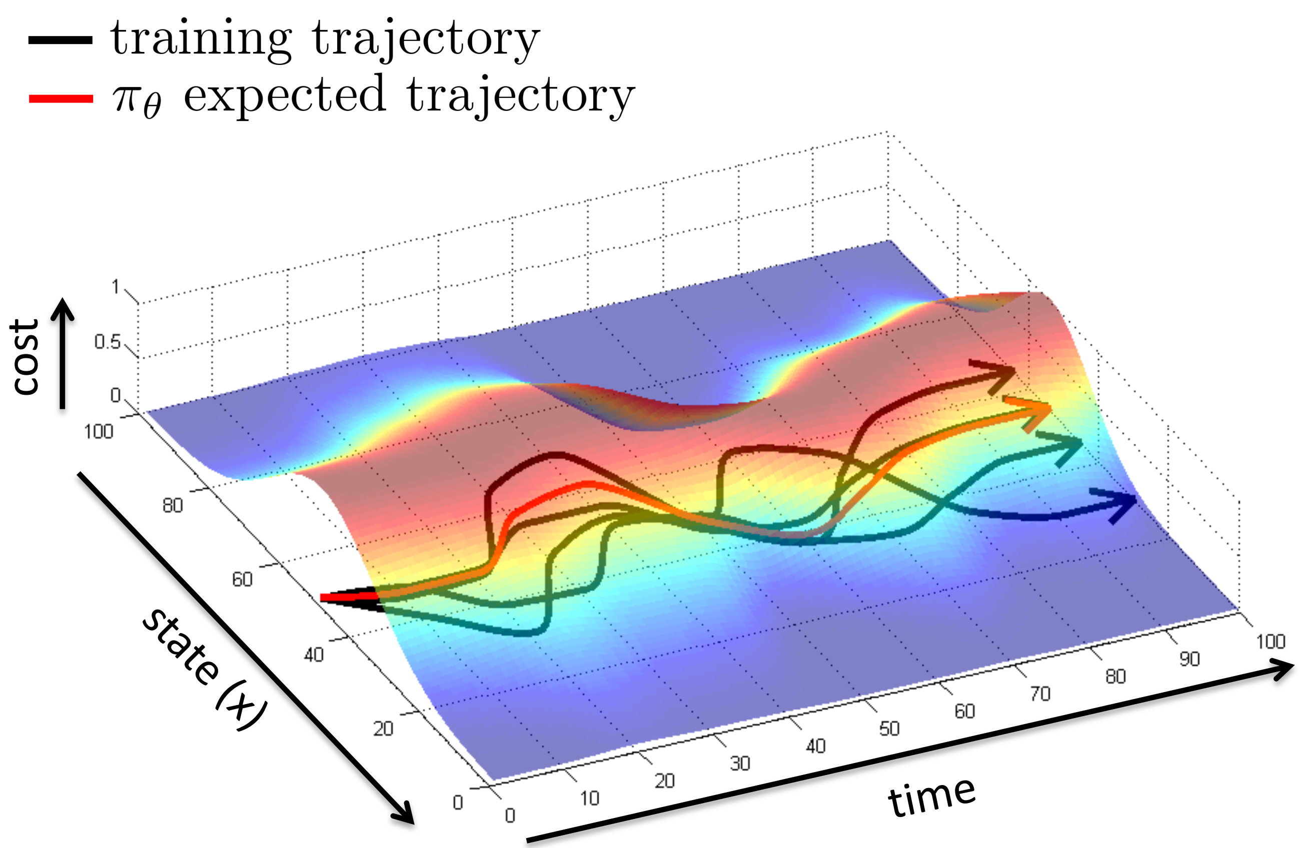
Why did that work?





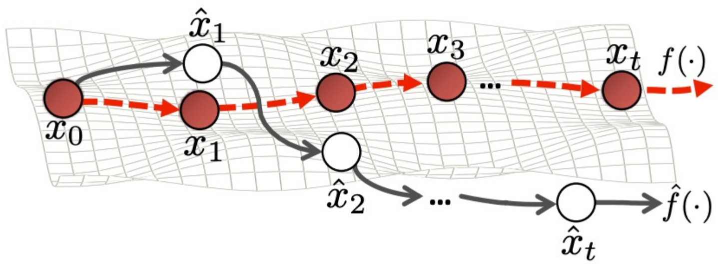
Bojarski et al. '16, NVIDIA

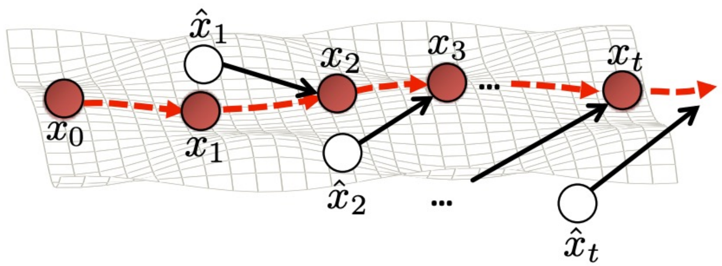
What is the general principle?



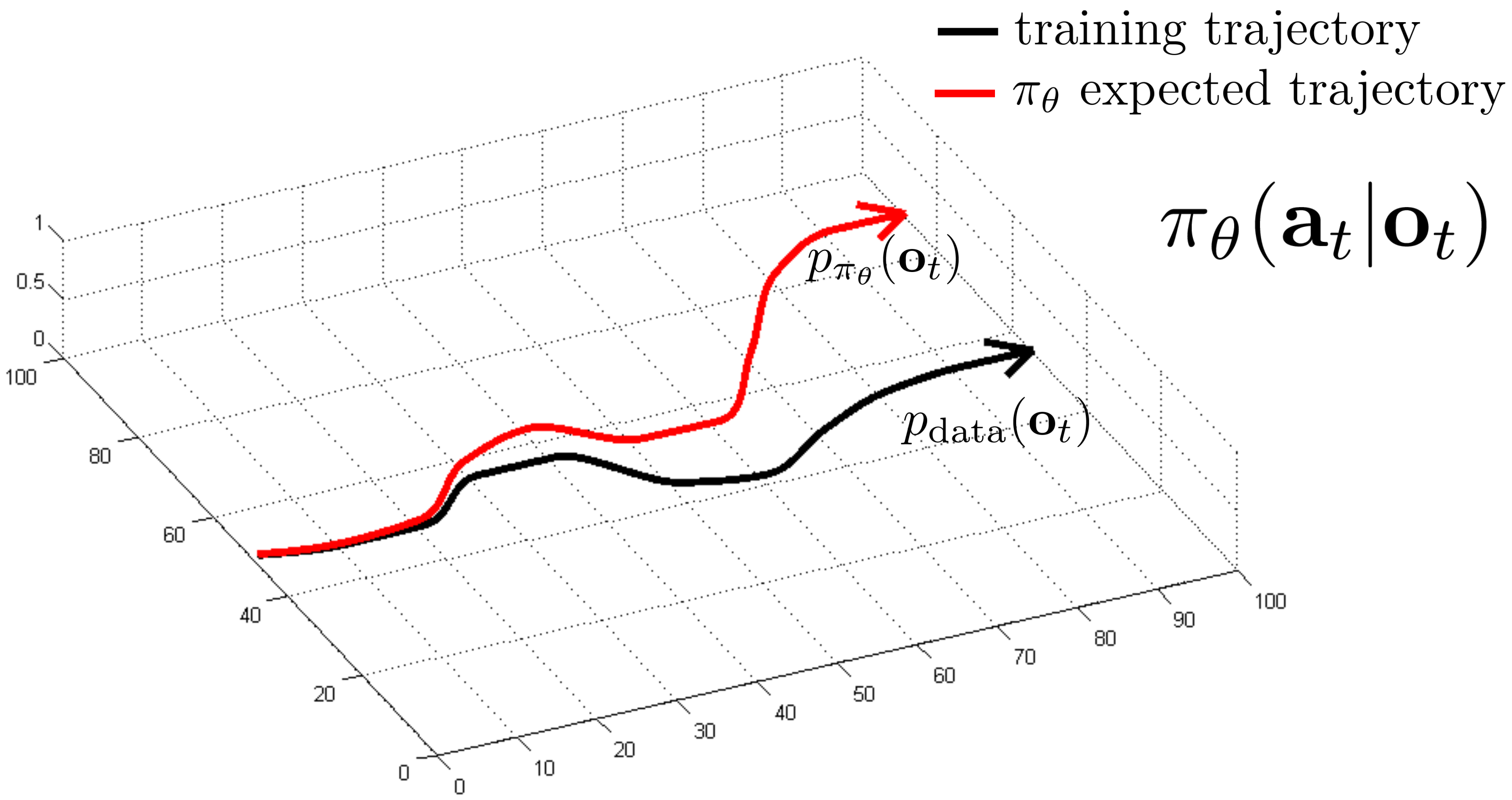
stability

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

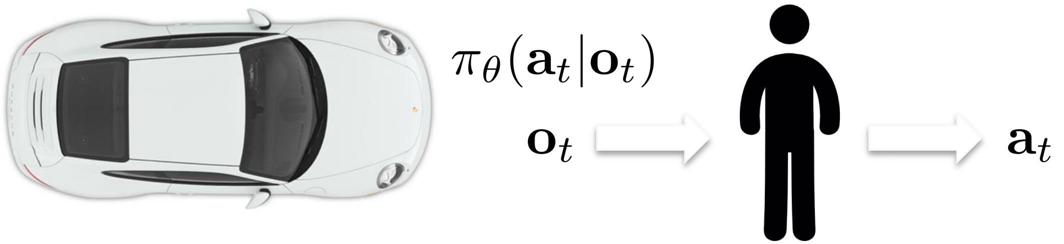
Ross et al. '11

DAgger Example

Ross et al. '13

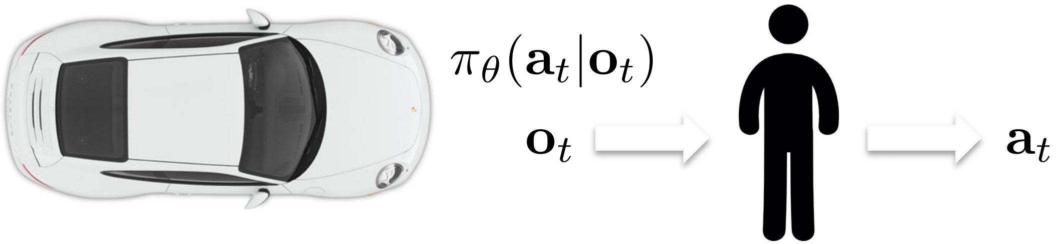
What's the problem?

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?

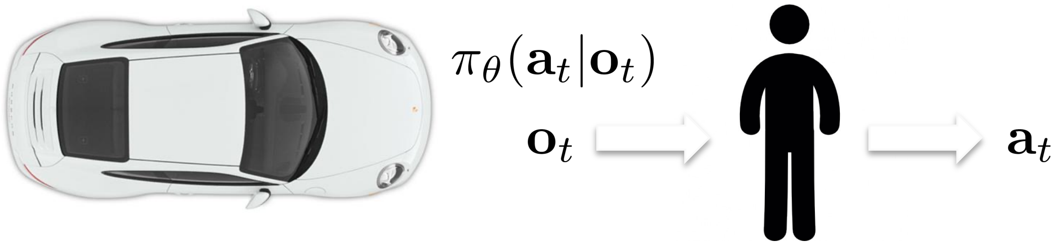
1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$ ← Do at data collection time
3.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



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

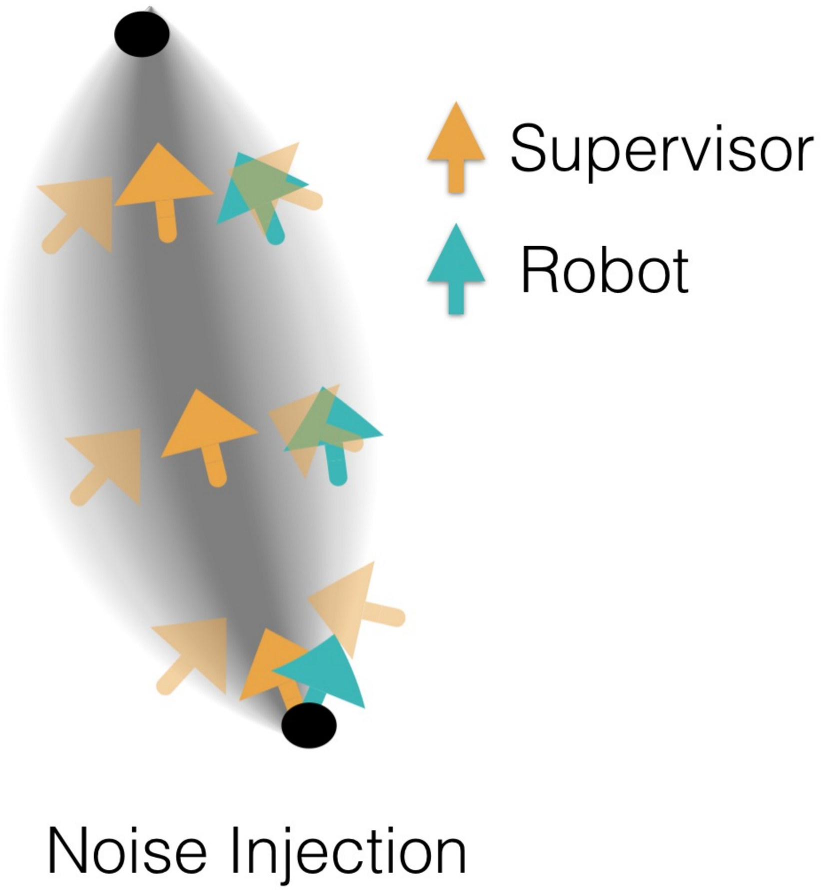
Do at data collection time



Ross et al. '11

Noising the Data Collection Process

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



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

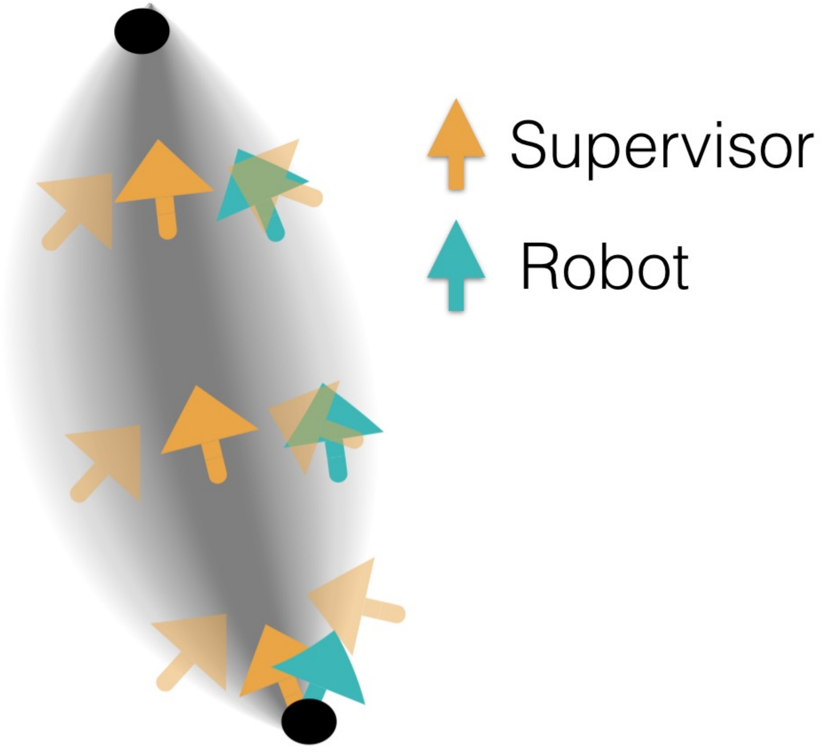
Maximize likelihood

Under noise during data collection

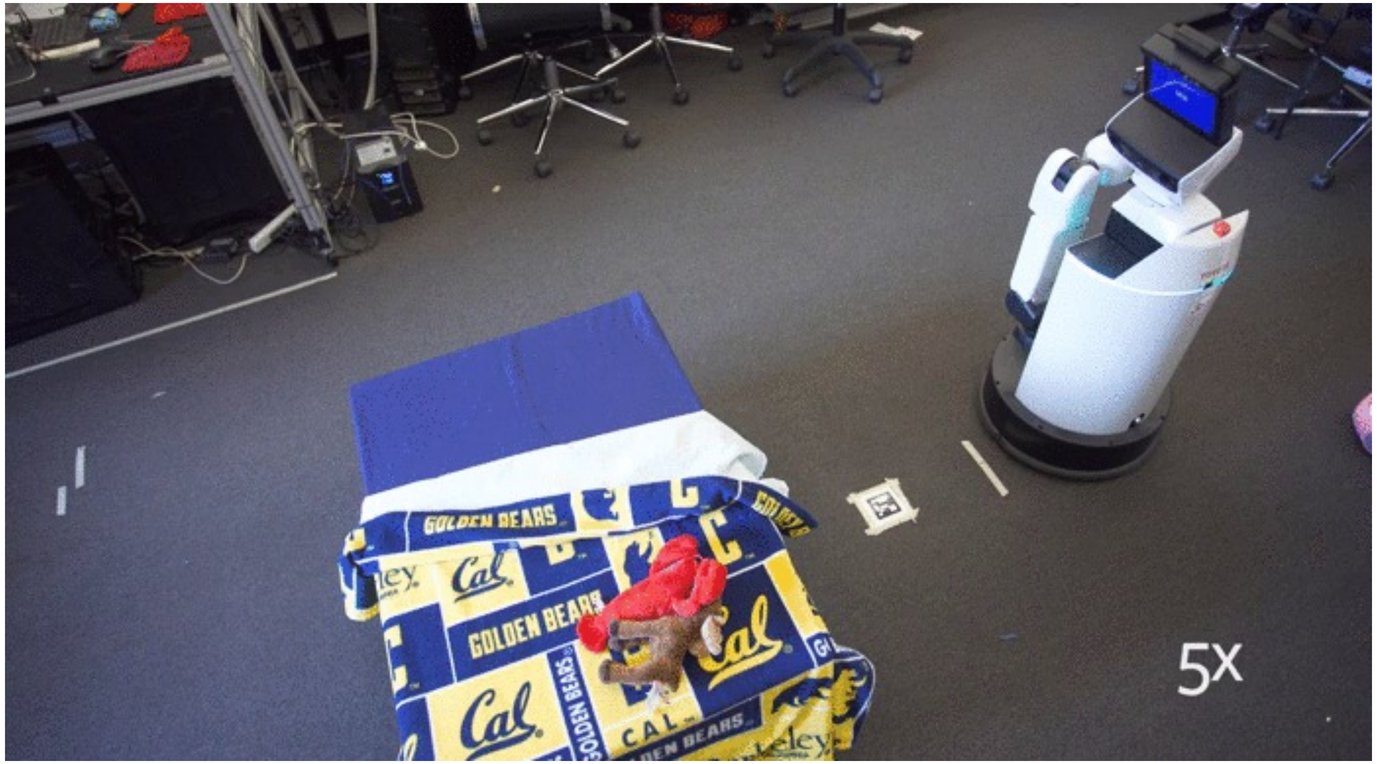
DART: Noise Injection for Robust Imitation Learning, Laskey et al CoRL '17

Noising the Data Collection Process

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



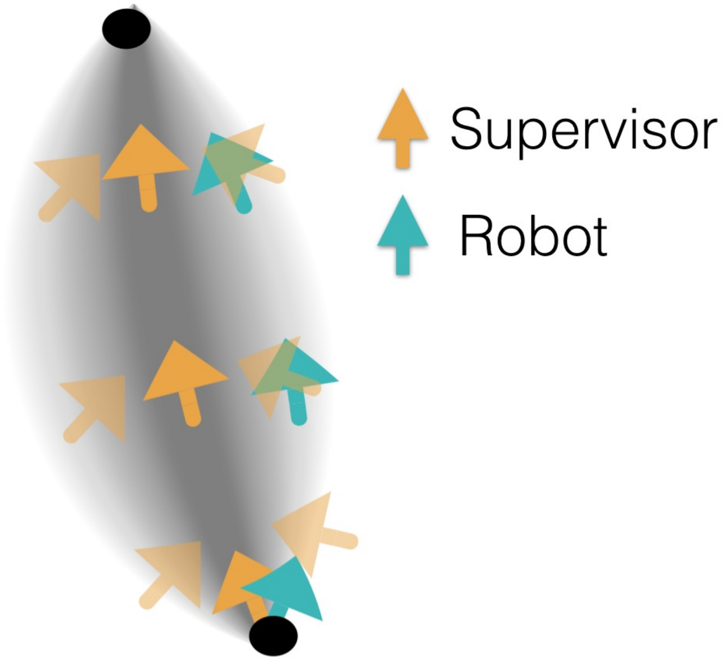
Noise Injection



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 **during** training



Noise Injection

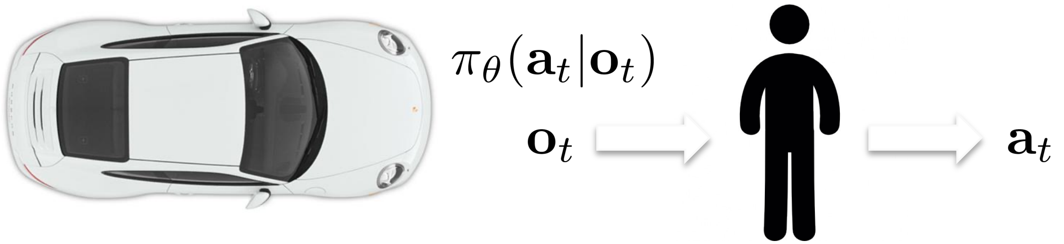
Assumes that the expert **can** actually perform behaviors under noise
→ Not always possible!

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

How might we fix this?

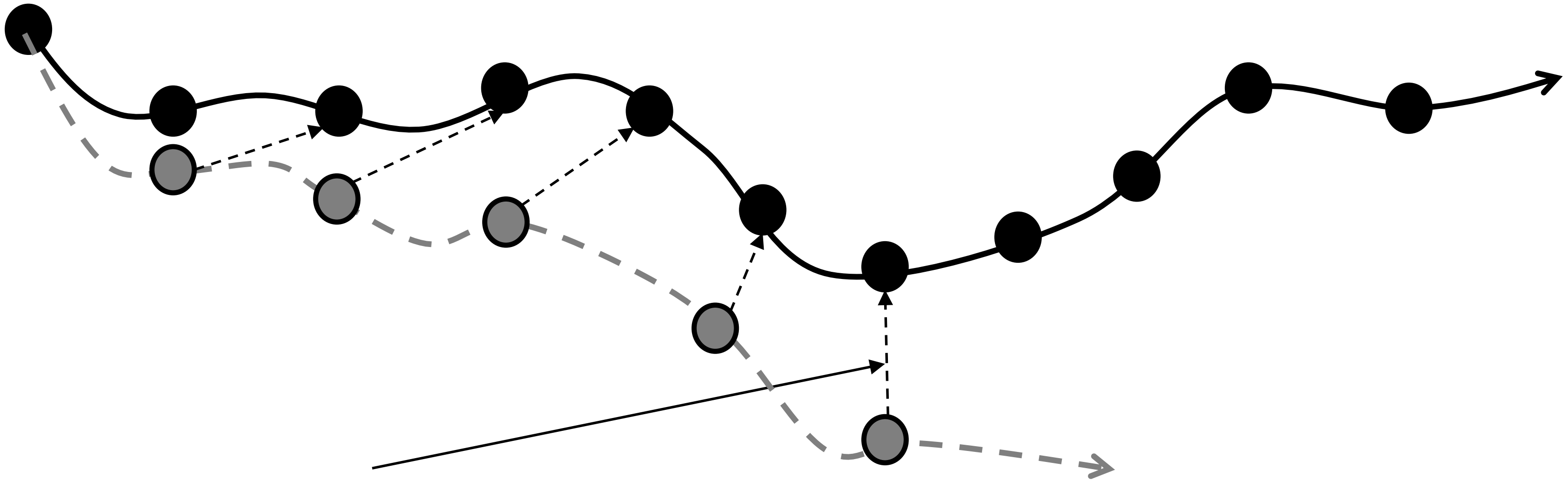
"Generate" corrective labels automatically →

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

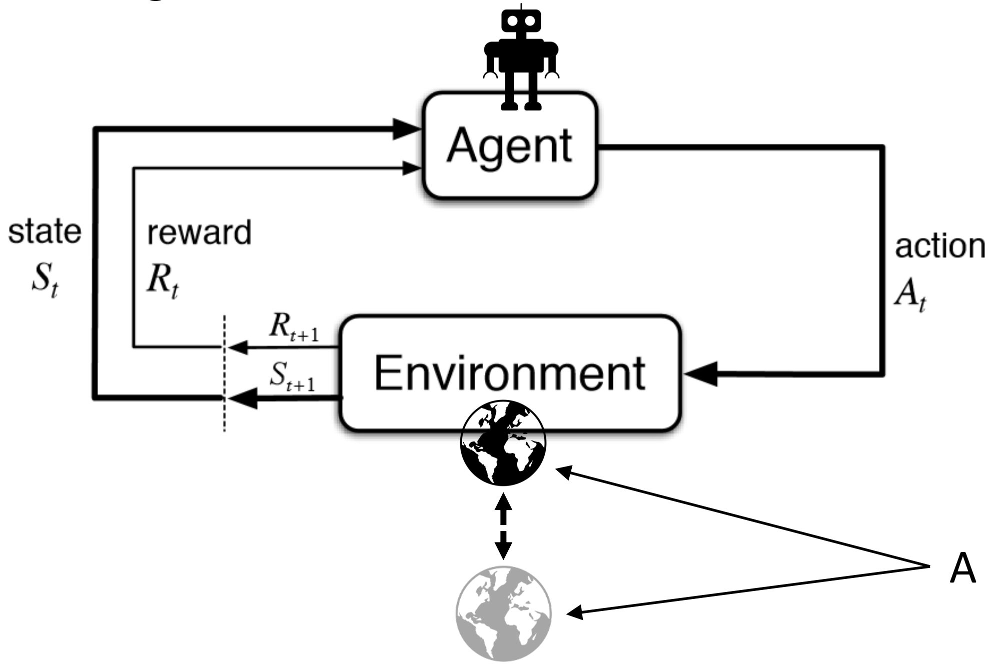


Ross et al. '11

How can we find corrective labels?



How might we obtain these corrections?



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

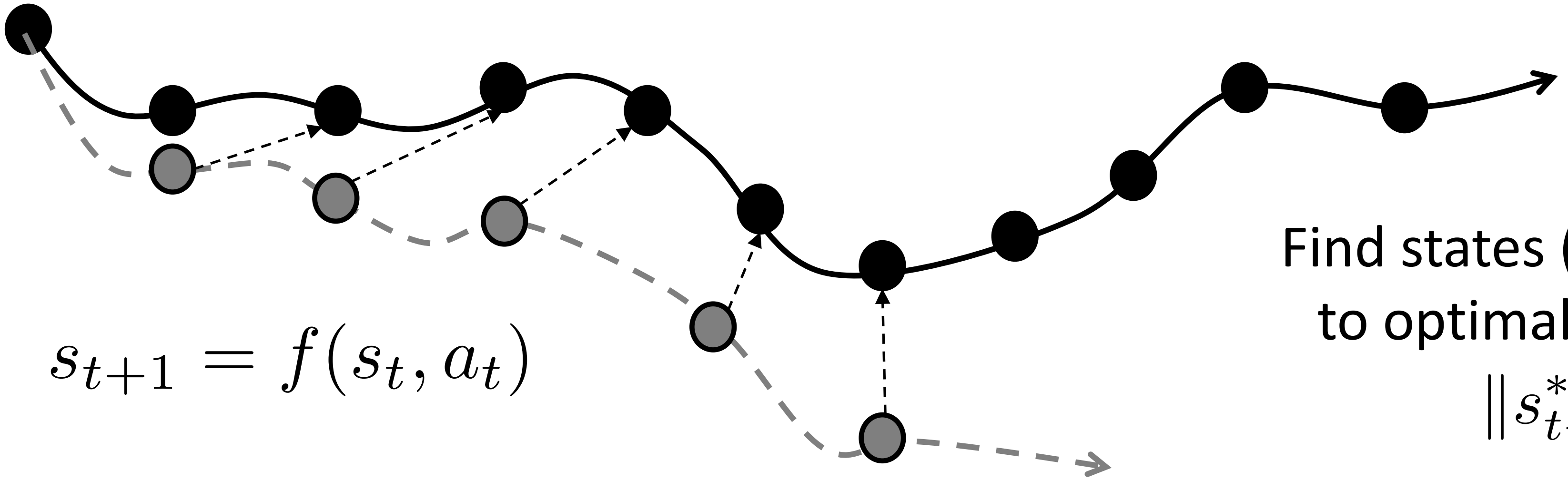
$$\|s^*_{t+1} - f(s_t, a_t)\| \leq \epsilon$$

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

A known/approximate dynamics model can help find corrective labels

CCIL: Continuity-based data augmentation for corrective imitation learning, Ke et al ICLR '24

Generating Corrective Labels for Imitation Learning



$f(s_t, a_t)$

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

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

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

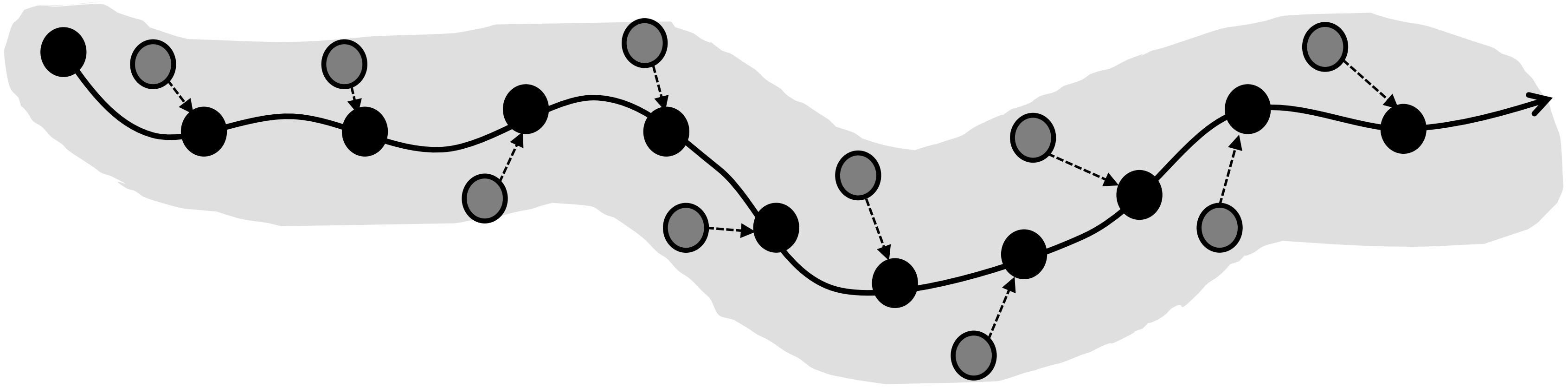
$$\min_{s_t, a_t} \|s^*_{t+1} - f(s_t, a_t)\|$$

Intuition: find labels to bring OOD states back in distribution (where policy can be trusted)

Easy with known dynamics

But dynamics are not known! → More machinery needed with learned dynamics!

Generating Corrective Labels for Imitation Learning with Learned Dynamics



$\hat{f}_\phi$

minimizing MSE on expert data
+ spectral norm

When can we trust learned dynamics $\hat{f}_\phi$?

↓

Under approximately Lipschitz smooth models, trust models around training data

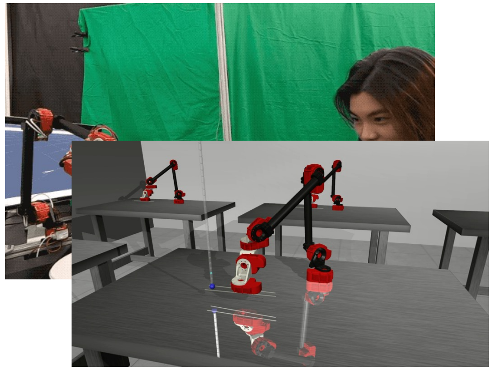
$$\|s^*_{t+1} - \hat{f}_\phi(s_t, a_t)\| \leq \epsilon$$

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

CCIL: Continuity-based data augmentation for corrective imitation learning, Ke et al ICLR '24

Overall Learning Pipeline with Corrective Labels

Expert Data Collection



$D = \{s, a, s'\}$

Supervised learning

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

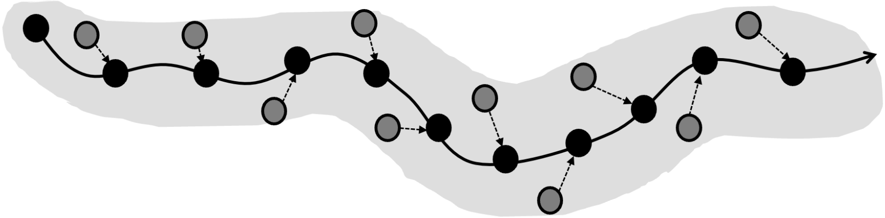
Learning Approximate Dynamics

$$\hat{f}_\phi$$

$$\arg\min_{\hat{f}} \mathbb{E}_{s_j^*, a_j^*, s_{j+1}^* \sim \mathcal{D}^*} \left[\text{MSE} + \lambda_j \cdot \mathit{Lipschitz}(s_j^*, a_j^*) + ||\lambda_j - \bar{\lambda}||_0 \right]$$

$$\mathit{Lipschitz}(s_j^*, a_j^*) = \mathbb{E}_{\Delta_s \sim \mathcal{N}} \left(\frac{\|\hat{f}'(s_j^* + \Delta_s, a_j^*) - \hat{f}'(s_j^*, a_j^*)\|}{\|\Delta_s\|} \right)$$

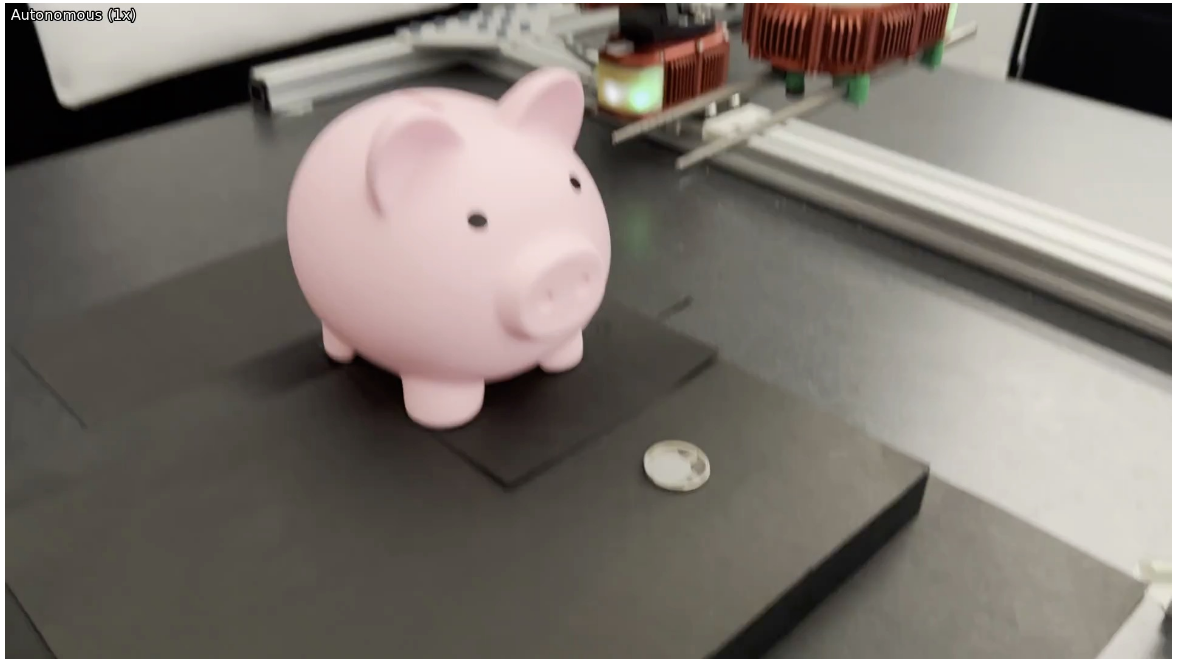
Generating Corrective Labels



How well does generating corrective labels work?

With corrective labels

Without corrective labels

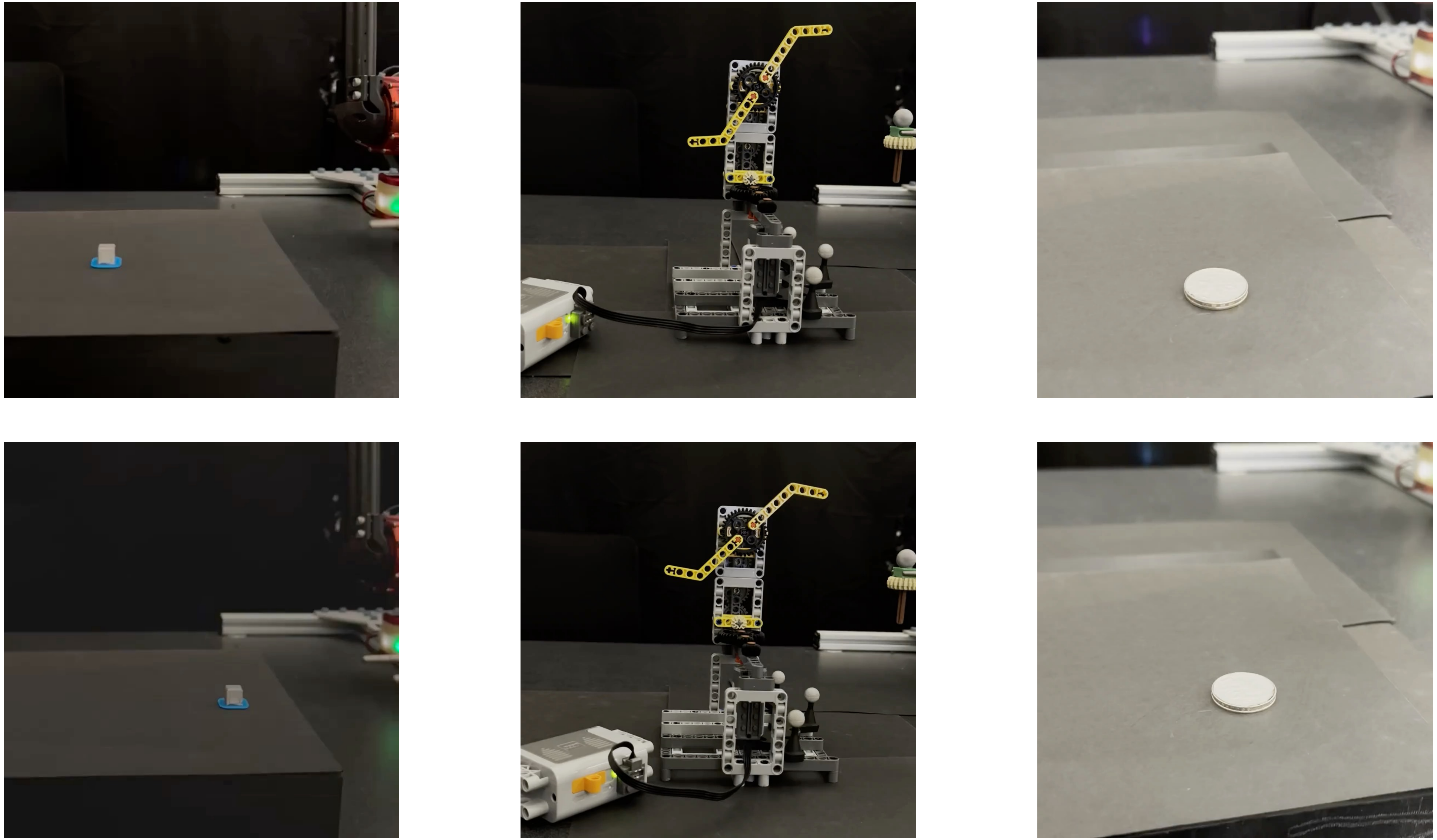


CCIL: Continuity-based Data Augmentation for Corrective Imitation Learning, Ke et al '24
Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels, Deshpande et al '24

How well does generating corrective labels work?

With corrective labels

Without corrective labels



So does this solve all the issues in imitation?

Lecture Outline

A Formalism for Sequential Decision Making

↓

Imitation Learning: Behavior Cloning

↓

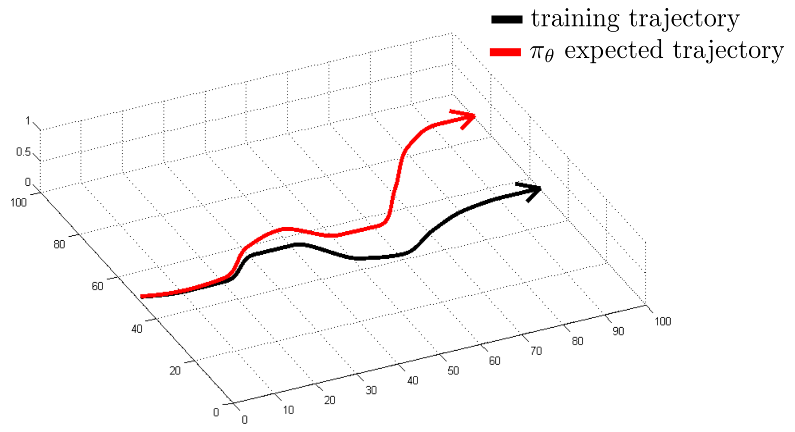
Imitation Learning: Improvements – Compounding Error

↓

Imitation Learning: Improvements – Multimodality

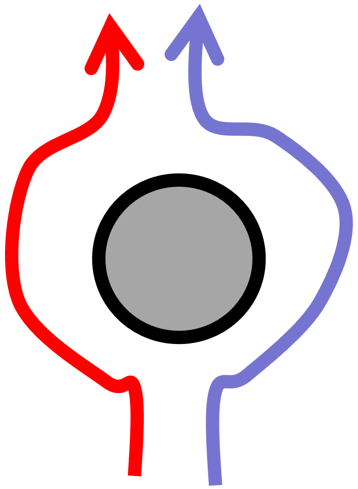
Can we make it work without more data?

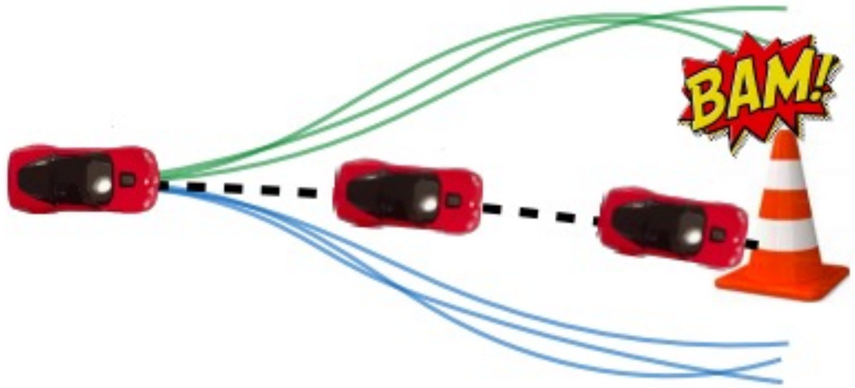
- DAgger addresses the problem of distributional "drift"
- What if our model is so good that it doesn't drift?
- Need to mimic expert behavior very accurately
- But don't overfit!

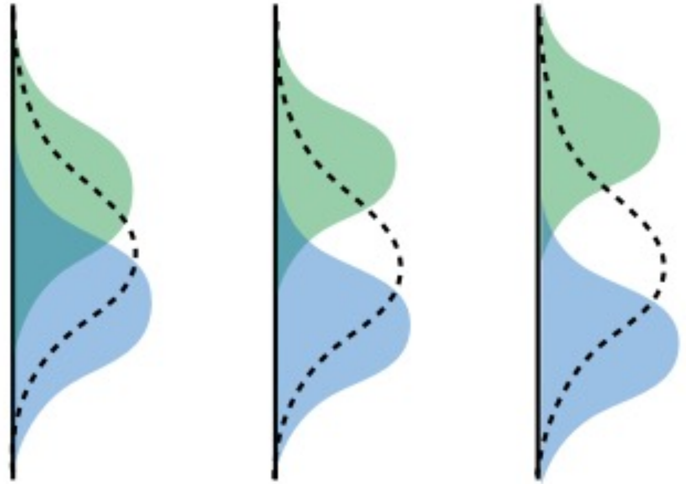


Why might we fail to fit the expert?

Multimodal behavior.. amongst other reasons



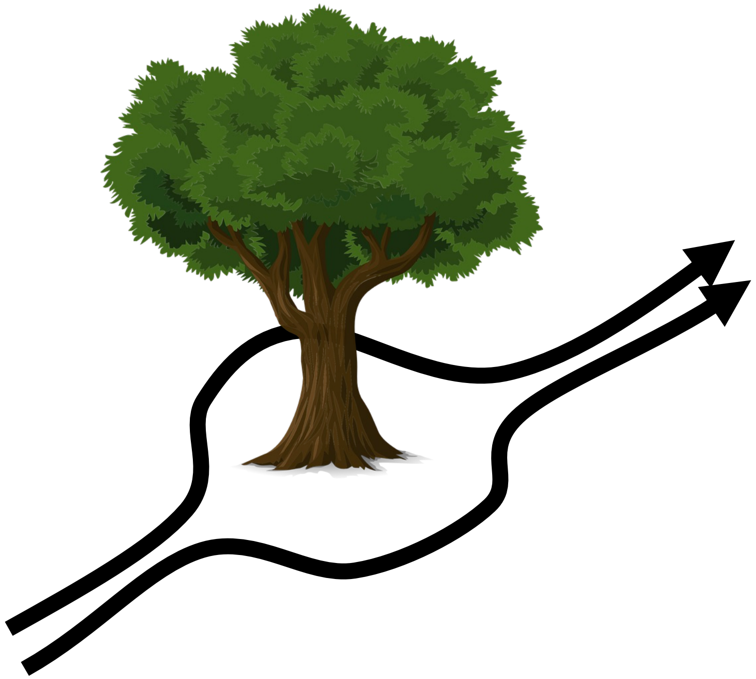


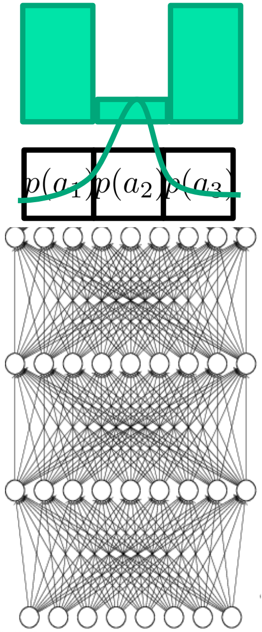


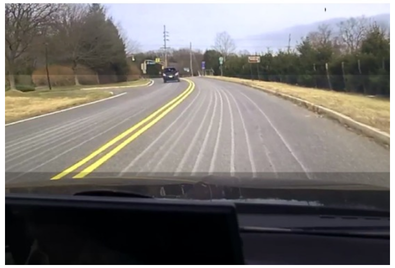
Not a matter of network size! It's about distributional expressivity

Why might we fail to fit the expert?

Multimodal behavior → use more **expressive** probability distributions





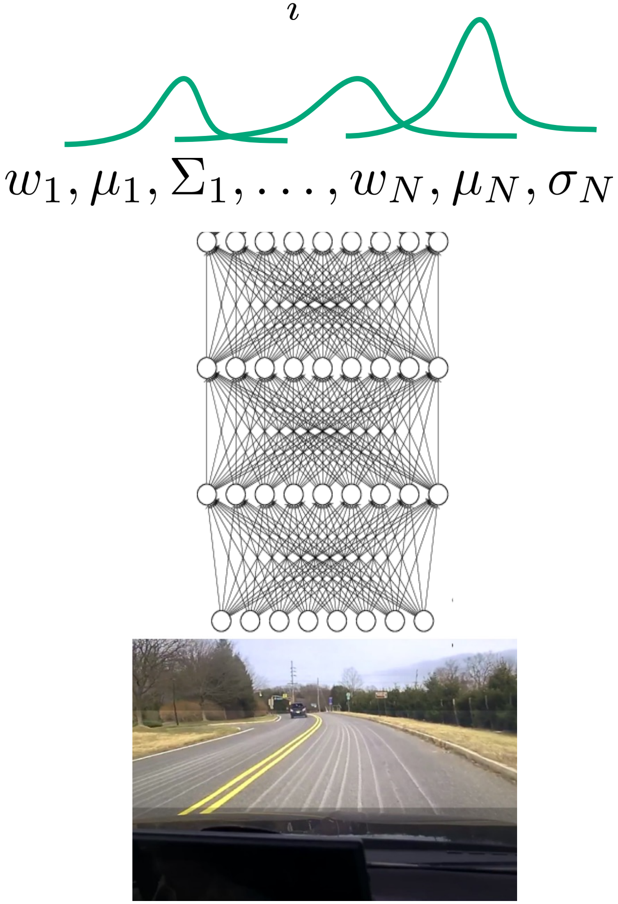


1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. …

Why might we fail to fit the expert?

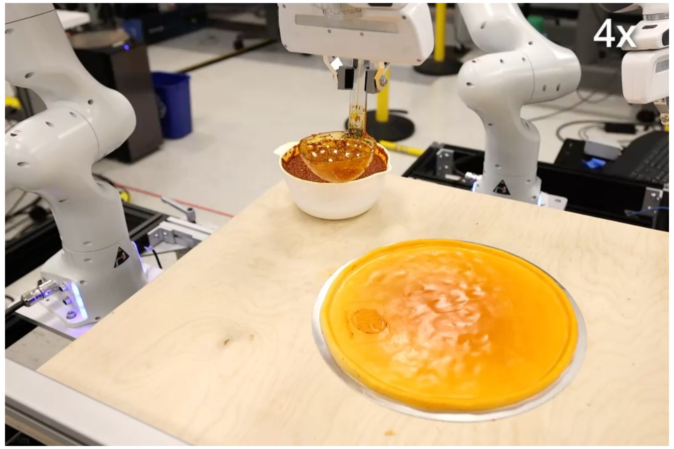
1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. …

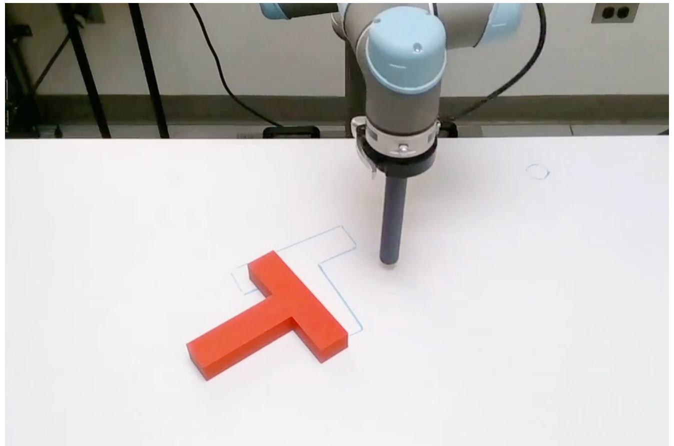
$$\pi(\mathbf{a}|\mathbf{o}) = \sum_i w_i \mathcal{N}(\mu_i, \Sigma_i)$$

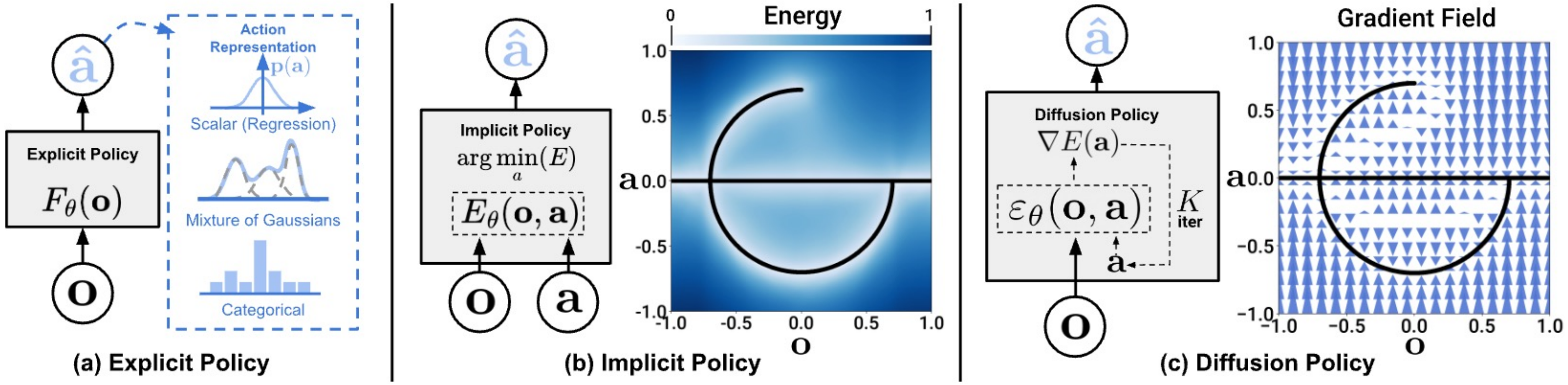


Why might we fail to fit the expert?

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







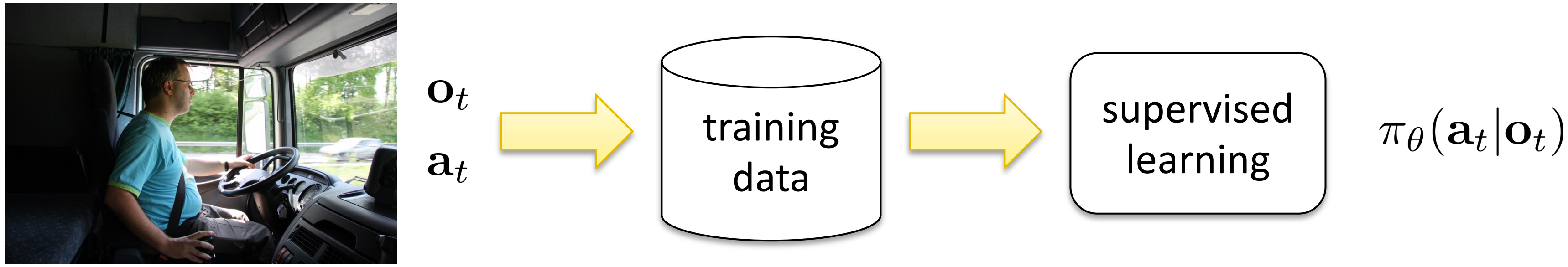
Some cool imitation videos

1x and tesla humanoid robots

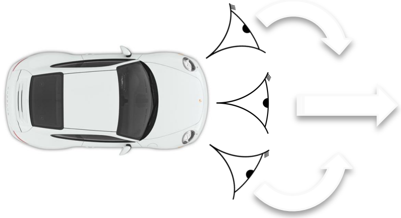
ALOHA Manipulation

TRI Diffusion Policies

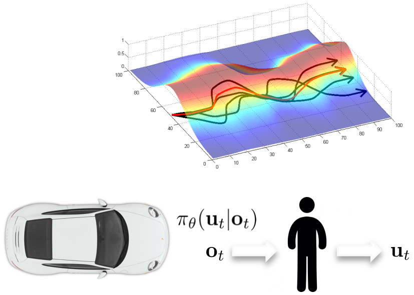
Perspectives on Imitation – don't believe everything you see online



- 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!



Lecture Outline

A Formalism for Sequential Decision Making

↓

Imitation Learning: Behavior Cloning

↓

Imitation Learning: Improvements – Compounding Error

↓

Imitation Learning: Improvements – Multimodality

Class Outline

State Estimation

- Robotic System Design
- Filtering
- Localization
- SLAM

Control

- Feedback Control
- PID Control
- MPC
- LQR

Planning

- Search
- Heuristic Search
- Motion Planning
- Lazy Search

Learning

- Imitation Learning
- Policy Gradient
- Actor-Critic
- Model-Based RL