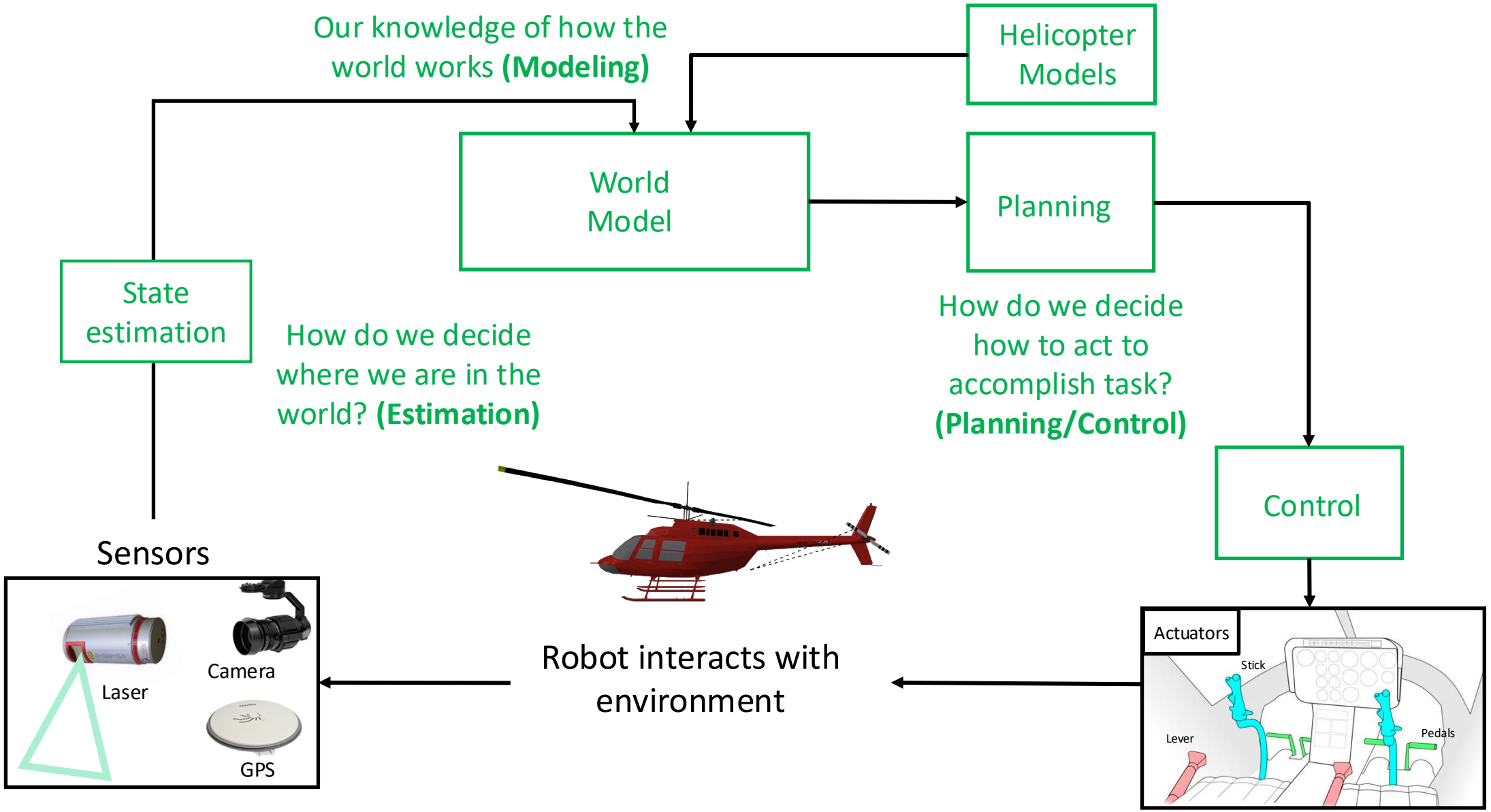
CSE 478 Robot Autonomy
Imitation Learning

Siddhartha Srinivasa (siddh@)
Abhishek Gupta (abhgupta@)

TAs:
Rohan Baijal (rbaijal@)
Sidhartha Talia (sidtalia@)
Christopher Tan (tan7271@)
Helen Wang (yiruwang@)

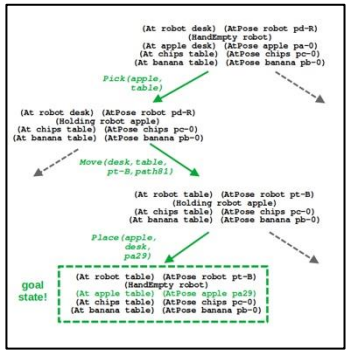
We have built a model-based control system!

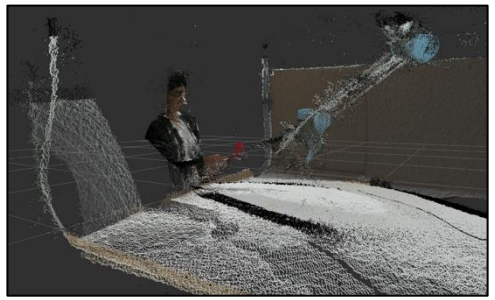


Are we done?

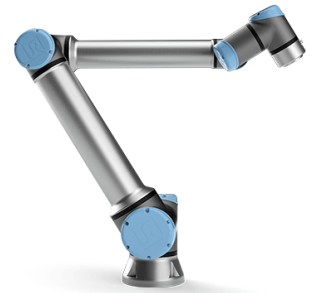
Model-based Control for Robotics

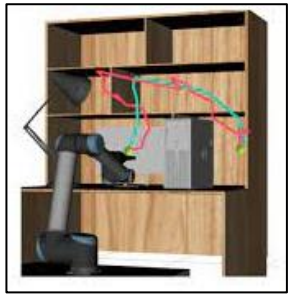
$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \tau_g(\mathbf{q}) + \mathbf{B}\mathbf{u},$$

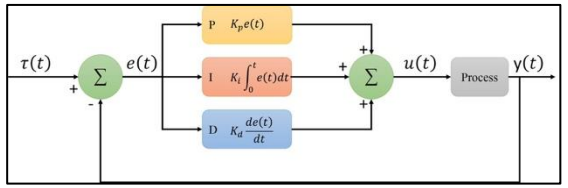




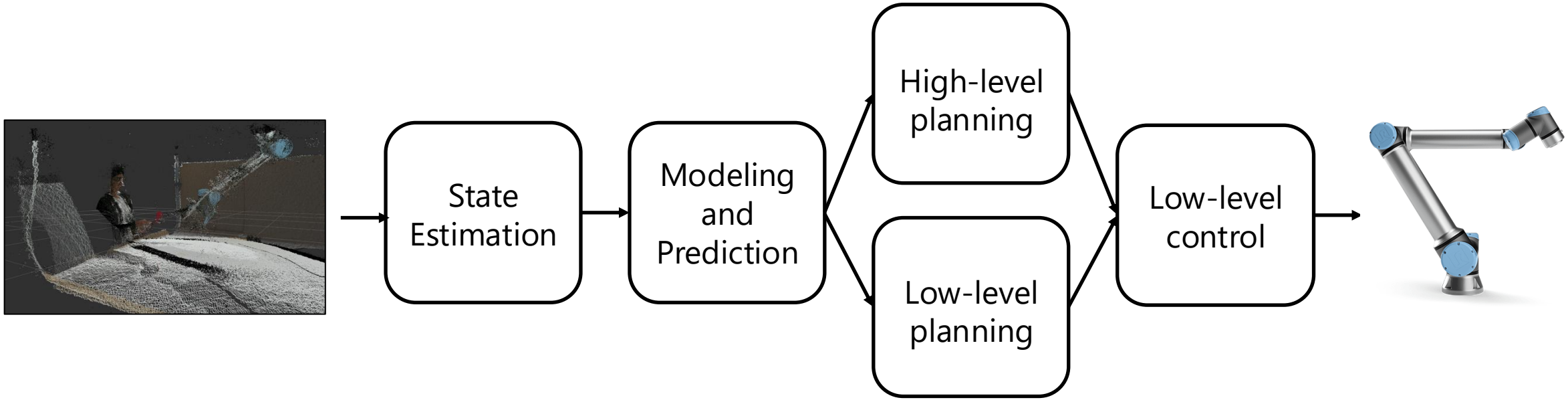
State Estimation → Modeling and Prediction → Planning → Low-level control



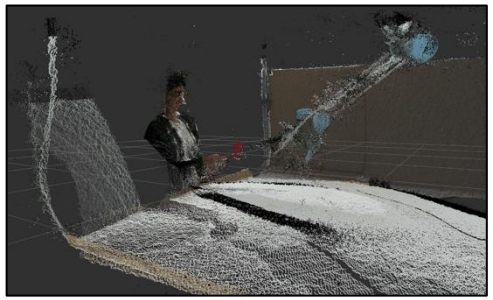




What does a typical model based look like?



End-to-End Learning Based Control for Robots



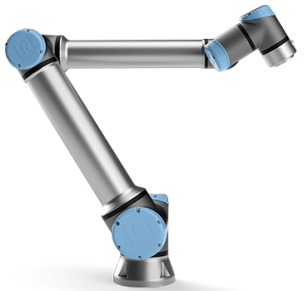
State Estimation

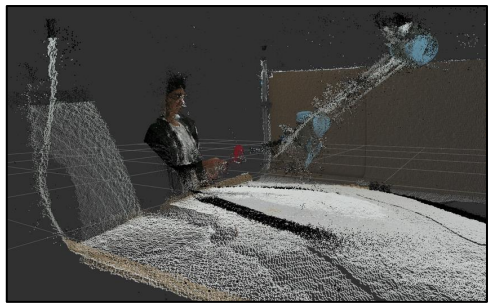
Modeling and Prediction

High-level planning

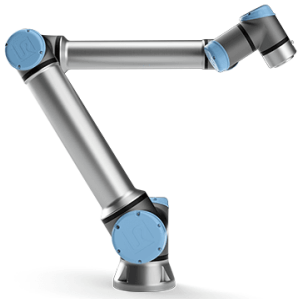
Low-level planning

Low-level control





End to end policy – perception + control



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

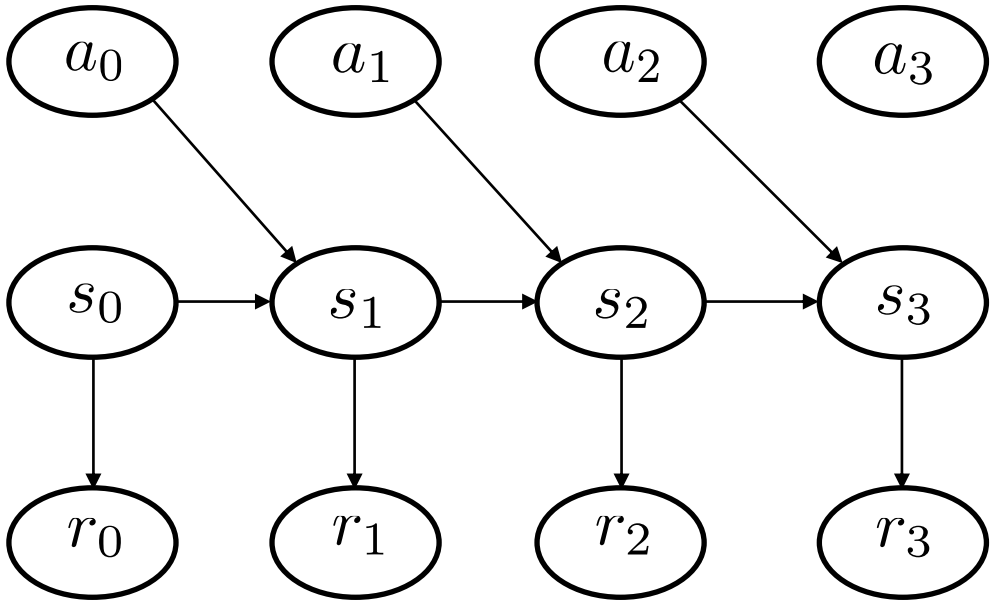
A Formalism for Sequential Decision Making

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

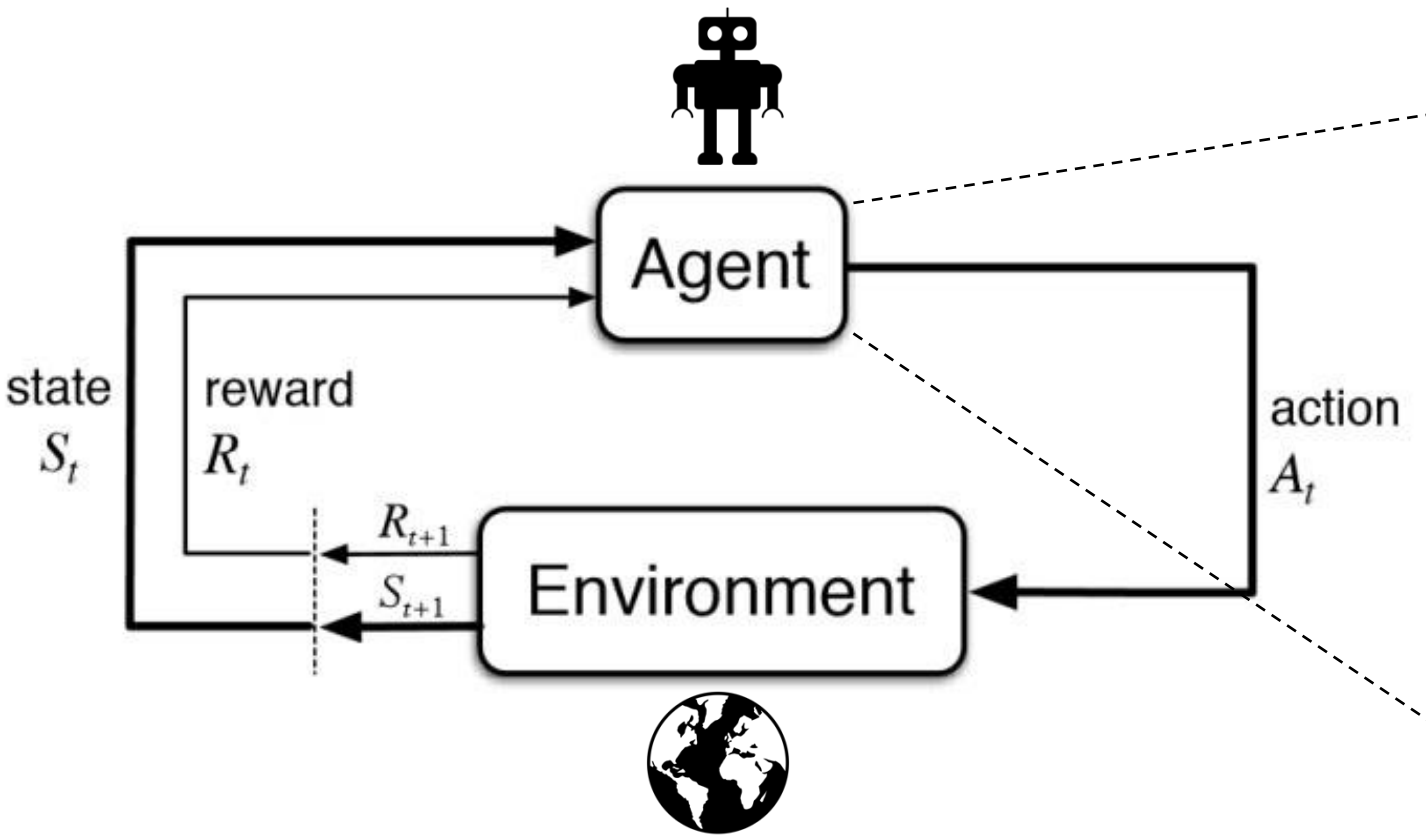
Reward: Success or Failure

Transition: World physics

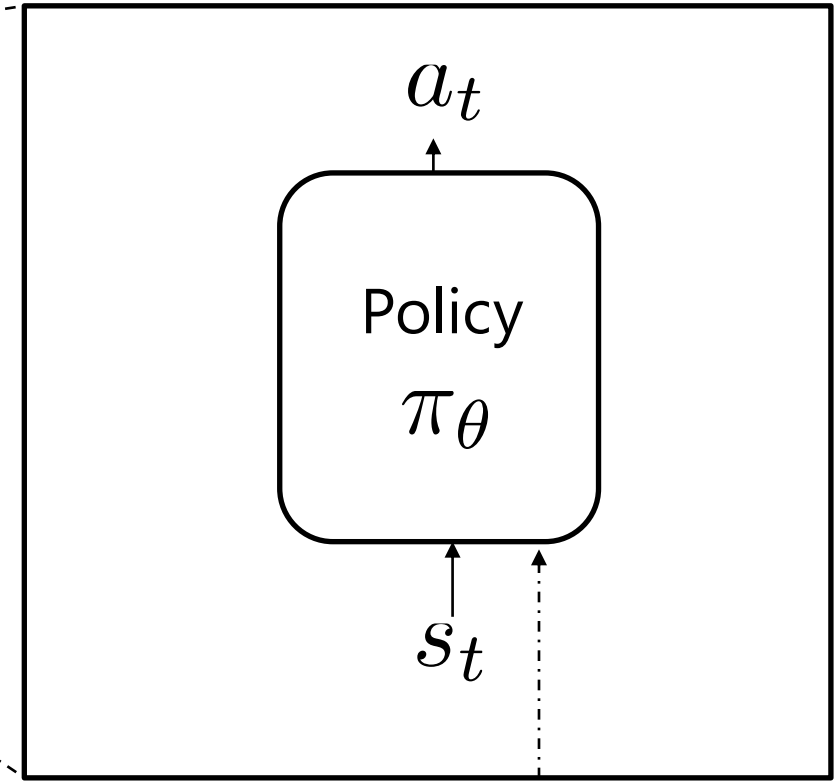
Rewards and Resets

- Sparse reward – 1 when you succeed and 0 elsewhere
- Dense reward – sort-of like a heuristic
  - Increase reward as you approach the goal
  - Negative reward for approaching obstacles
- Reward hacking – encoding the policy in the reward 
- Reset function – when do you give up?
- Reset function hacking – OpenAI gym 

Reinforcement Learning Formalism



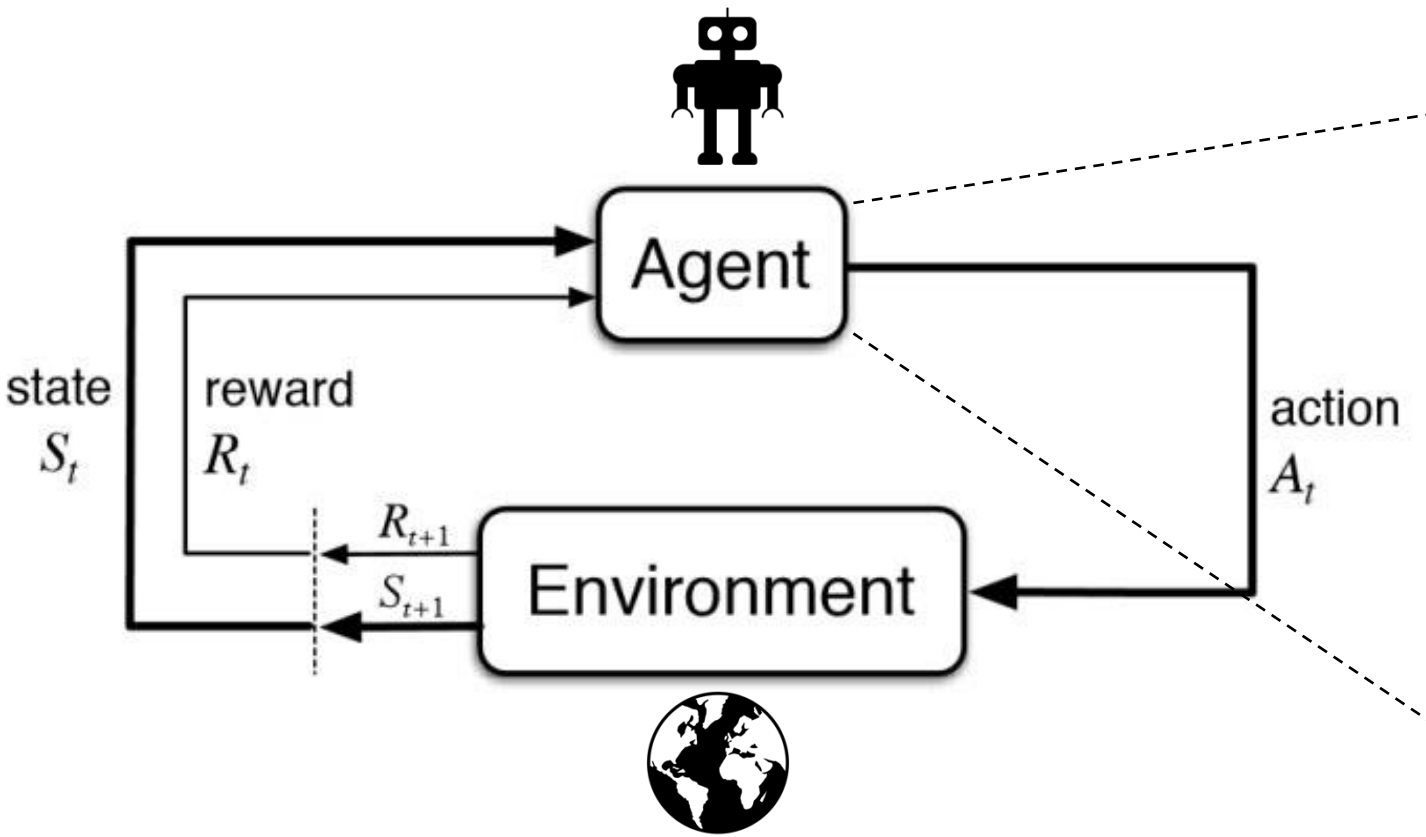
Rules for choosing actions



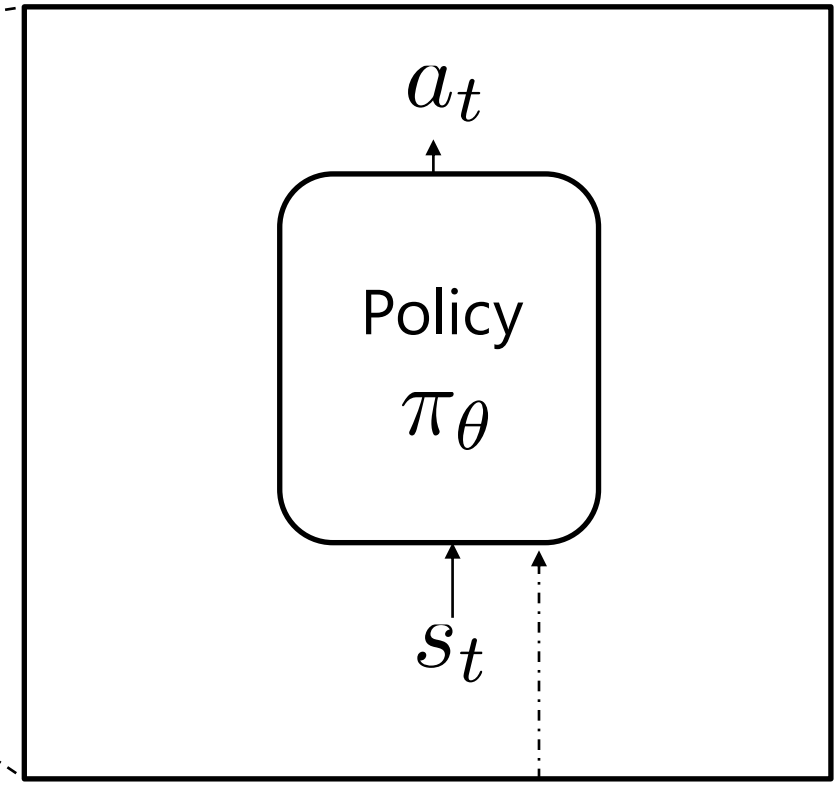
Maximize the sum of expected rewards under policy

Needs to be learned

Reinforcement Learning Formalism



Rules for choosing actions



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

Trajectory sampled using policy

Needs to be learned

Why isn't this just optimal control?

Optimal control

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

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

Reinforcement Learning

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

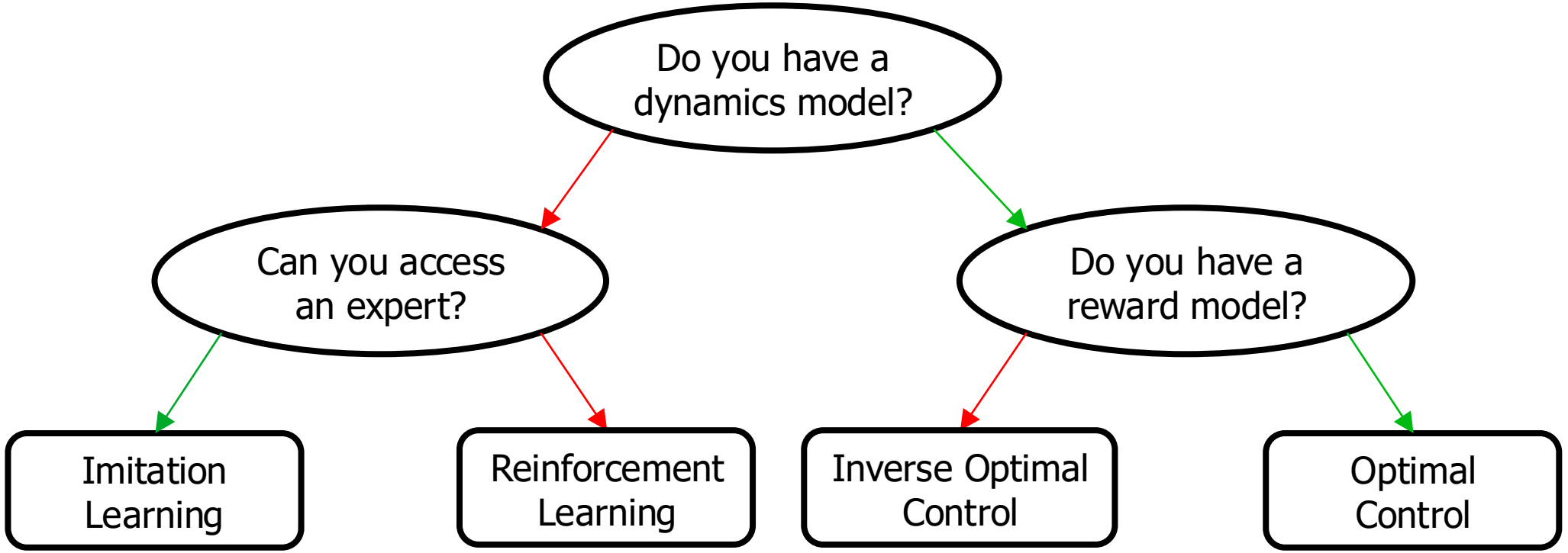
Cosmetic differences:

- Costs vs rewards
- Often discrete vs continuous time

Real differences:

- Known model vs sample-able model

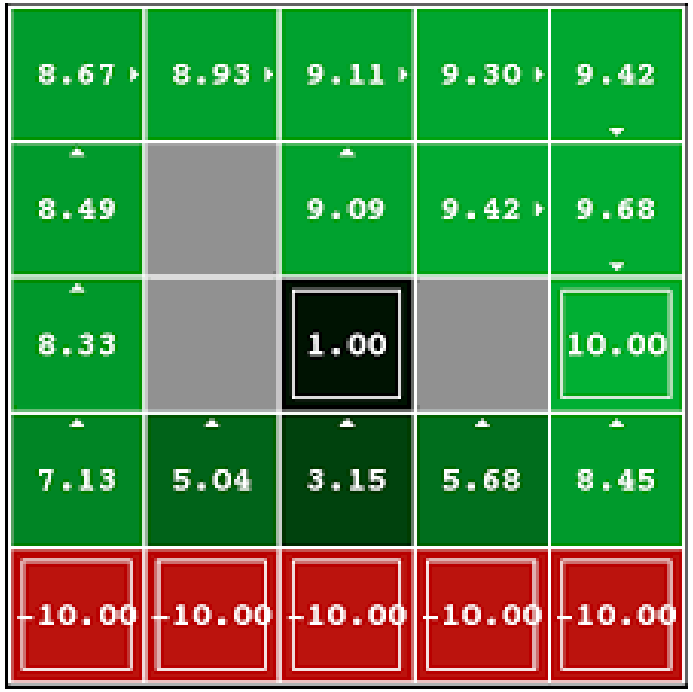
Roadmap for Control and Learning



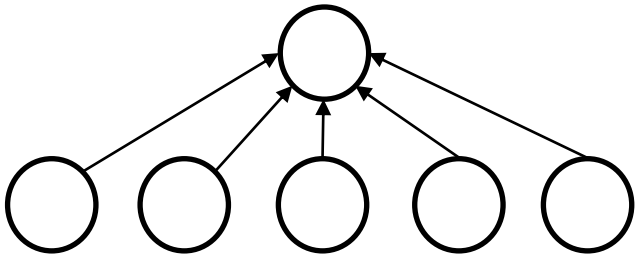
Main thing to learn - Policies

Policies are mappings from states to optimal actions

Tabular

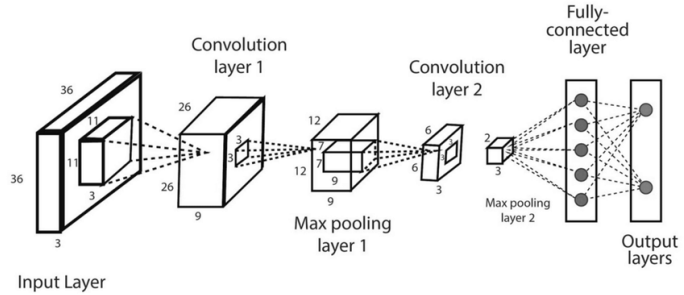


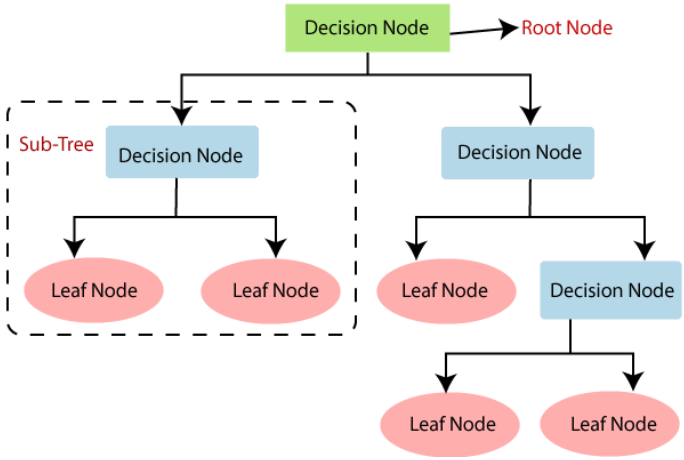
Linear



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

Arbitrary function approx

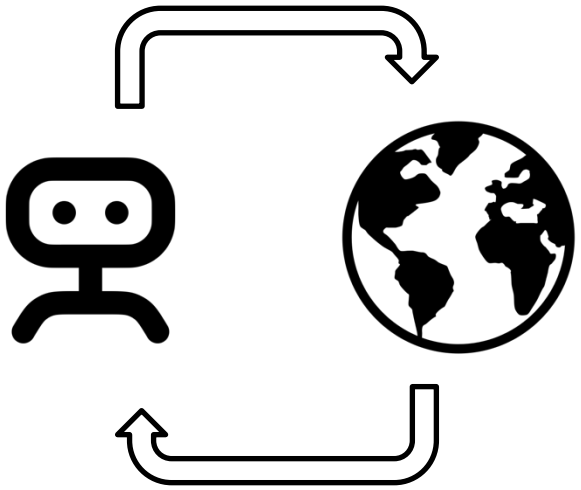




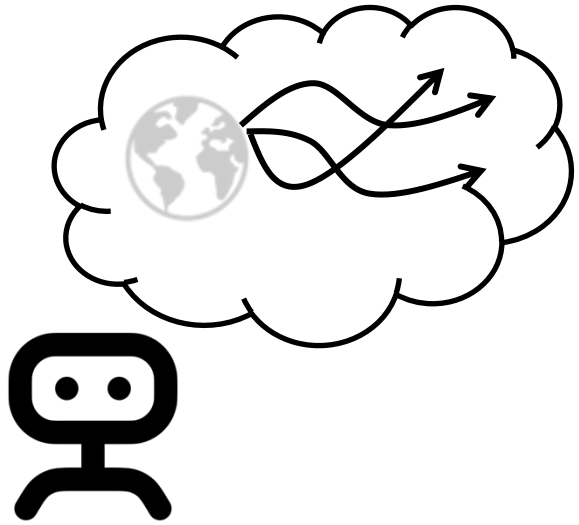
Ok so how can we learn policies?

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

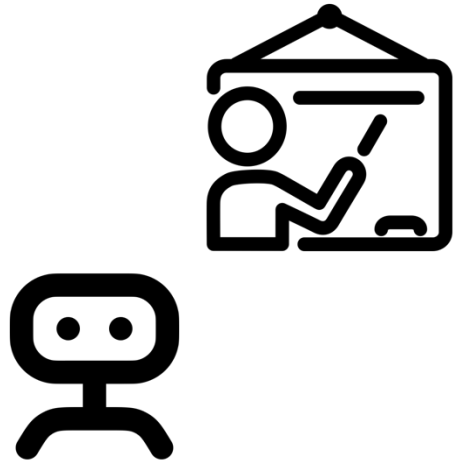
Model-free RL



Model-based RL



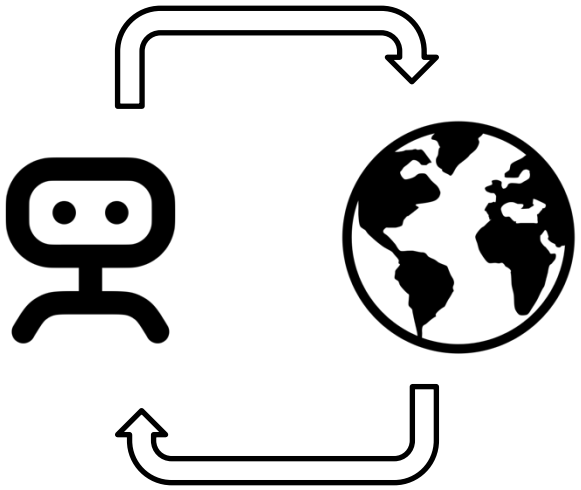
Imitation Learning



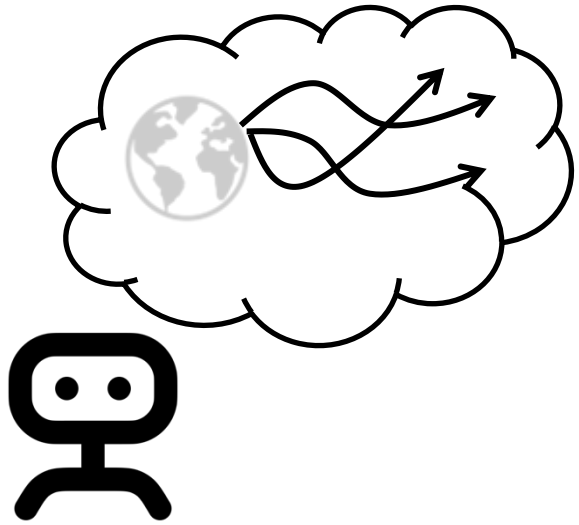
Ok so how can we learn policies?

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

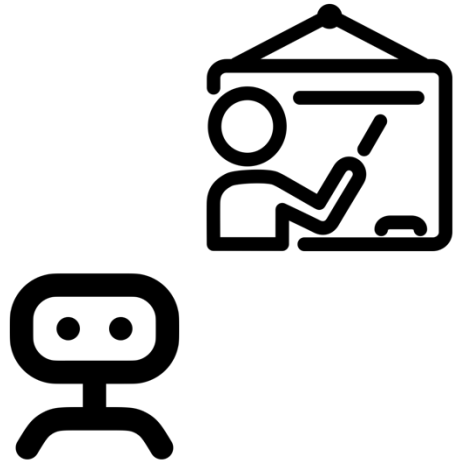
Model-free RL



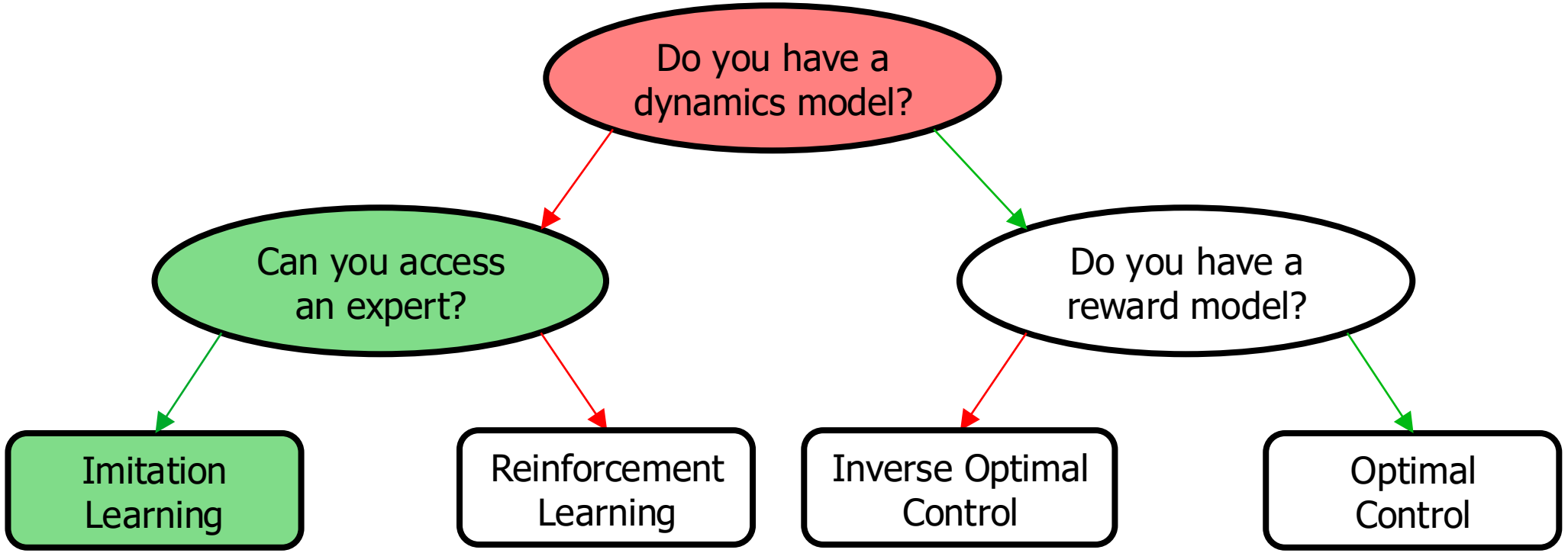
Model-based RL



Imitation Learning



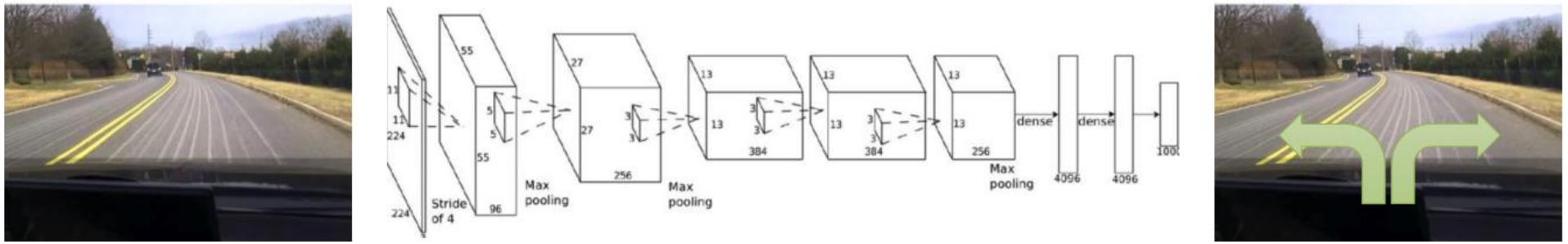
Roadmap for Control and Learning



Imitation Learning: Intuition

Given: Demonstrations of optimal behavior from expert

Goal: Train a policy to mimic the demonstrator



Pros: No rewards, online experience needed (?)

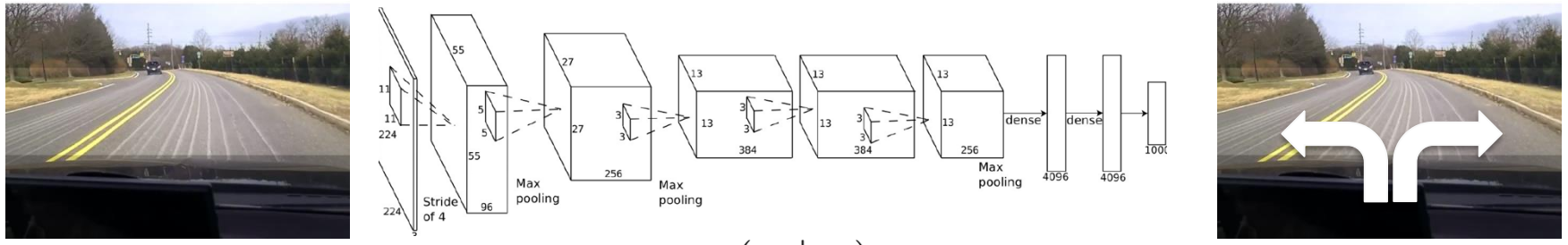
Idea 1: Imitation Learning via Behavior Cloning

Given: Demonstrations of optimal behavior

Goal: Train a policy to mimic the demonstrator

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

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

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

Supervised Loss Function

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

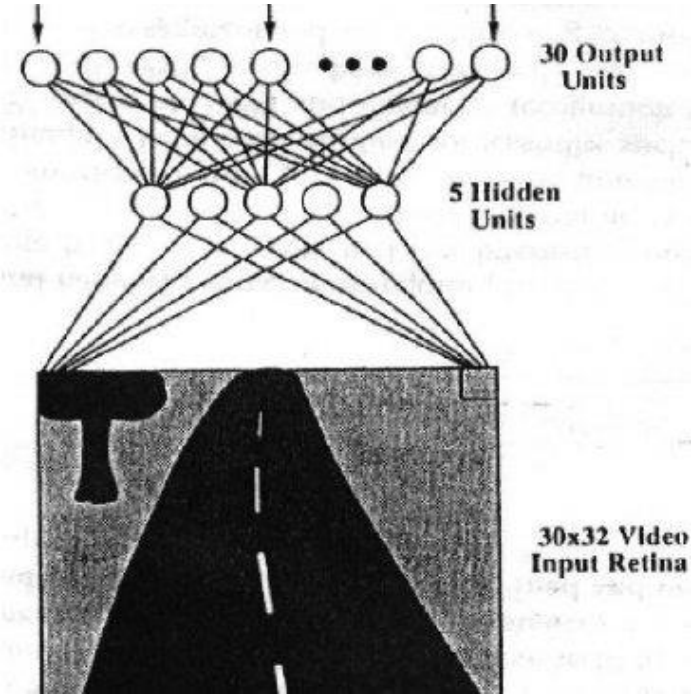
Find the best policy from our policy class

Iterate over expert demonstrations

Probability of our current policy taking the same action the expert took at the same state the expert visited

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?

Does Behavior Cloning work well in practice?

Expert Intervention Learning: An online framework for robot learning from explicit and implicit human feedback.

J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S.S. Srinivasa. *Autonomous Robots*, 46, 2022.

Does Behavior Cloning work well in practice?

Expert Intervention Learning: An online framework for robot learning from explicit and implicit human feedback.

J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S.S. Srinivasa. *Autonomous Robots*, 46, 2022.

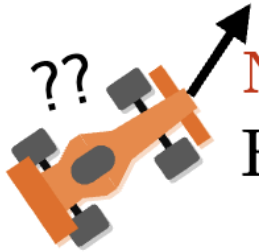
Does Behavior Cloning work well in practice?

Why does Behavior Cloning fail?

Why does Behavior Cloning fail?

No training data

Error: 1.0

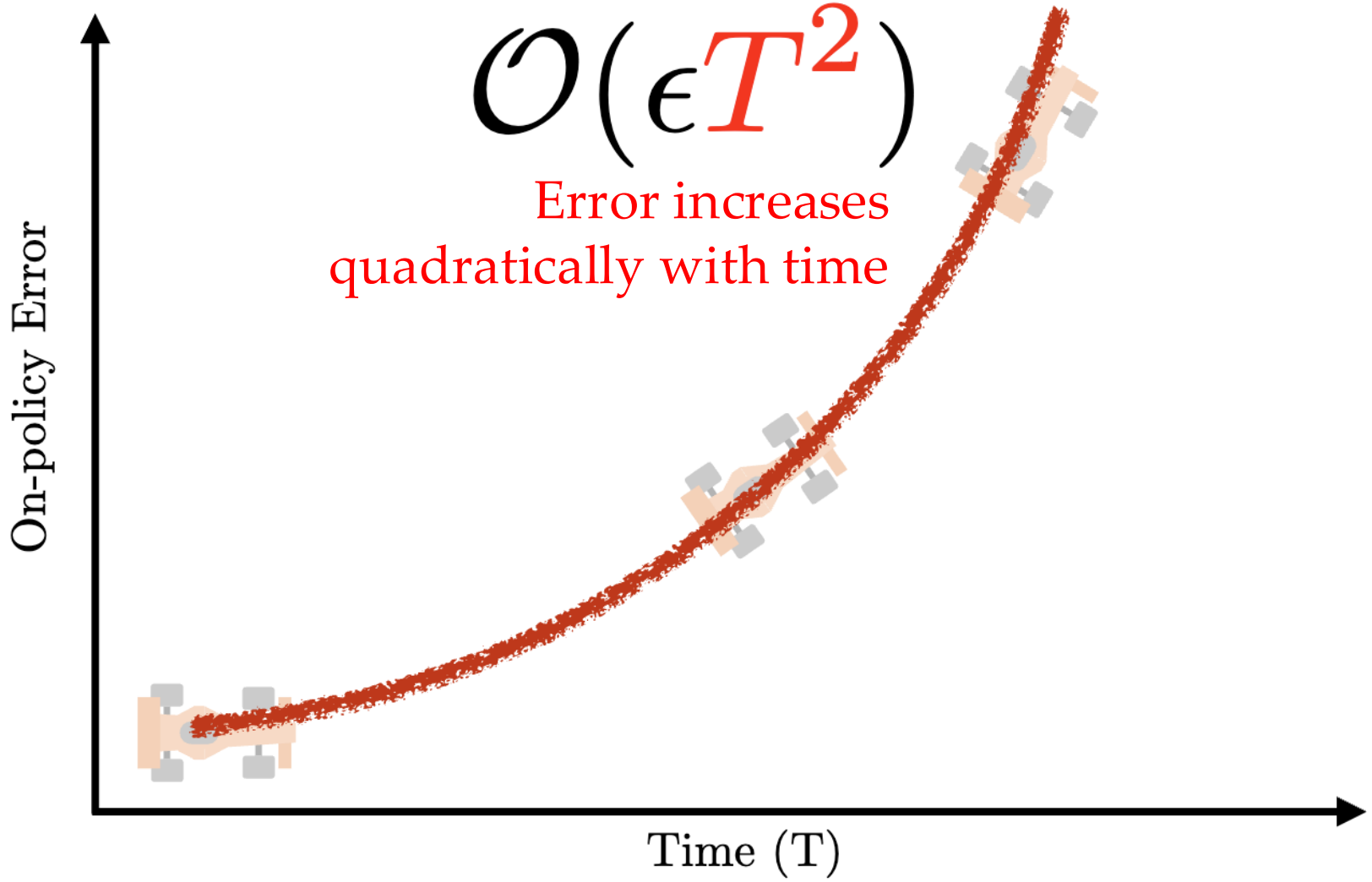


No training data

Error: 1.0

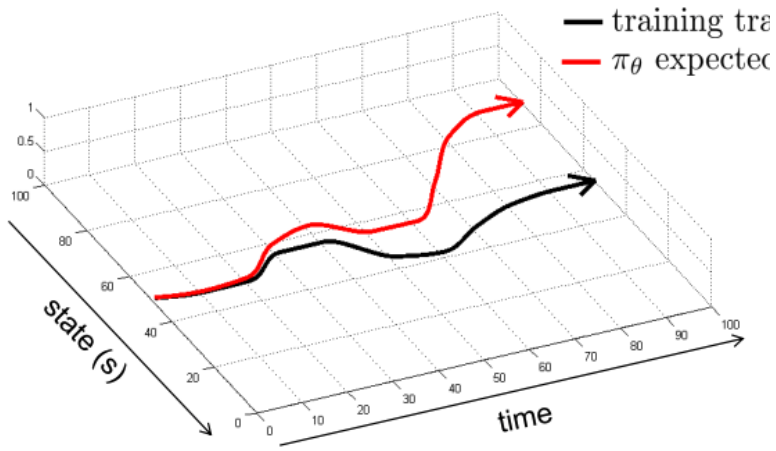
Demonstrations

Errors feedback and compound



Covariate Shift

- Imitation Learning ≠ Supervised Learning

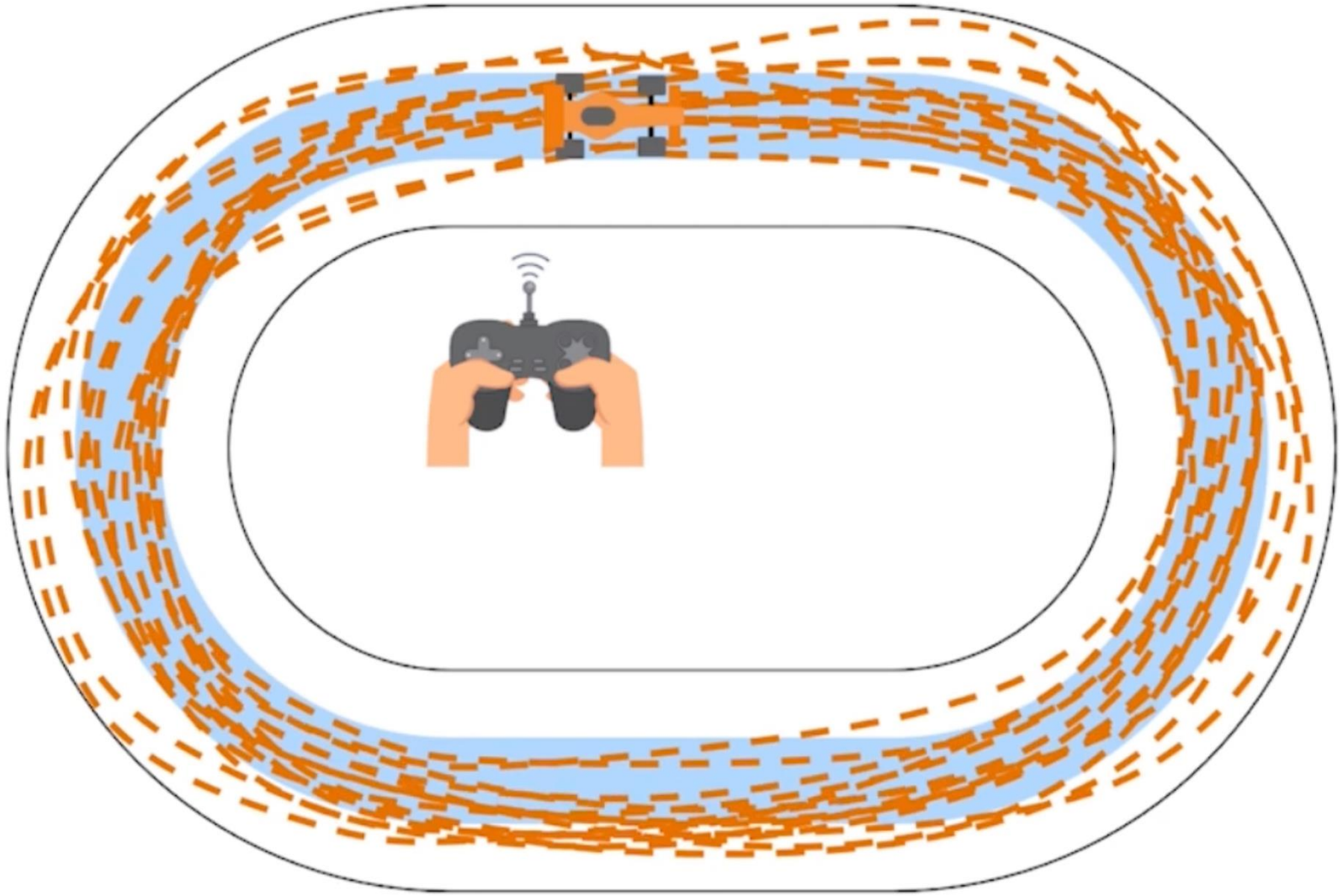


$$\arg\max_\theta \mathbb{E}_{(s^*,a^*)\sim\mathcal{D}}\left[\log \pi_\theta(a^*|s^*)\right] \qquad \mathbb{E}_{(s,a)\sim\rho(\pi)}\left[1(a=a^*)\right]$$

Not the same!

So what's with all the cool demos?

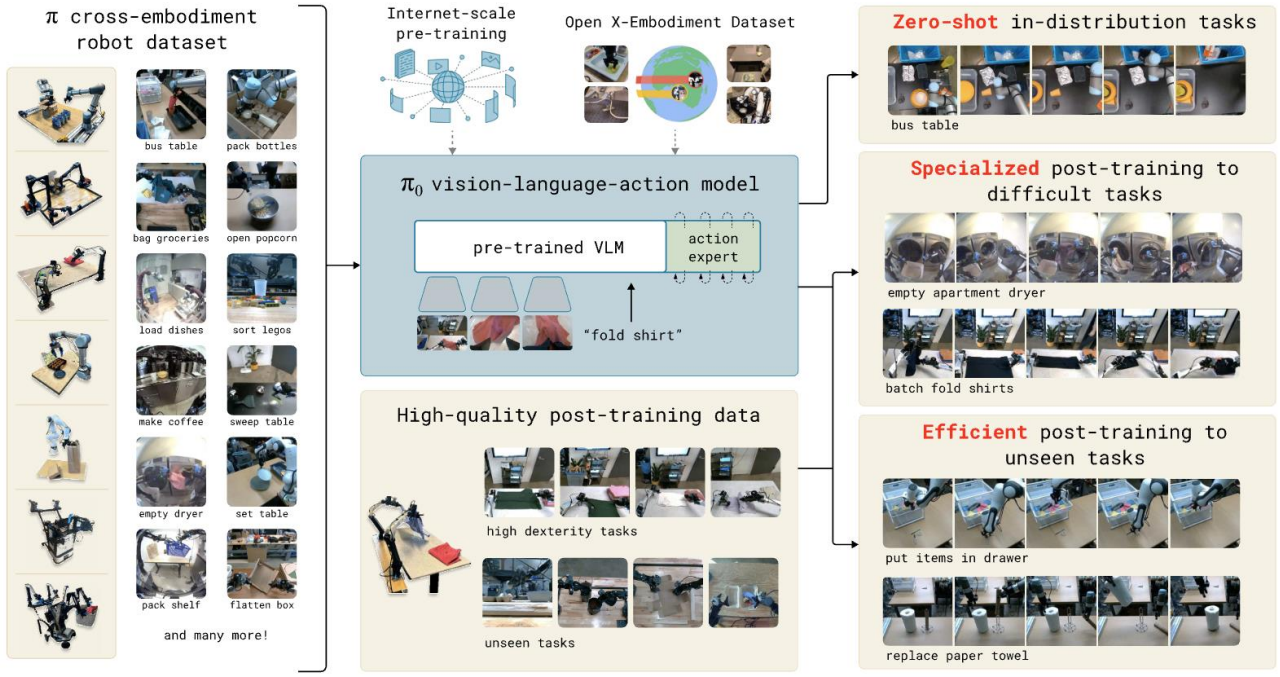
Need a quadratic amount of data!
Might have poor performance even then!



So is all hope lost?

π_0: A Vision-Language-Action Flow Model for General Robot Control

Physical Intelligence
Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, Ury Zhilinsky
https://physicalintelligence.company/blog/pi0



The contributions of our work consist of a novel generalist robot policy architecture based on VLM pre-training and flow matching, and an empirical investigation of pre-training/post-training recipes for such robot foundation models. We evaluate our model out of the box with language commands, with fine-tuning to downstream tasks, and in combination with a high-level semantic policy that outputs intermediate language commands to perform complex and temporally extended tasks. While our model and system make use of a variety of ideas presented in recent work, the combination of ingredients is novel, and the empirical evaluation demonstrates a level of dexterity and generality that goes significantly beyond previously demonstrated robot foundation models. We evaluate our approach by pre-training on over 10,000 hours of robot data, and fine-tuning to a variety of dexterous tasks, including laundry folding (see Figure 2), clearing a table, putting dishes in a microwave, stacking eggs into a carton, assembling a box, and bagging groceries.

Lecture Outline

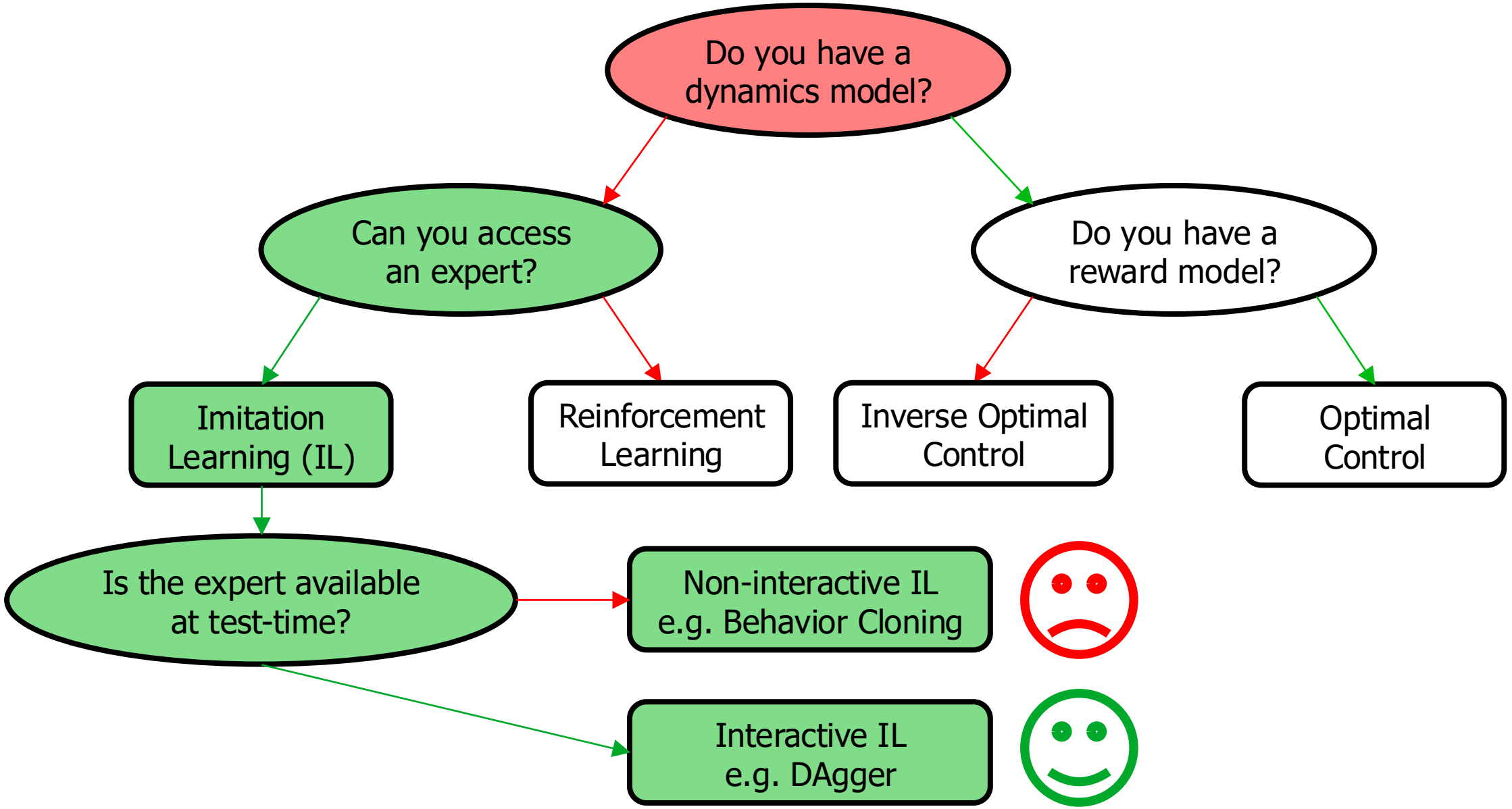
A Formalism for Sequential Decision Making

Imitation Learning: Behavior Cloning

Imitation Learning: Improvements – Compounding Error

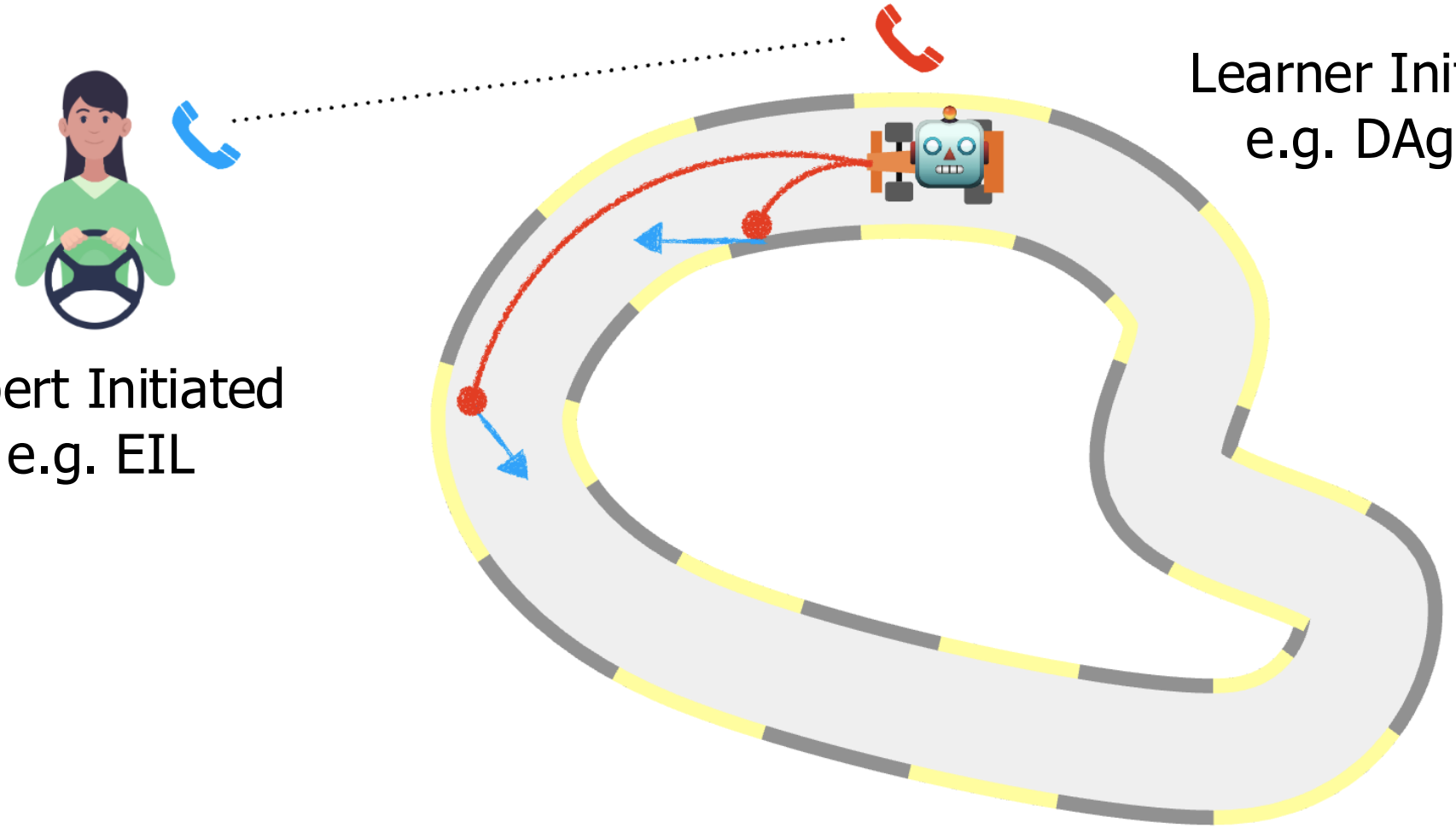
Imitation Learning: Improvements – Multimodality

Roadmap for Control and Learning



Interactive Imitation Learning

An interactive game between the learner and the expert



Learner Initiated
e.g. DAgger

Expert Initiated
e.g. EIL

DAgger – a meta-algorithm for learner-initiated IL

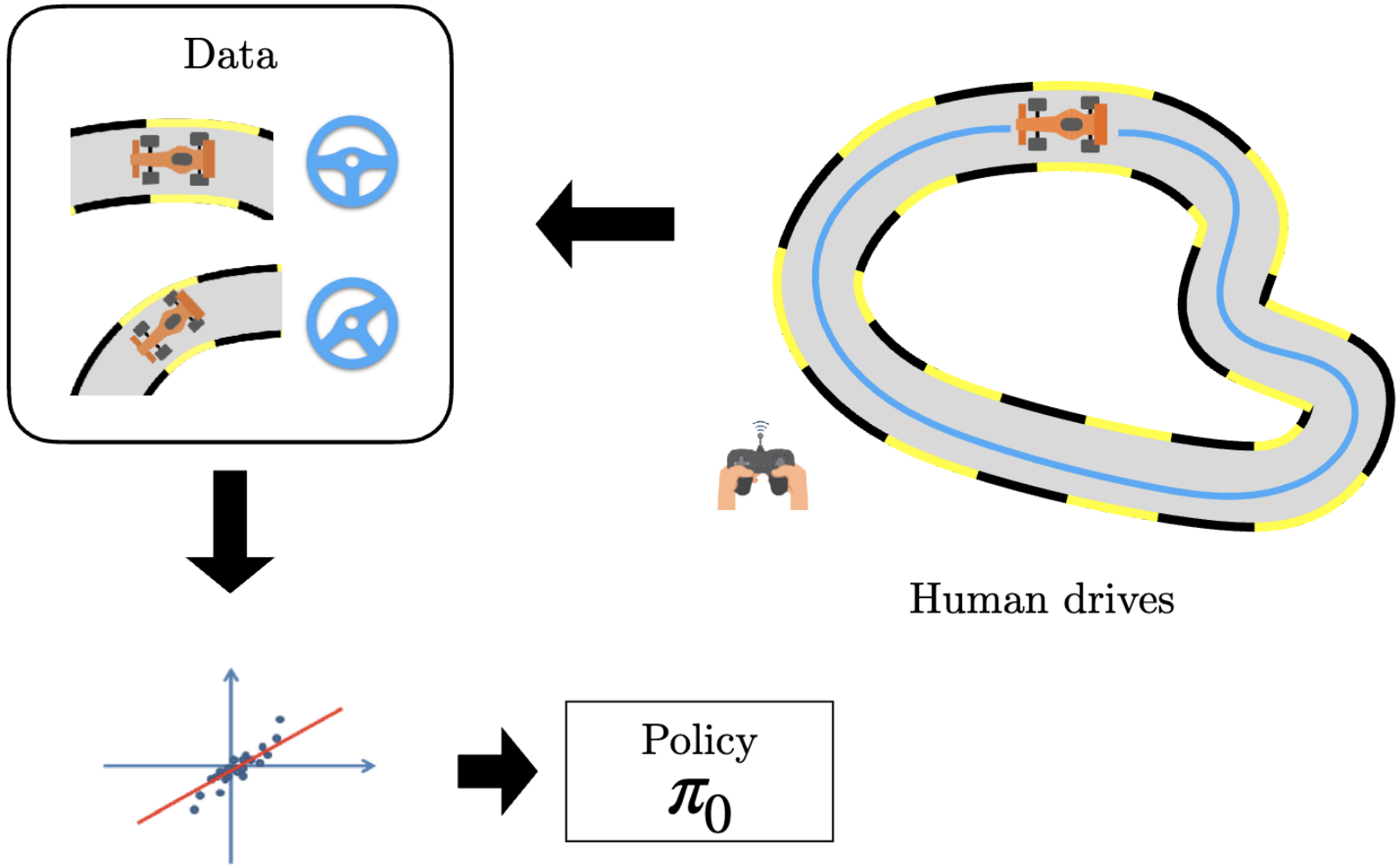
A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning

Stéphane Ross
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
stephaneross@cmu.edu

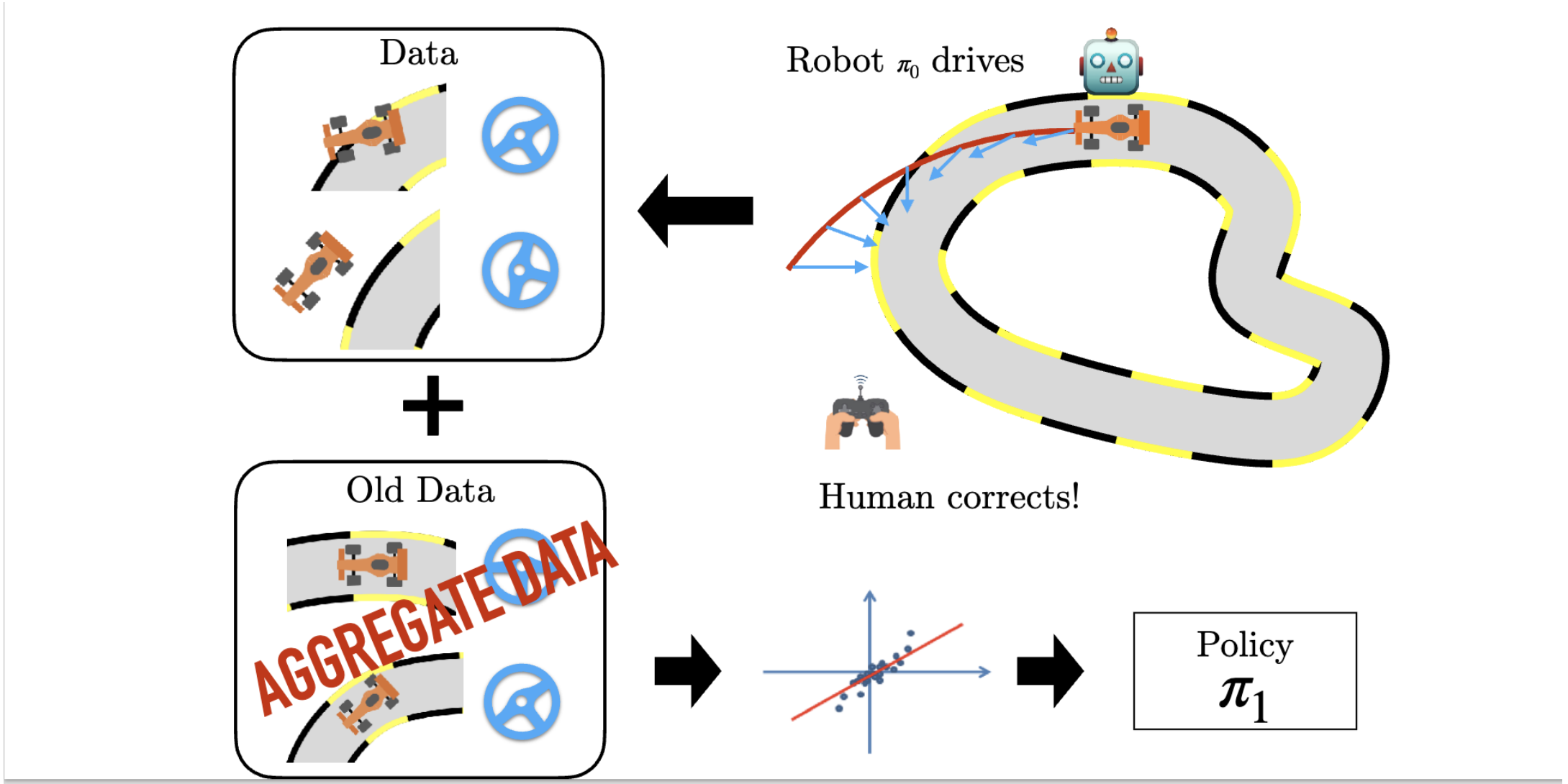
Geoffrey J. Gordon
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
ggordon@cs.cmu.edu

J. Andrew Bagnell
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
dbagnell@ri.cmu.edu

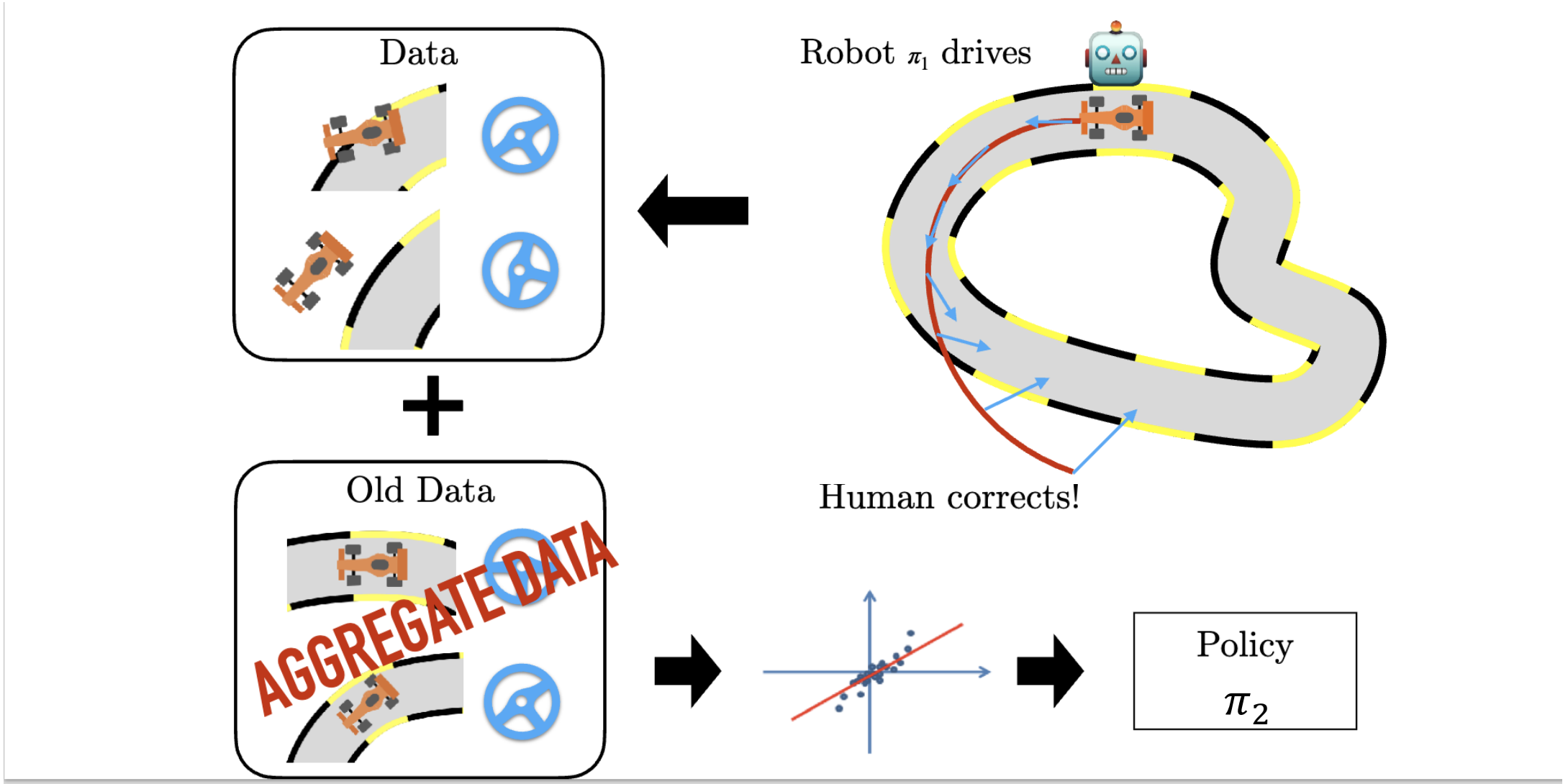
DAgger Iteration 0 – Identical to BC



DAgger Iteration 1

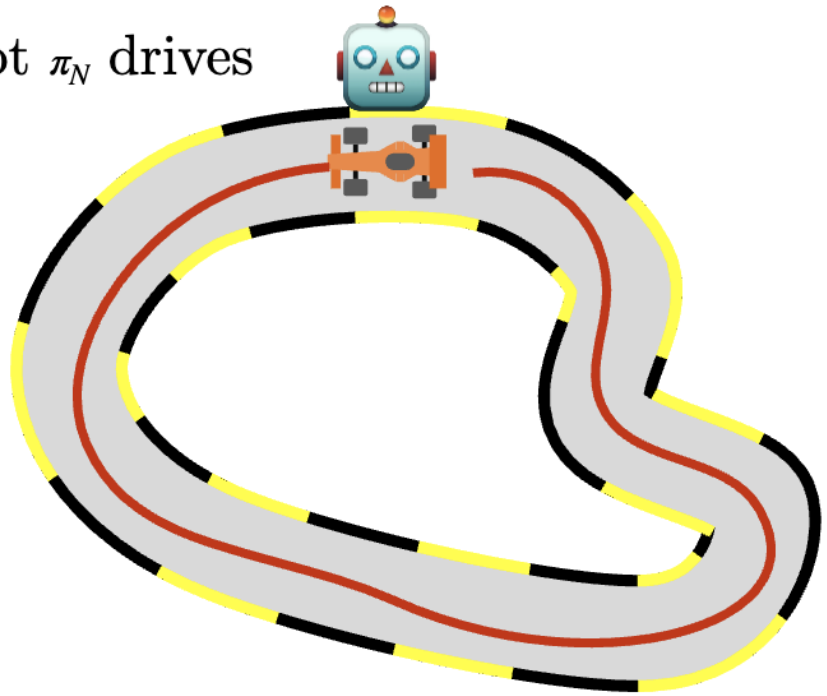


DAgger Iteration 2



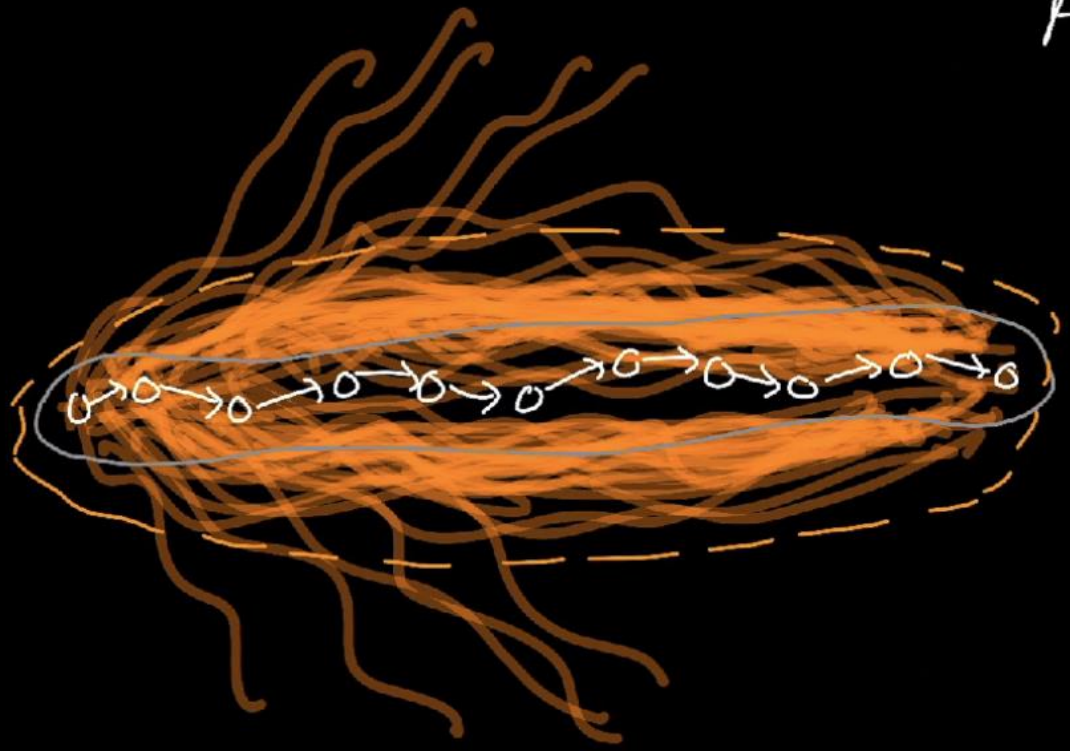
DAgger Iteration N

Robot π_N drives



After many iterations ....
we are able to drive like a human!

DAGGER (DATASET AGGREGATION)



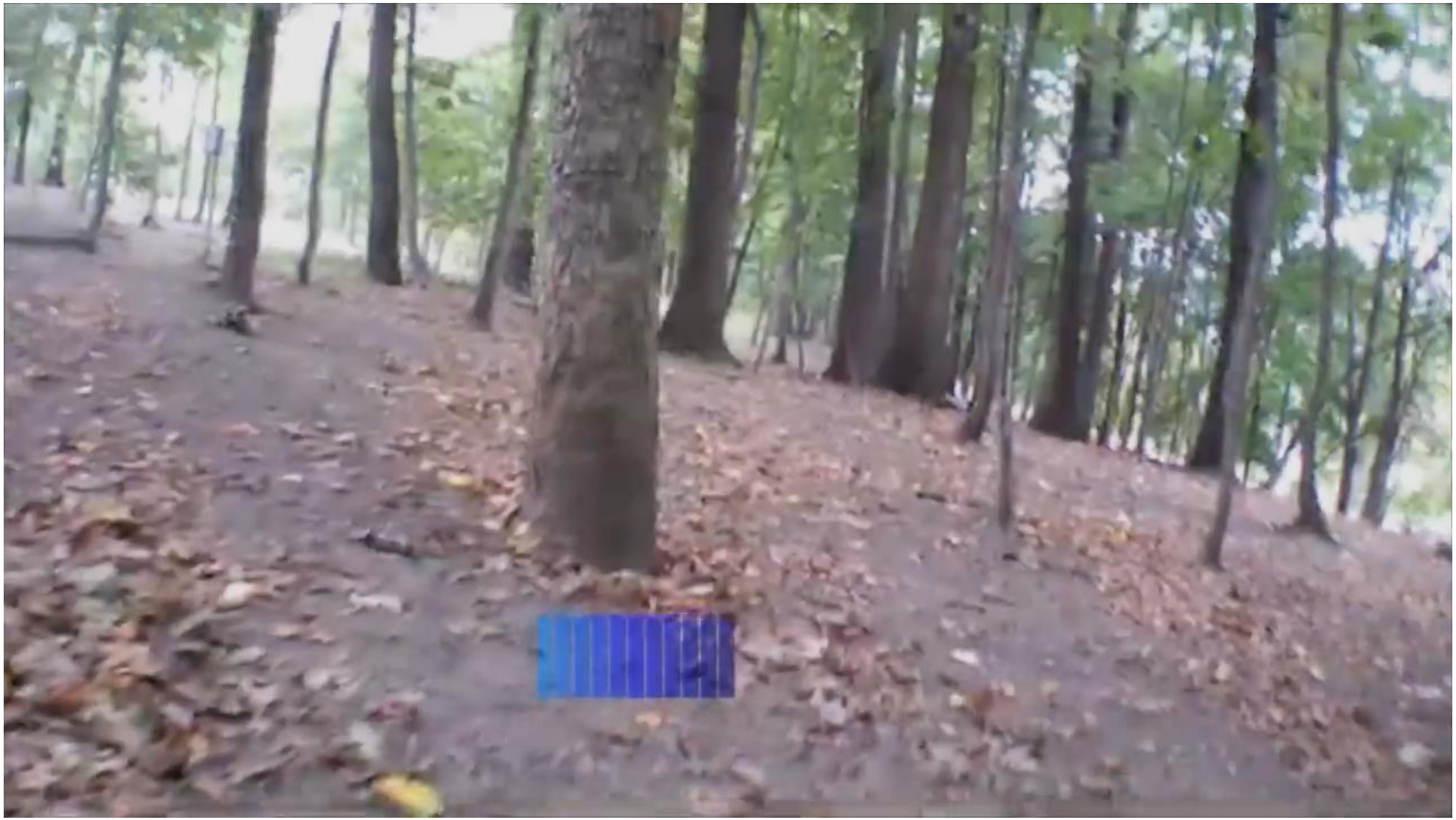
AGGREGATED TRAINING DISTRIBUTION

≈

TEST DISTRIBUTION

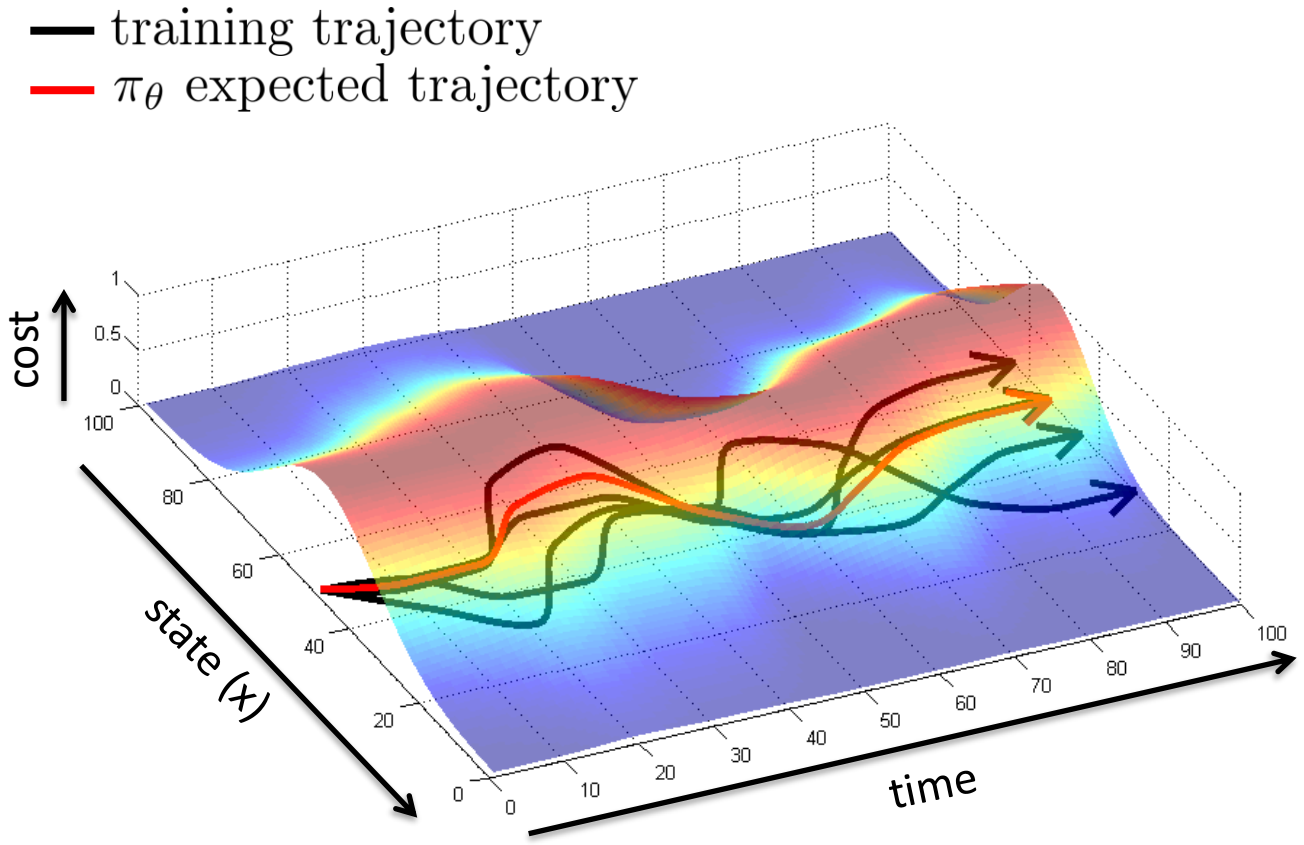
HUMAN DISTRIBUTION

DAgger Example



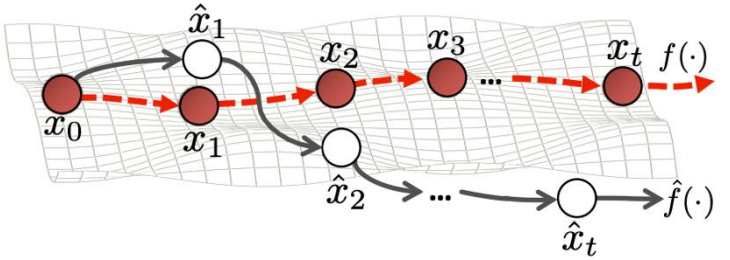
Ross et al. '13

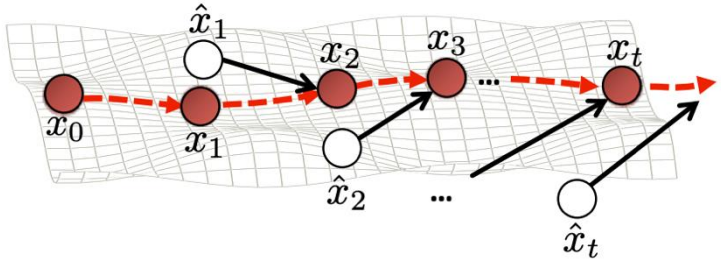
What is the general principle?



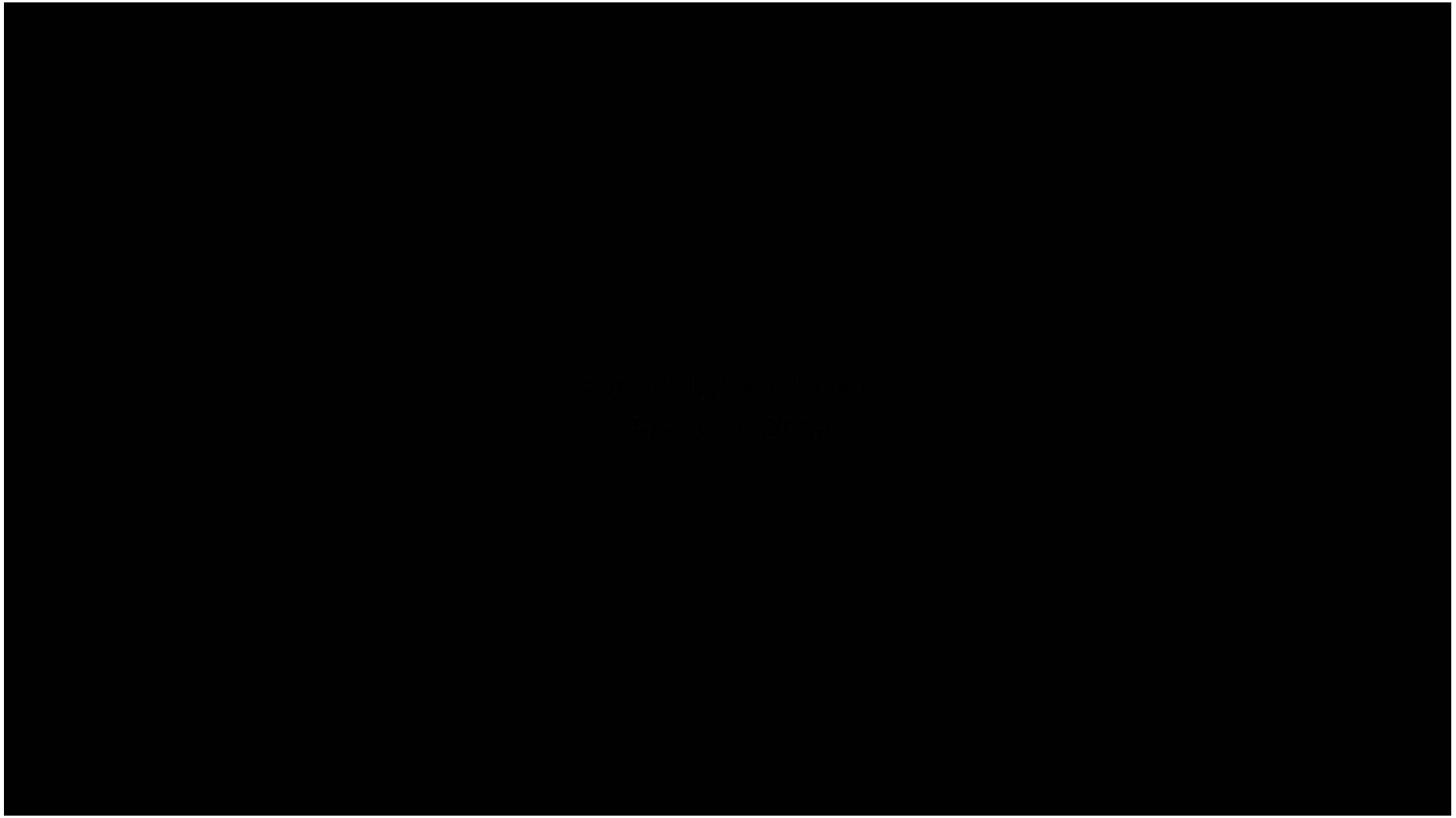
stability

Corrective labels that bring you back to the data



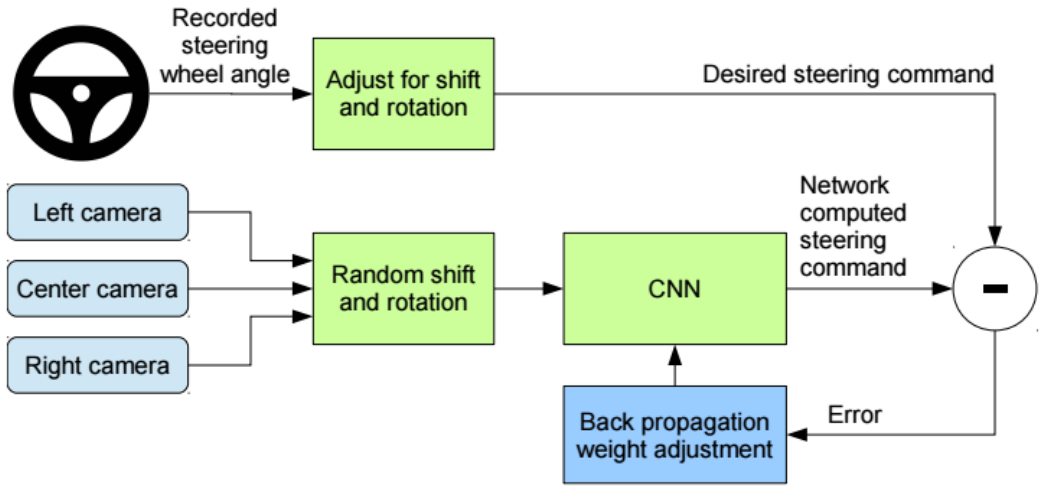


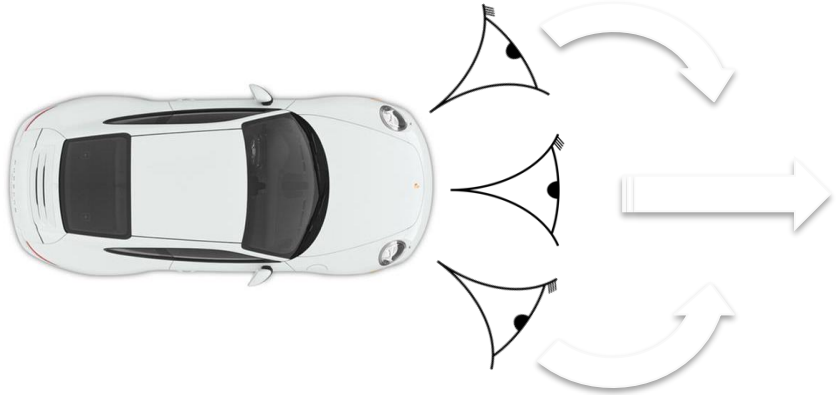
Sometimes BC lucks out!



Video: Bojarski et al. `16, NVIDIA

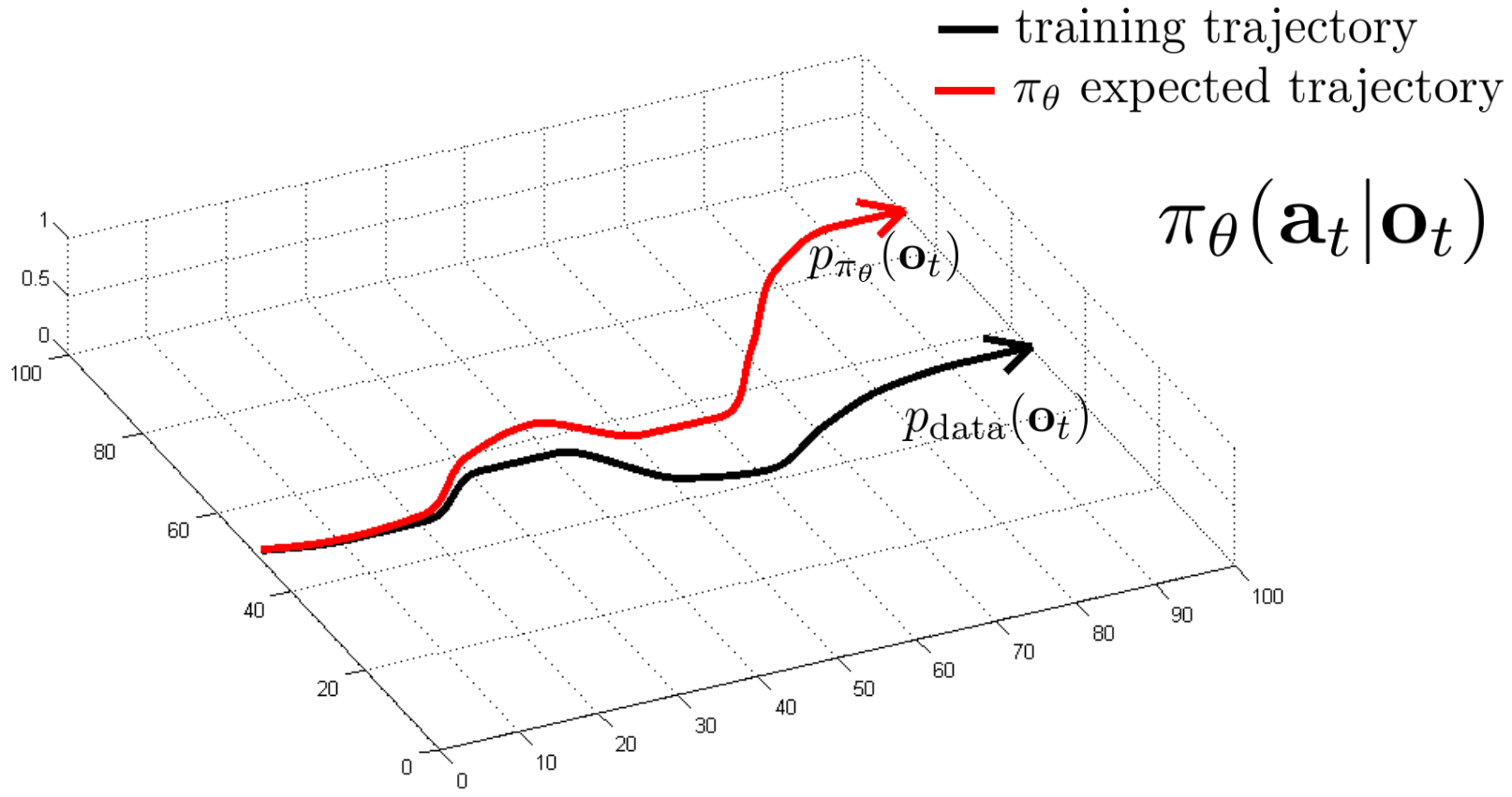
Why did that work?





Bojarski et al. `16, NVIDIA

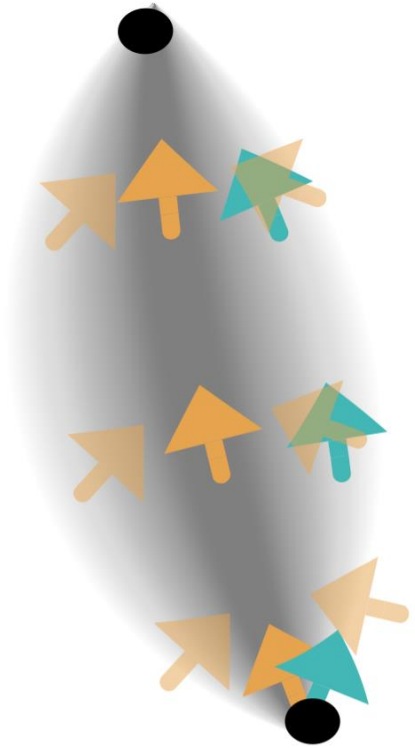
What might this mean mathematically?



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

Noising the Data Collection Process

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



Supervisor

Robot

Noise Injection

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

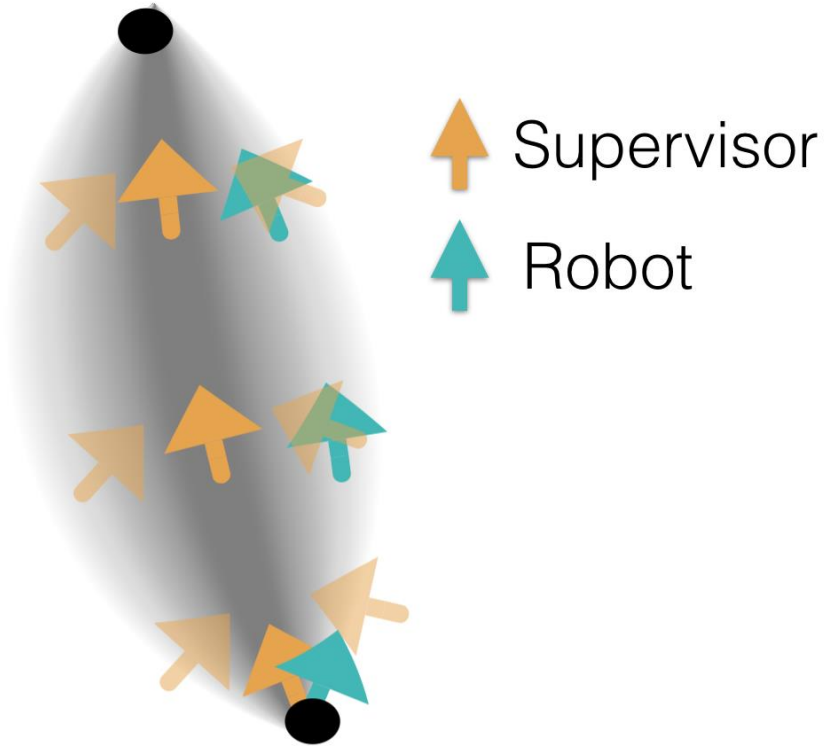
Maximize likelihood

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

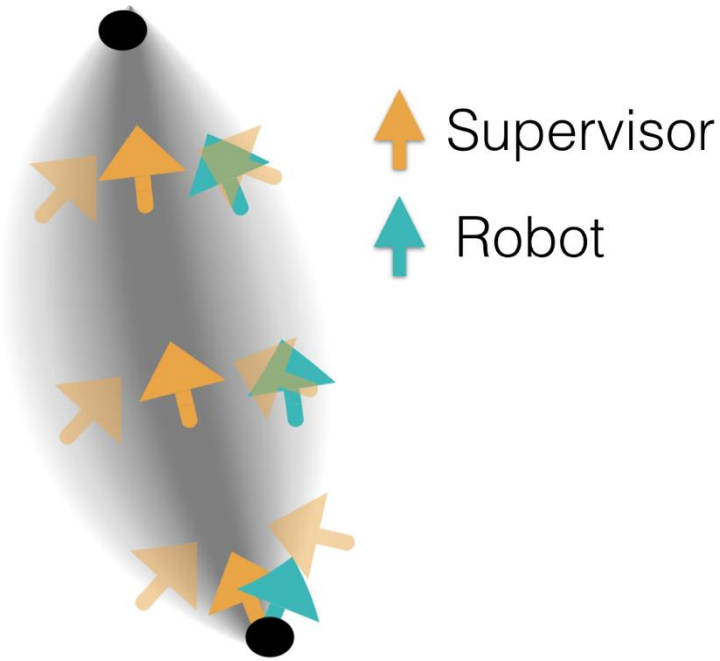


Noise Injection

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

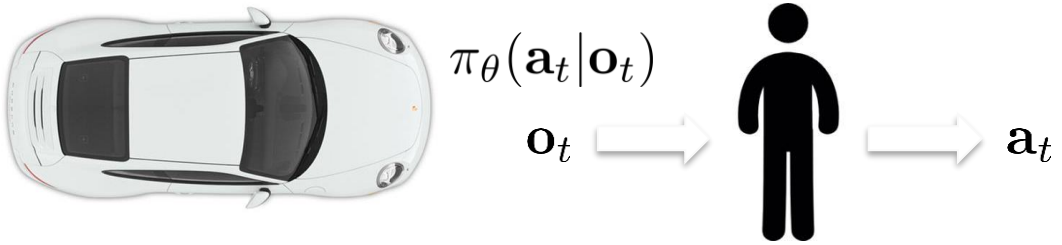
Assumes that the expert **can** actually perform behaviors under noise
→ Not always possible!

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

How might we fix this?

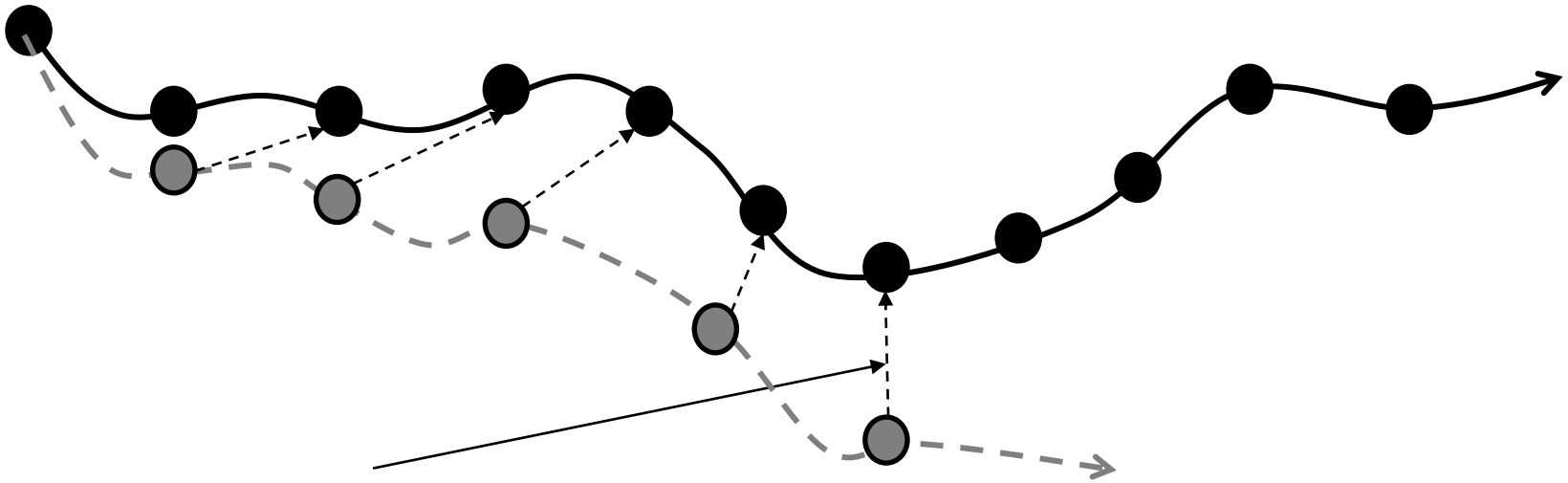
"Generate" corrective labels automatically

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

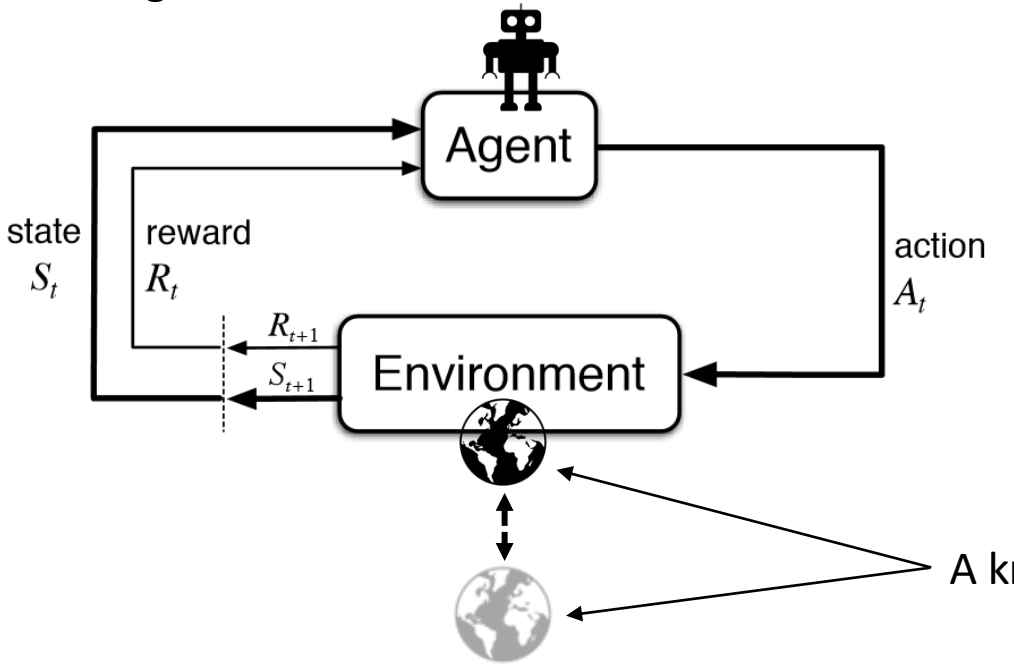


Ross et al. '11

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

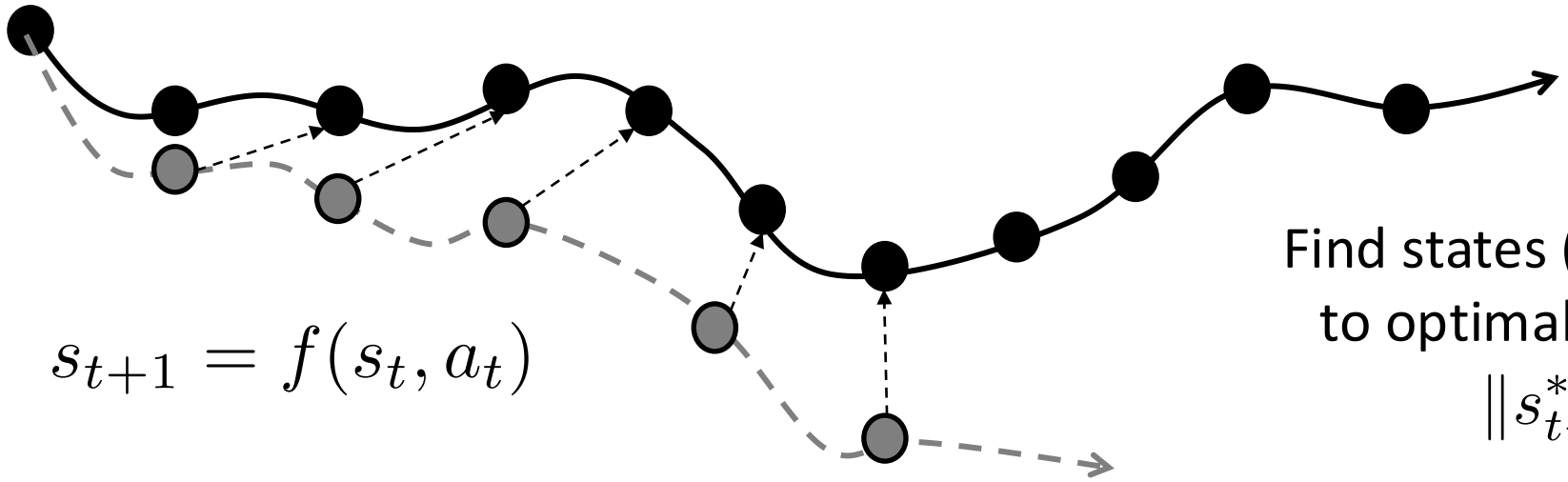
$$\|s^*_{t+1} - f(s_t, a_t)\| \leq \epsilon$$

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

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



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

$f(s_t, a_t)$

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

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

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

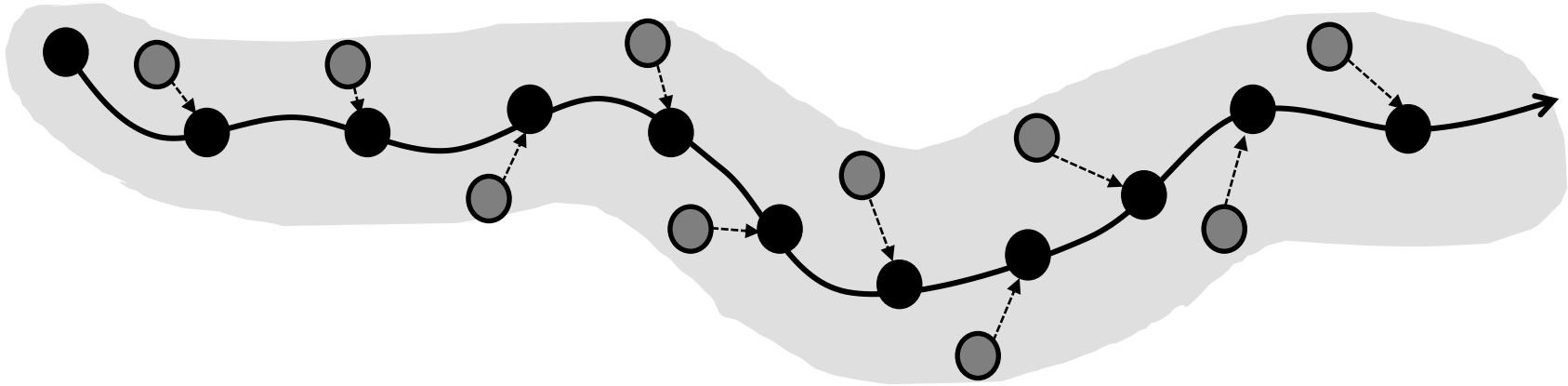
Intuition: find labels to bring OOD states back in distribution (where policy can be trusted)

Easy with known dynamics

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

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

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

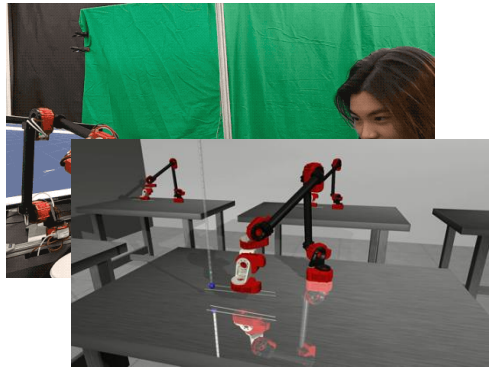
$$\|s^*_{t+1} - \hat{f}_\phi(s_t, a_t)\| \leq \epsilon$$

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



Learning Approximate Dynamics

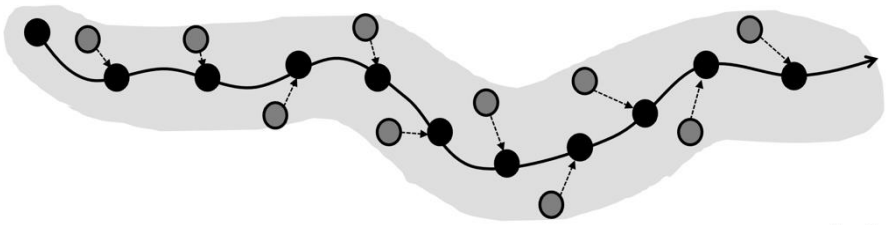
$\hat{f}_\phi$

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

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

D = {s, a, s'}

Generating Corrective Labels

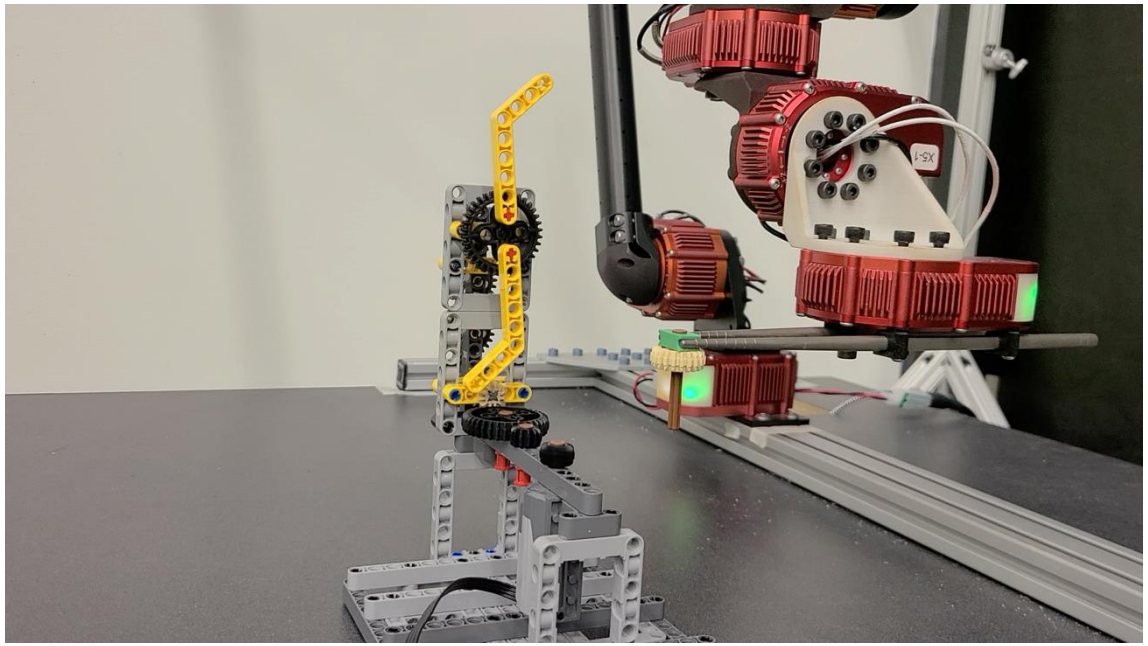


Supervised learning

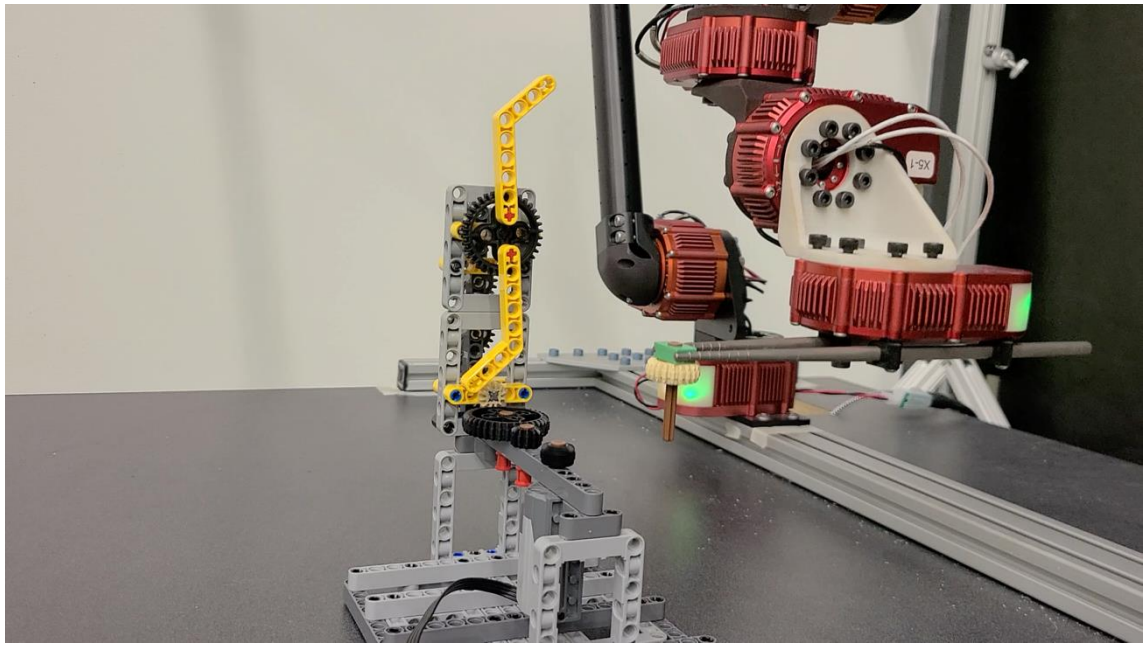
$$\max_\theta \mathbb{E}_{(s,a)\sim\mathcal{D}} \log \pi_\theta(a|s)$$

Overall Learning Pipeline with Corrective Labels

Standard behavior cloning



Corrective labels



So does this solve all the issues in imitation?

Lecture Outline

A Formalism for Sequential Decision Making

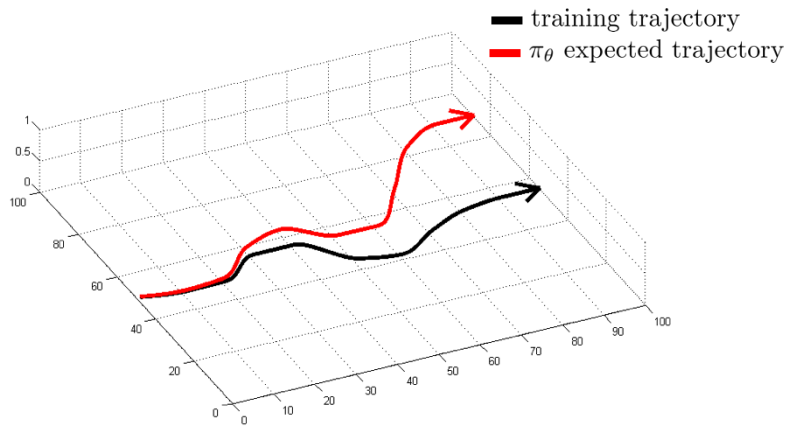
Imitation Learning: Behavior Cloning

Imitation Learning: Improvements – Compounding Error

Imitation Learning: Improvements – Multimodality

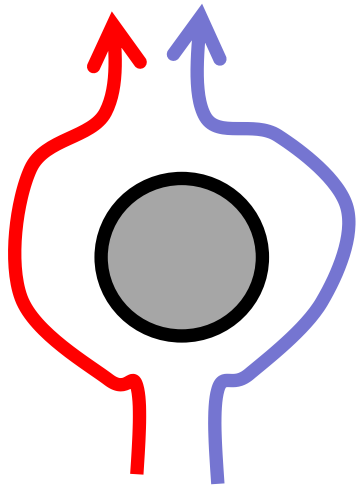
Can we make it work without more data?

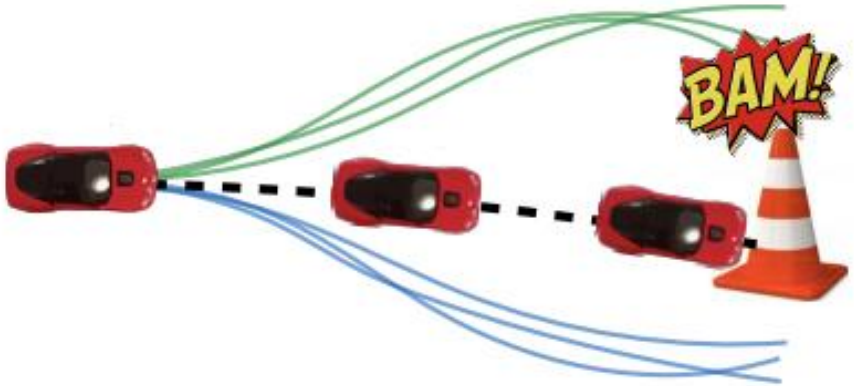
- DAgger addresses the problem of distributional “drift”
- What if our model is so good that it doesn’t drift?
- Need to mimic expert behavior very accurately
- But don’t overfit!

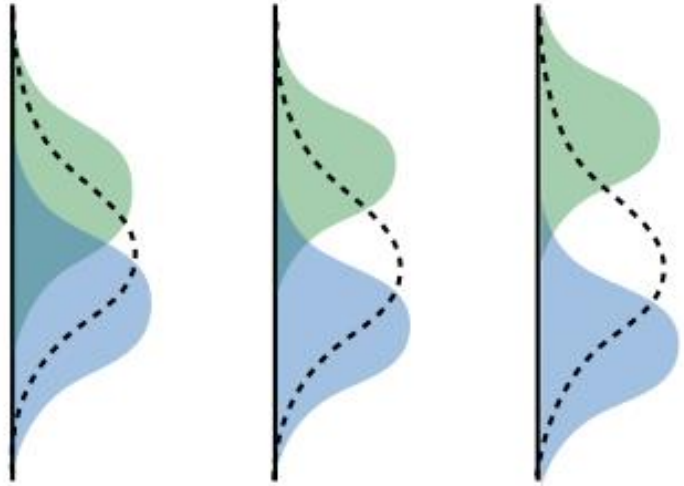


Why might we fail to fit the expert?

Multimodal behavior.. amongst other reasons



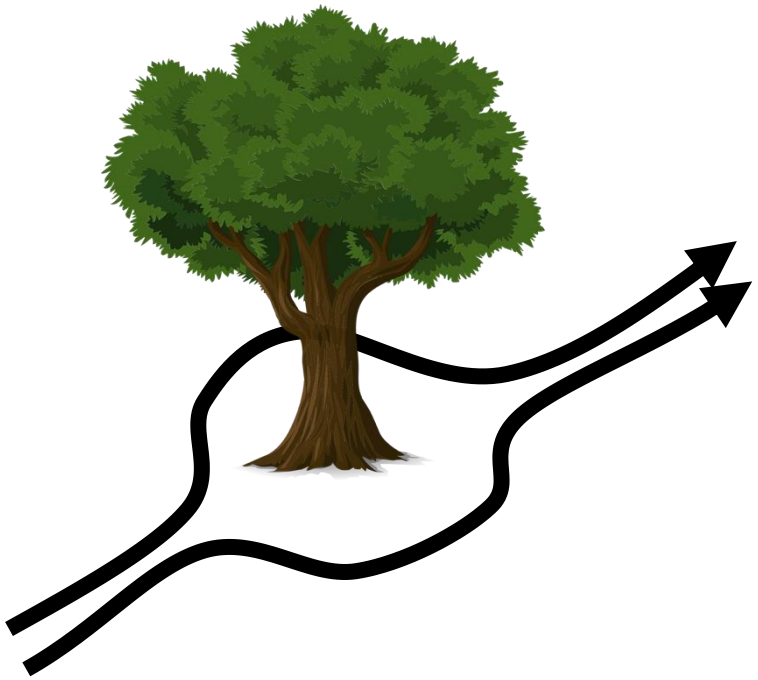


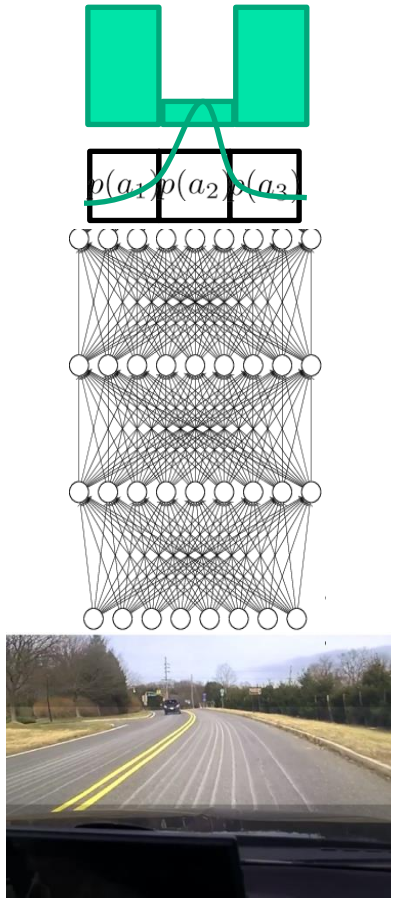


Not a matter of network size! It’s about distributional expressivity

Why might we fail to fit the expert?

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



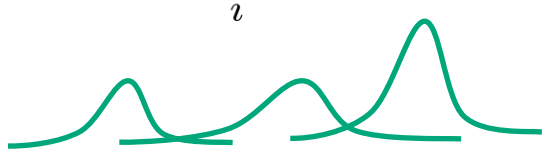


1. Output mixture of Gaussians
2. Latent variable models
3. Autoregressive discretization
4. Diffusion models
5. …

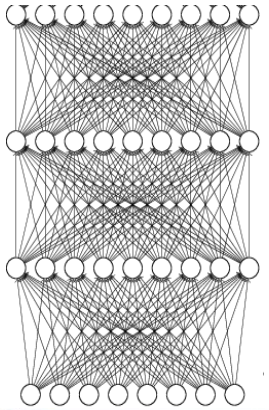
Why might we fail to fit the expert?

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

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

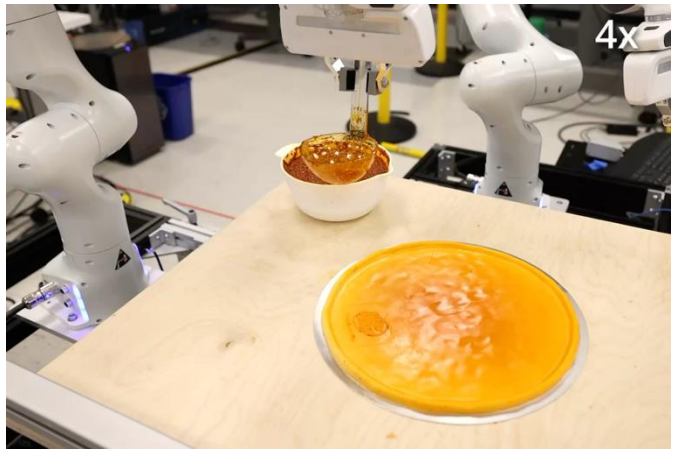


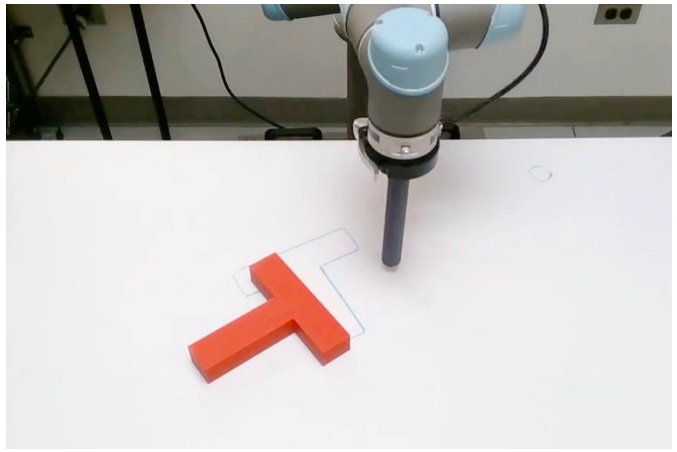
$$w_1, \mu_1, \Sigma_1, \ldots, w_N, \mu_N, \sigma_N$$

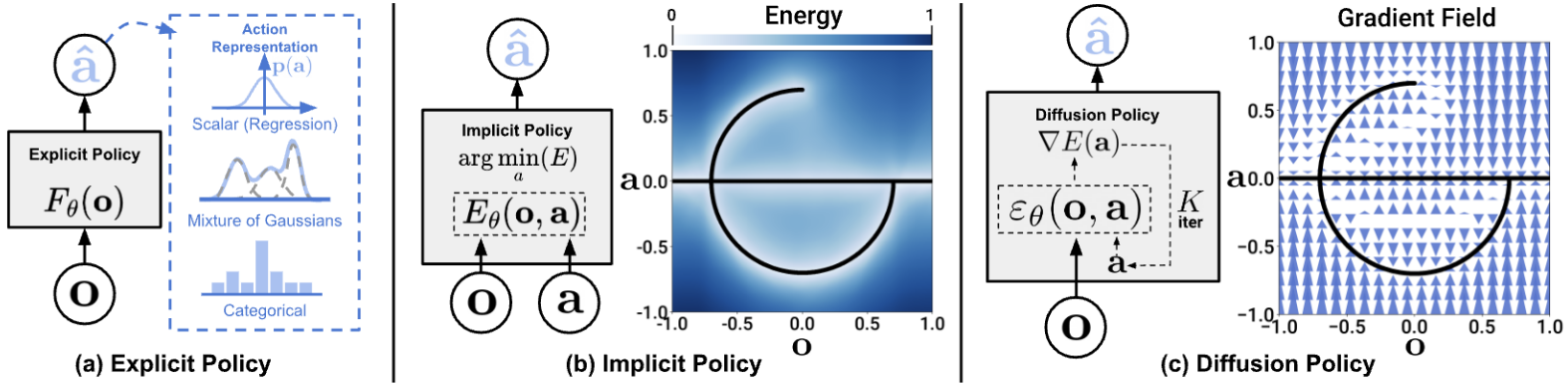


Why might we fail to fit the expert?

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







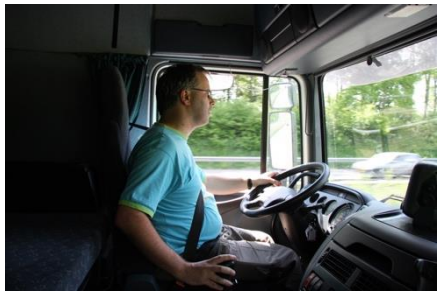
Some cool imitation videos

1x and tesla humanoid robots

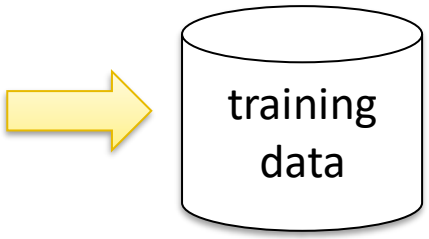
ALOHA and CherryBot Fine Manipulation

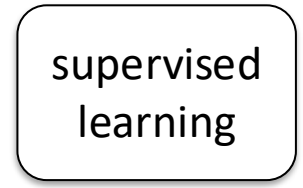
TRI Diffusion Policies

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



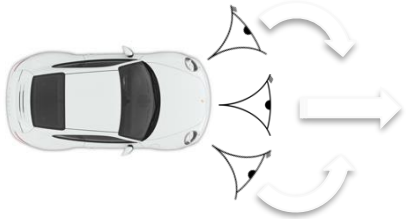
$\mathbf{o}_t$
$\mathbf{a}_t$



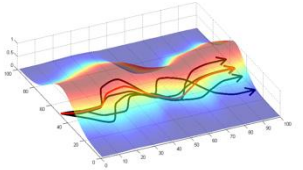


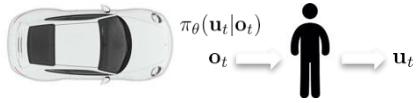
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

- 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

A Formalism for Sequential Decision Making

Imitation Learning: Behavior Cloning

Imitation Learning: Improvements – Compounding Error

Imitation Learning: Improvements – Multimodality

CSE 478 Robot Autonomy
Imitation Learning

Siddhartha Srinivasa (siddh@)
Abhishek Gupta (abhgupta@)

TAs:
Rohan Baijal (rbaijal@)
Sidhartha Talia (sidtalia@)
Christopher Tan (tan7271@)
Helen Wang (yiruwang@)