DAgger: Interactive Experts and No-Regret Learning

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Easy 😄

Hard 😱

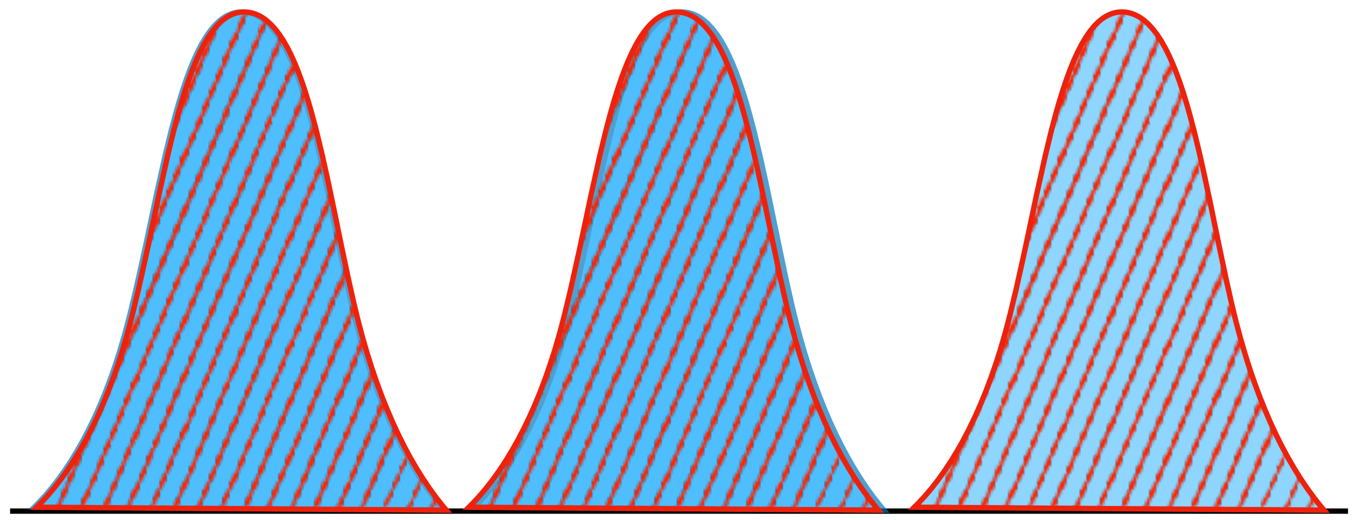
Setting

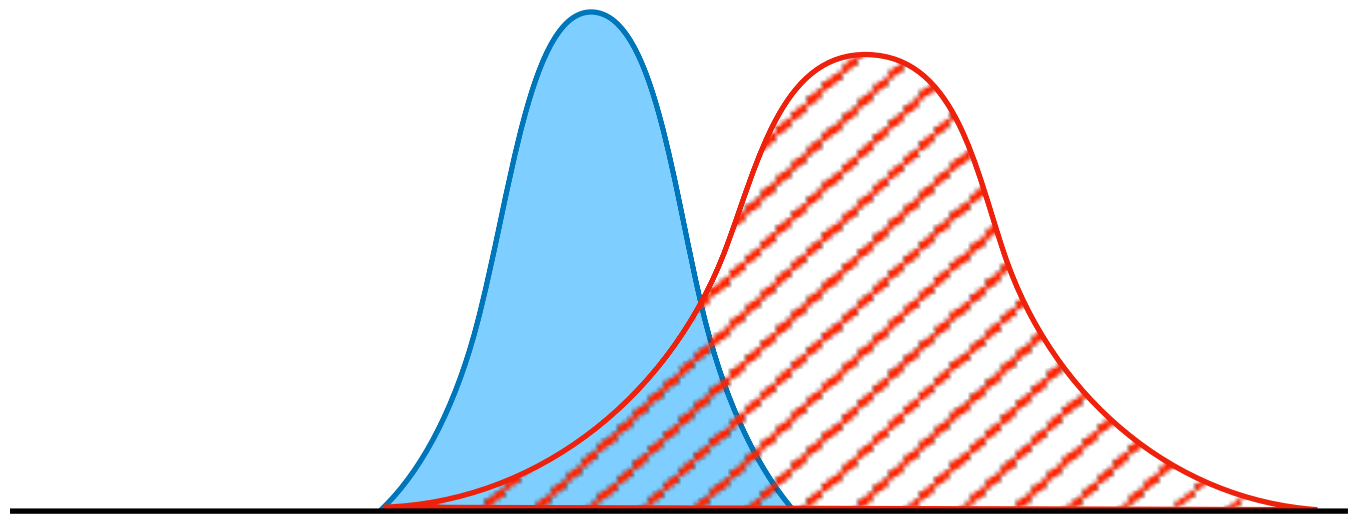
Expert is realizable
$\pi^E \in \Pi$

As $N \to \infty$, drive down
$\epsilon = 0$ (or Bayes error)

Non-realizable expert +
limited expert support

Even as $N \to \infty$,
behavior cloning $O(\epsilon T^2)$





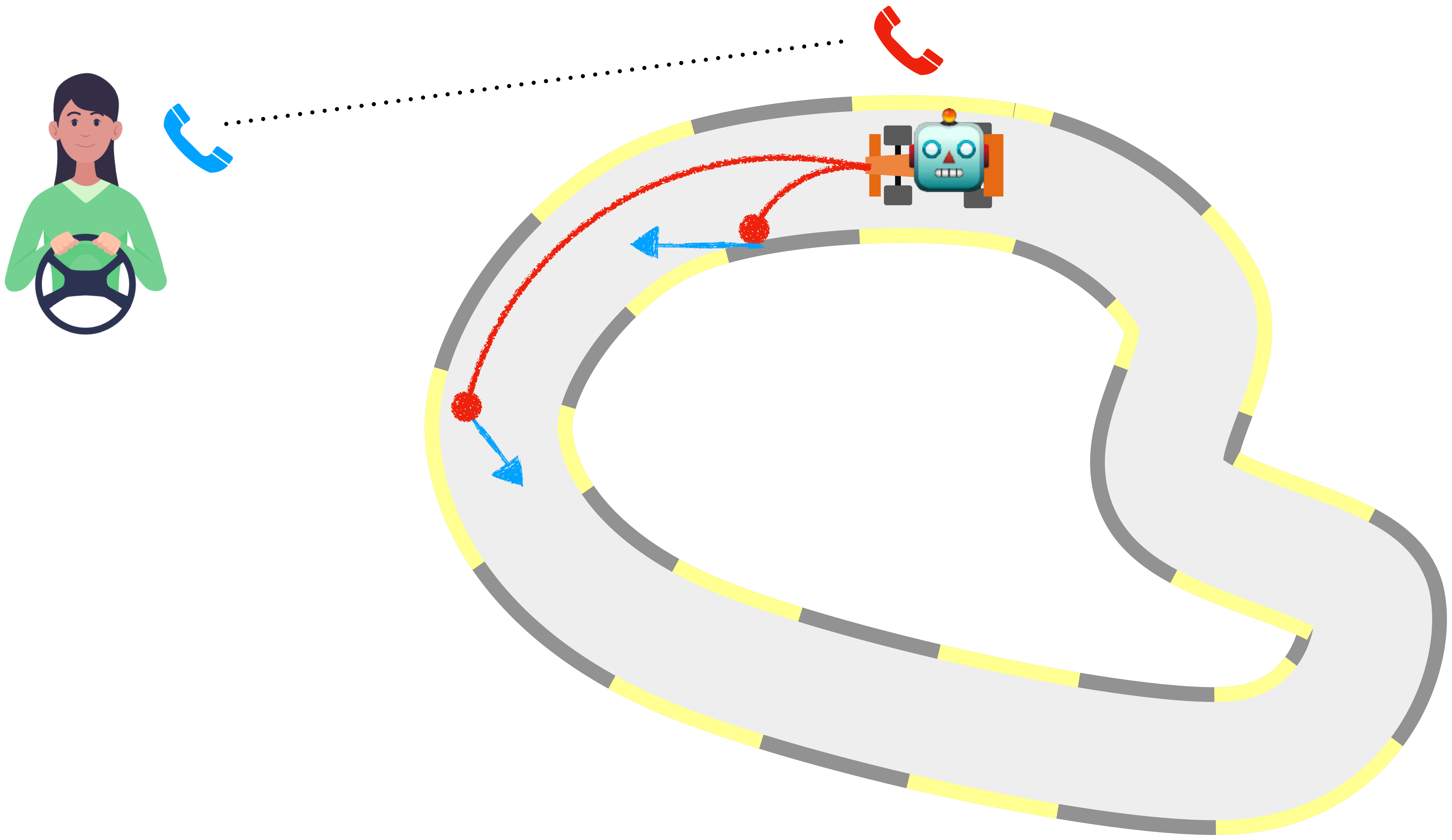
Solution

Nothing special.
Collect lots of data and
do Behavior Cloning

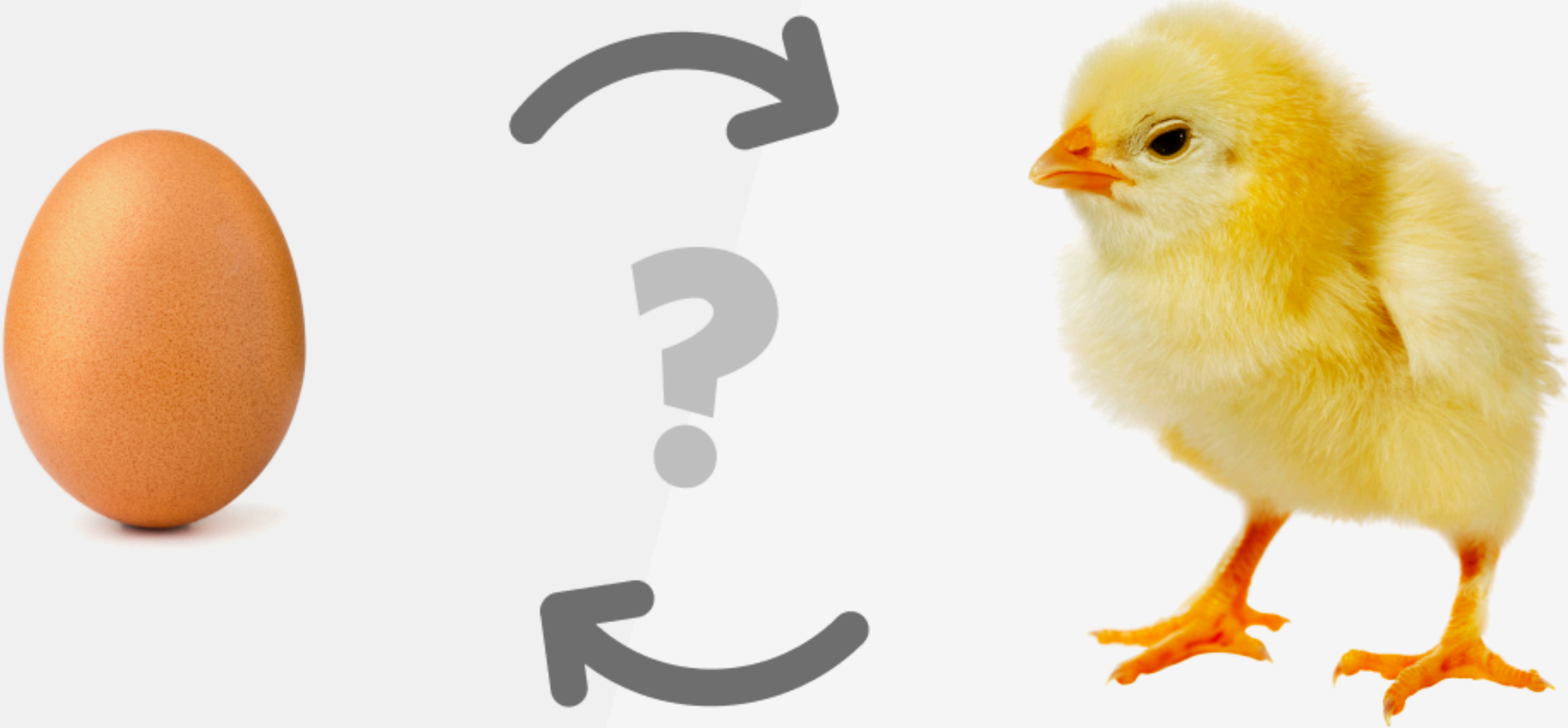
?

Why can't we just collect data (s, a^*) on the distribution of states the learner visits?

Introducing an *interactive* expert!



To know the distribution, you need a learner
To train a learner, you need a distribution



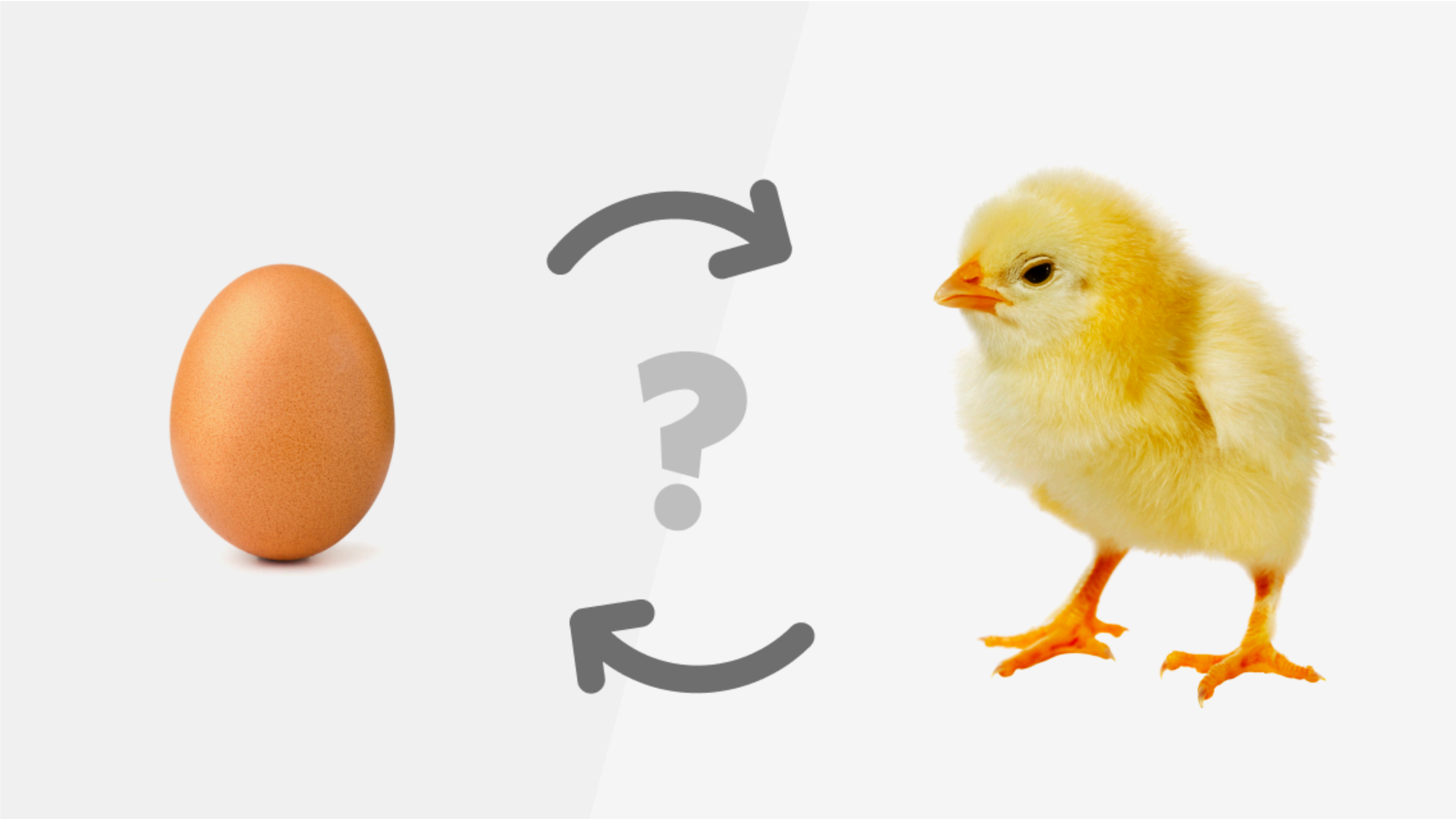
Activity!

Think-Pair-Share!

Think (30 sec): How can we solve the chicken and egg problem, i.e. train the learner on a distribution of states it visits?

Pair: Find a partner

Share (45 sec): Partners exchange ideas



An *embarrassingly* simple algorithm: FORWARD

Idea: Train a different learner policy at every timestep by interactively querying expert

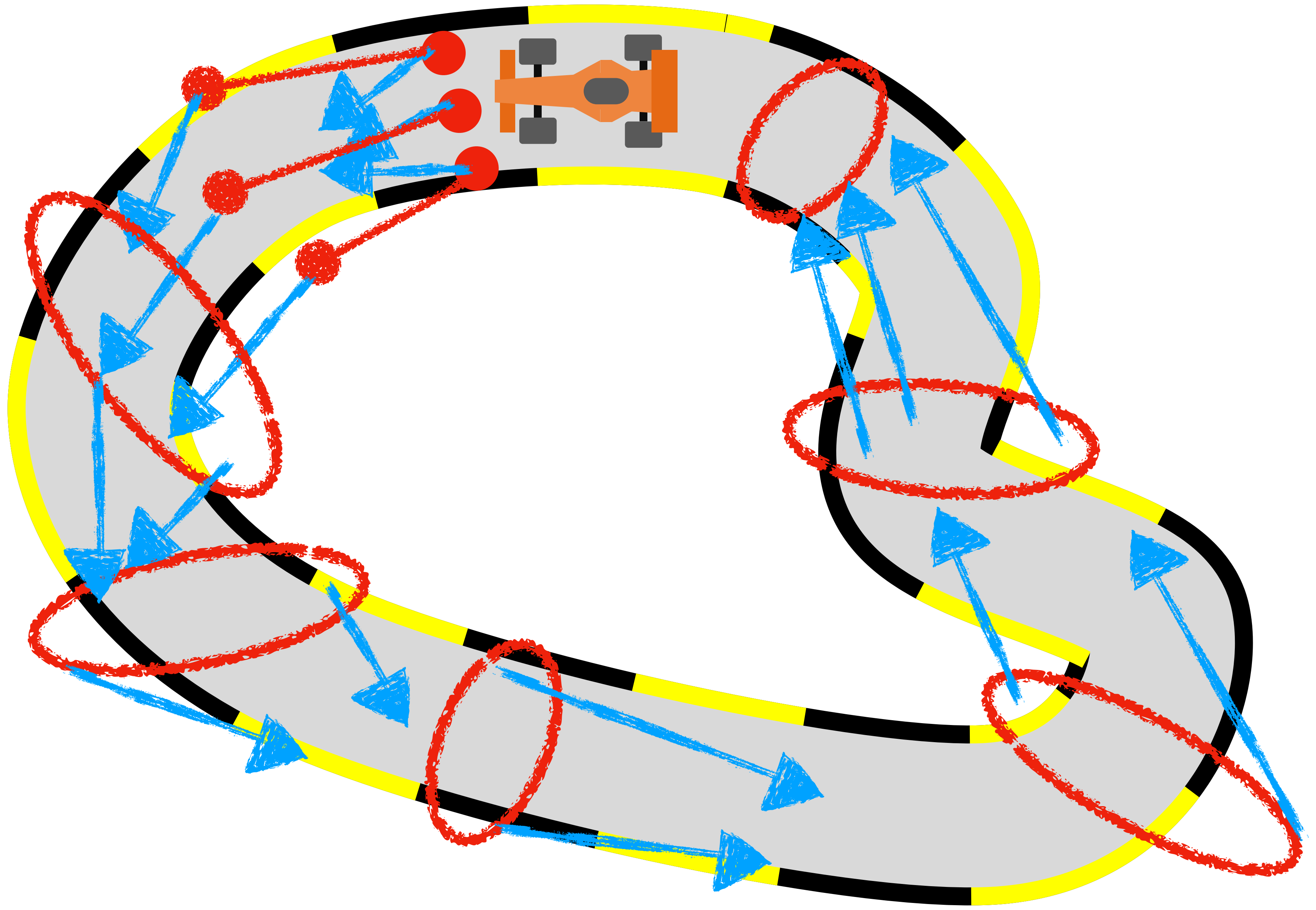
Get start state samples $s_0 \sim d^0(\,.\,)$

for t = 0 … T-1

Query interactive expert to get
$a_t^* = \pi^*(s_t)$

Train a learner policy at time t
$\pi^t = \text{Train}(s_t, a_t^*)$

Execute learner policy π^t to get
next state samples $s_{t+1} \sim d_\pi^{t+1}(\,.\,)$



But what if we want ONE policy?

DAGGER

Episode IV

A NEW HOPE

DAGGER: A meta-algorithm for imitation learning

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

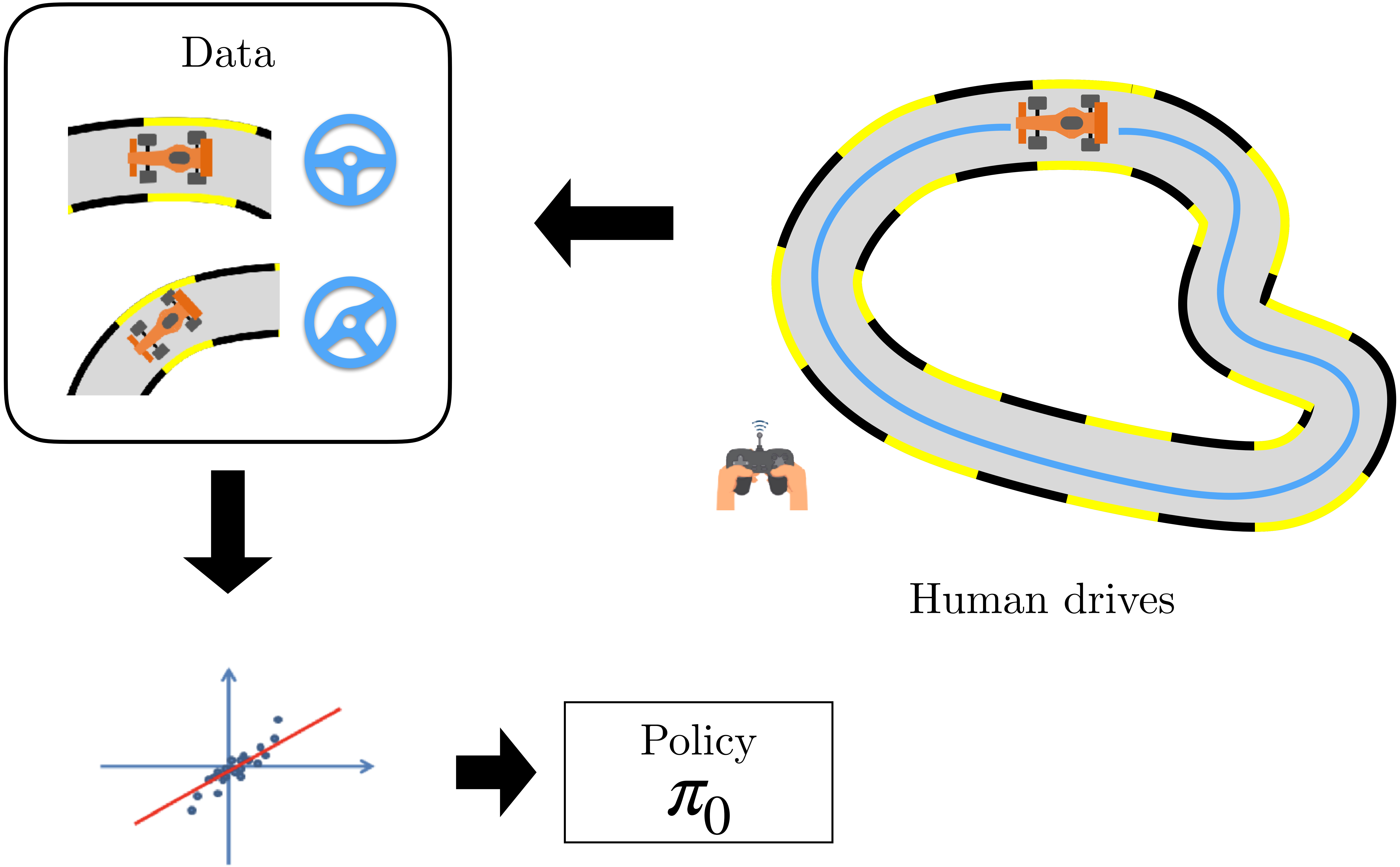
Stéphane Ross
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
stephaneross@cmu.edu

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

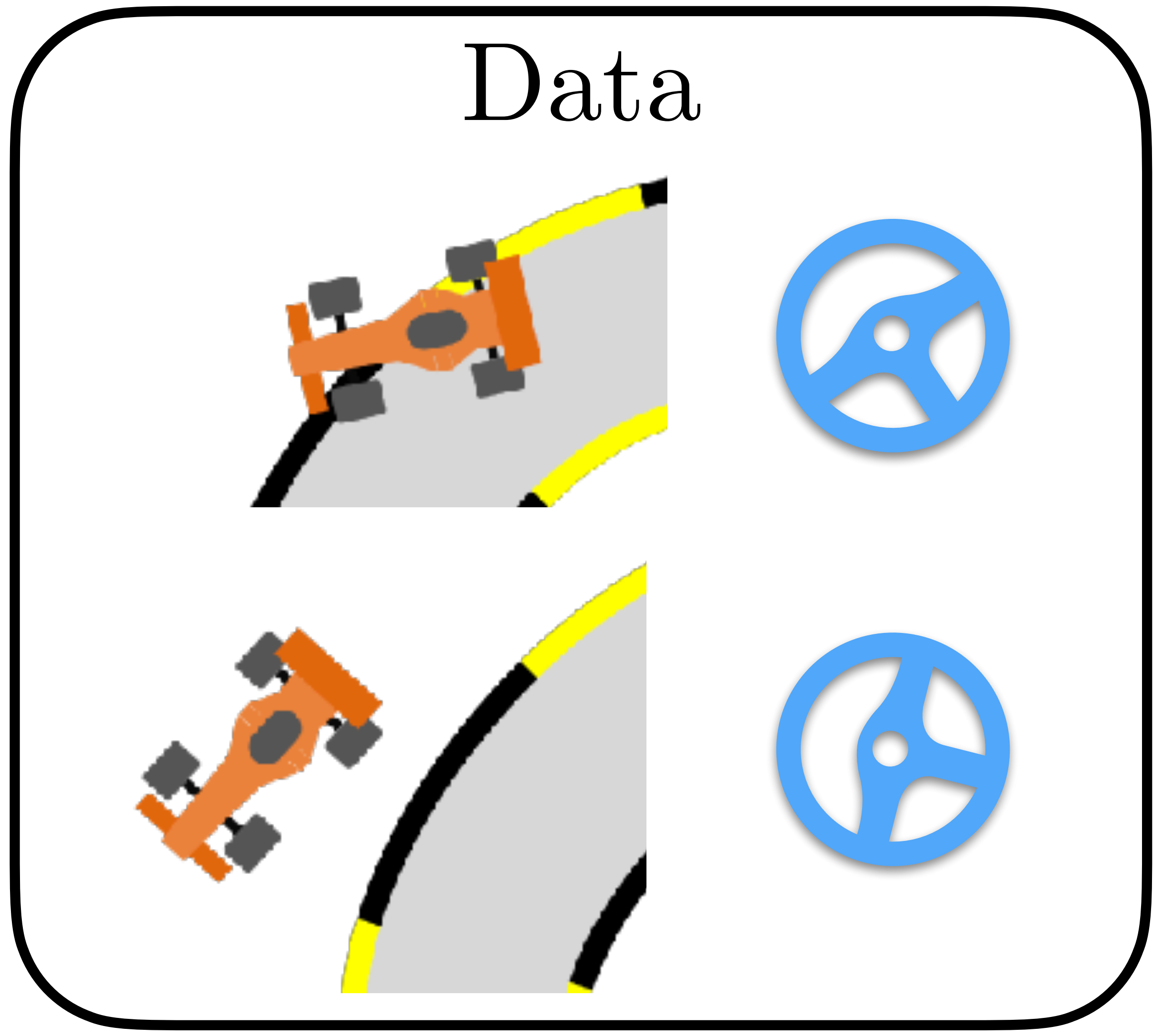
J. Andrew Bagnell
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
dbagnell@ri.cmu.edu

[Ross et al'11]

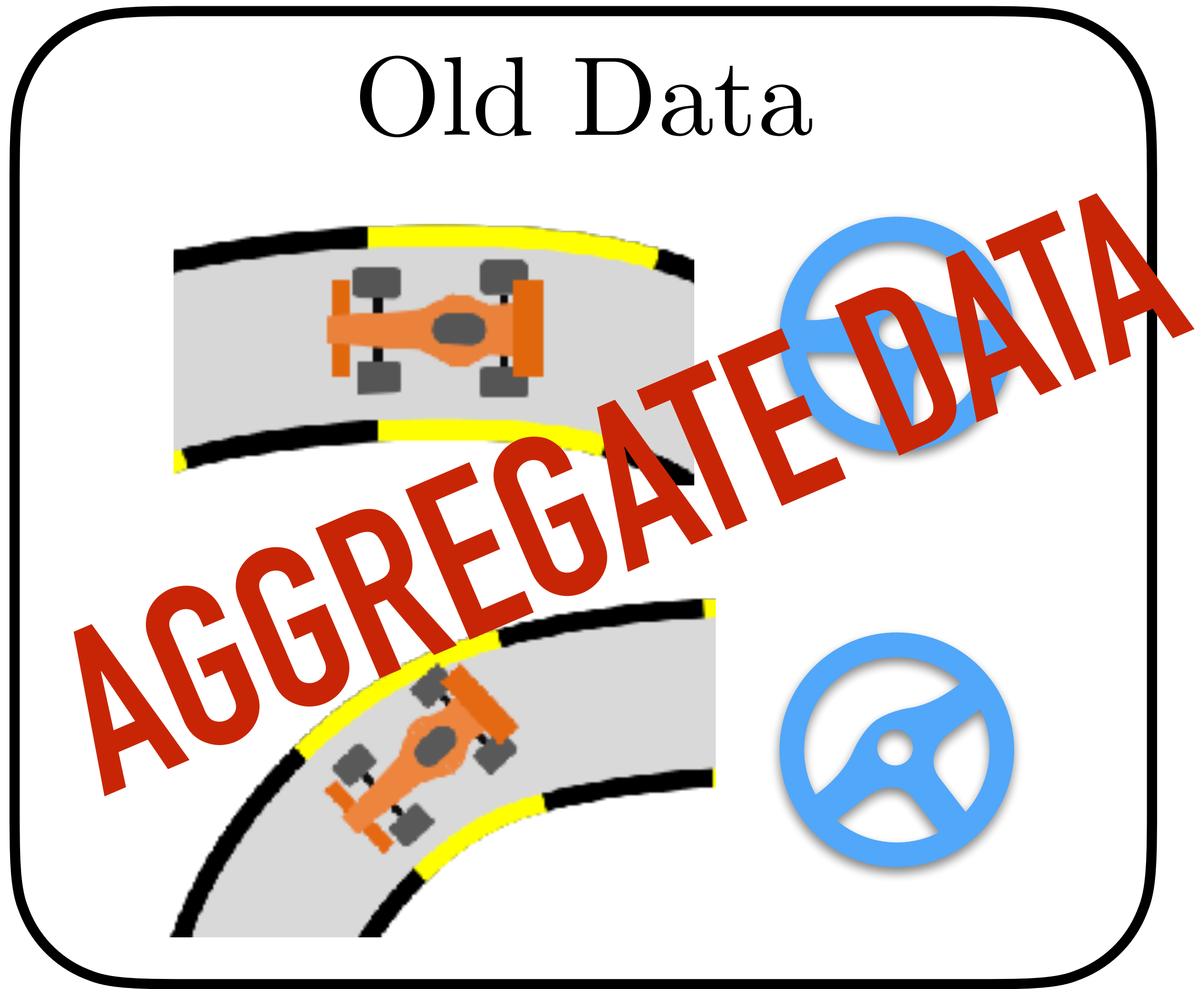
DAgger: Iteration 0

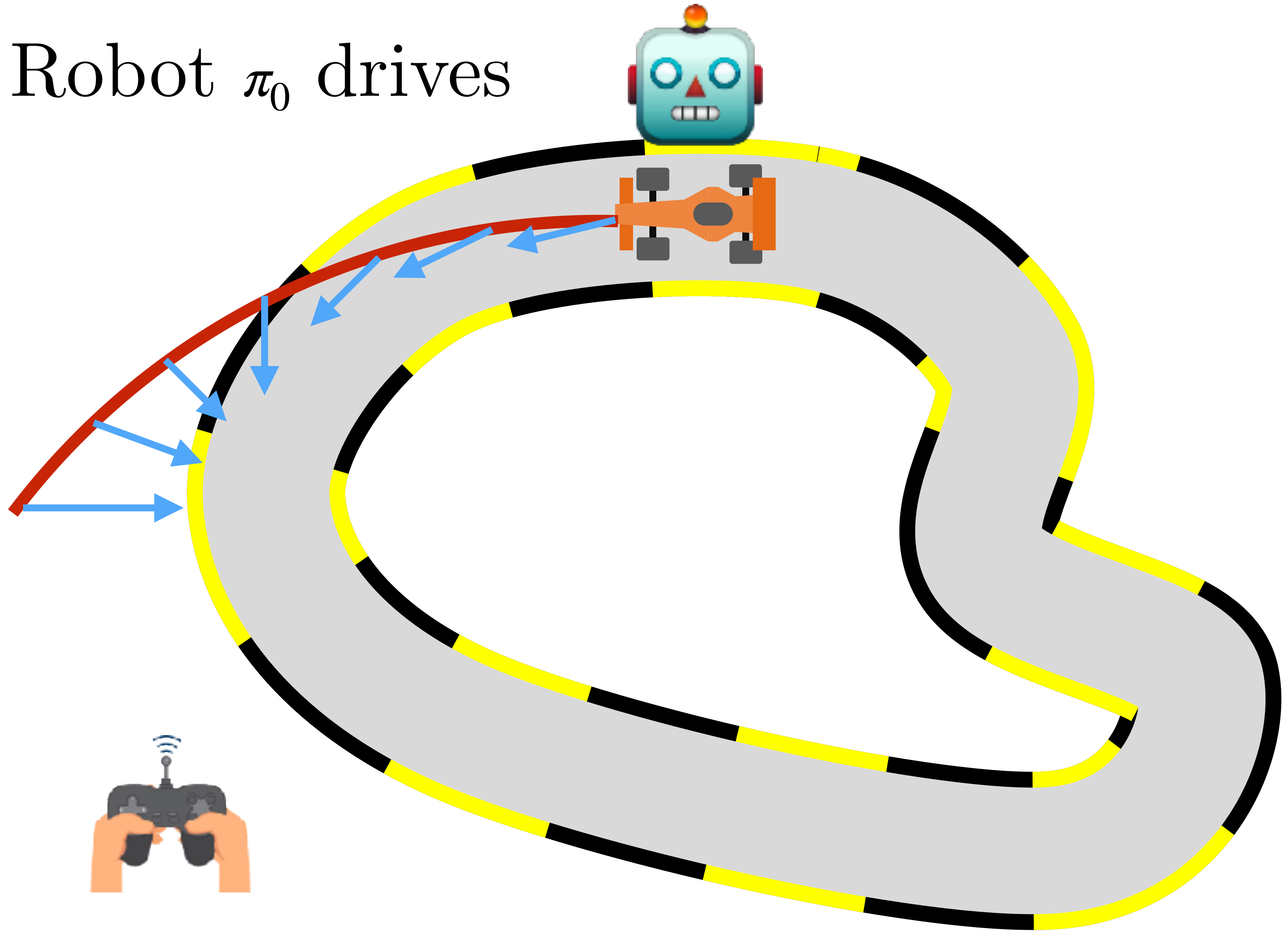


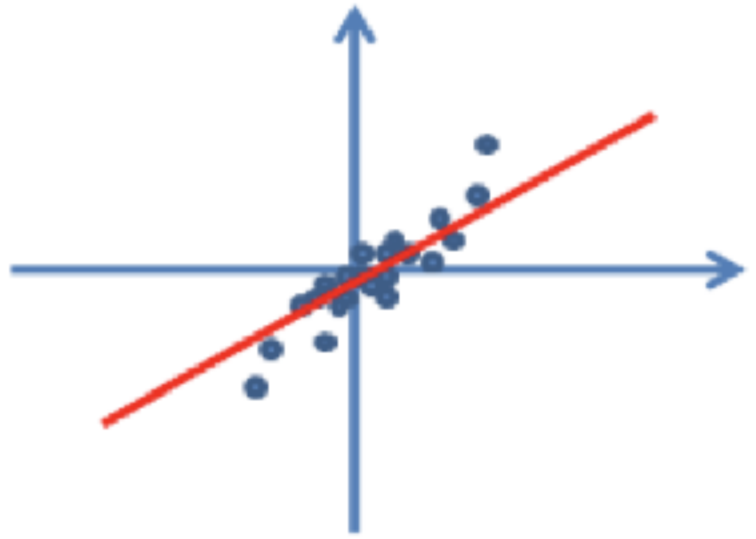
DAgger: Iteration 1



+





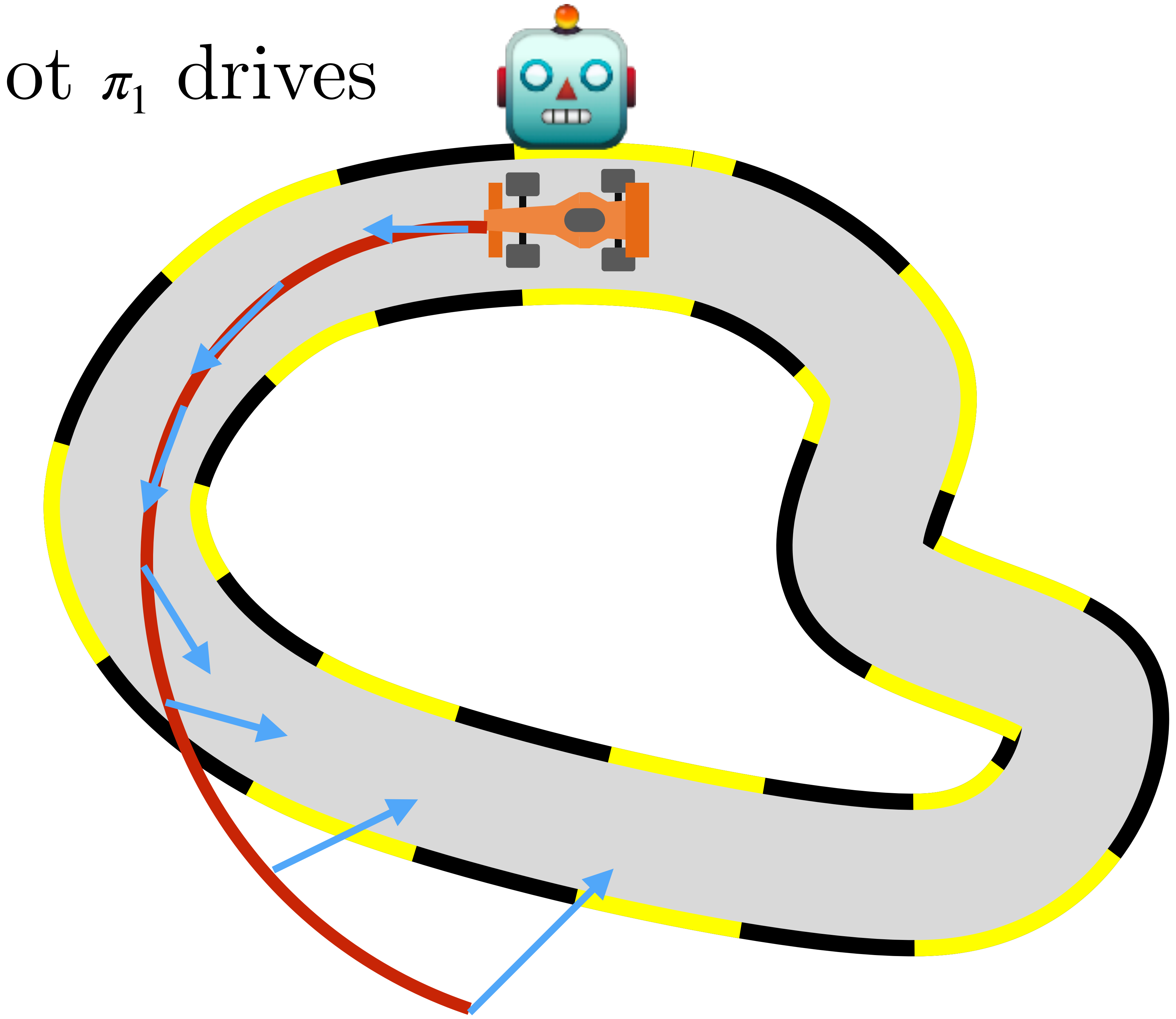


Policy
π_1

DAgger: Iteration 2

[Ross et al'11]

Robot π_1 drives

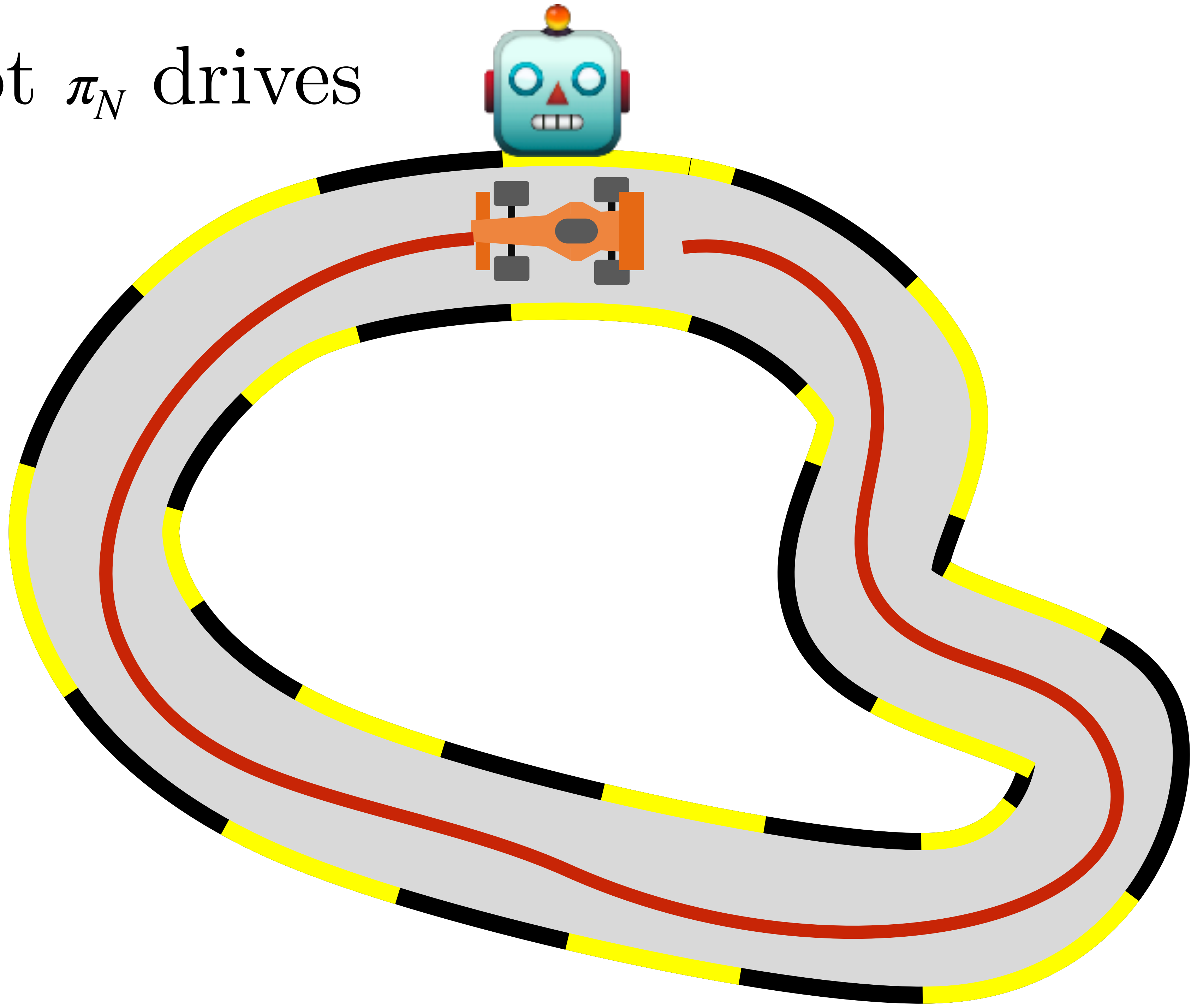


AGGREGATE DATA

[Ross et al'11]

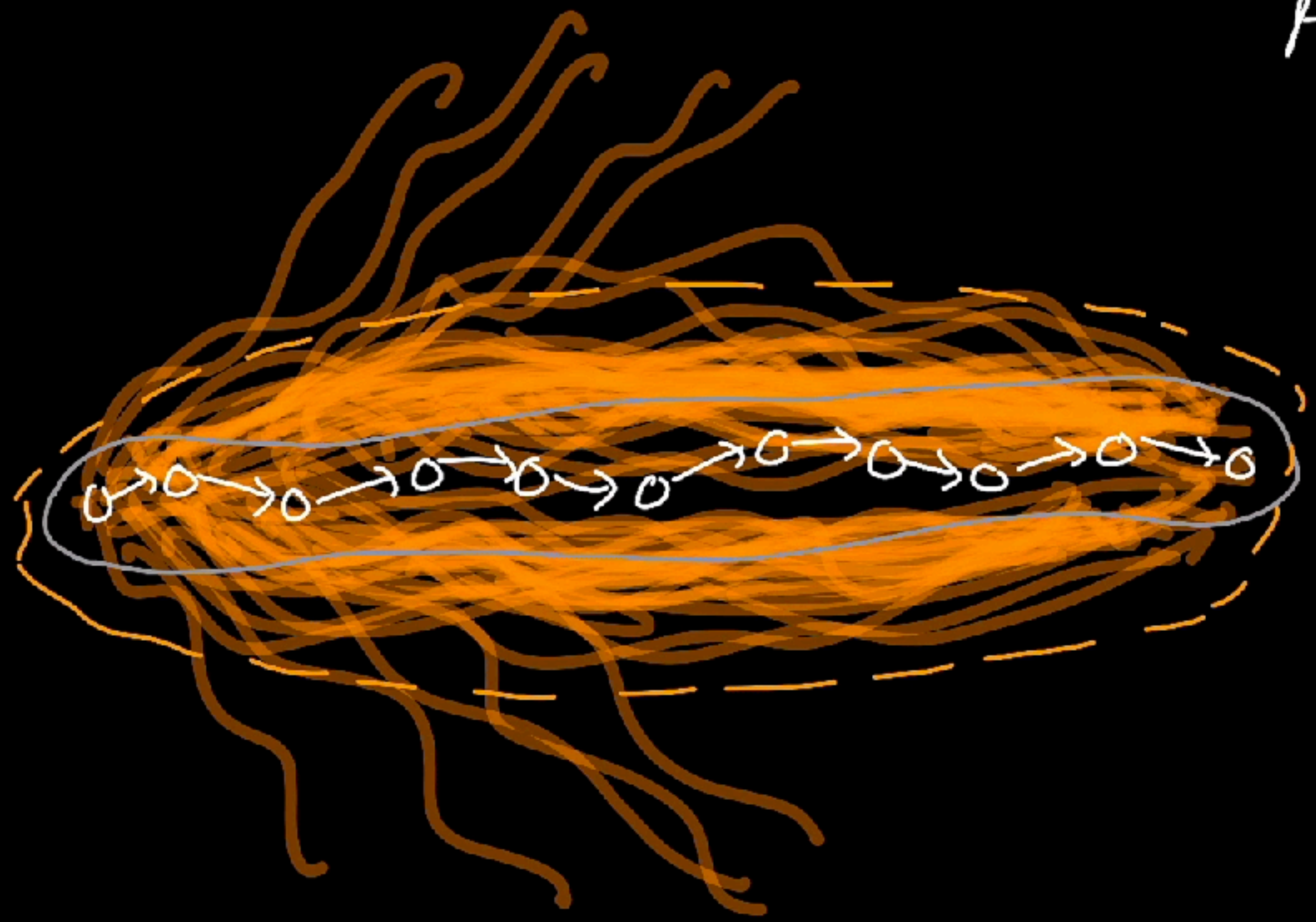
DAgger: Iteration 1

Robot π_N drives



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

DAGGER (DATASET AGGREGATION)



AGGREGATED
TRAINING
DISTRIBUTIO

≈

TEST DISTRIBUTION

HUMAN DISTRIBUTION

But why does *aggregating* data work?

Imitation learning is just a game

Be stable

Slowly change predictions

The Imitation Game

Learner

Adversary

Initialize policy

π_1 [policy]

Chooses loss

$l_1(\,.\,)$ [loss]

Update policy

π_2

Chooses loss

$l_2(\,.\,)$

⋮

⋮

Let's prove!

How can I customize DAGGER to be more practical?

Customizing your DAGGER

Q1. The policy iteration at step 1 is crappy and visits irrelevant states. What do I do?

Blend the expert and learner policy $\pi_i = (1 - \beta_i)\hat{\pi}_i + \beta_i \pi^*$

Q2. What if I can't afford to store all the aggregated data?

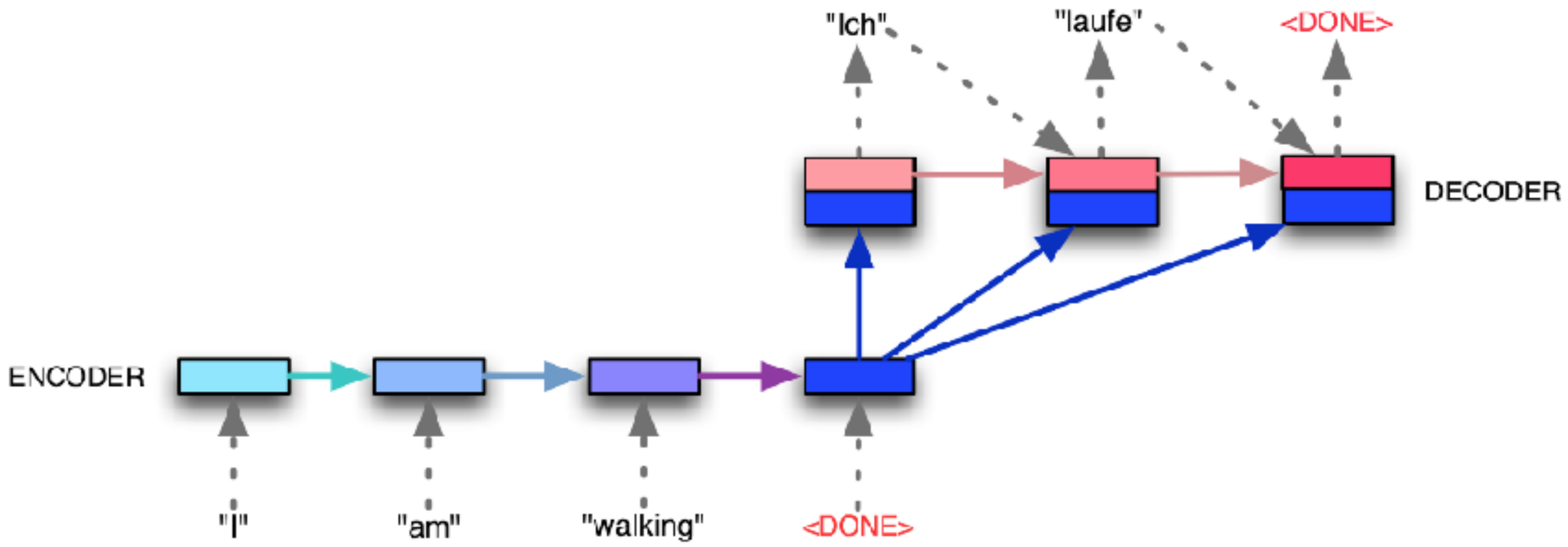
Online gradient descent!

Original results from DAGGER!

DAgger here reacts dynamically to an untrained obstacle

A brief history of DAGGER

NLP folks (Hal Daume III in particular) were first to identify feedback effects in sequential prediction tasks



- Feedback: Previous word predictions feed back in as inputs for future inputs. Data is non-IID.
- Search-based Structured Prediction (SEARN, Daume 2009) looks at this problem for part-of-speech (POS) tagging and handwriting recognition
- DAGGER made the connected the fields on sequential prediction, imitation learning and online learning

DAGGER is a foundation

Imitation under uncertainty

SAIL

ExPLORE STROLL

Counterfactual Teaching

Agnostic SysID

DaaD

Model learning

DPI LOLS

NRPI

Reinforcement Learning

DAEQUIL

AGGREVATE(D)

Imitation learning

EIL

HG-DAGGER

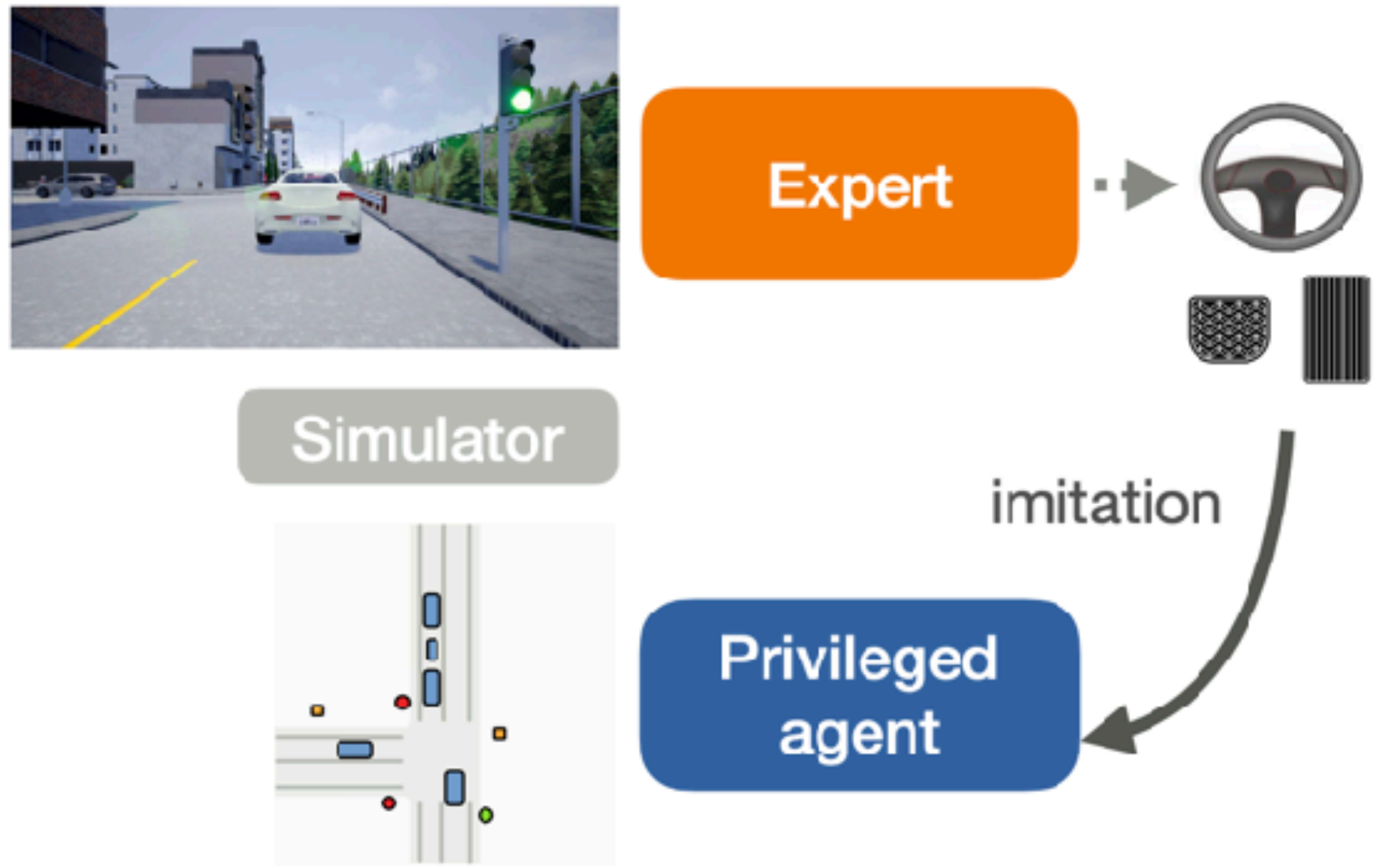
SHIV

Query efficient imitation learning

DAGGER

Many cool applications of DAGGER in robotics

Lee et al, Learning quadrupedal locomotion over challenging terrain (2020)

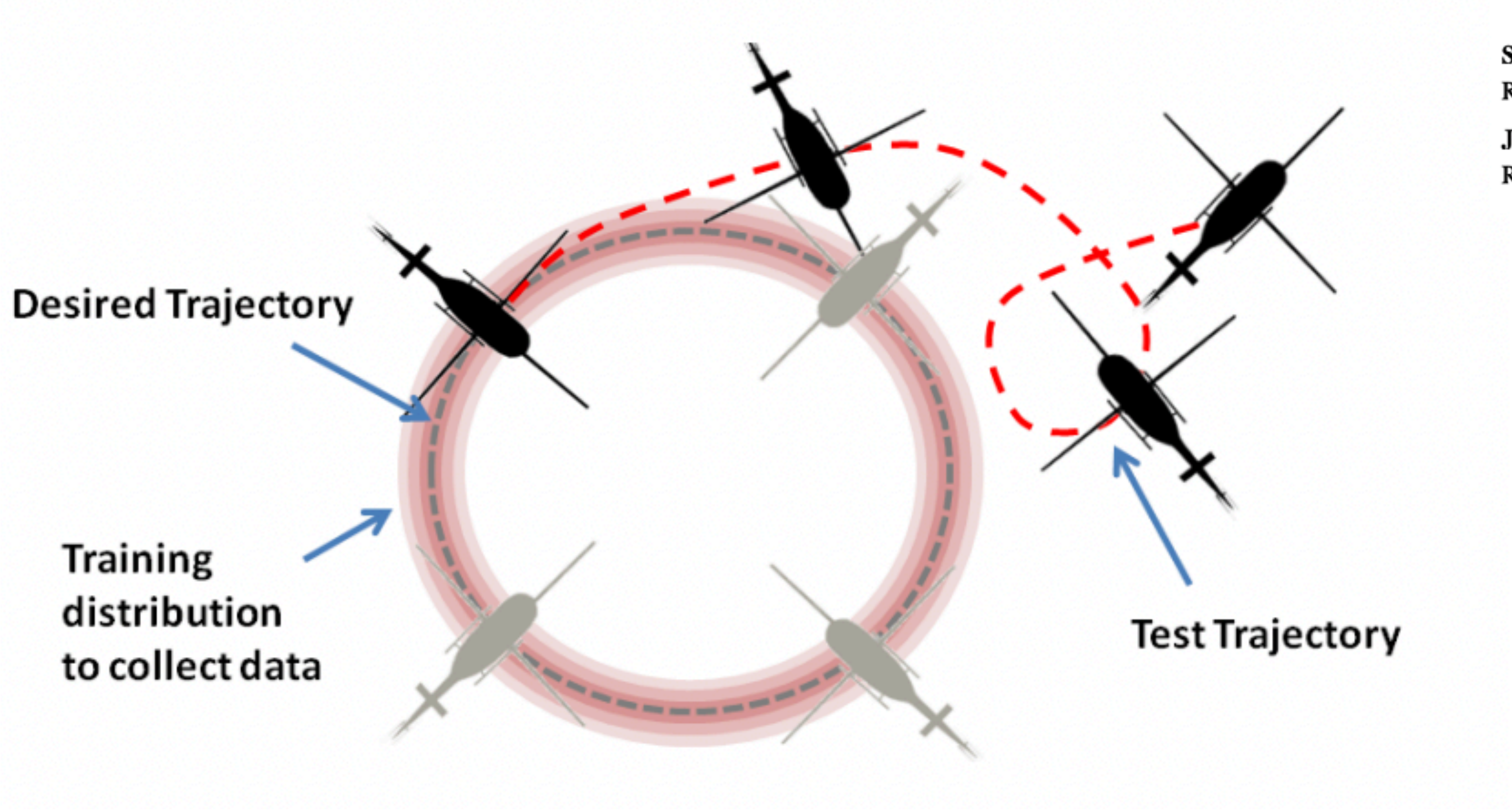


Chen et al Learning by Cheating(2020)

Choudhury et al, Data Driven Planning via Imitation Learning (2018)

Pan et al Imitation learning for agile autonomous driving (2019)

DAGGER is not *just* for imitation learning!



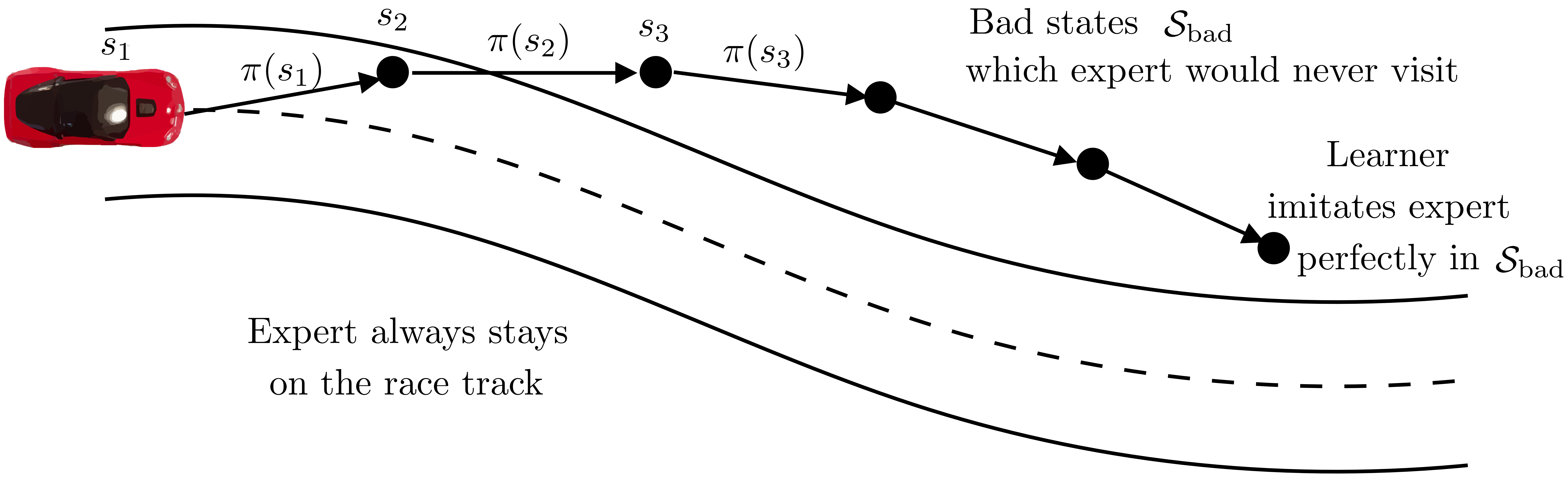

Agnostic System Identification for Model-Based Reinforcement Learning

Stéphane Ross STEPHANEROSS@CMU.EDU
Robotics Institute, Carnegie Mellon University, PA USA

J. Andrew Bagnell DBAGNELL@RI.CMU.EDU
Robotics Institute, Carnegie Mellon University, PA USA

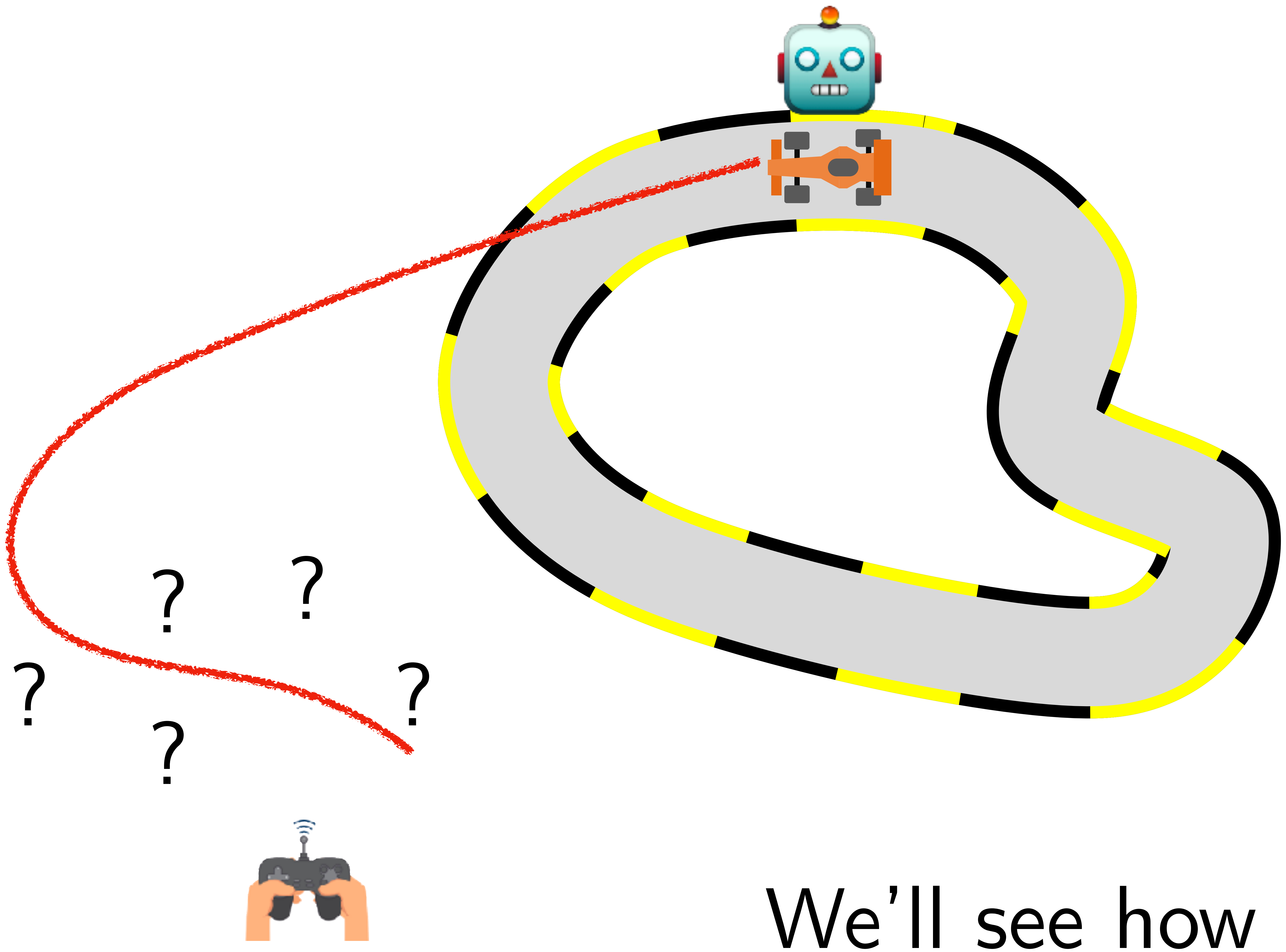

Hidden charges from DAGGER

Hidden charge #1: Not all mistakes are equal



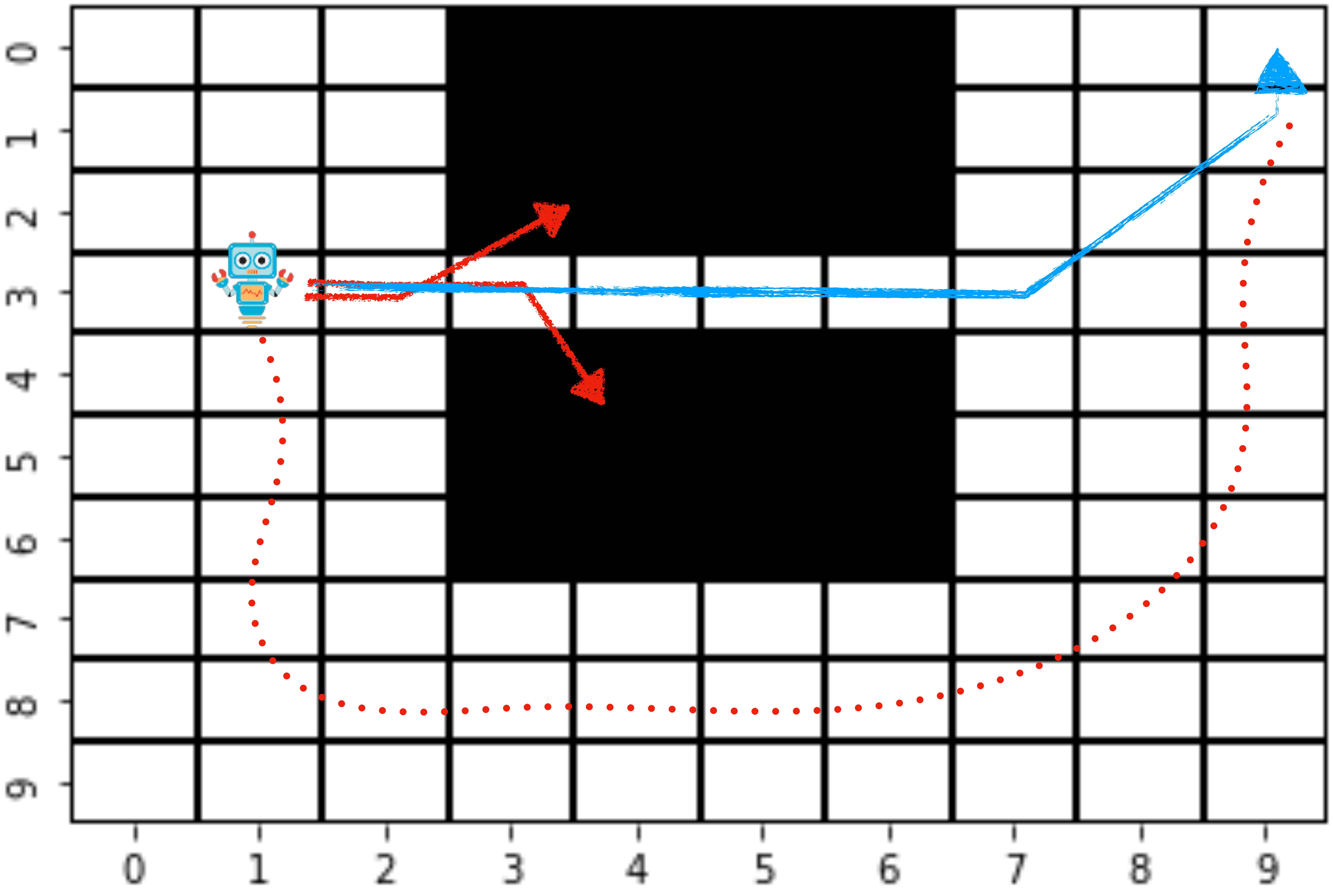
Dagger minimizes 0-1 loss, but what we really want to optimize are advantages! (More next lecture)

Hidden charge #2: Dagger asks the expert for queries *everywhere*



We'll see how to learn from
limited human feedback (interventions)

Hidden charge #3: Dagger expects at least one policy to be good *everywhere*



Learner simply can't cross the bridge! …

… but can take the long way round.