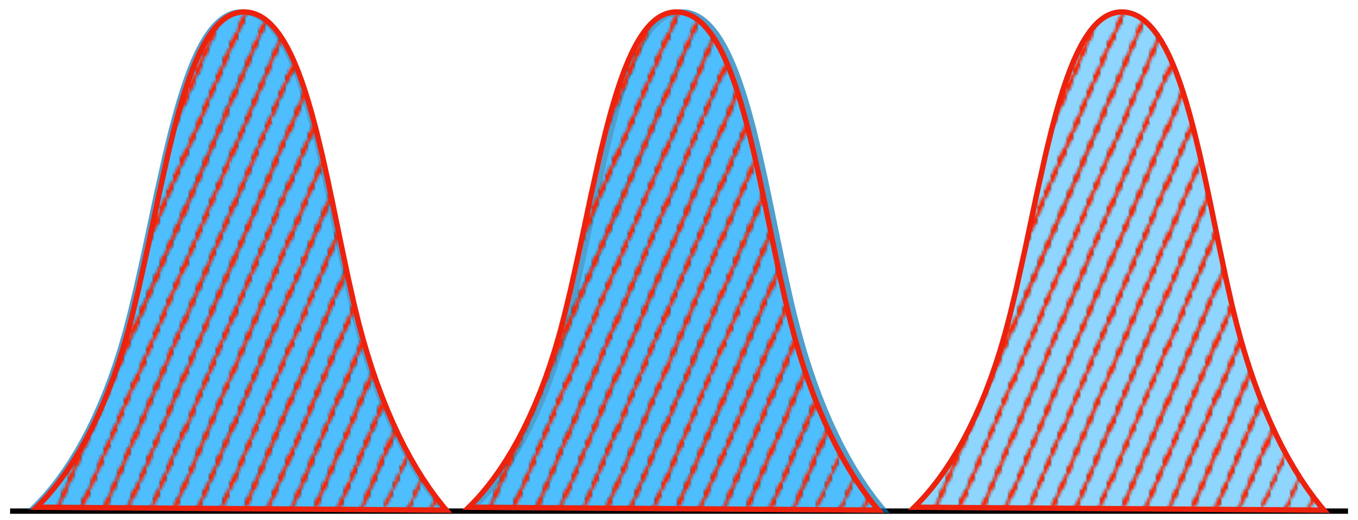
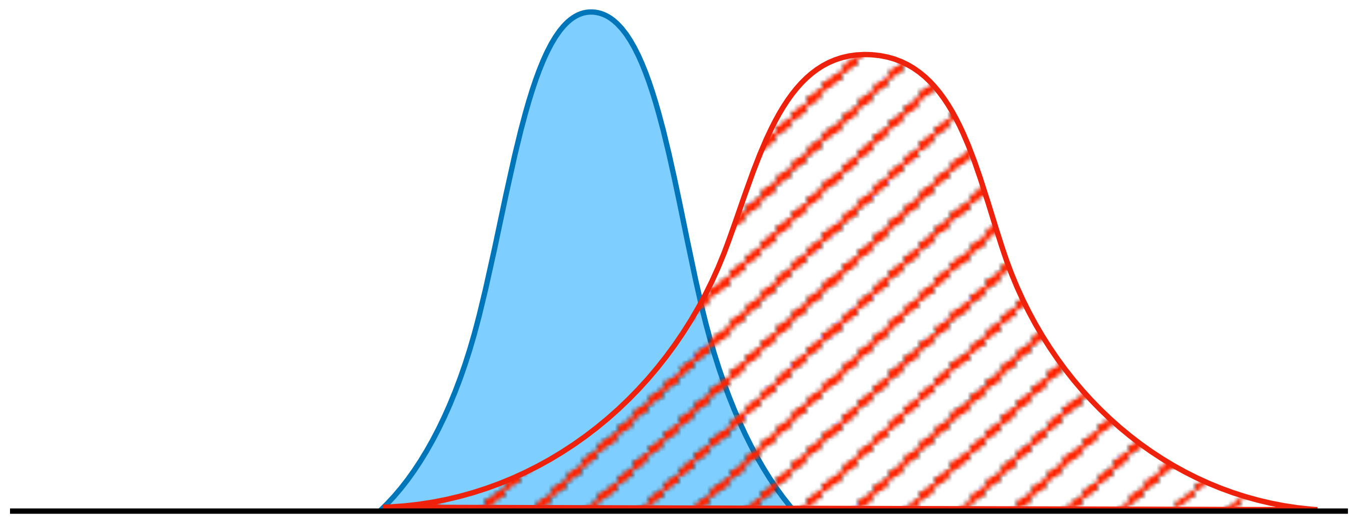
Inverse Reinforcement Learning: From Maximum Margin to Maximum Entropy

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Let's travel to the INFINITE data limit!

The
Three Regimes
of
Covariate
Shift

	Easy 😄	**Hard** 😱
Setting	Expert is realizable $\pi^E \in \Pi$	Non-realizable expert + limited expert support
	As $N \to \infty$, drive down $\epsilon = 0$ (or Bayes error)	Even as $N \to \infty$, behavior cloning $O(\epsilon T^2)$
		
Solution	Nothing special. Collect lots of data and do Behavior Cloning	Requires interactive expert (DAGGER/EIL) to provide labels $\Rightarrow O(\epsilon T)$

Setting

Solution

Easy 😄

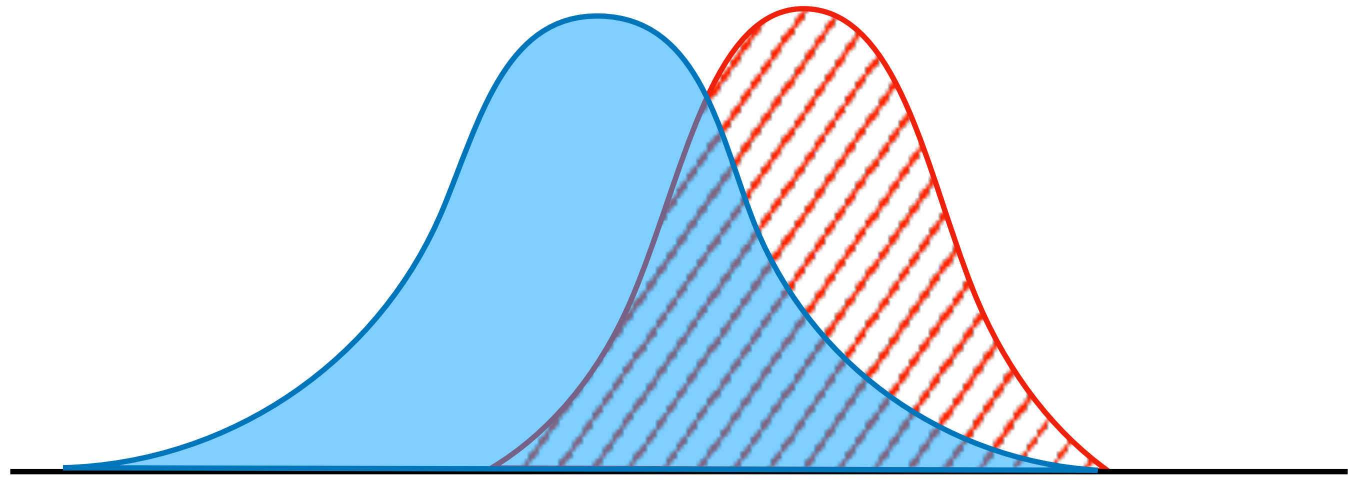
Expert is realizable $\pi^E \in \Pi$

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

Nothing special. Collect lots of data and do Behavior Cloning

Medium 🤔

Non-realizable expert but full expert support



?

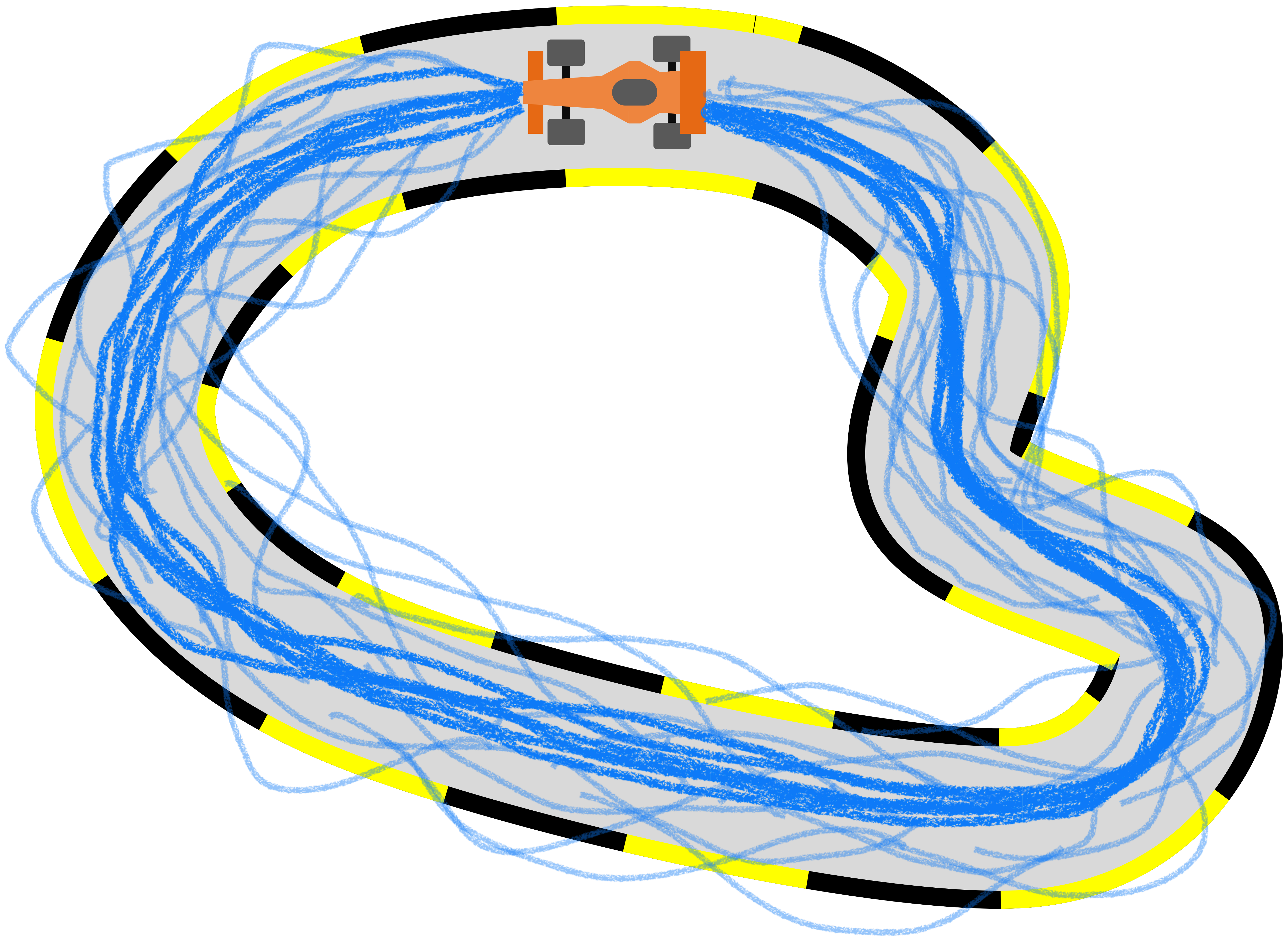
Hard 😱

Non-realizable expert + limited expert support

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

Requires interactive expert (DAGGER/EIL) to provide labels $\Rightarrow O(\epsilon T)$

Expert demonstrations have full coverage



.. but expert runs away after demonstrations

So expert data has
full coverage ...

.. why don't we just do
Behavior Cloning?

Activity!

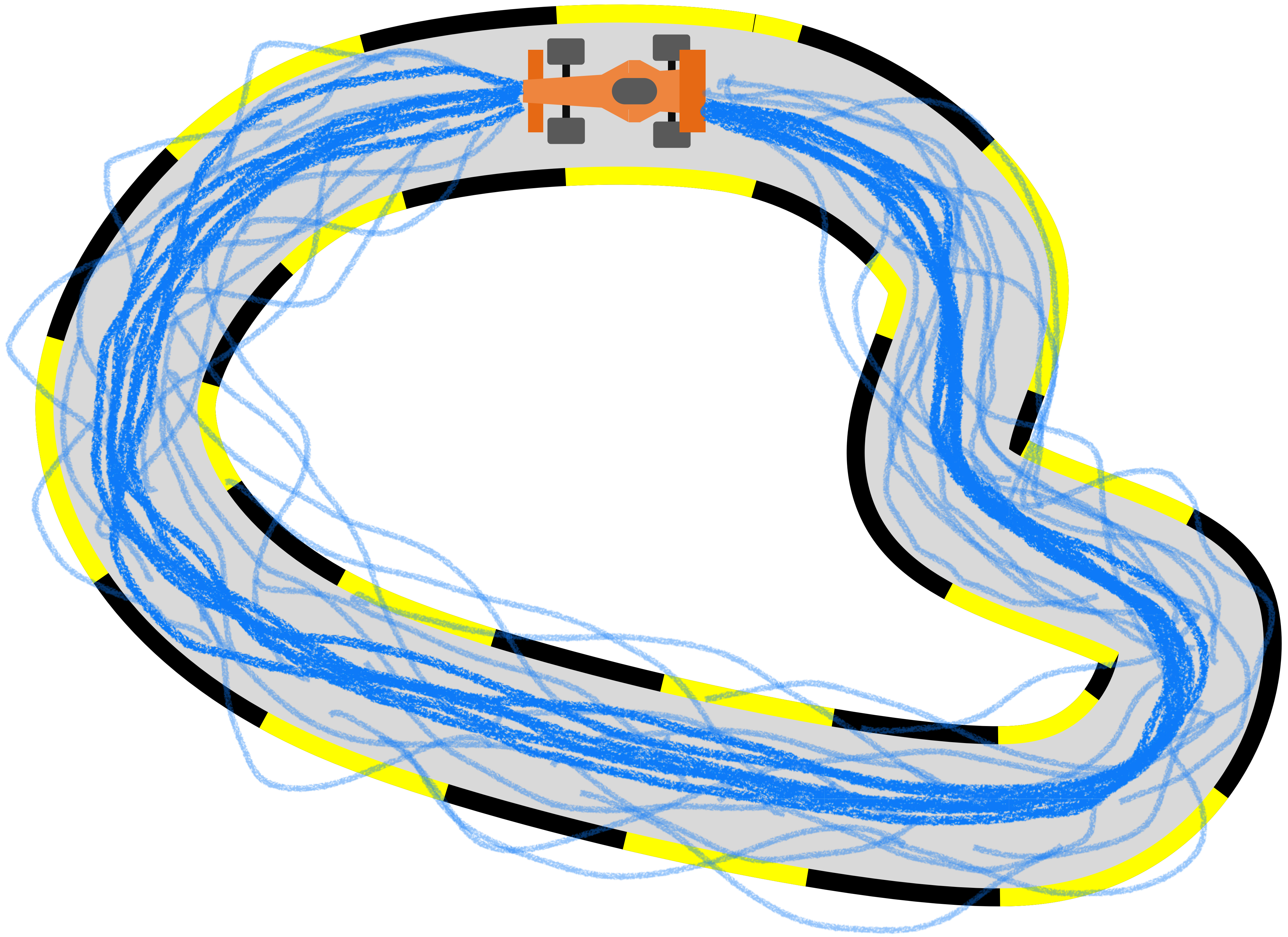
Think-Pair-Share!

To BC, or not to BC!

Think (30 sec): Will BC work? $O(\epsilon T)$ or $O(\epsilon T^2)$?
Make the argument!

Pair: Find a partner

Share (45 sec): Partners exchange ideas



BC results in compounding errors

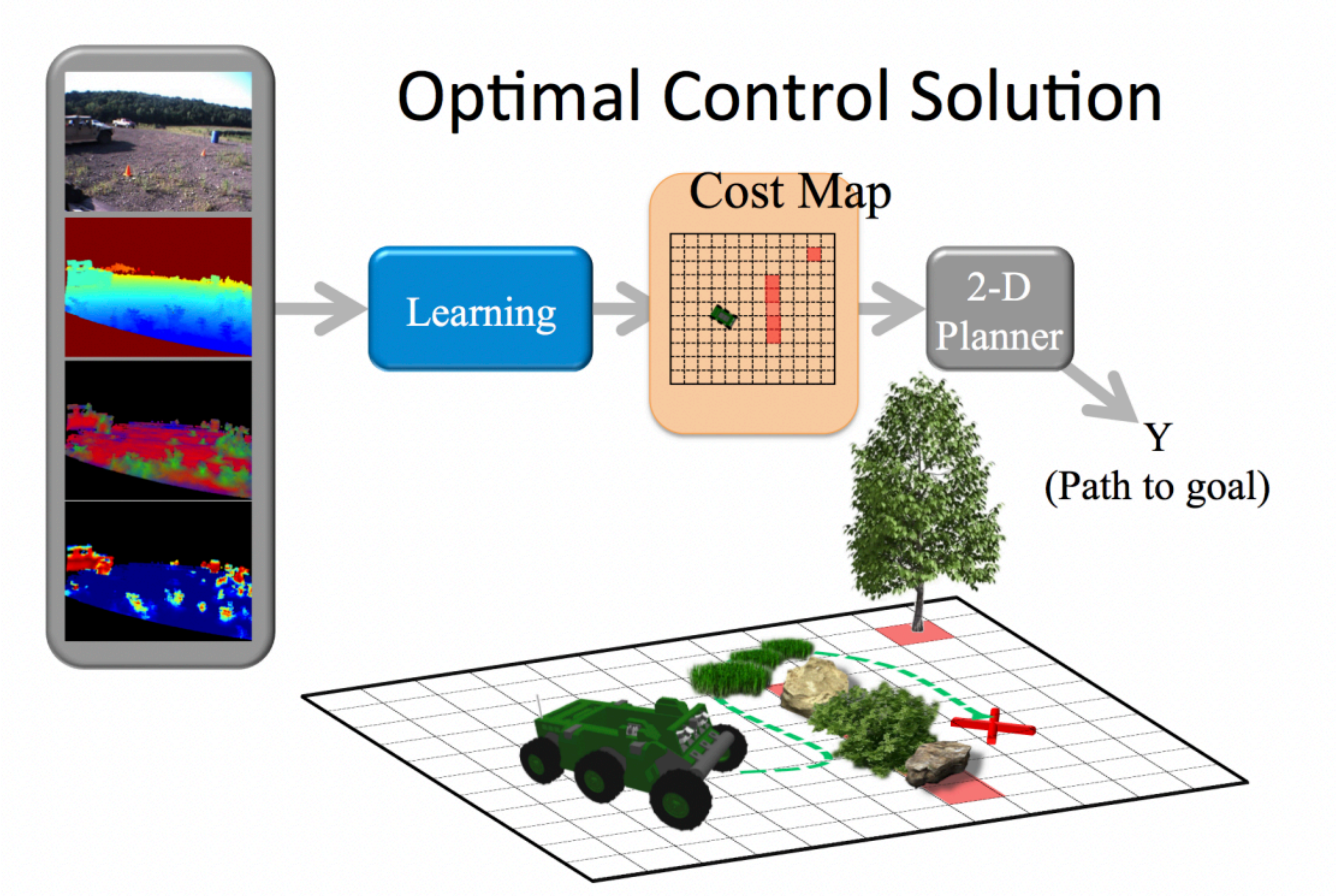
We don't have an interactive expert

What if we knew our MDP (except the cost)?

Or what if we had an interactive simulator?

CRUSHER robot from CMU

Can we learn a cost function for CRUSHER navigation?



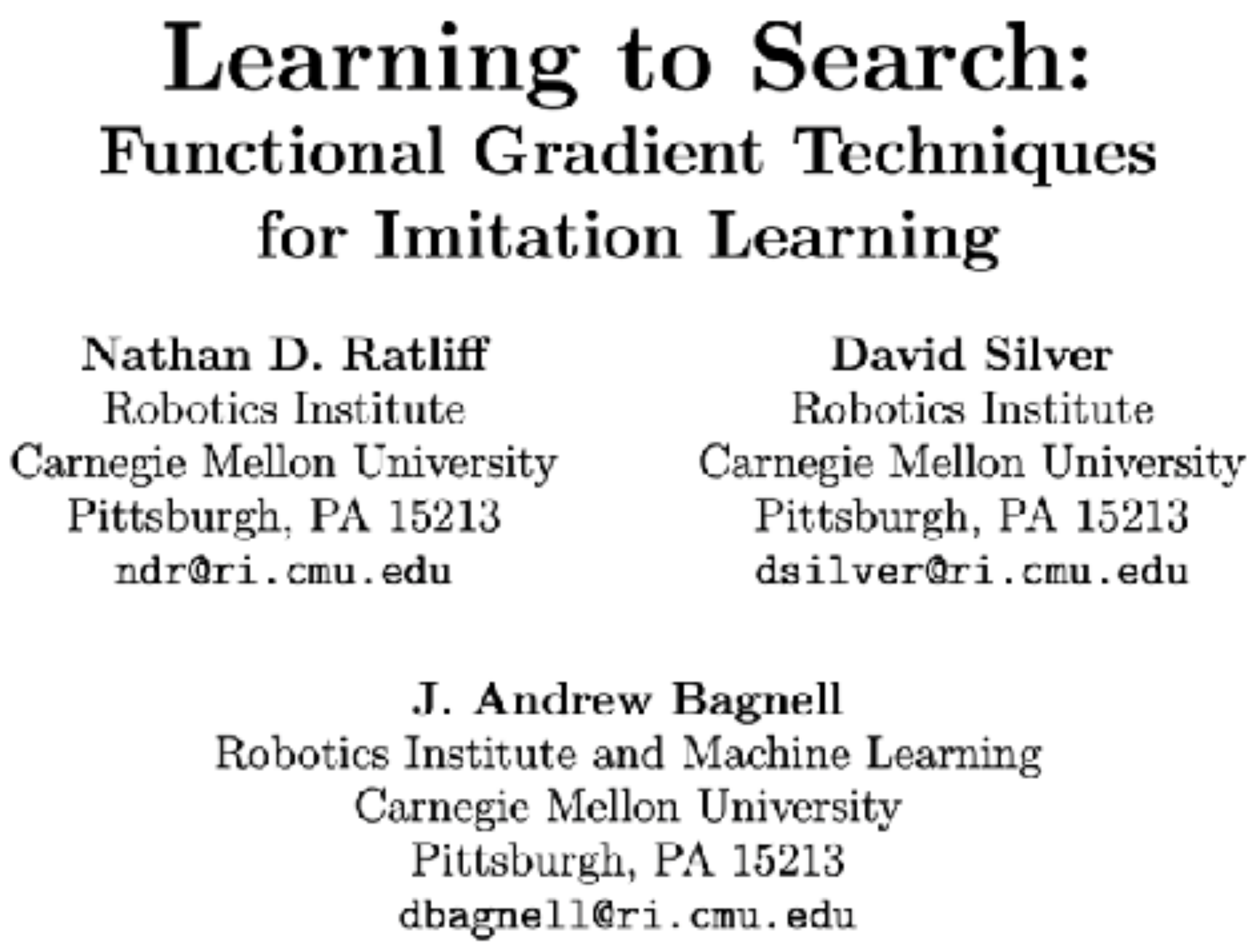
Let's formalize!

Learning to Search (LEARCH)

Min distance

Stay on roads

Stay near trees



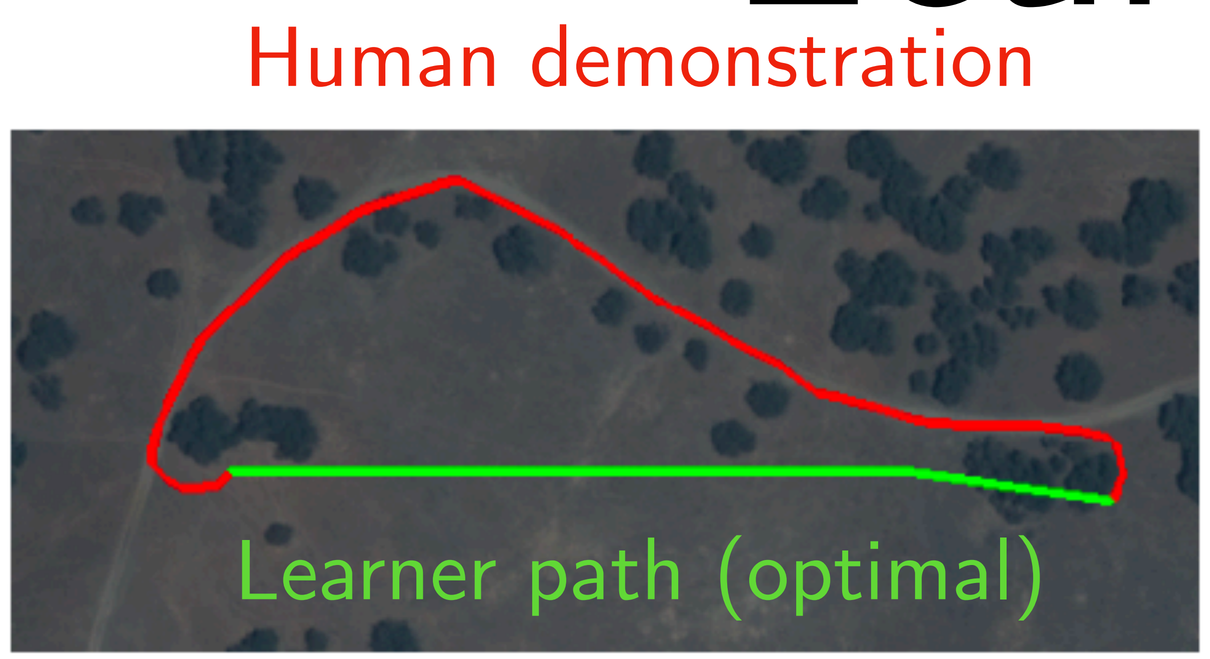
Learning to Search:
Functional Gradient Techniques for Imitation Learning

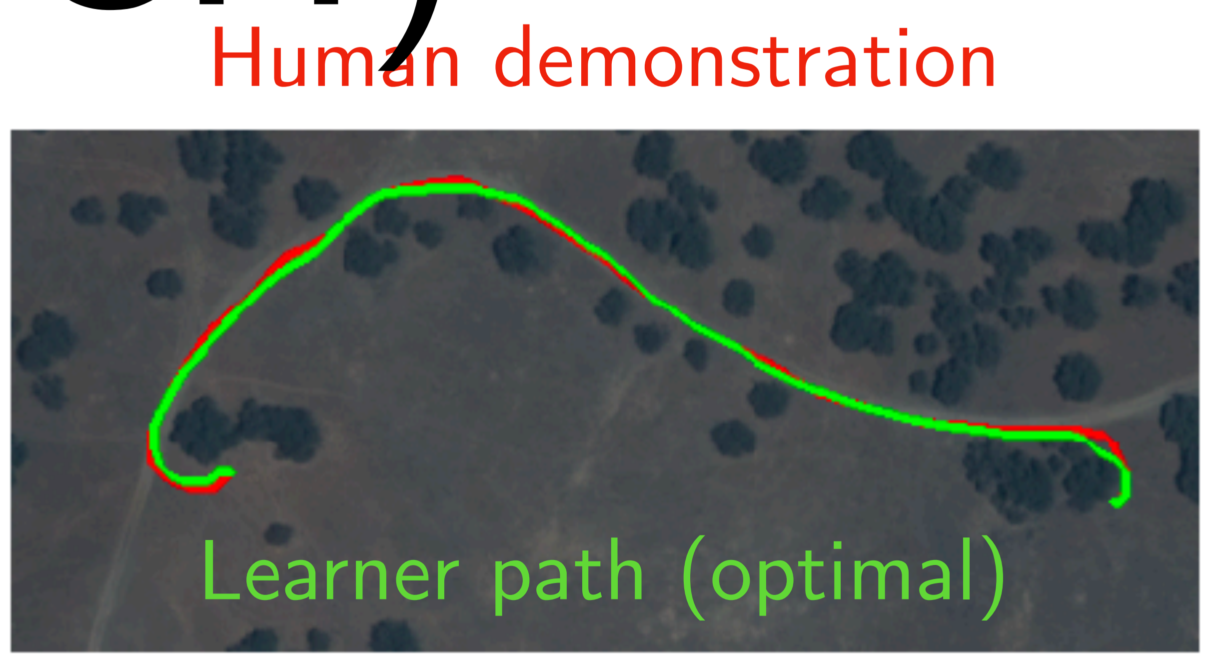
Nathan D. Ratliff
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
ndr@ri.cmu.edu

David Silver
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
dsilver@ri.cmu.edu

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

Learning to Search (LEARCH)





Given dataset: $\{\xi_i^h, \phi_i\}_{i=1}^N$

(Human demo) *(Map)*

Solve for cost $C_\theta(\xi)$

$$\min_\theta \frac{1}{N} \sum_{i=1}^N \left(C_\theta(\xi_i^h, \phi_i) - \min_\xi [C_\theta(\xi, \phi_i) - \gamma(\xi, \xi^h)] \right) + R(\theta)$$

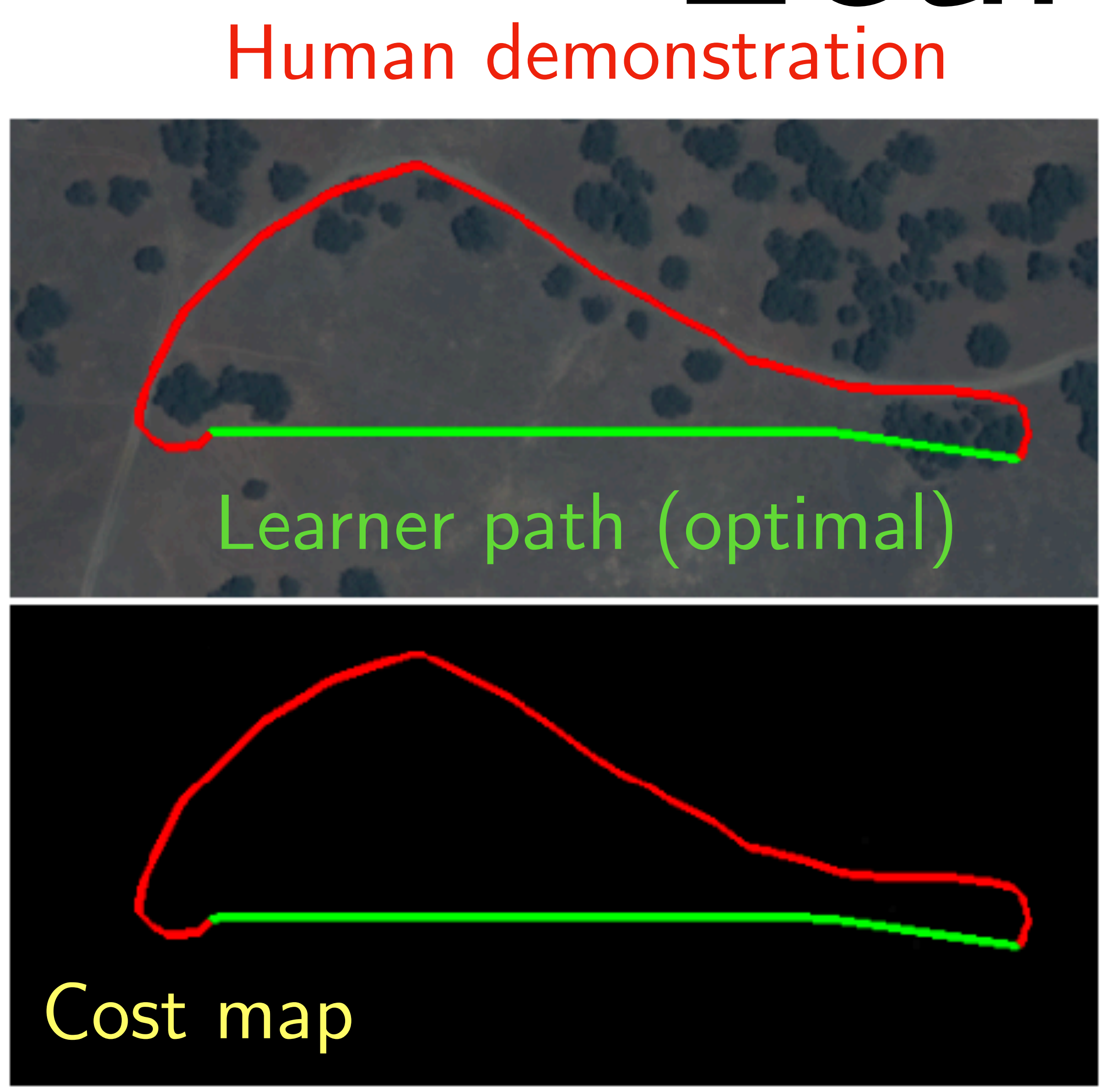
(Margin)

Human Cost

Learner Cost

Regularizer

Learning to Search (LEARCH)



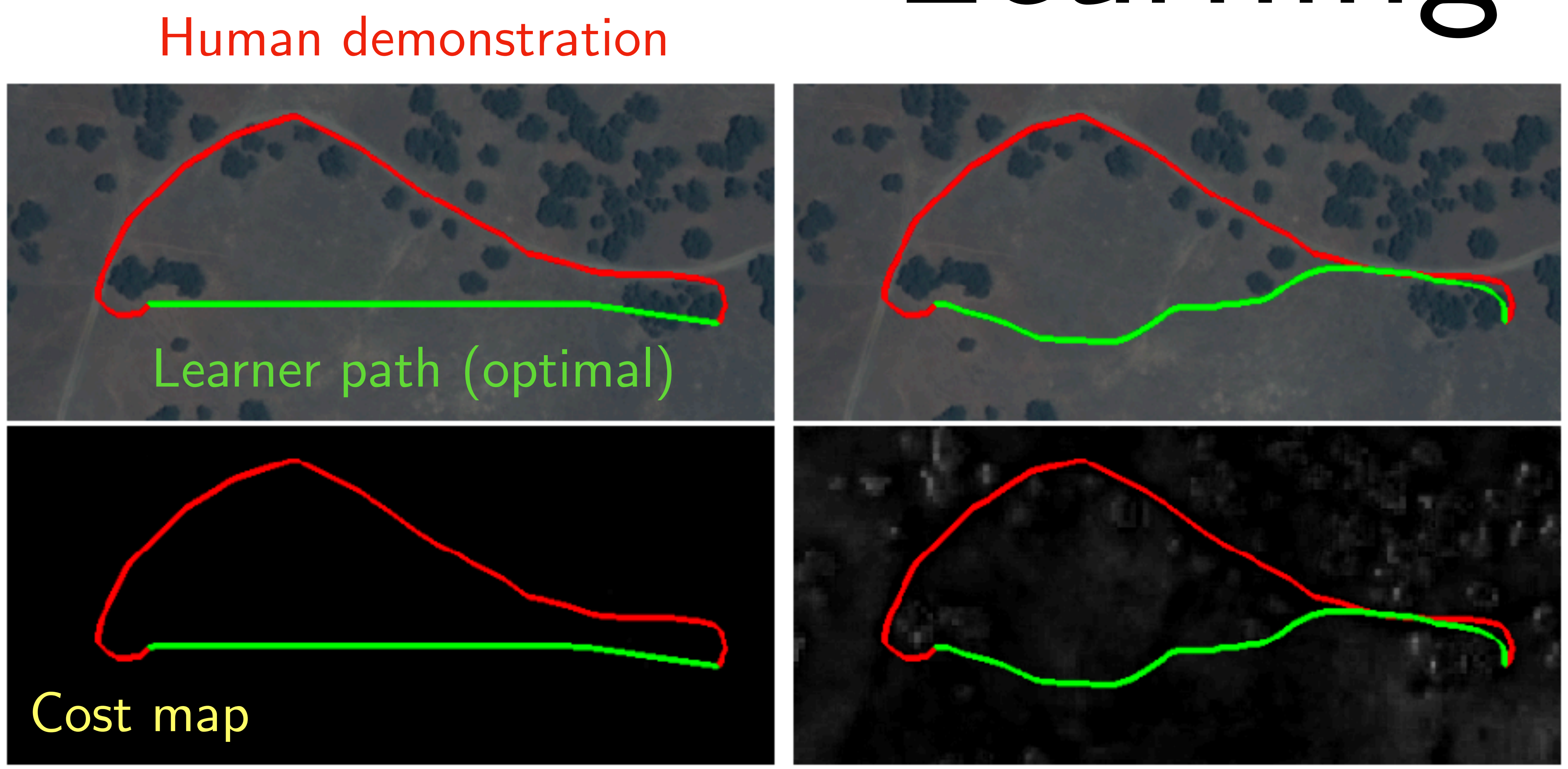
for $i = 1, \ldots, N$ # Loop over datapoints

$$\xi_i^* = \min_{\xi} [C_\theta(\xi, \phi_i) - \gamma(\xi, \xi^h)]$$ # Call planner!

$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i^*, \phi_i) + \nabla_\theta R(\theta)]$$

(Push down human cost) *(Push up planner cost)* # Update cost

Learning to Search



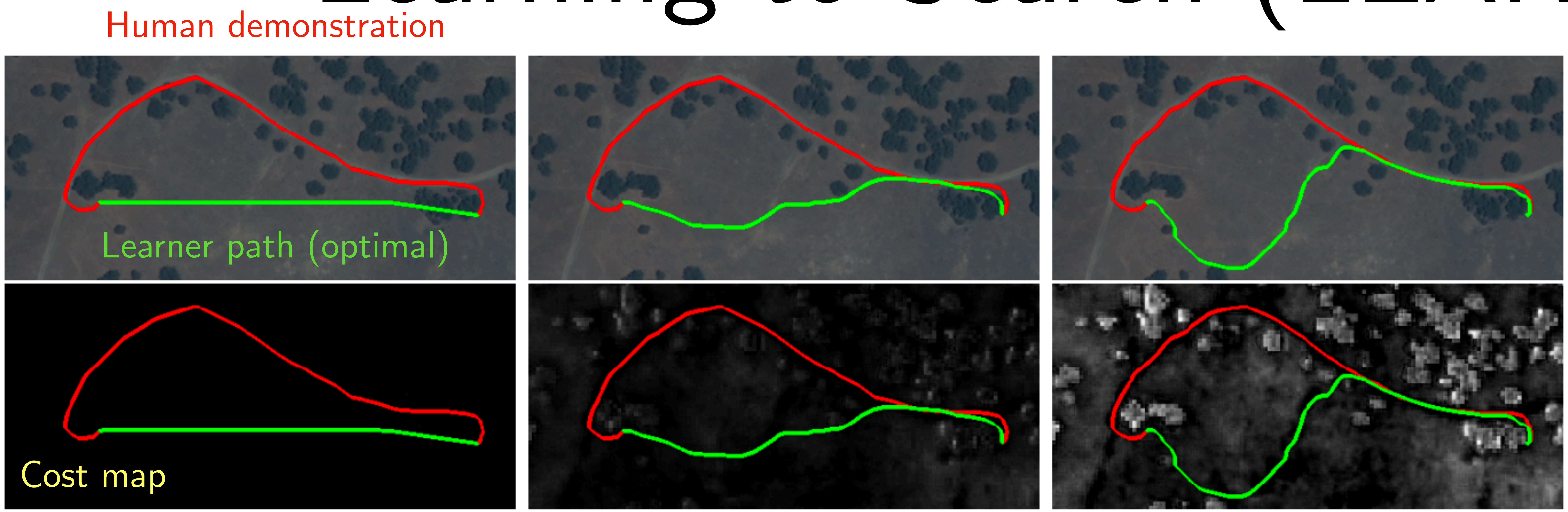
for $i = 1,\ldots,N$ # Loop over datapoints

$$\xi_i^* = \min_{\xi}[C_\theta(\xi, \phi_i) - \gamma(\xi, \xi^h)]$$ # Call planner!

$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i^*, \phi_i) + \nabla_\theta R(\theta)]$$

(Push down human cost) *(Push up planner cost)* # Update cost

Learning to Search (LEARCH)



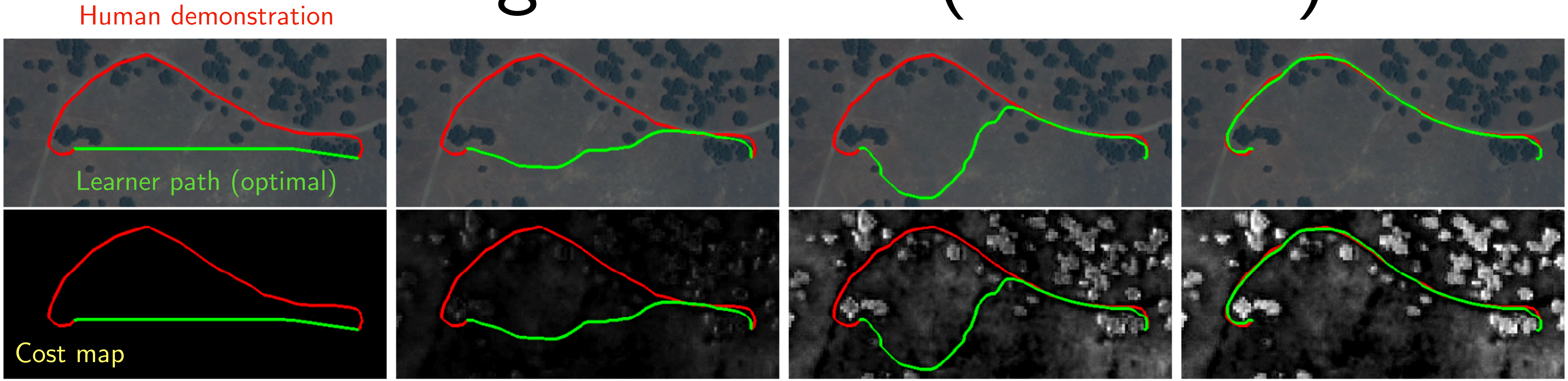
for $i = 1, \ldots, N$ # Loop over datapoints

$$\xi_i^* = \min_{\xi} [C_\theta(\xi, \phi_i) - \gamma(\xi, \xi^h)]$$ # Call planner!

$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i^*, \phi_i) + \nabla_\theta R(\theta)]$$

(Push down human cost) *(Push up planner cost)* # Update cost

Learning to Search (LEARCH)



$\text{for } i = 1, \ldots, N$ # Loop over datapoints

$$\xi_i^* = \min_{\xi}[C_\theta(\xi, \phi_i) - \gamma(\xi, \xi^h)]$$ # Call planner!

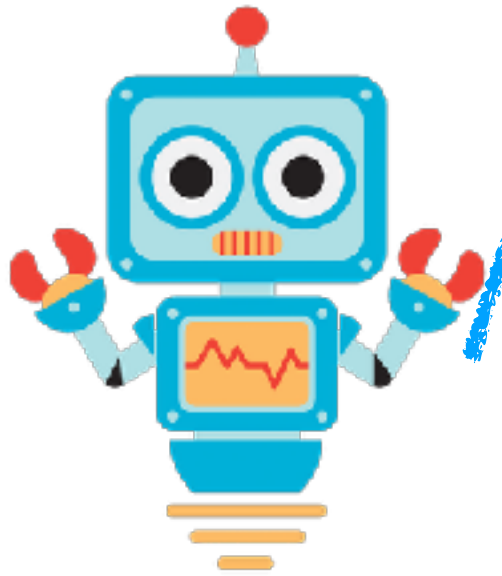
$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i^*, \phi_i) + \nabla_\theta R(\theta)]$$

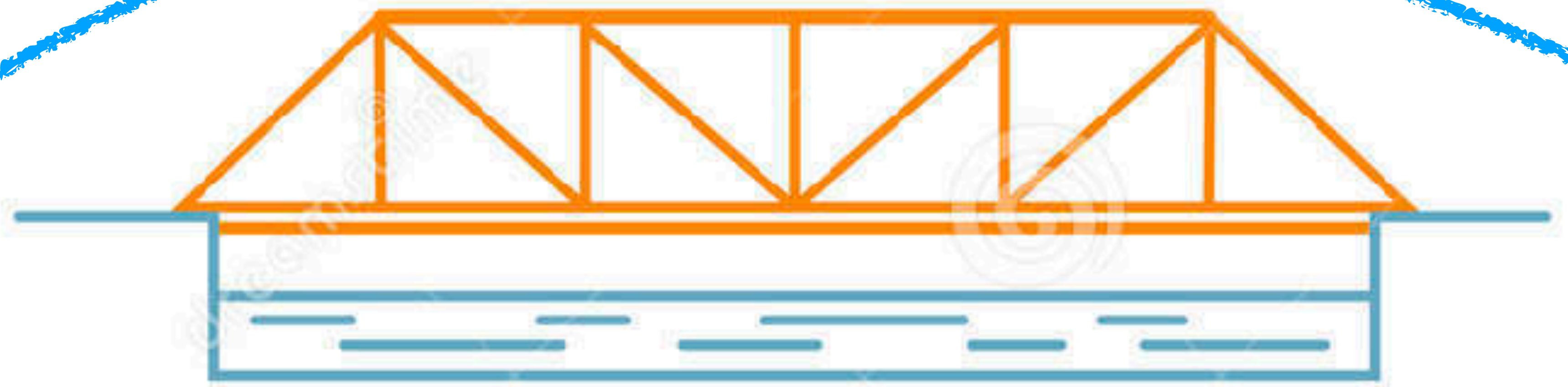
(Push down human cost) *(Push up planner cost)* # Update cost

What happens when the expert is stochastic / noisy / suboptimal?

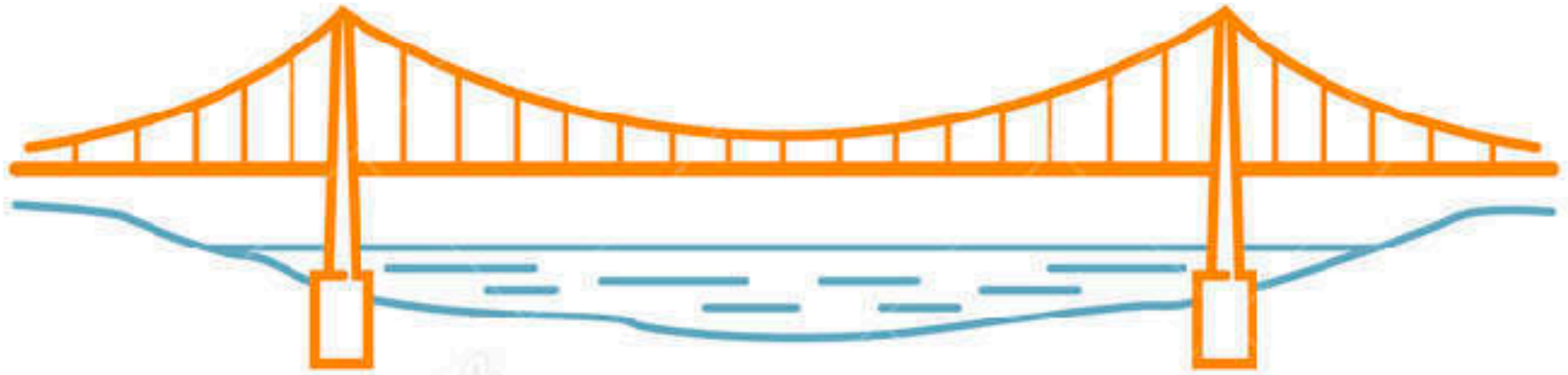
Learning which bridge to cross

Demonstrations always pick Bridge 1





$f = [1 \quad 0 \quad 0]$



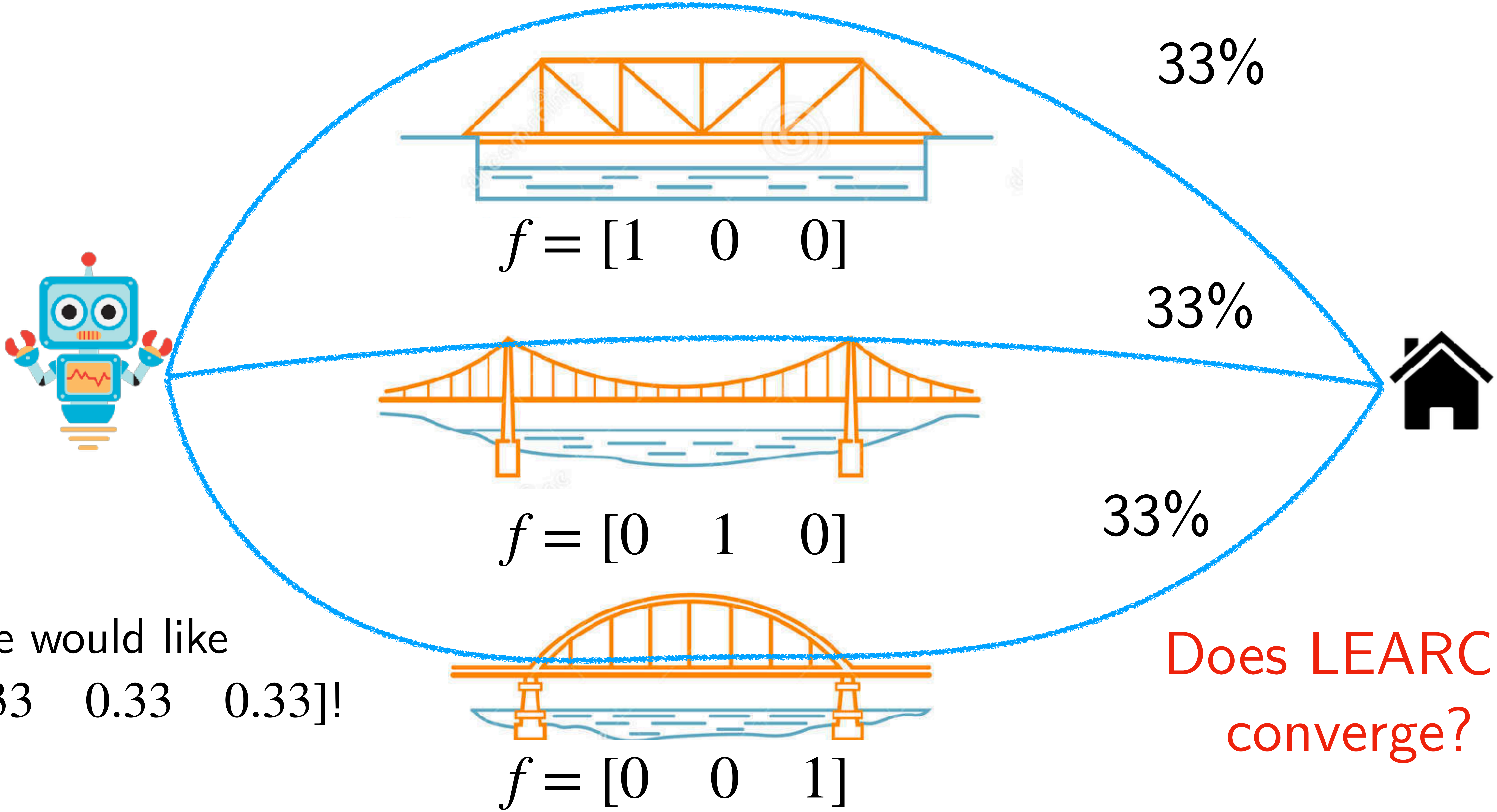
$f = [0 \quad 1 \quad 0]$

$f = [0 \quad 0 \quad 1]$

LEARCH converges to

$w = [1 \quad 0 \quad 0]$!

Learning which bridge to cross



We would like
$w = [0.33 \quad 0.33 \quad 0.33]$!

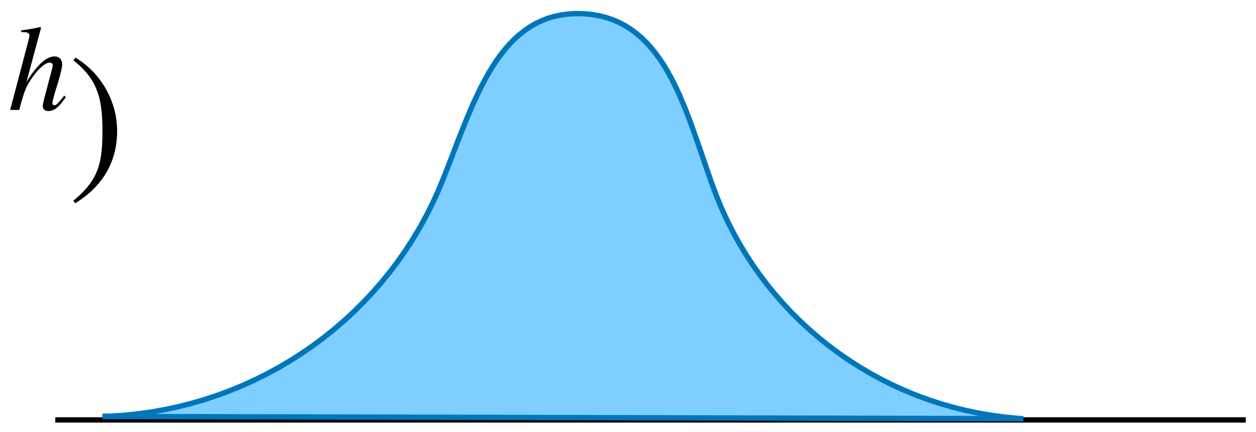
Does LEARCH converge?

Expert demonstrations are coming from some (unknown) distribution ..

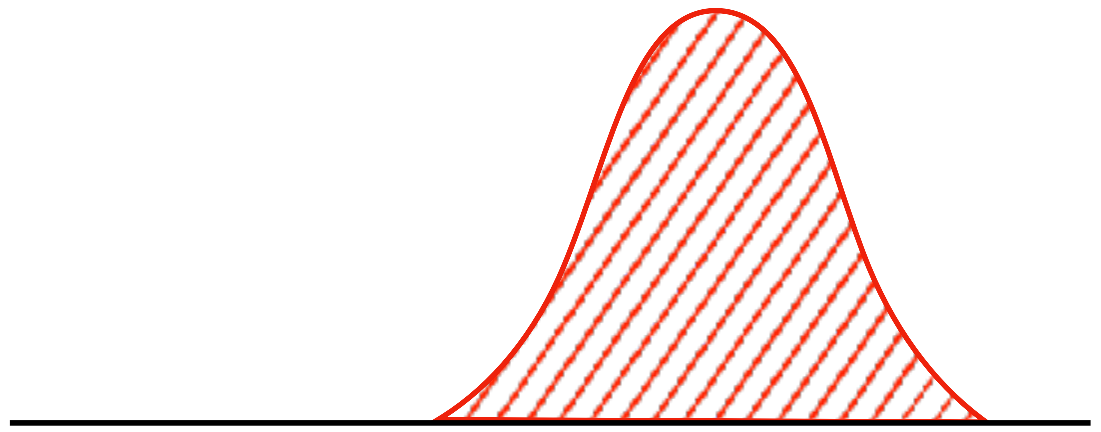
Can we learn this distribution?

The Distribution Matching Problem

$P_{expert}(\xi^h)$



(Unknown) expert distribution

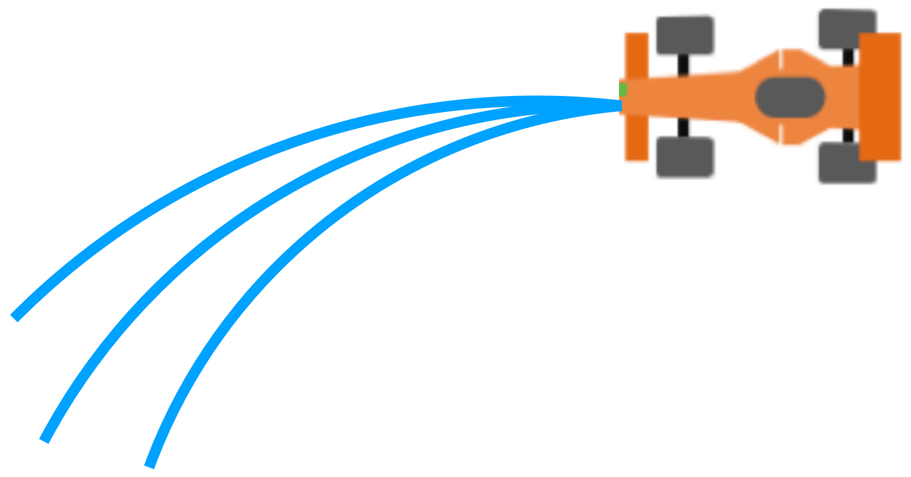


$P_\theta(\xi)$

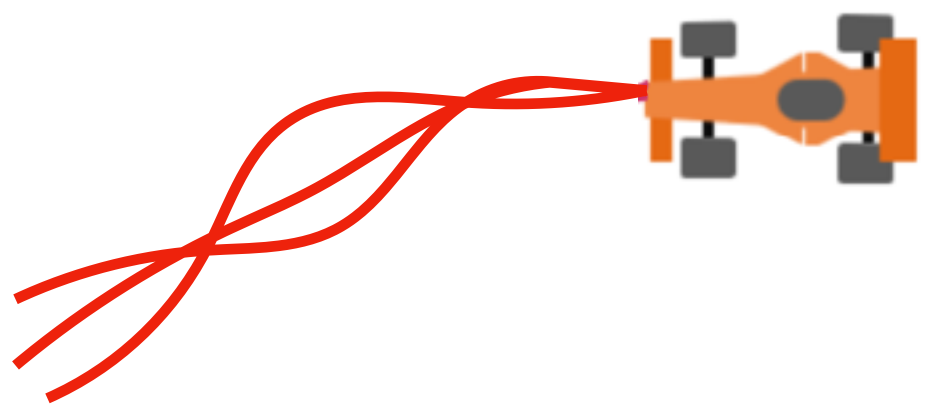
Learn distribution over trajectories

All we see are expert samples

Learner can also generate samples



What loss should we use?



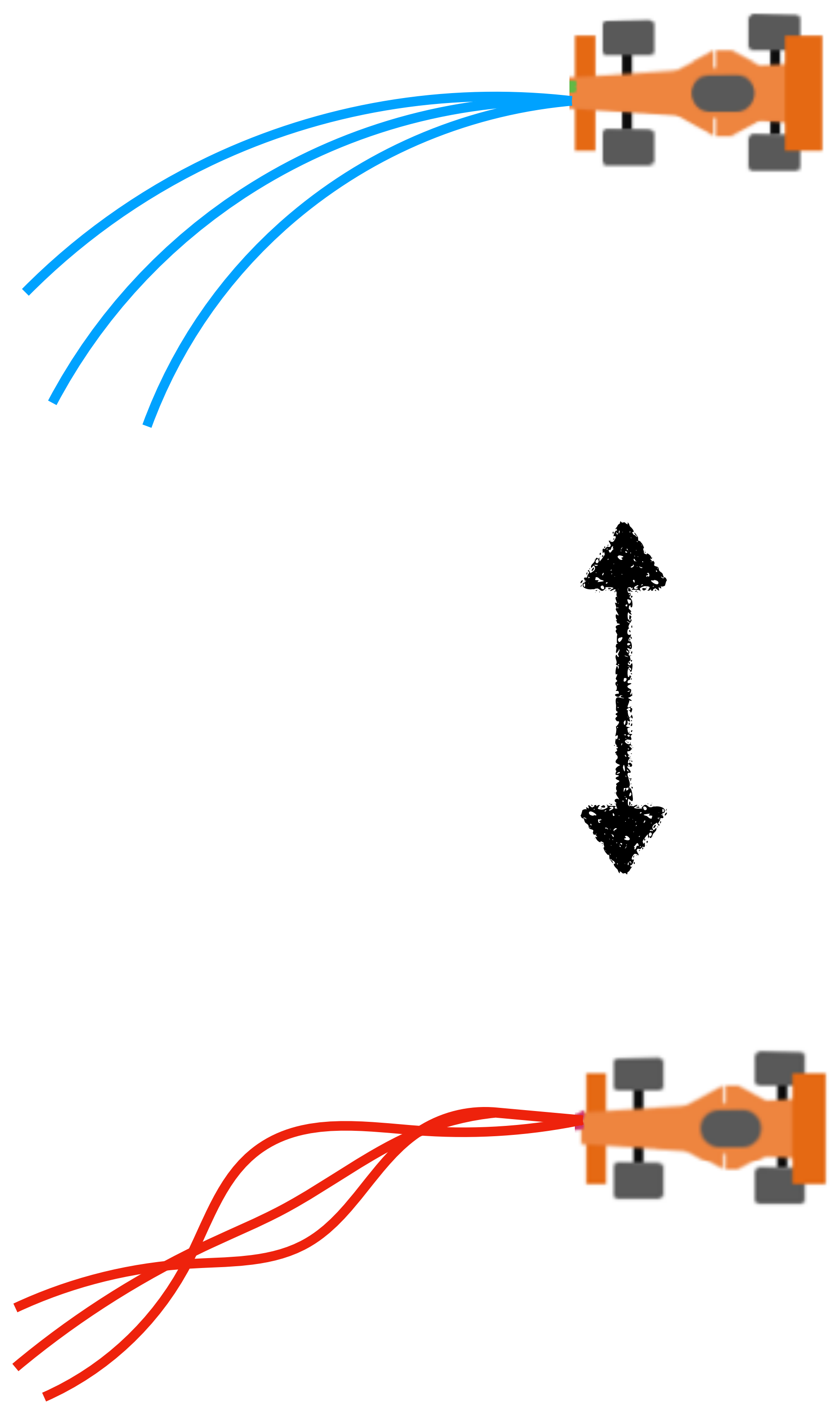
Activity!

Think-Pair-Share!

Think (30 sec): Given samples from expert and learner, what loss should we define to get learner to match expert distribution?

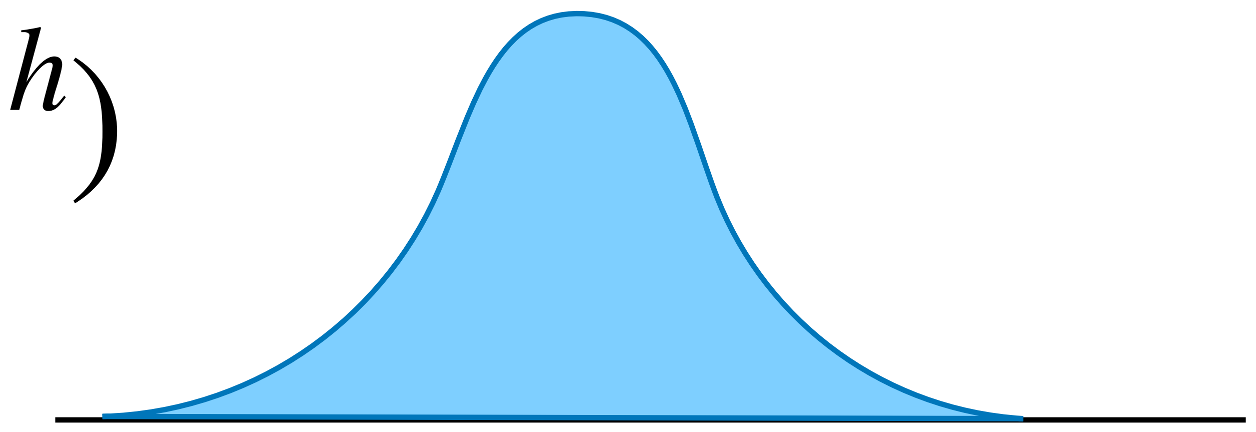
Pair: Find a partner

Share (45 sec): Partners exchange ideas

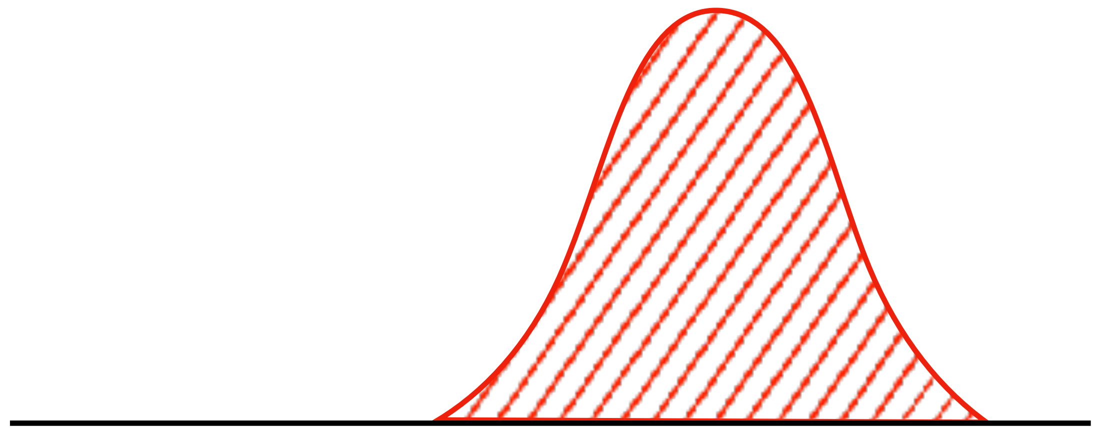


Proposal: Match expected costs?

$P_{expert}(\xi^h)$



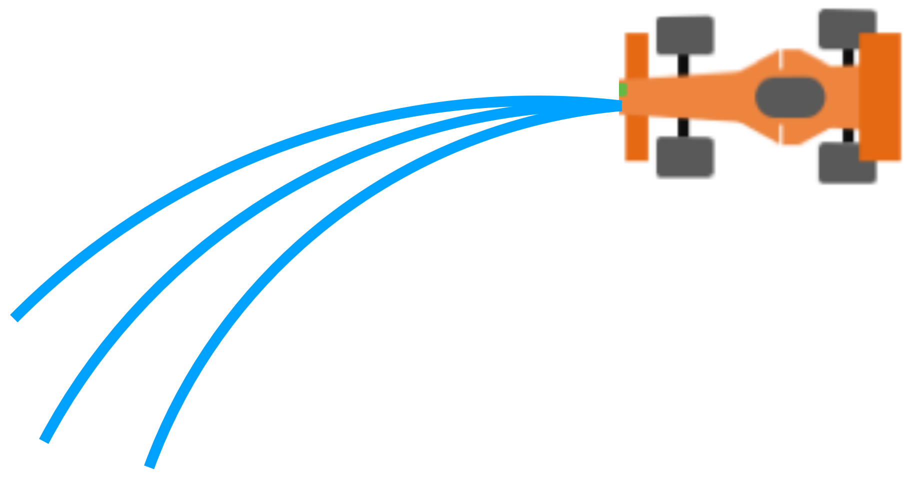
(Unknown) expert distribution



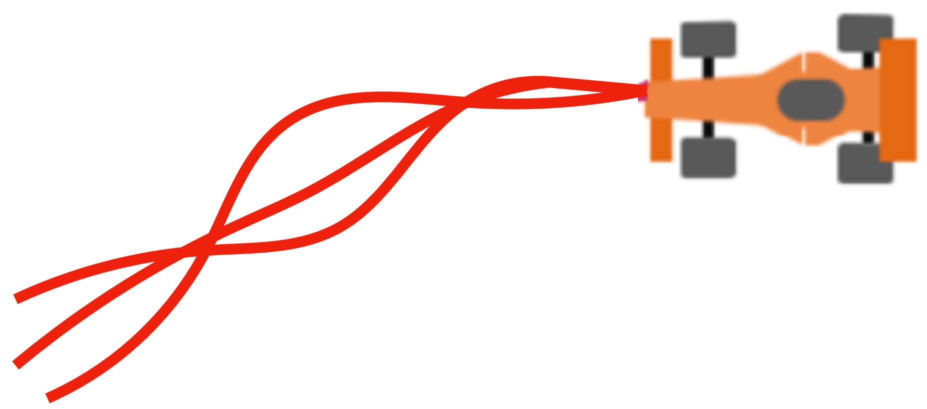
$P_\theta(\xi)$

Learn distribution over trajectories

All we see are expert samples



Learner can also generate samples



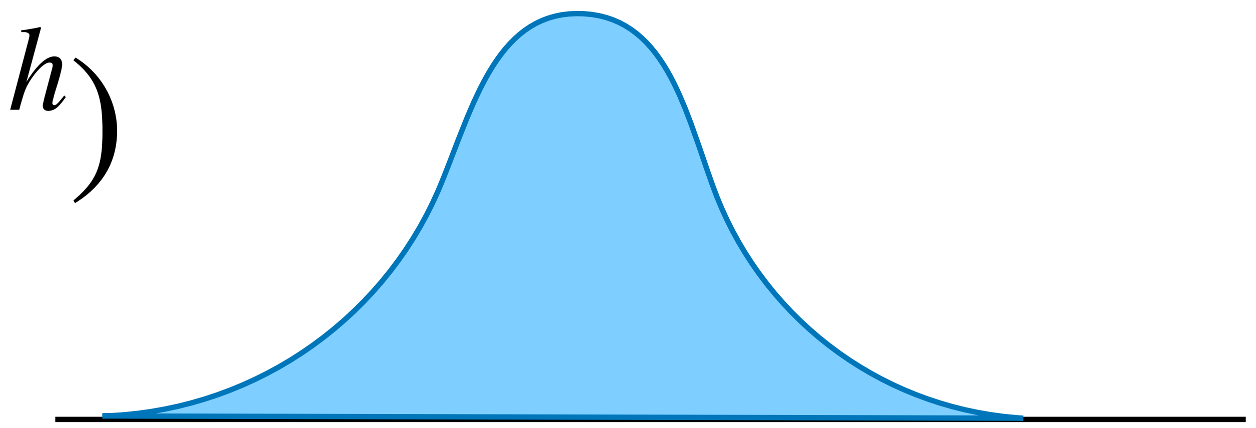
$$\mathbb{E}_{\xi^h \sim P_{expert}(.)} \text{cost}(\xi^h) = \mathbb{E}_{\xi \sim P_\theta(.)} \text{cost}(\xi)$$

But wait .. how can we match costs if we don't know the weights w?

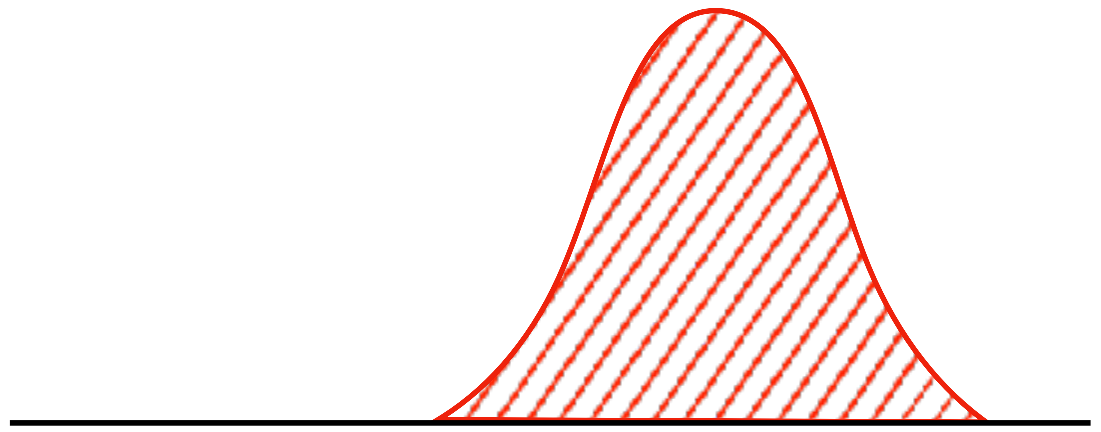
$$\text{cost}(\xi) = w^T f$$

Proposal: Match expected features!

$P_{expert}(\xi^h)$



(Unknown) expert distribution

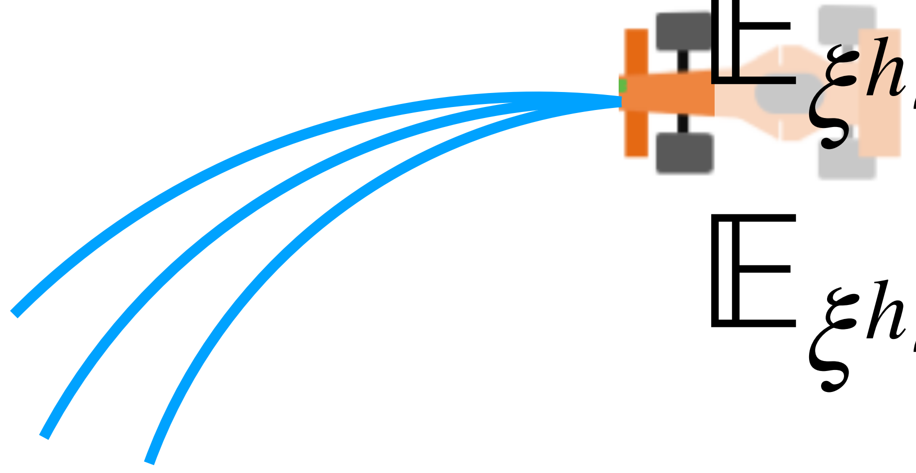


$P_\theta(\xi)$

Learn distribution over trajectories

All we see are expert samples

Learner can also generate samples

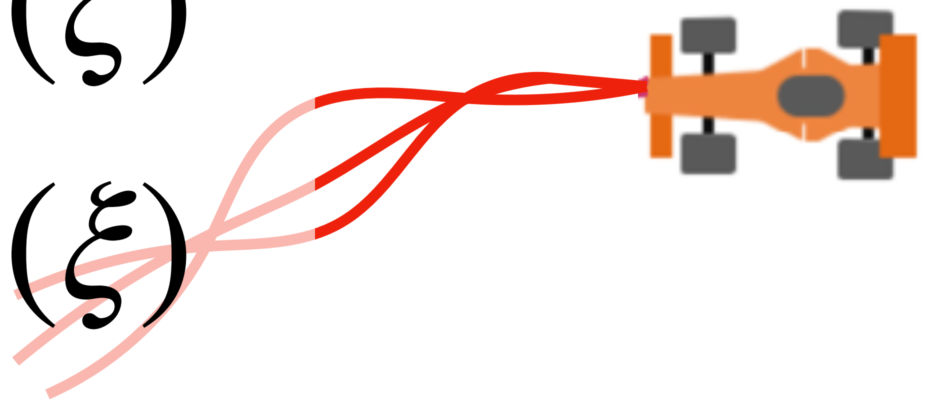


$$\mathbb{E}_{\xi^h \sim P_{expert}(.)}\, f_1(\xi^h) = \mathbb{E}_{\xi \sim P_\theta(.)}\, f_1(\xi)$$

$$\mathbb{E}_{\xi^h \sim P_{expert}(.)}\, f_2(\xi^h) = \mathbb{E}_{\xi \sim P_\theta(.)}\, f_2(\xi)$$

$$\vdots$$

$$\mathbb{E}_{\xi^h \sim P_{expert}(.)}\, f_k(\xi^h) = \mathbb{E}_{\xi \sim P_\theta(.)}\, f_k(\xi)$$



Let’s formalize!

Maximum Entropy Inverse Optimal Control

Maximum Entropy Inverse Reinforcement Learning

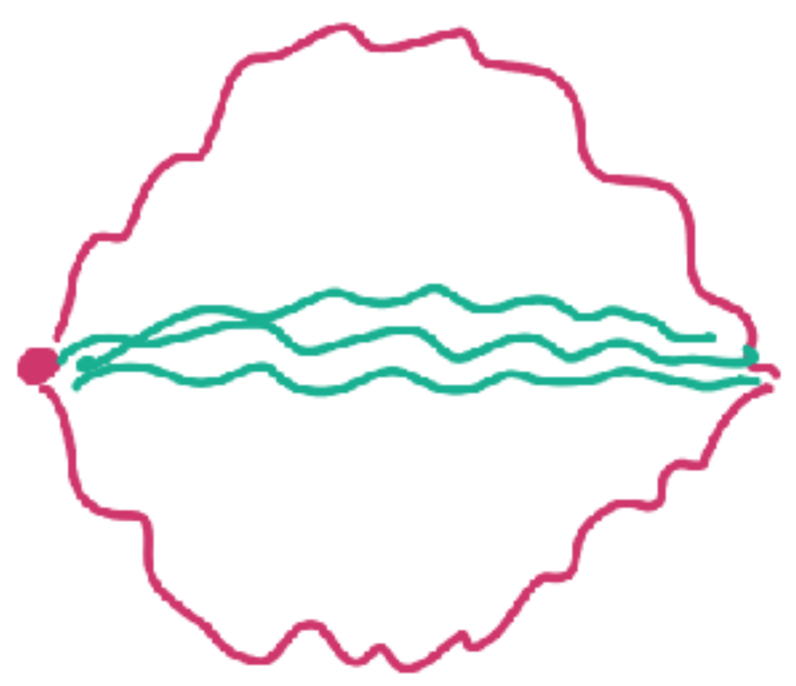
Brian D. Ziebart, Andrew Maas, J.Andrew Bagnell, and **Anind K. Dey**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
bziebart@cs.cmu.edu, amaas@andrew.cmu.edu, dbagnell@ri.cmu.edu, anind@cs.cmu.edu

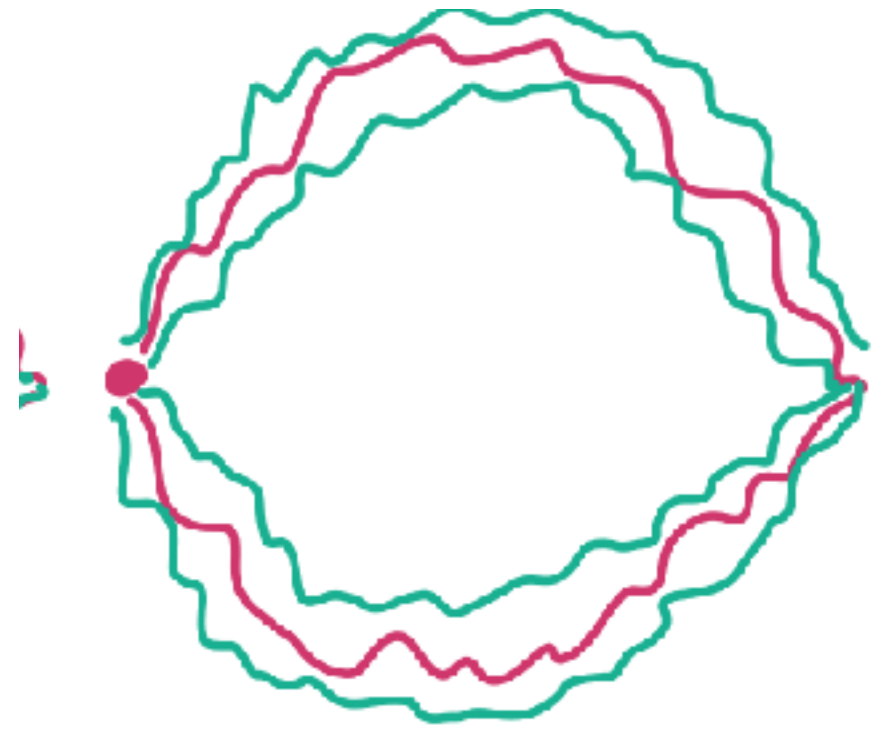
Maximum Entropy Inverse Optimal Control

LEO: Learning Energy-based Models in Factor Graph Optimization

Paloma Sodhi[1,2], Eric Dexheimer[1], Mustafa Mukadam[2], Stuart Anderson[2], Michael Kaess[1]
[1]Carnegie Mellon University, [2] Facebook AI Research

Maximum Entropy Inverse Optimal Control





Given dataset: $\{\xi_i^h, \phi_i\}_{i=1}^N$

(Human demo) *(Map)*

Solve for cost $C_\theta(\xi)$

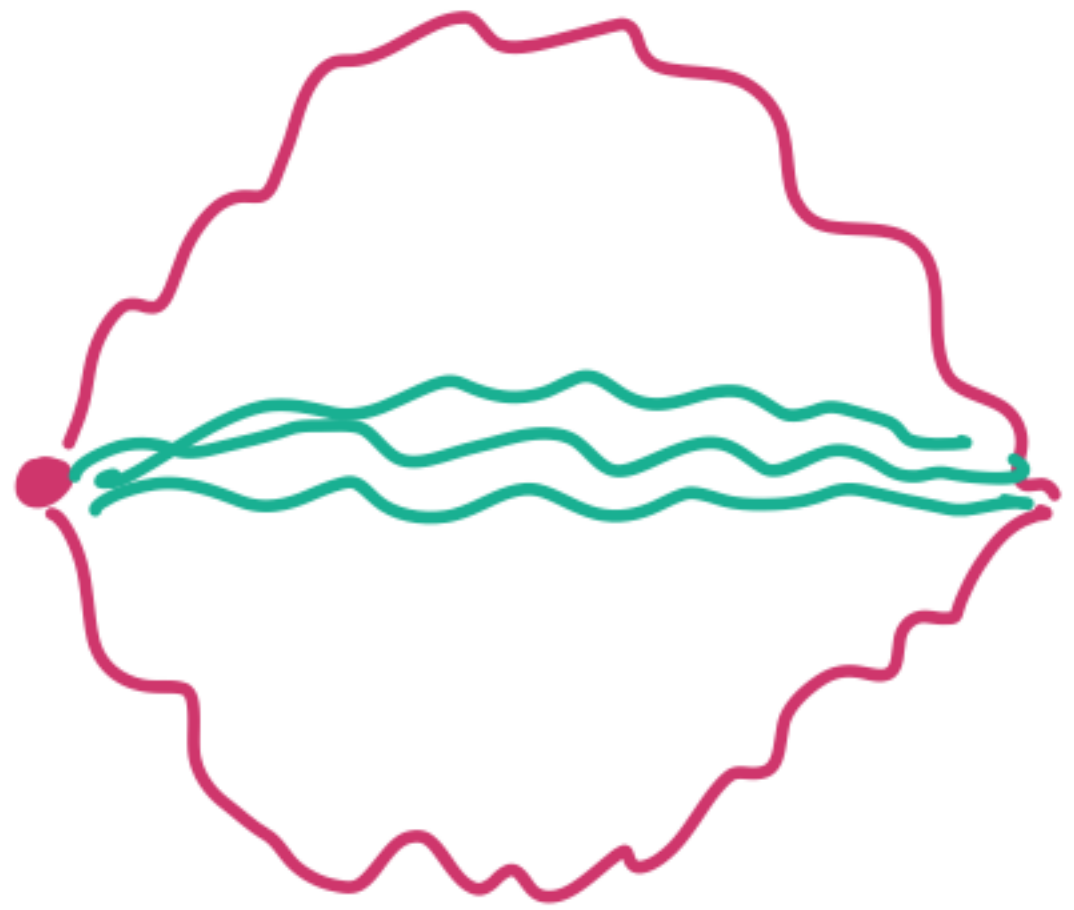
$$\min_\theta \frac{1}{N}\sum_{i=1}^{N} -\log P_\theta(\xi_i^h \mid \phi_i)$$

Max lik. of human traj

$$P_\theta(\xi \mid \phi) = \frac{1}{Z(\theta, \phi)} \exp(-C_\theta(\xi, \phi))$$

More costly traj, less likely

Maximum Entropy Inverse Optimal Control



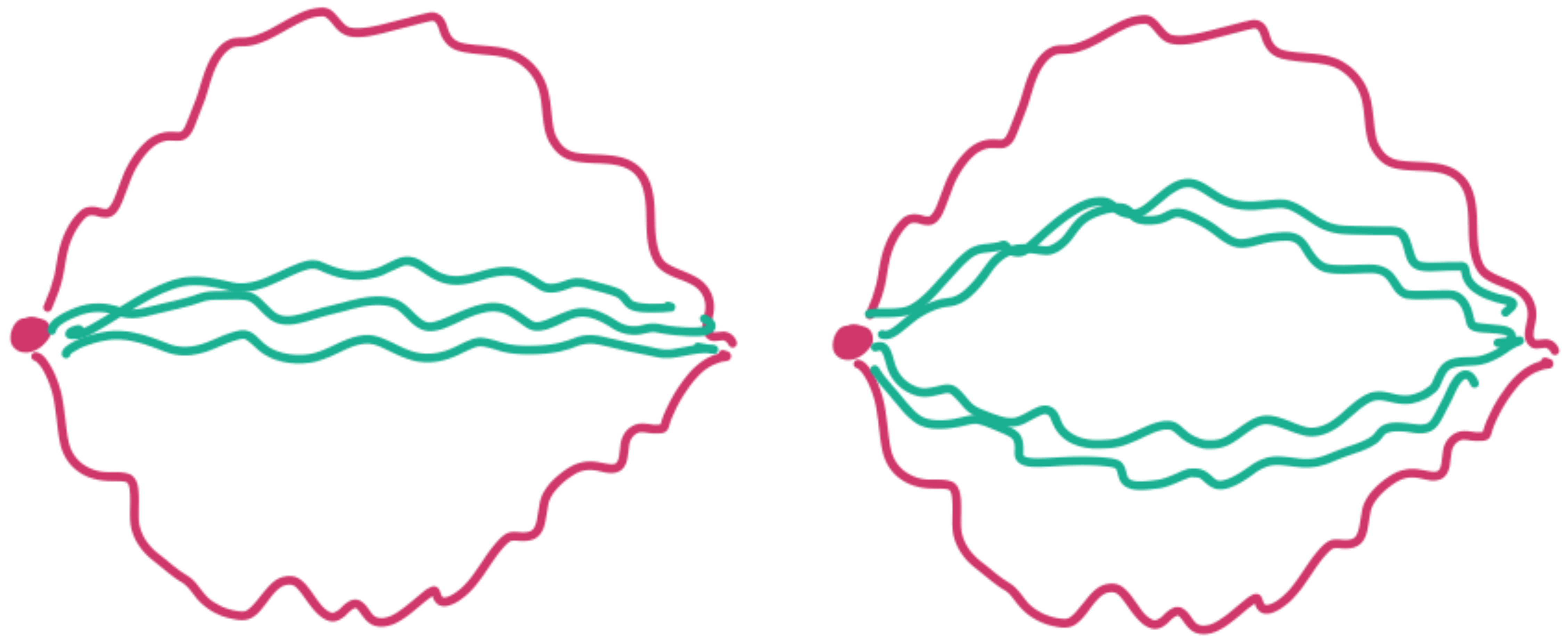
for $i = 1, \ldots, N$ # Loop over datapoints

$$\xi_i \sim \frac{1}{Z} \exp\left(-C_\theta(\xi, \phi_i)\right)$$ # Call planner!

$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i, \phi_i)]$$ # Update cost

(Push down human cost) *(Push up planner cost)*

Maximum Entropy Inverse Optimal Control



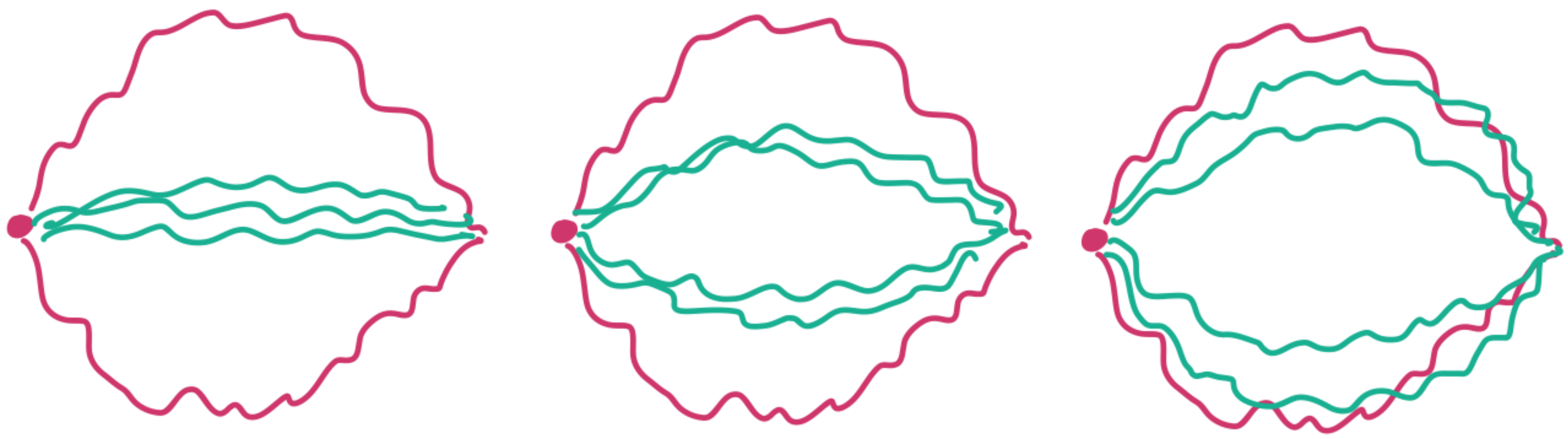
for $i = 1, \ldots, N$ # Loop over datapoints

$$\xi_i \sim \frac{1}{Z} \exp\left(-C_\theta(\xi, \phi_i)\right)$$ # Call planner!

$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i, \phi_i)]$$ # Update cost

(Push down human cost) *(Push up planner cost)*

Maximum Entropy Inverse Optimal Control



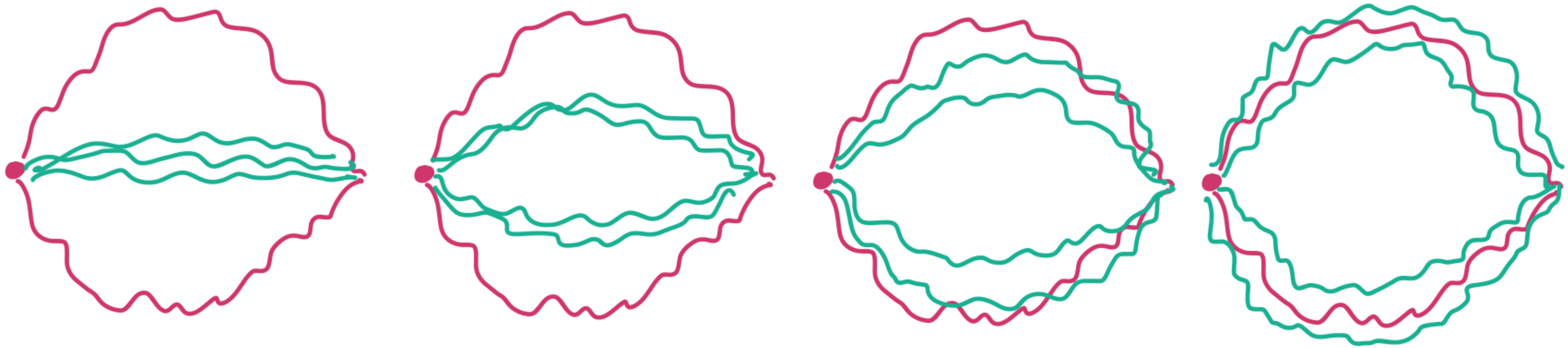
for $i = 1, \ldots, N$ # Loop over datapoints

$$\xi_i \sim \frac{1}{Z} \exp\left(-C_\theta(\xi, \phi_i)\right)$$ # Call planner!

$$\theta^+ = \theta - \eta[\nabla_\theta C_\theta(\xi_i^h, \phi_i) - \nabla_\theta C_\theta(\xi_i, \phi_i)]$$ # Update cost

(Push down human cost) *(Push up planner cost)*

Maximum Entropy Inverse Optimal Control



for $i = 1,\ldots,N$ # Loop over datapoints

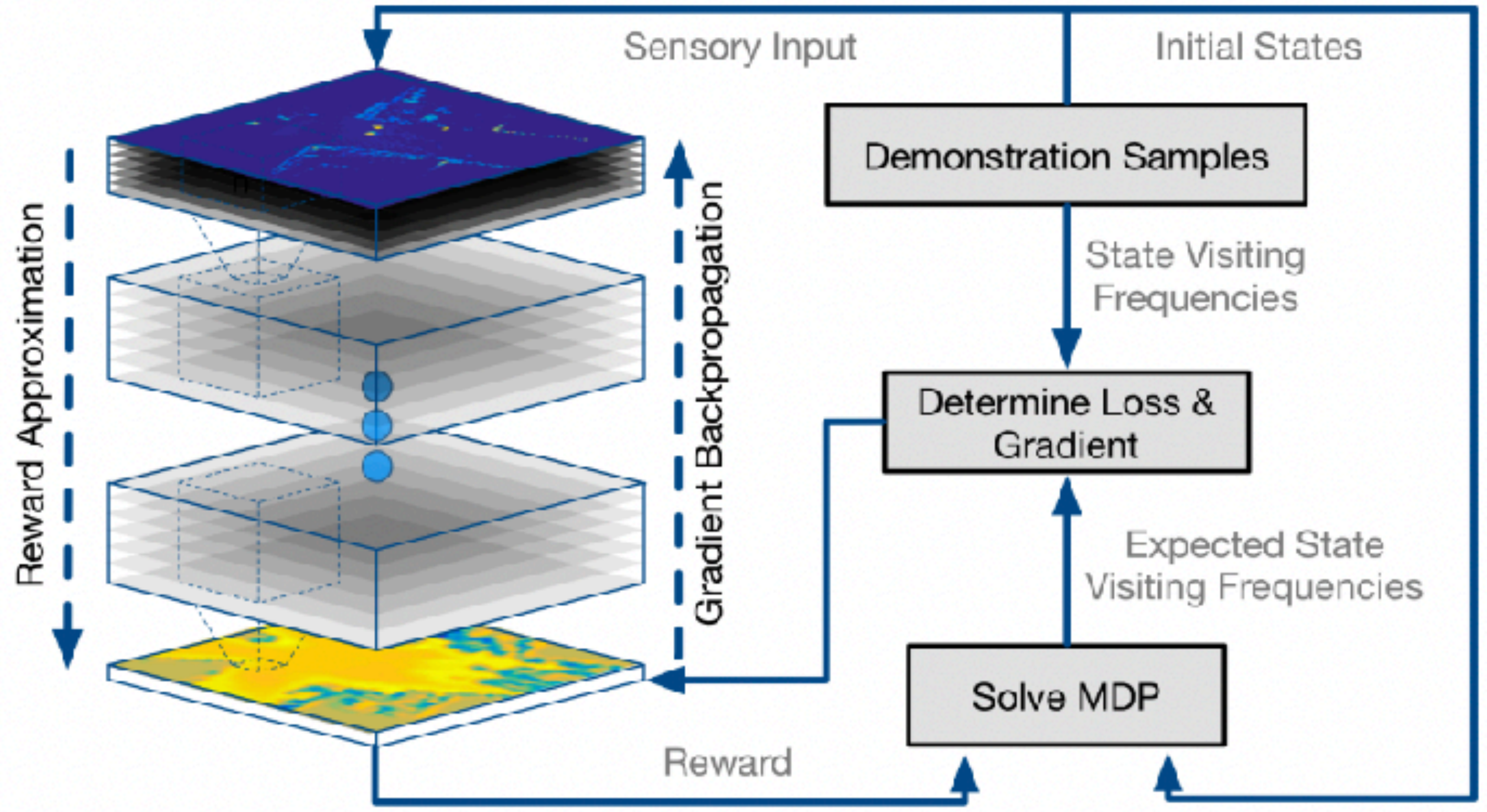
$$\xi_i \sim \frac{1}{Z}\exp\left(-C_\theta(\xi,\phi_i)\right)$$ # Call planner!

$$\theta^+ = \theta - \eta[\underbrace{\nabla_\theta C_\theta(\xi_i^h,\phi_i)}_{\text{(Push down human cost)}} - \underbrace{\nabla_\theta C_\theta(\xi_i,\phi_i)}_{\text{(Push up planner cost)}}]$$ # Update cost

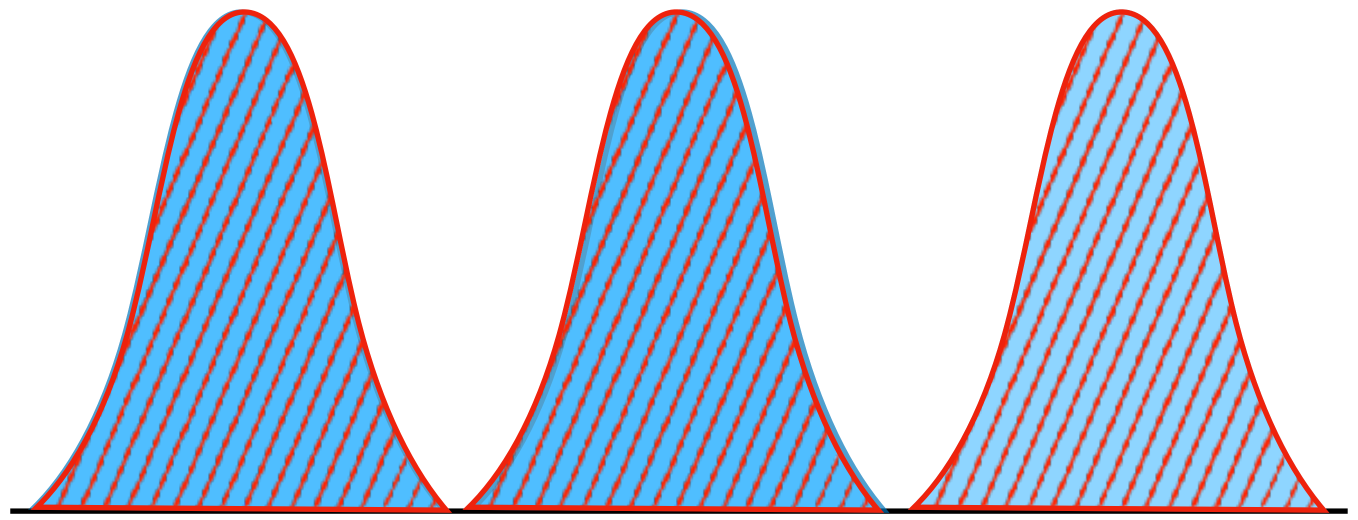
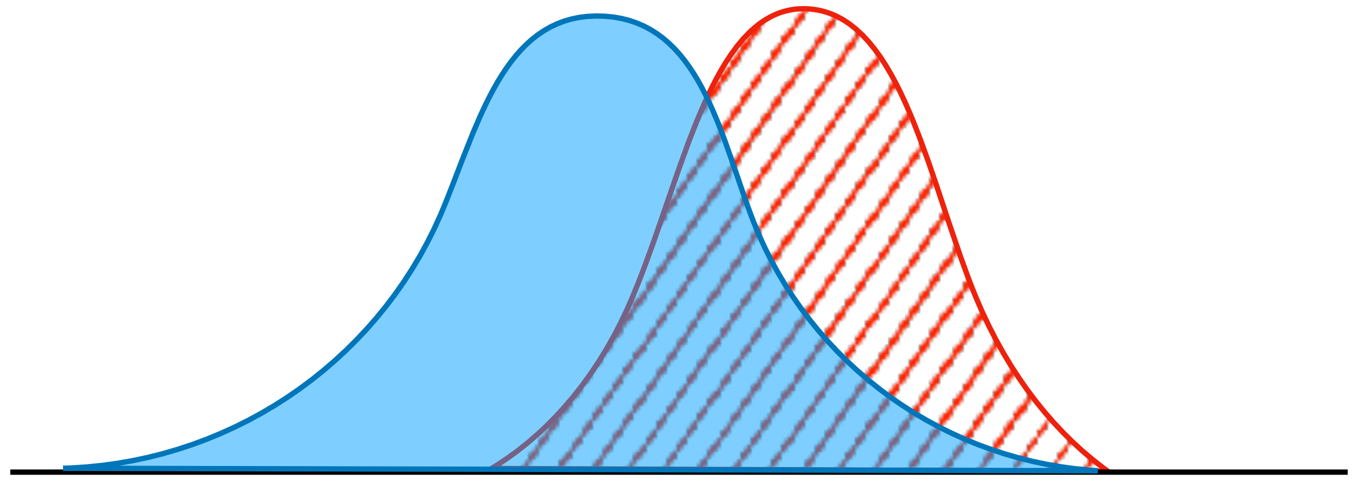
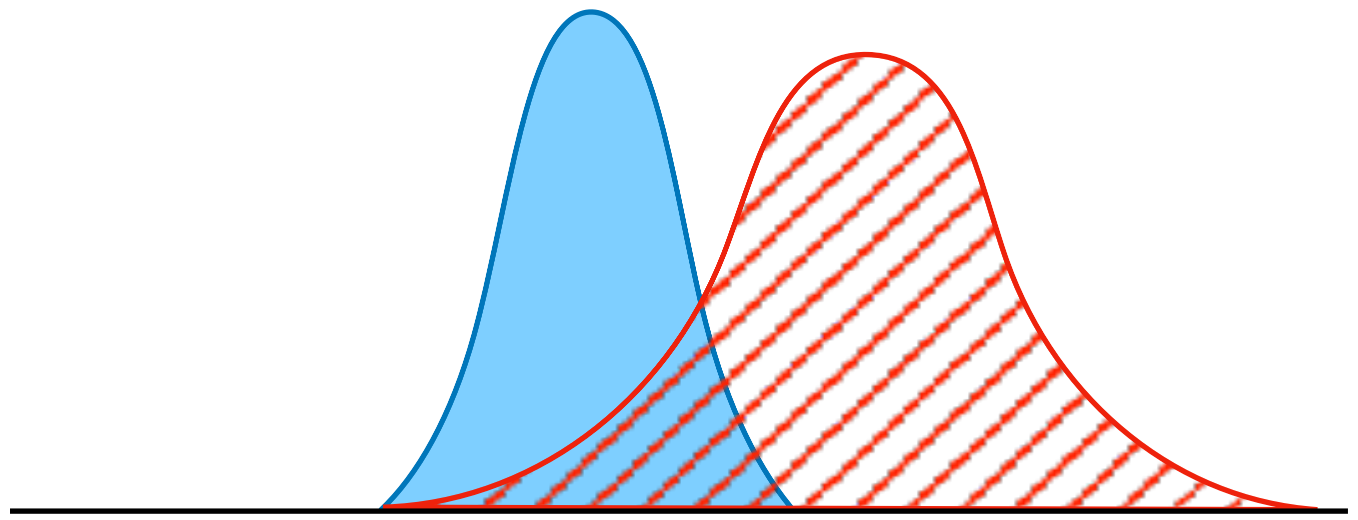
Deep Max Ent

Watch This: Scalable Cost-Function Learning for Path Planning in Urban Environments

Markus Wulfmeier[1], Dominic Zeng Wang[1] and Ingmar Posner[1]

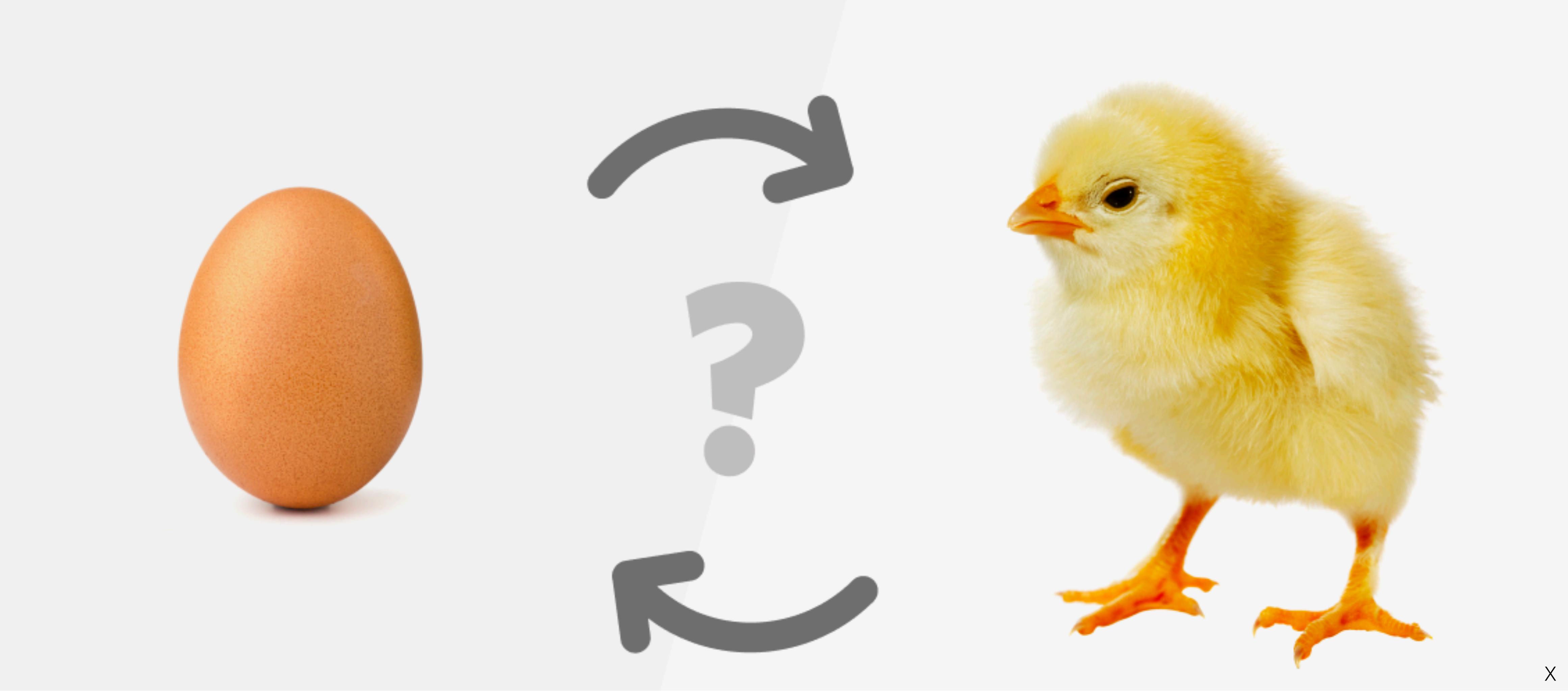


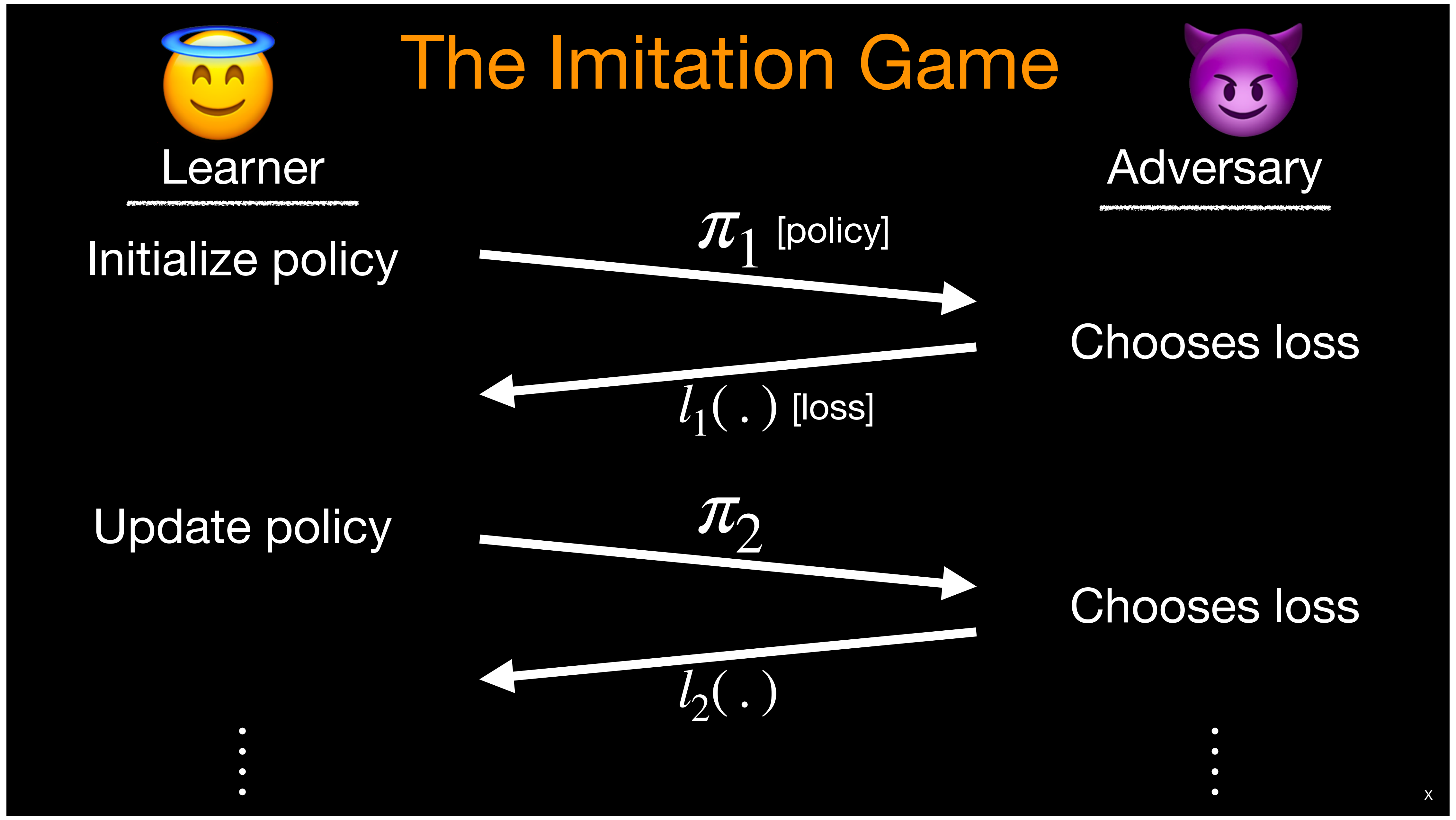

autonomous execution
1x real-time
PR2
goal
our method
IOC samples from q(u_t|x_t)

	Easy 😄	**Medium** 🤔	**Hard** 😱
Setting	Expert is realizable $\pi^E \in \Pi$ As $N \to \infty$, drive down $\epsilon = 0$ (or Bayes error)	Non-realizable expert but full expert support Even as $N \to \infty$, behavior cloning $O(\epsilon C T)$ where C is conc. coeff	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	Requires interactive simulator (MaxEntIRL) to match distribution $\Rightarrow O(\epsilon T)$	Requires interactive expert (DAGGER / EIL) to provide labels $\Rightarrow O(\epsilon T)$

tl;dr

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





[Ross et al'11]

DAgger: Iteration 1