Happy Halloween!

Principle of Maximum Entropy in Decision Making (From IRL to RL and back)

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Maximum Entropy Inverse Reinforcement Learning

How do we imitate noisy / suboptimal experts?

Collect dataset $\mathscr{D} = \{\xi_i^h\}$ of expert trajectories

Update cost / reward function doing gradient descent on :

$$\mathbb{E}_{\xi_i^h \sim \mathscr{D}} \nabla_\theta C_\theta(\xi_i^h) - \mathbb{E}_{\xi_i \sim \frac{1}{Z}\exp\left(-C_\theta(\xi)\right)} \nabla_\theta C_\theta(\xi_i)$$

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

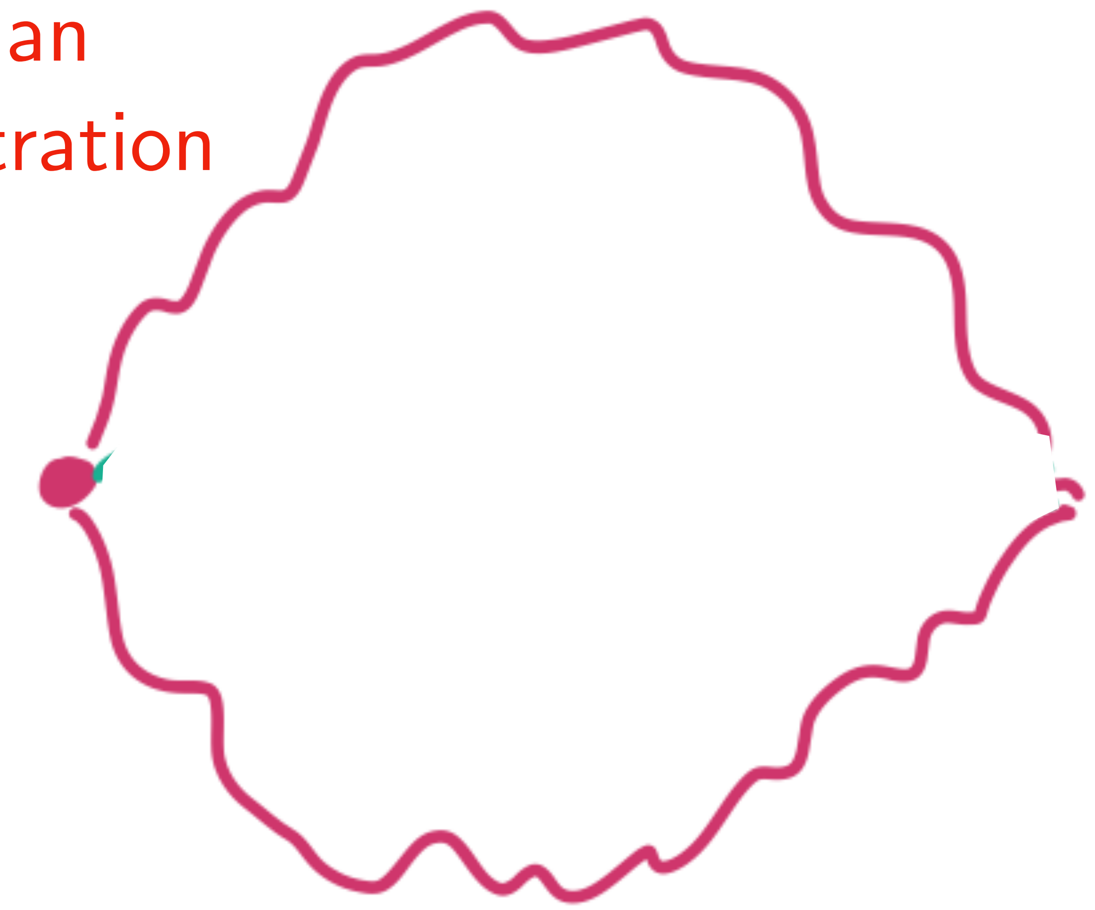
How do we sample from

$$\xi \sim \frac{1}{Z} \exp\left(-C_\theta(\xi)\right)$$

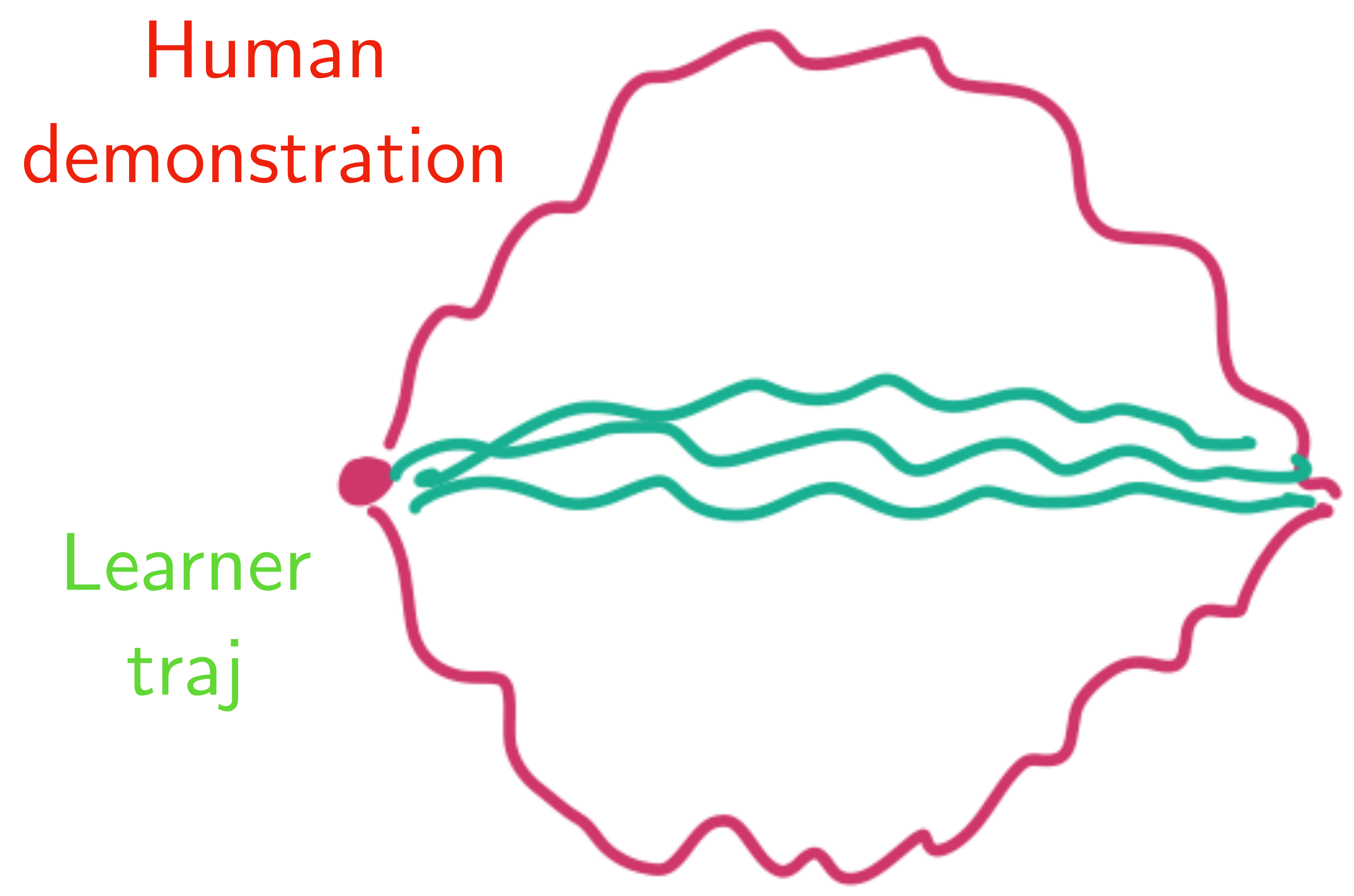
Is it intuitively like calling a planner?

Maximum Entropy Inverse Reinforcement Learning

Human demonstration



Maximum Entropy Inverse Reinforcement Learning



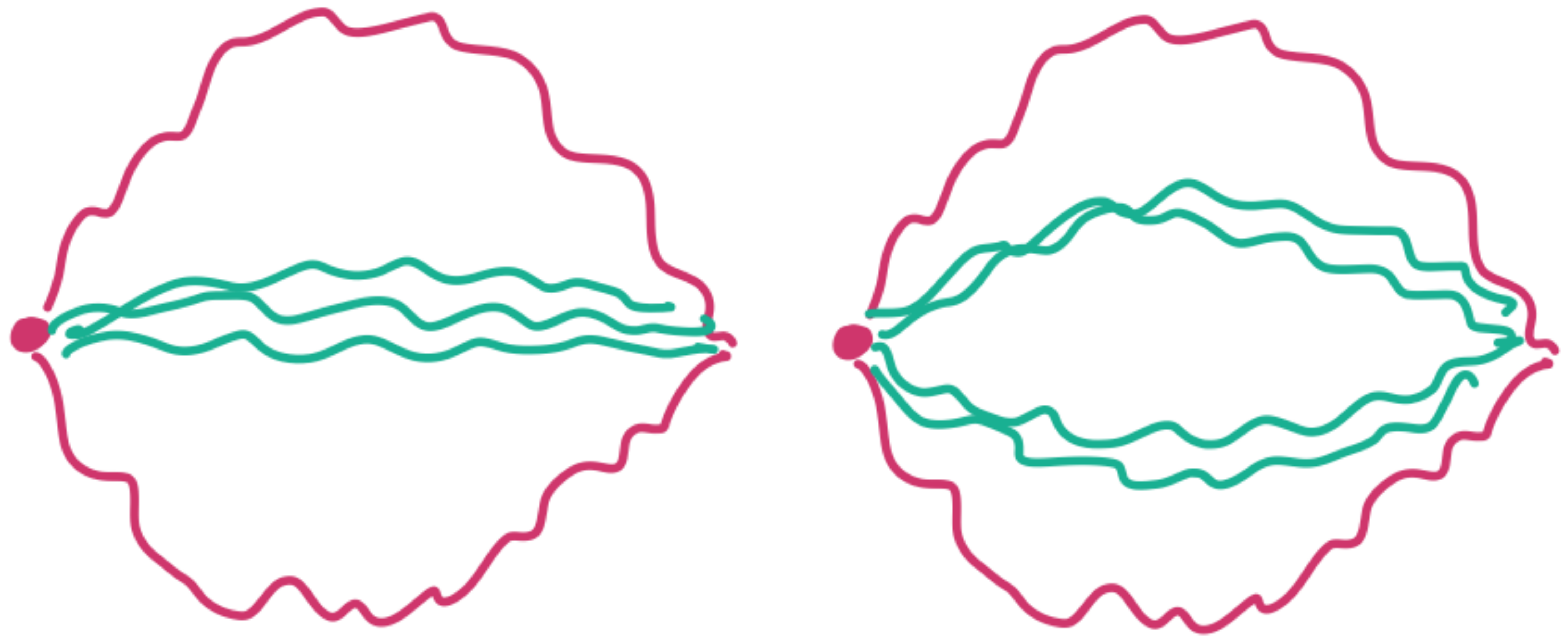
for $i = 1, \ldots, N$ # Loop over datapoints

$$\xi_i \sim \frac{1}{Z} \exp\left(-C_\theta(\xi)\right)$$ # Call "Soft" Planner

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

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

Maximum Entropy Inverse Reinforcement Learning



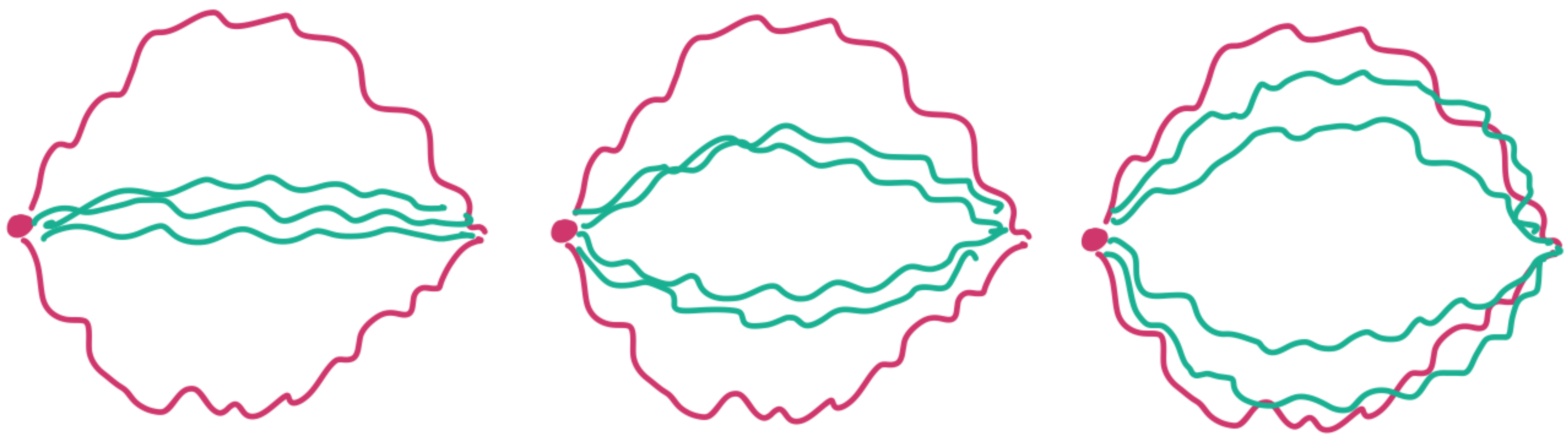
for $i = 1, \ldots, N$ # Loop over datapoints

$$\xi_i \sim \frac{1}{Z} \exp\left(-C_\theta(\xi)\right)$$ # Call "Soft" Planner

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

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

Maximum Entropy Inverse Reinforcement Learning



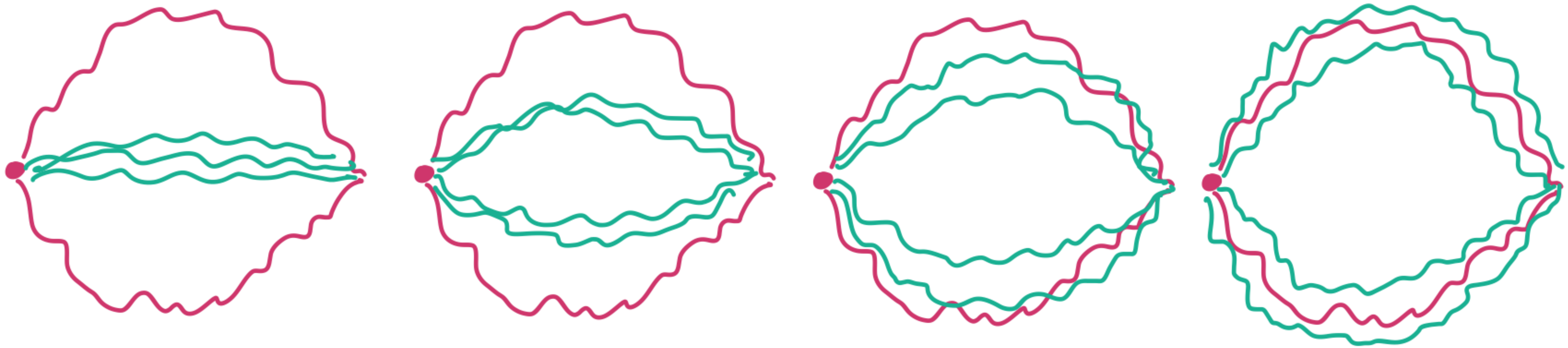
for $i = 1,\ldots,N$ # Loop over datapoints

$$\xi_i \sim \frac{1}{Z}\exp\left(-C_\theta(\xi)\right)$$ # Call "Soft" Planner

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

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

Maximum Entropy Inverse Reinforcement Learning



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

$$\xi_i \sim \frac{1}{Z} \exp\left(-C_\theta(\xi)\right)$$ # Call "Soft" Planner

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

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

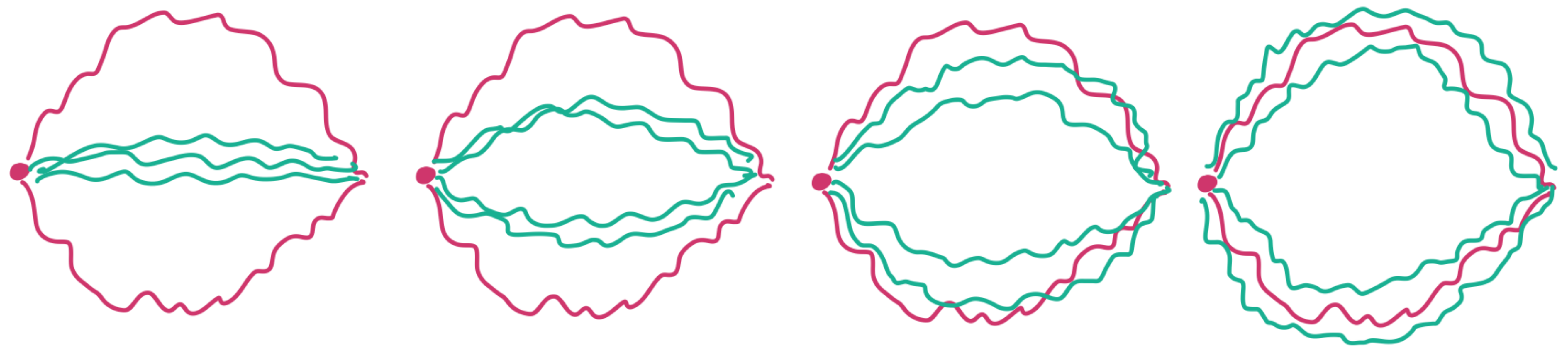
Activity!

Think-Pair-Share

Think (30 sec): What if we called a hard/optimal planner rather than a soft planner, i.e. $\xi_i = \arg\min C_\theta(\xi)$

Would you converge?

Pair: Find a partner



Share (45 sec): Partners exchange ideas

Okay...

But how do we *actually* sample from

$$\xi \sim \frac{1}{Z} \exp\left(-C_\theta(\xi)\right)$$

Let's derive soft value iteration!

How do we do soft value iteration with deep networks?

Soft Actor Critic

1. Q-function update

Update Q-function to evaluate current policy:

$$Q(\mathbf{s},\mathbf{a}) \leftarrow r(\mathbf{s},\mathbf{a}) + \mathbb{E}_{\mathbf{s}'\sim p_{\mathbf{s}},\ \mathbf{a}'\sim\pi}\left[Q(\mathbf{s}',\mathbf{a}') - \log\pi(\mathbf{a}'|\mathbf{s}')\right]$$

This converges to Q^{π}.

2. Update policy

Update the policy with gradient of information projection:

$$\pi_{\text{new}} = \arg\min_{\pi'} \mathrm{D}_{\mathrm{KL}}\left(\pi'(\,\cdot\,|\mathbf{s})\,\middle\|\,\frac{1}{Z}\exp Q^{\pi_{\text{old}}}(\mathbf{s},\,\cdot\,)\right)$$

In practice, only take one gradient step on this objective

3. Interact with the world, collect more data

"Soft" Critic

Recall Nightmare!

Haarnoja, Zhou, Hartikainen, Tucker, Ha, Tan, Kumar, Zhu, Gupta, Abbeel, L. **Soft Actor-Critic Algorithms and Applicatic**

Credit S.Levine.

Back to Inverse Reinforcement Learning

(But with deep networks)

Maximum Entropy Inverse Reinforcement Learning

$R_\theta(s, a) \longrightarrow$ $\boxed{\text{Soft Actor Critic}}$ $\longrightarrow \pi(a \mid s)$

Roll-out π to collect trajectory $\xi = \{s_0, a_0, \ldots .\}$

$$\theta^+ = \theta + \eta[\nabla_\theta R_\theta(\xi_i^h) - \nabla_\theta R_\theta(\xi_i)]$$

MaxEntIRL has had many success stories over the years and been rediscovered a lot of times

Navigate Like a Cabbie: Probabilistic Reasoning from Observed Context-Aware Behavior

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

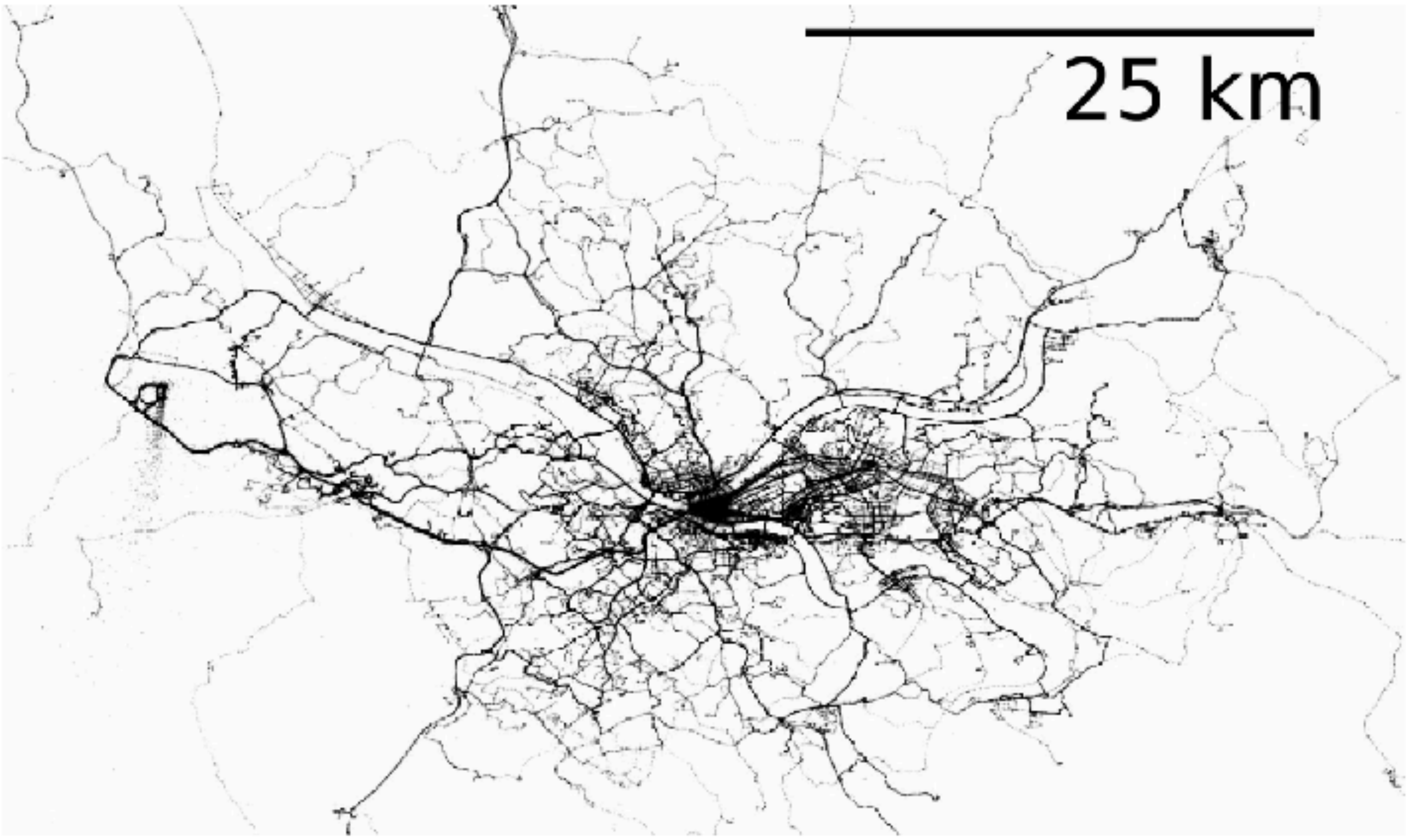


Figure 4. The collected GPS datapoints

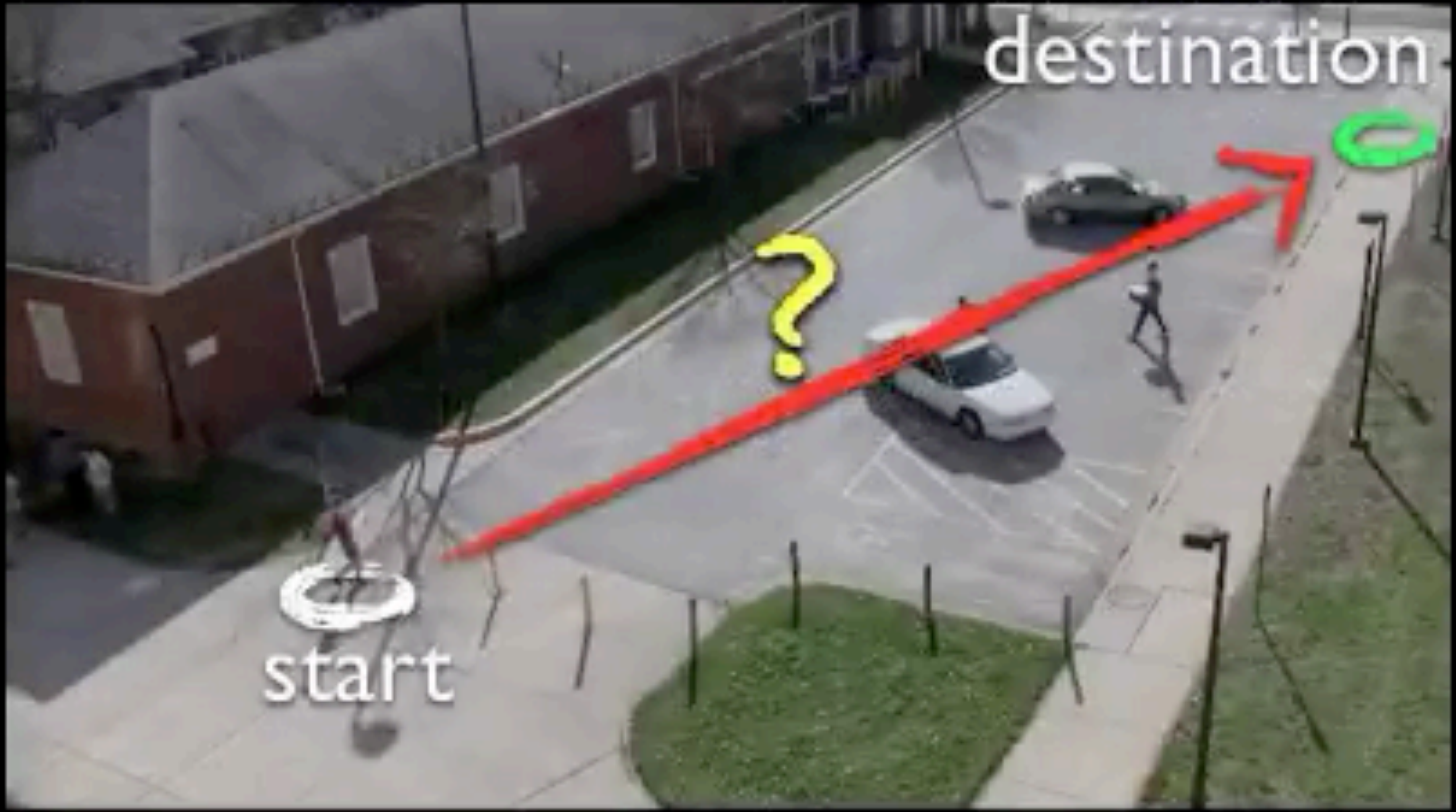
ABSTRACT

We present *PROCAB*, an efficient method for Probabilistically Reasoning from Observed Context-Aware Behavior. It models the context-dependent utilities and underlying reasons that people take different actions. The model generalizes to unseen situations and scales to incorporate rich contextual information. We train our model using the route preferences of 25 taxi drivers demonstrated in over 100,000 miles of collected data, and demonstrate the performance of our model by inferring: (1) decision at next intersection, (2) route to known destination, and (3) destination given partially traveled route.

Activity Forecasting

Kris M. Kitani, Brian D. Ziebart, J. Andrew Bagnell, and Martial Hebert

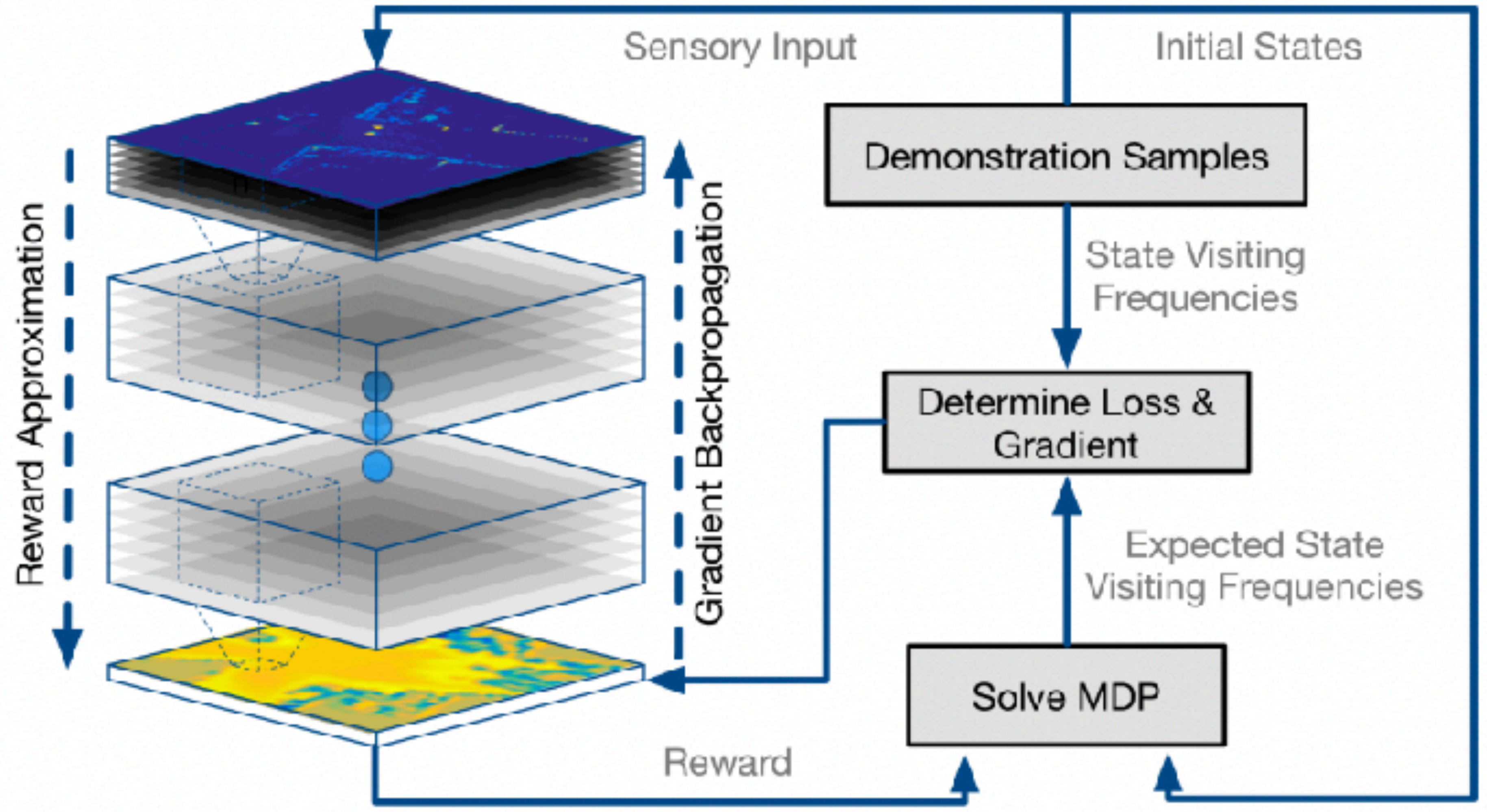
Carnegie Mellon University, Pittsburgh, PA 15213 USA
{kkitani,bziebart}@cs.cmu.edu, {dbagnell,hebert}@ri.cmu.edu



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]



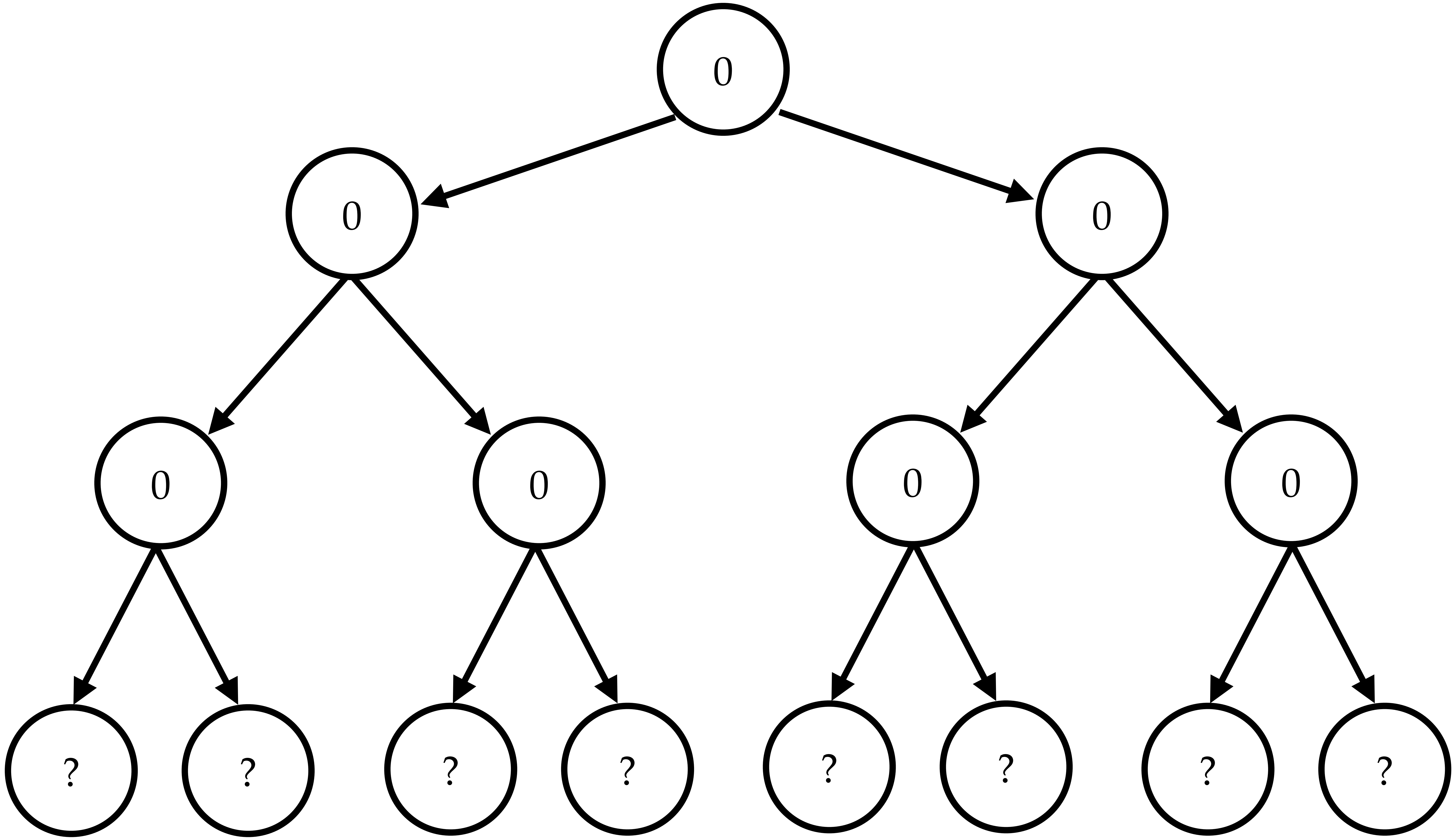
Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization

Chelsea Finn CBFINN@EECS.BERKELEY.EDU
Sergey Levine SVLEVINE@EECS.BERKELEY.EDU
Pieter Abbeel PABBEEL@EECS.BERKELEY.EDU
University of California, Berkeley, Berkeley, CA 94709 USA

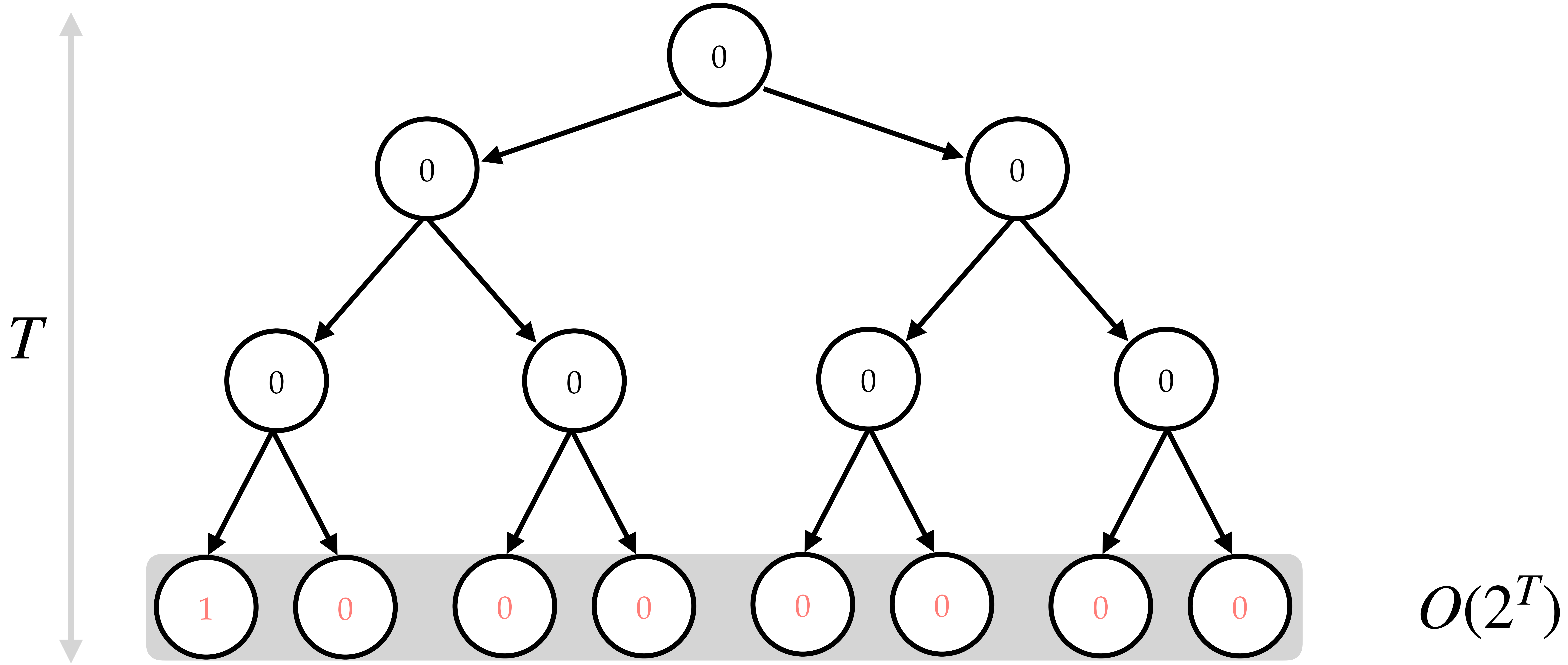
Is IRL running a RL algorithm in the inner loop ?!?

Won't that take very long??

Complexity of IRL for a tree MDP?



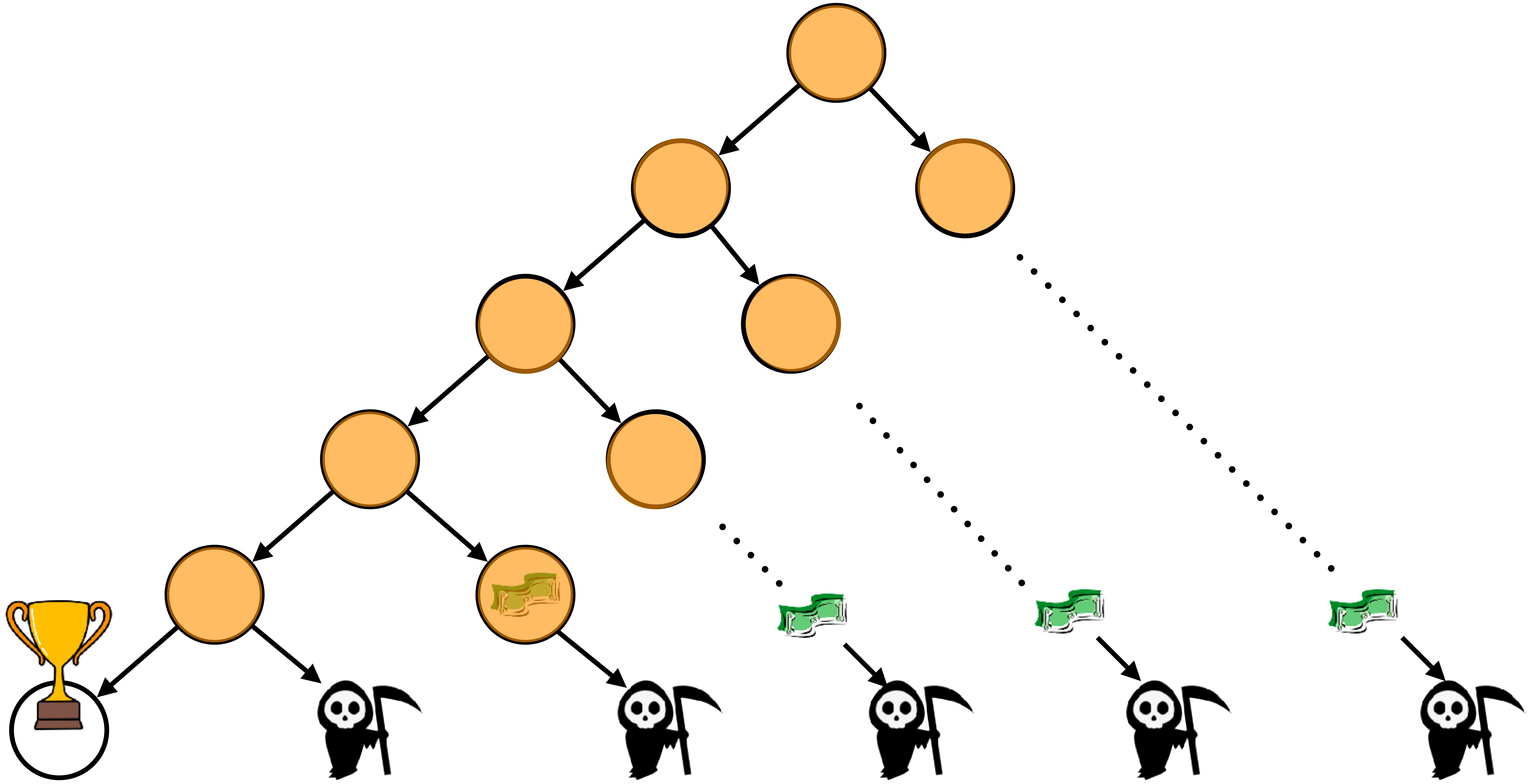
Complexity of IRL for a tree MDP?



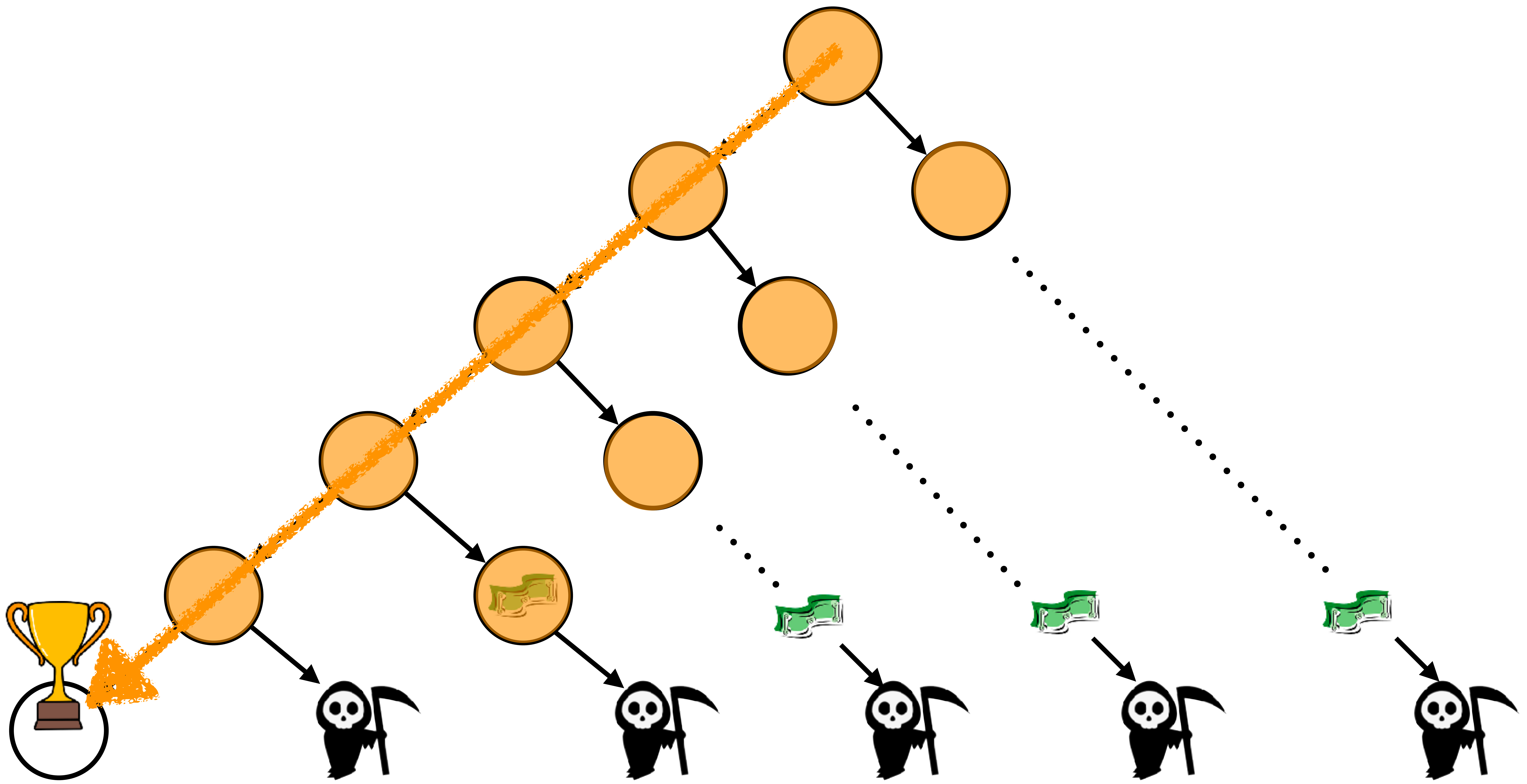
We have seen this movie before ...


RL is like finding a
needle in an exponential haystack

RL is exp(T)!



RL is exp(T)!



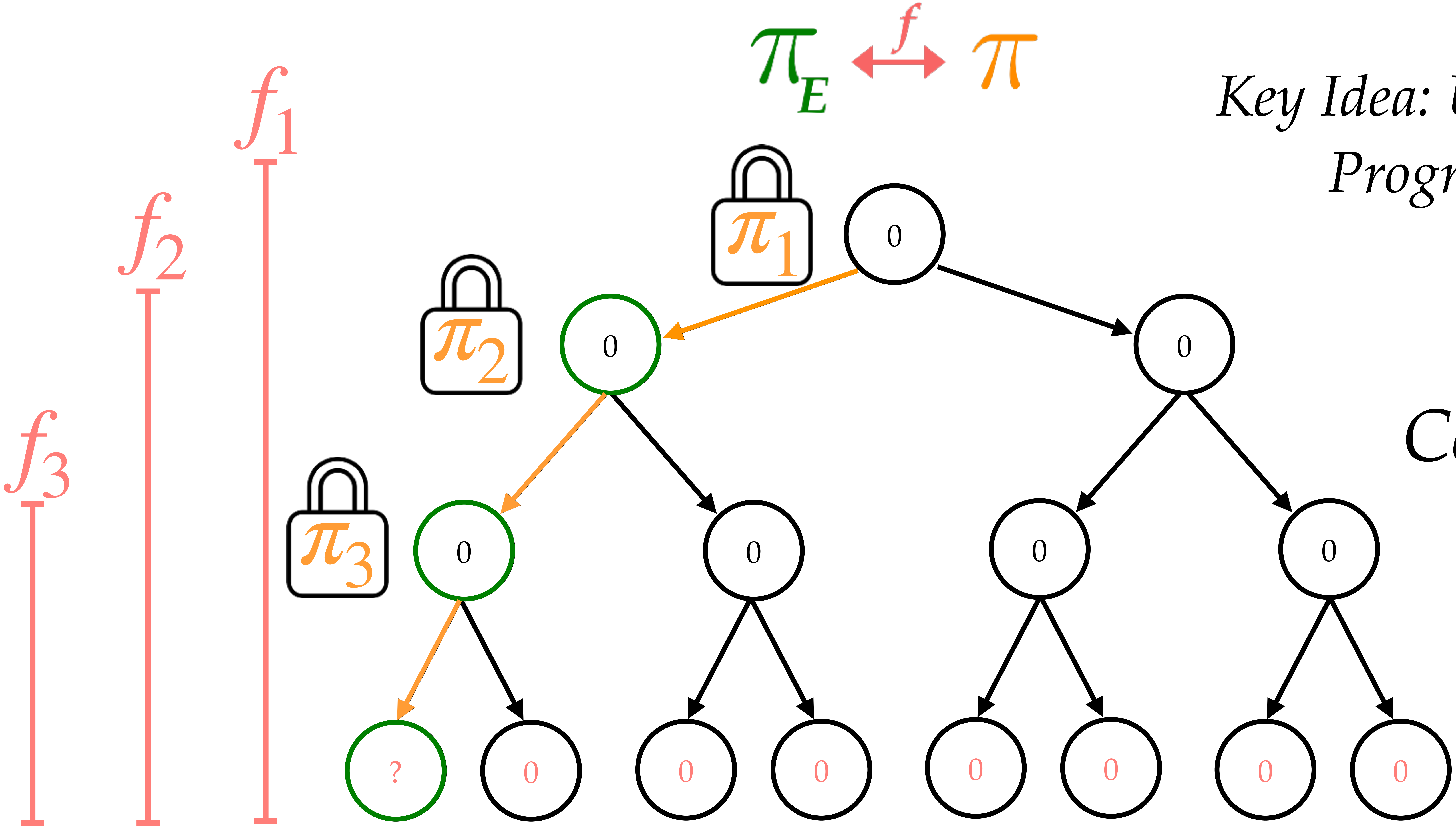
🔑 ***Insight****: We can reset the learner to states from the expert demonstrations to reduce unnecessary exploration.*

Inverse Reinforcement Learning without Reinforcement Learning




(Gokul Swamy, Sanjiban Choudhury, Drew Bagnell, and Steven Wu)

Speeding up IRL with Expert Resets

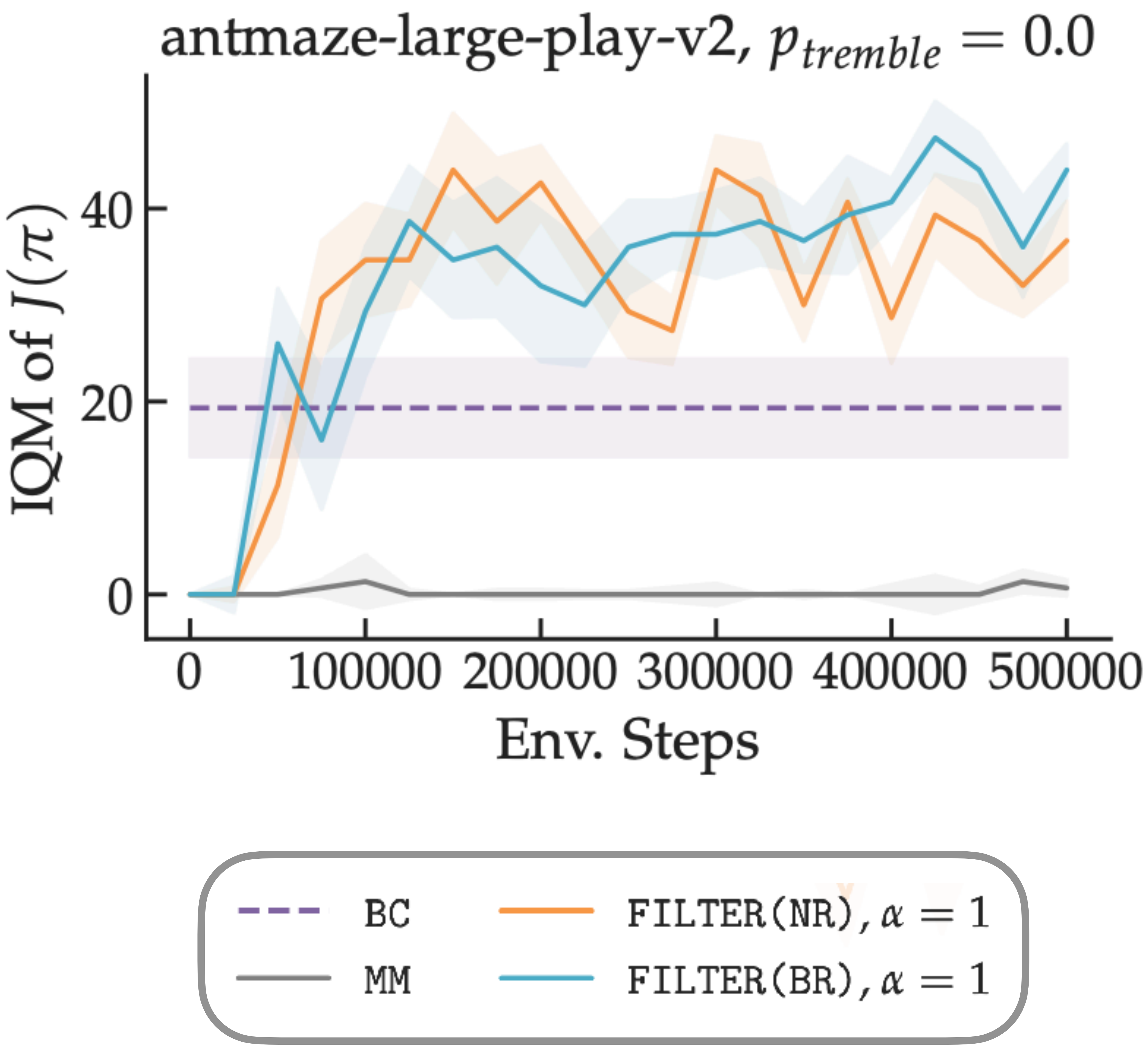


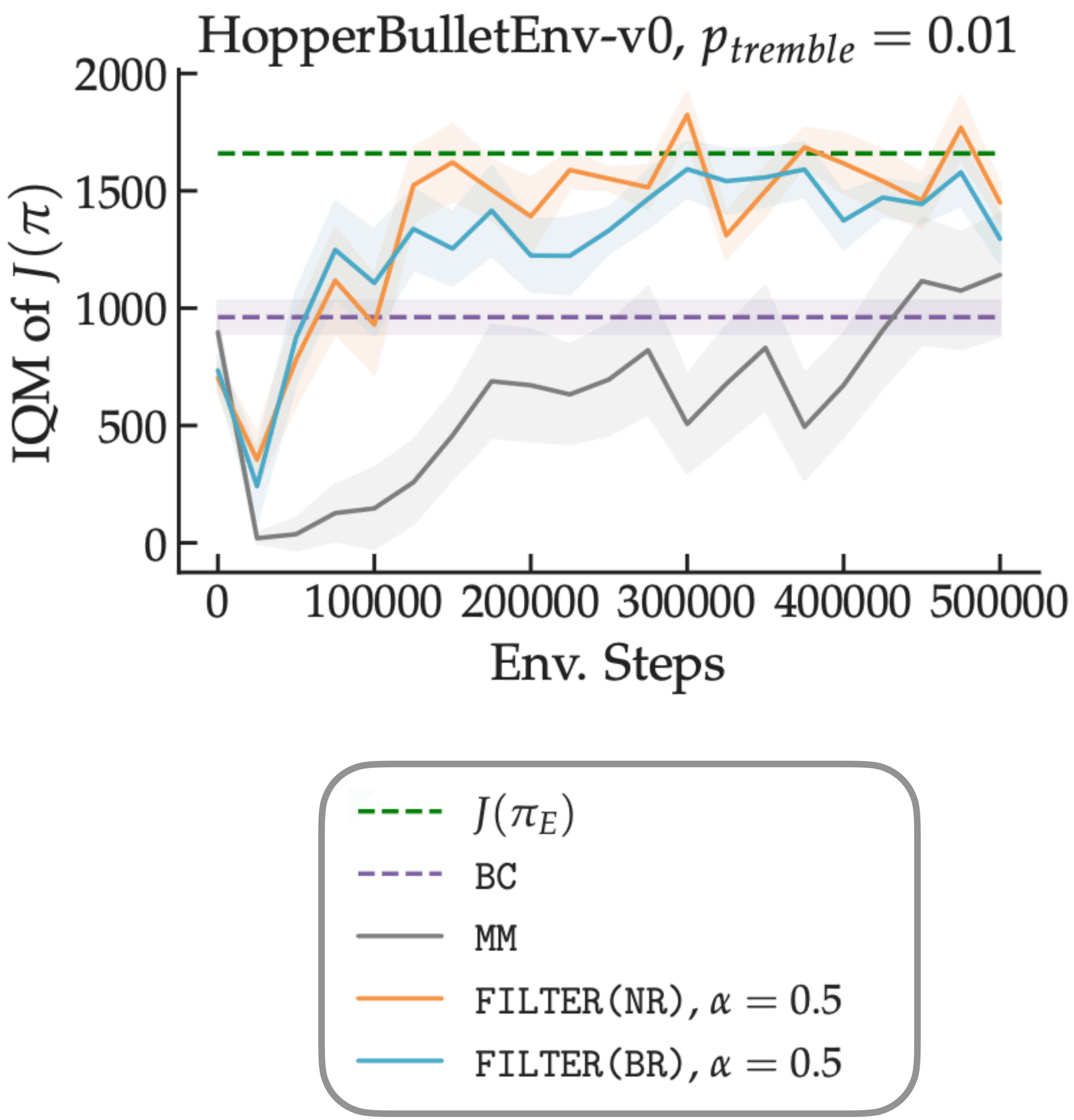
Key Idea: Use Dynamic Programming

$O(T^2)$

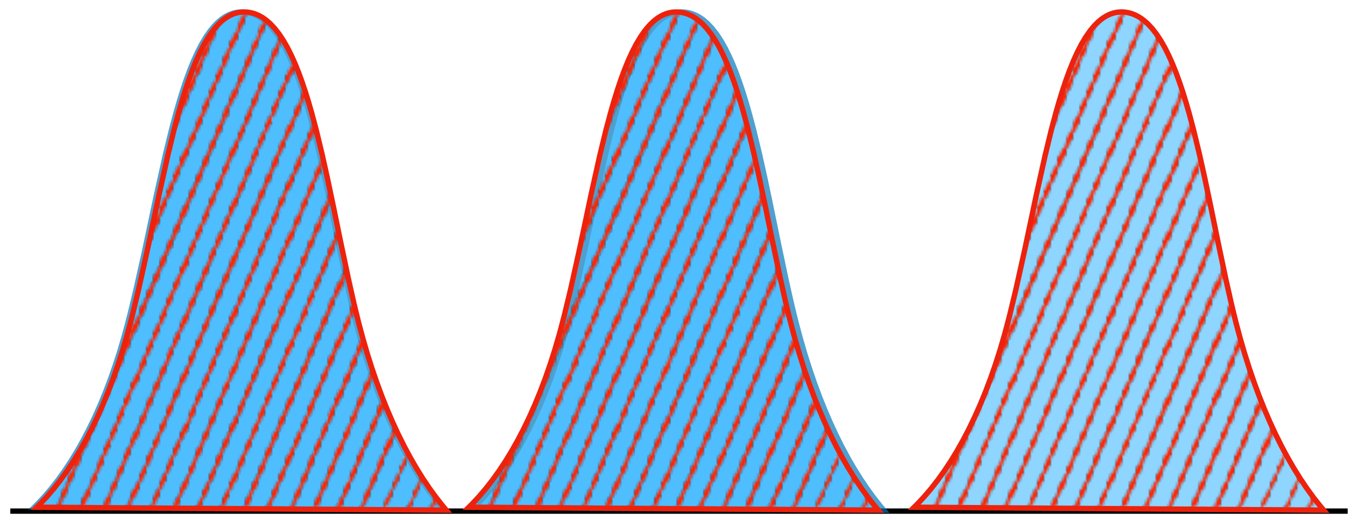
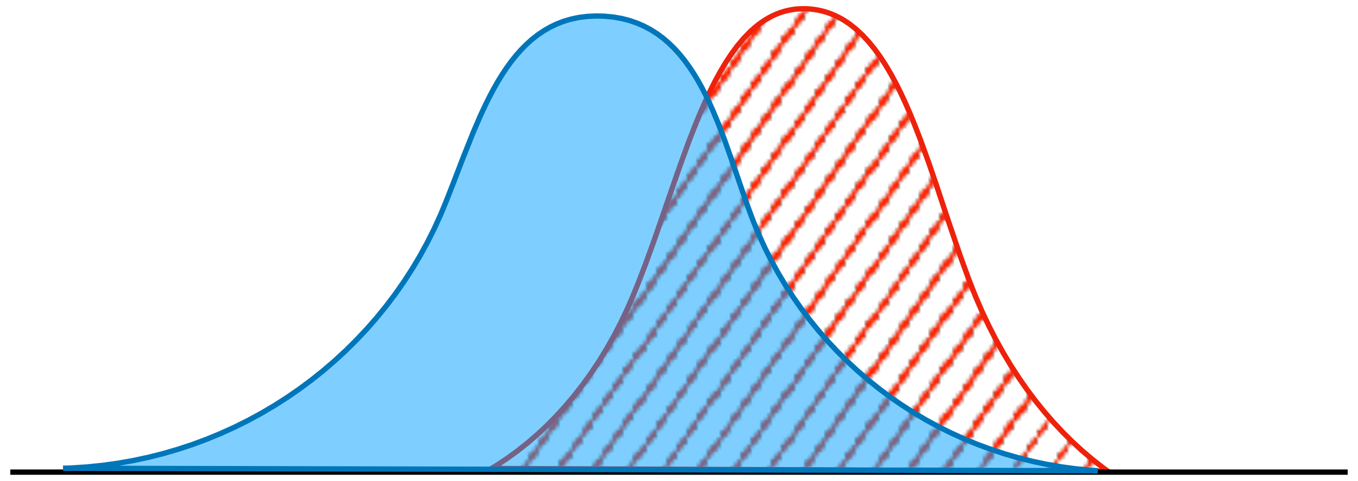
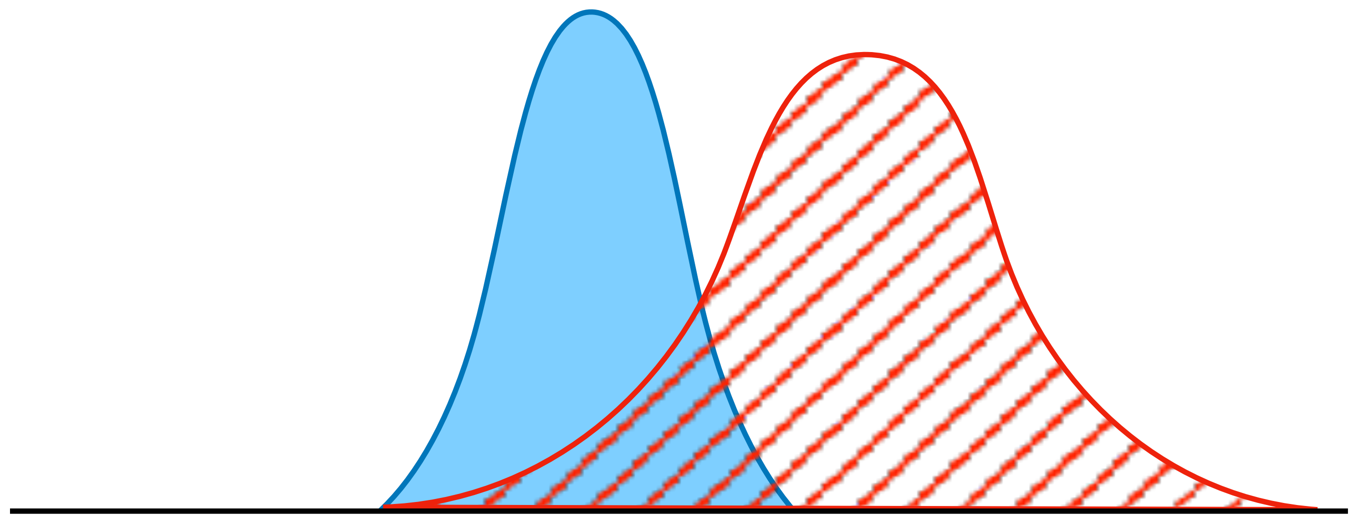
Complexity!

Expert Resets Speed Up IRL





The BIG Picture!

	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 $\Rightarrow O(\epsilon T)$