DAgger: Taming Covariate Shift with No Regret

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

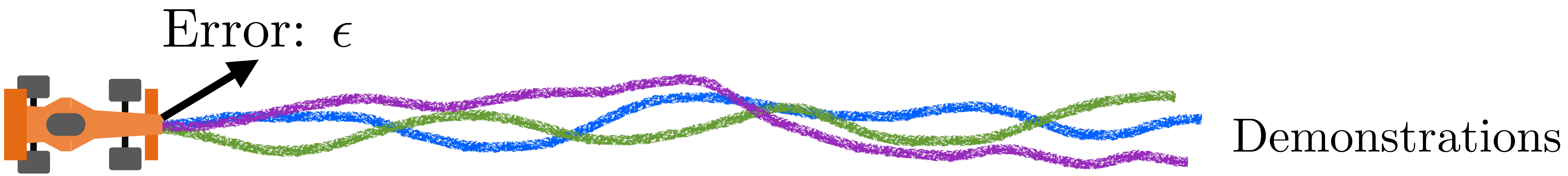
Behavior Cloning crashes into a wall

Train $\neq$ Test

Train Test Mismatch

Training / Validation Loss

$$\sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi^\star}} \underbrace{[\ell(s_t, \pi(s_t))]}_{\epsilon} \qquad O(\epsilon T)$$



$\neq$

Test Loss

$$\sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi}}[\ell(s_t, \pi(s_t))] \qquad O(\epsilon T^2)$$

Can we mathematically quantify how much worse BC is compared to the demonstrator?

First, let's define performance of a policy

$$J(\pi) = \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(s_t) \\ s_{t+1} \sim \mathcal{T}(s_t, a_t)}} \left[\sum_{t=0}^{T-1} c(s_t, a_t) \right]$$

(Performance)

Second, let's define performance difference

$$J(\pi) - J(\pi^*)$$

(Performance of my learner) (Performance of my demonstrator)

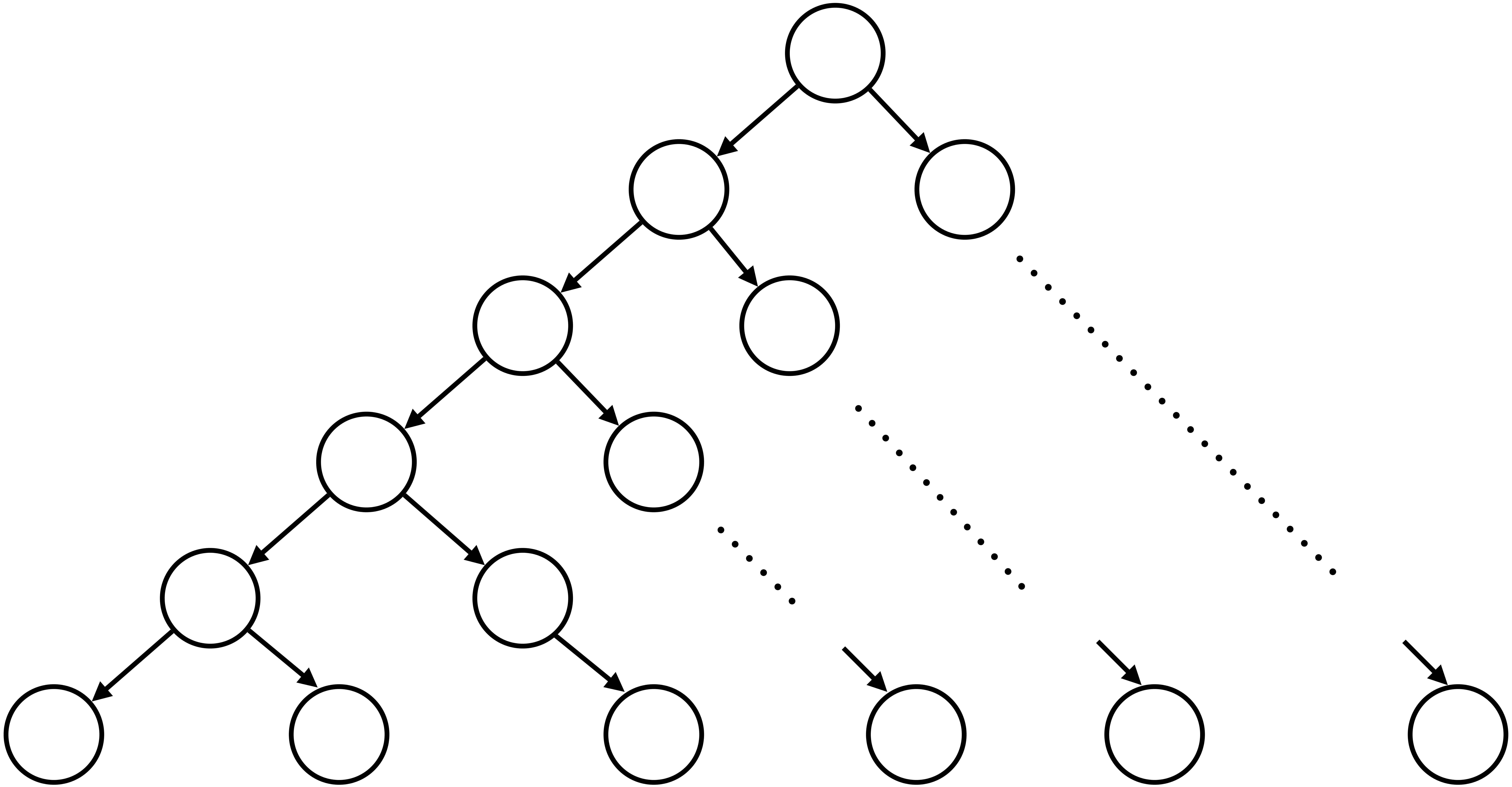
We want to *minimize* the performance difference

Behavior cloning hits the worst case!

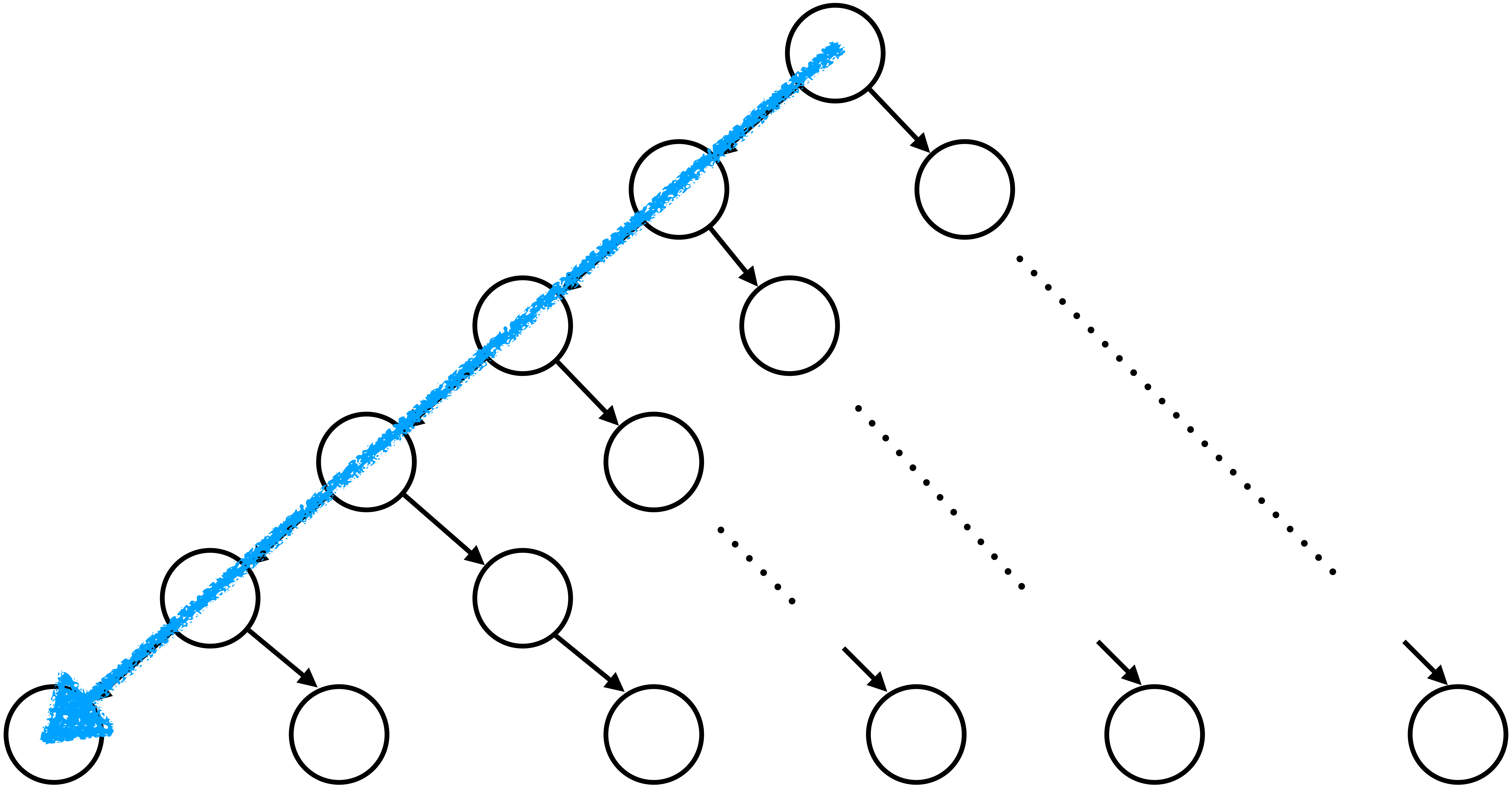
There exists an MDP where BC
has a performance difference of $O(\epsilon T^2)$

We are going to such a MDP right now,
and you will see more in A1!

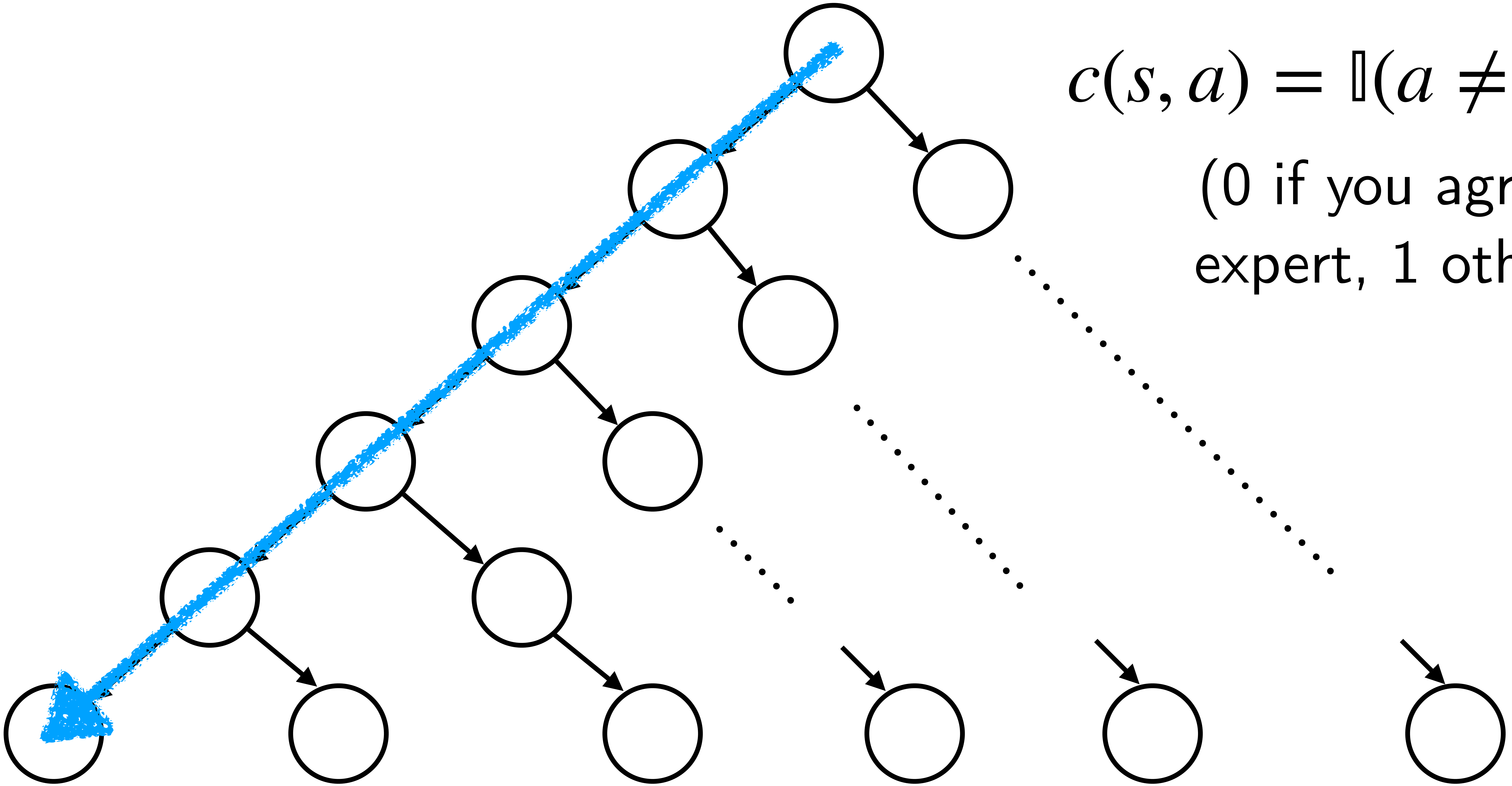
A Tree MDP



The demonstrator always takes a left



Assume the following cost function

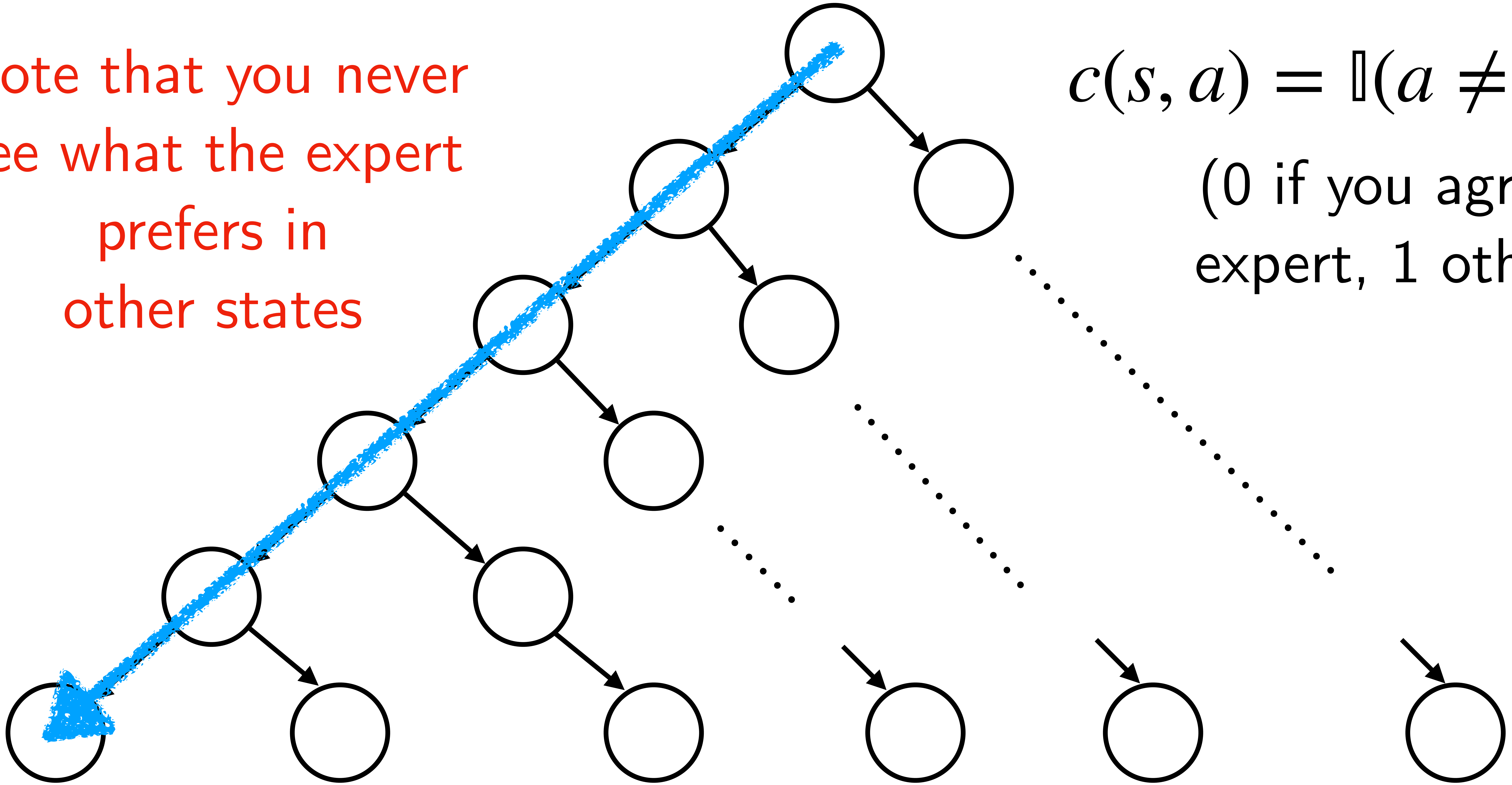


$$c(s, a) = \mathbb{I}(a \neq \pi^*(s))$$

(0 if you agree with expert, 1 otherwise)

Assume the following cost function

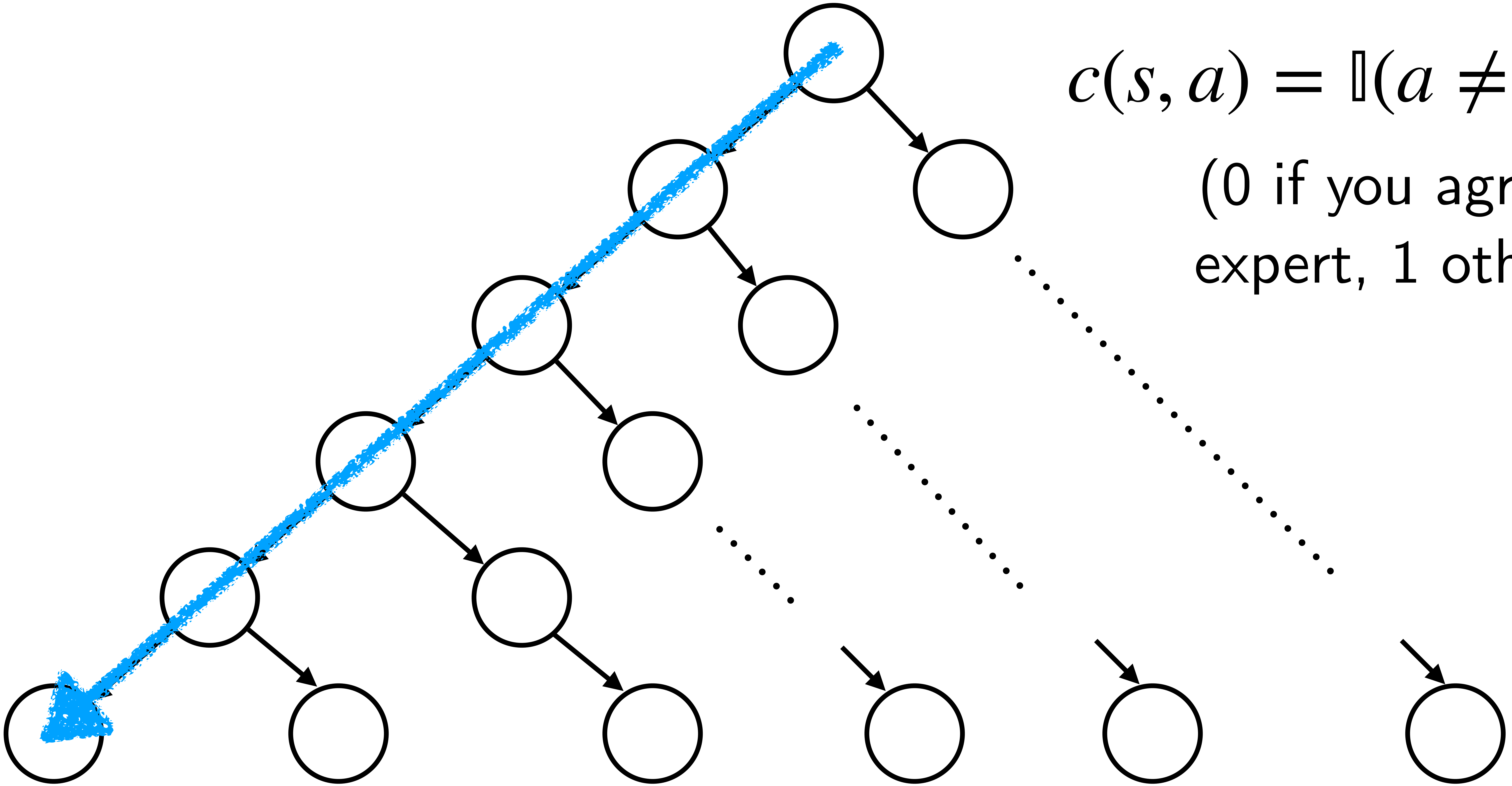
Note that you never see what the expert prefers in other states



$$c(s,a) = \mathbb{I}(a \neq \pi^*(s))$$

(0 if you agree with expert, 1 otherwise)

Show that BC has a performance difference of $O(\epsilon T^2)$



$$c(s, a) = \mathbb{I}(a \neq \pi^*(s))$$

(0 if you agree with expert, 1 otherwise)

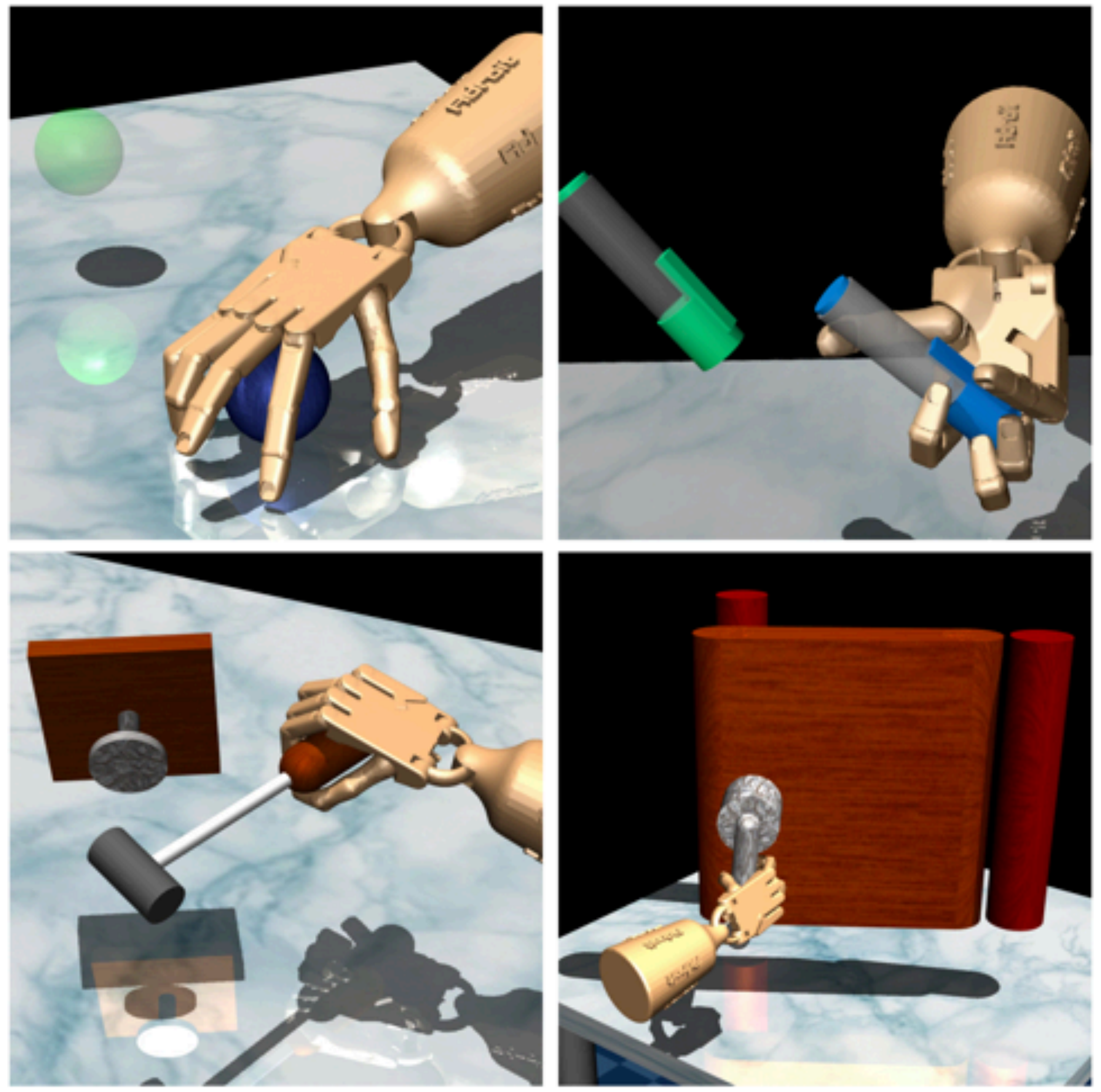
Proof

So, it seems BC is totally doomed ...

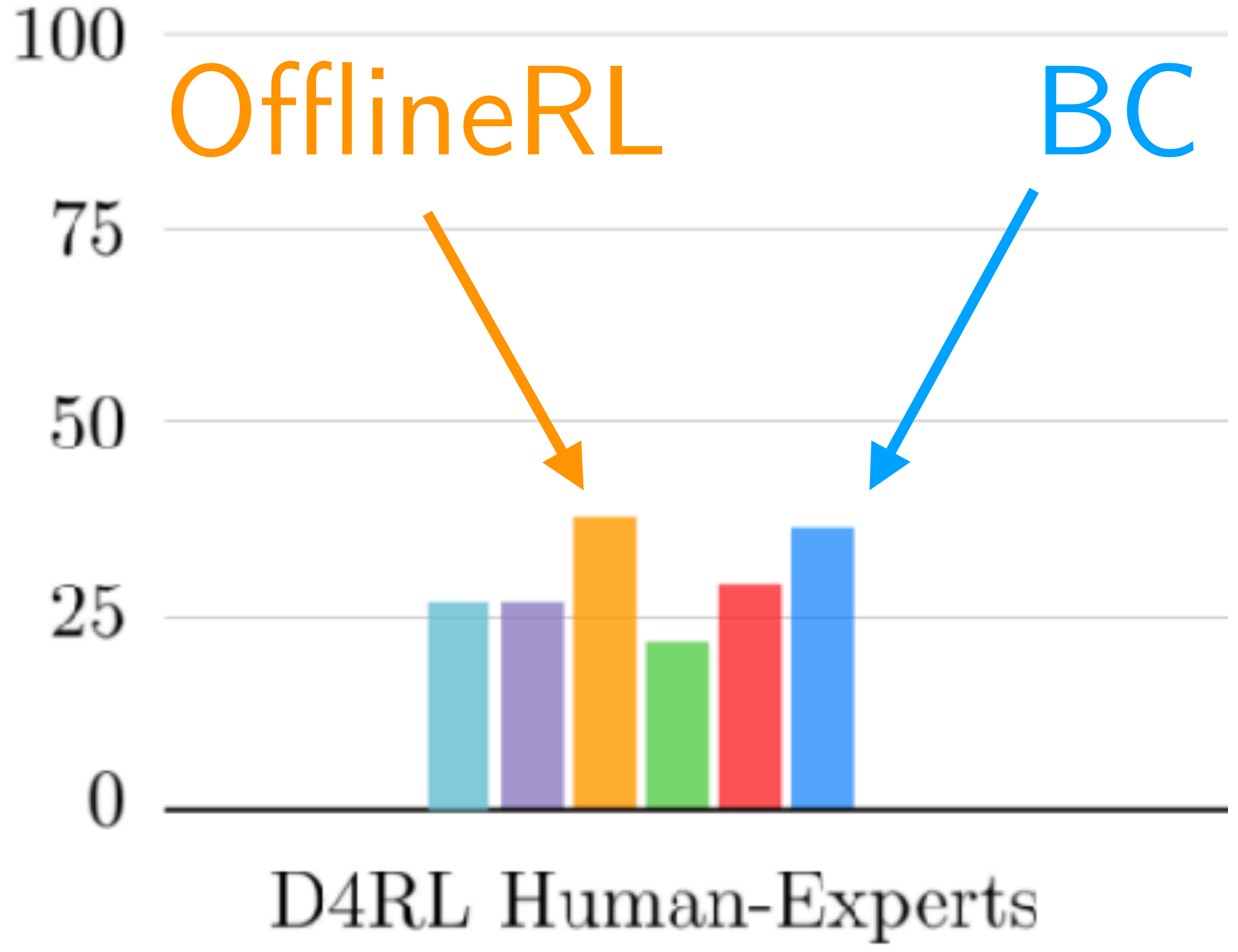
But, BC works surprisingly often!!

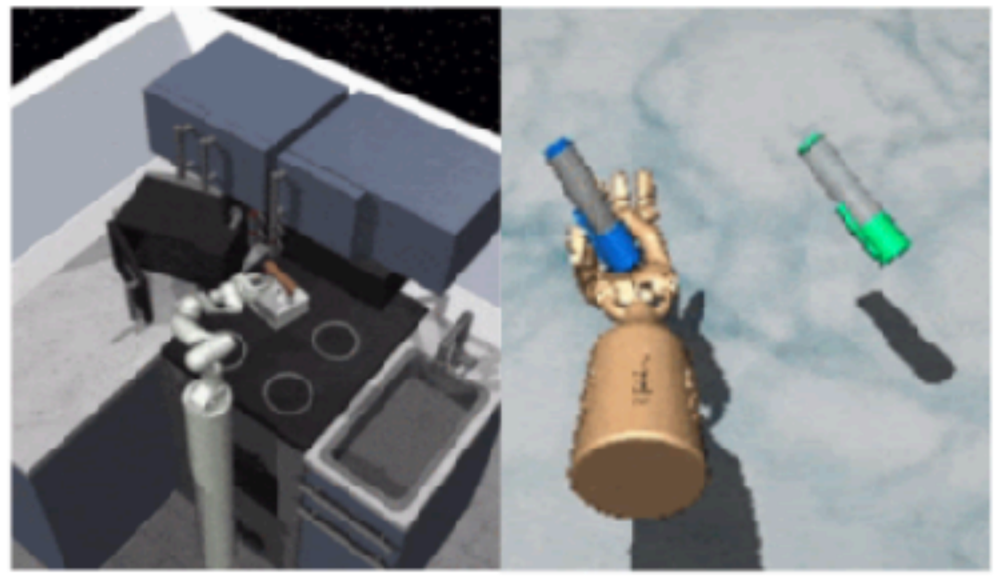
Environment	Expert	BC
CartPole	500 ± 0	500 ± 0
Acrobot	-71.7 ± 11.5	-78.4 ± 14.2
MountainCar	-99.6 ± 10.9	-107.8 ± 16.4
Hopper	3554 ± 216	3258 ± 396
Walker2d	5496 ± 89	5349 ± 634
HalfCheetah	4487 ± 164	4605 ± 143
Ant	4186 ± 1081	3353 ± 1801

[SCV+ arXiv '21]



[Rajeswaran et al. '17]





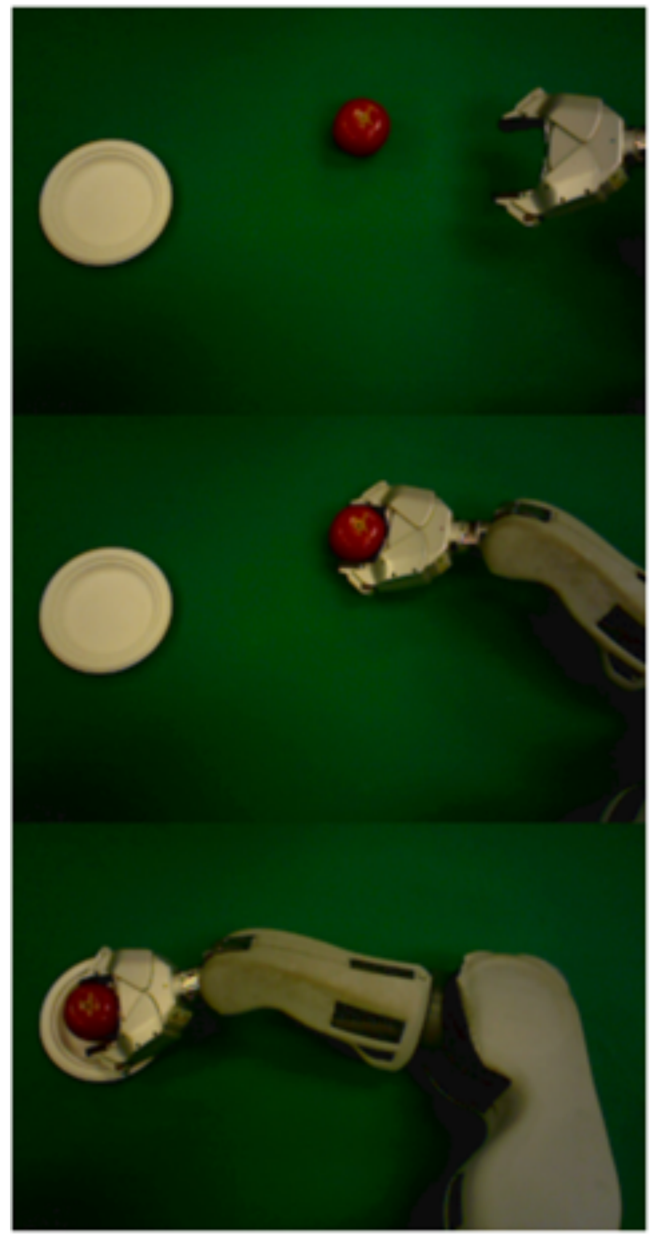
[Florence et al. '21]

But, BC works surprisingly often!!

Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation

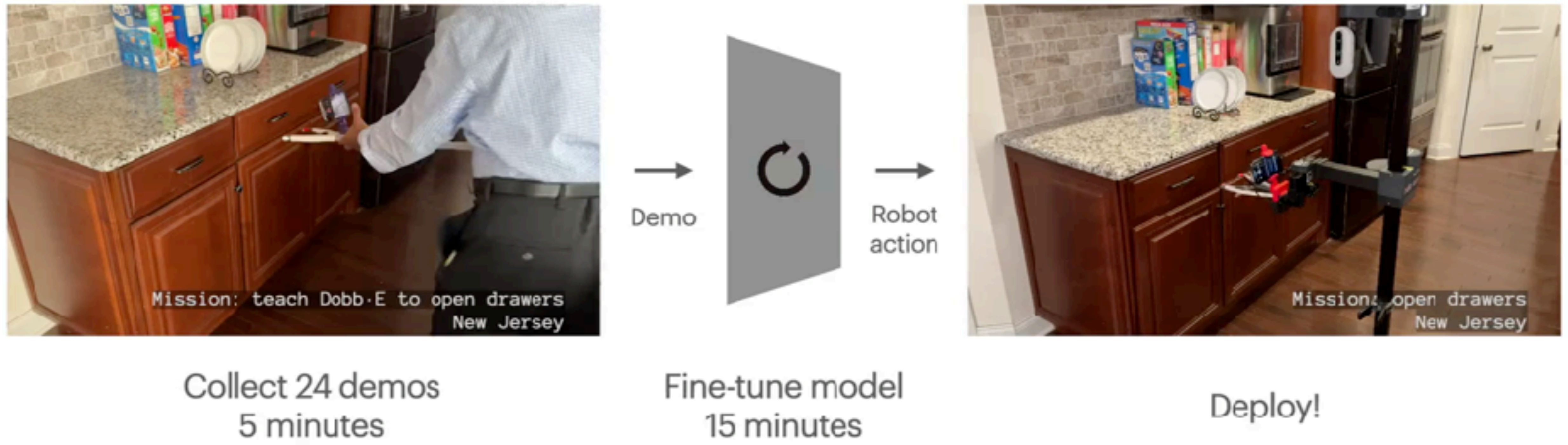
Tianhao Zhang*[12], Zoe McCarthy*[1], Owen Jow[1], Dennis Lee[1], Xi Chen[12], Ken Goldberg[1], Pieter Abbeel[1-4]

Fig. 1: Virtual Reality teleoperation in action



On Bringing Robots Home

Nur Muhammad (Mahi) Shafiullah*† NYU, Anant Rai* NYU, Haritheja Etukuru NYU, Yiqian Liu NYU, Ishan Misra Meta, Soumith Chintala Meta, Lerrel Pinto NYU



🤔 Why does BC work in these cases?

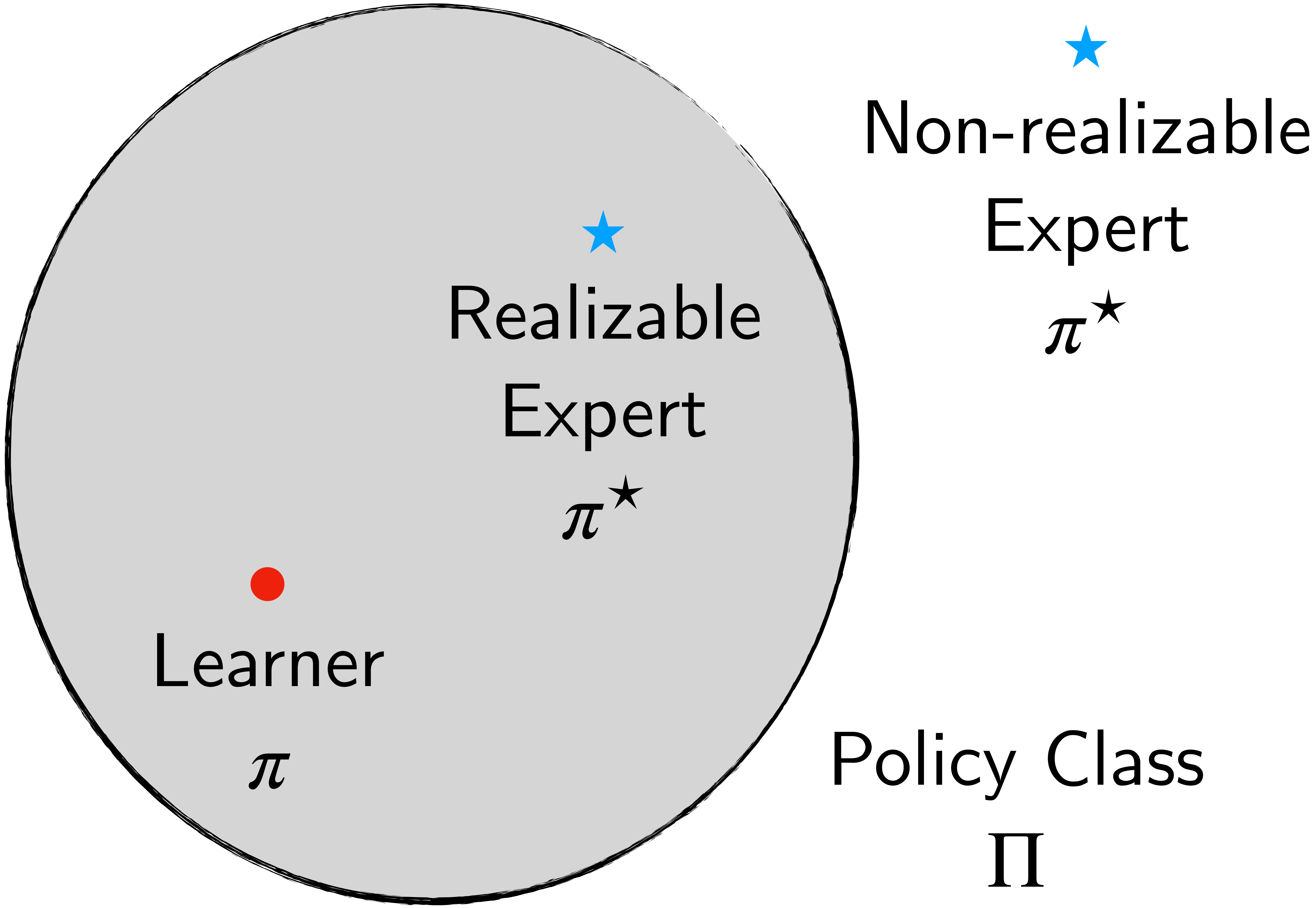
$$O(\epsilon T^2)$$

Drive ϵ to 0!

When can we actually do this?

The Realizable Setting

With infinite data and a realizable expert, can drive $\epsilon \to 0$

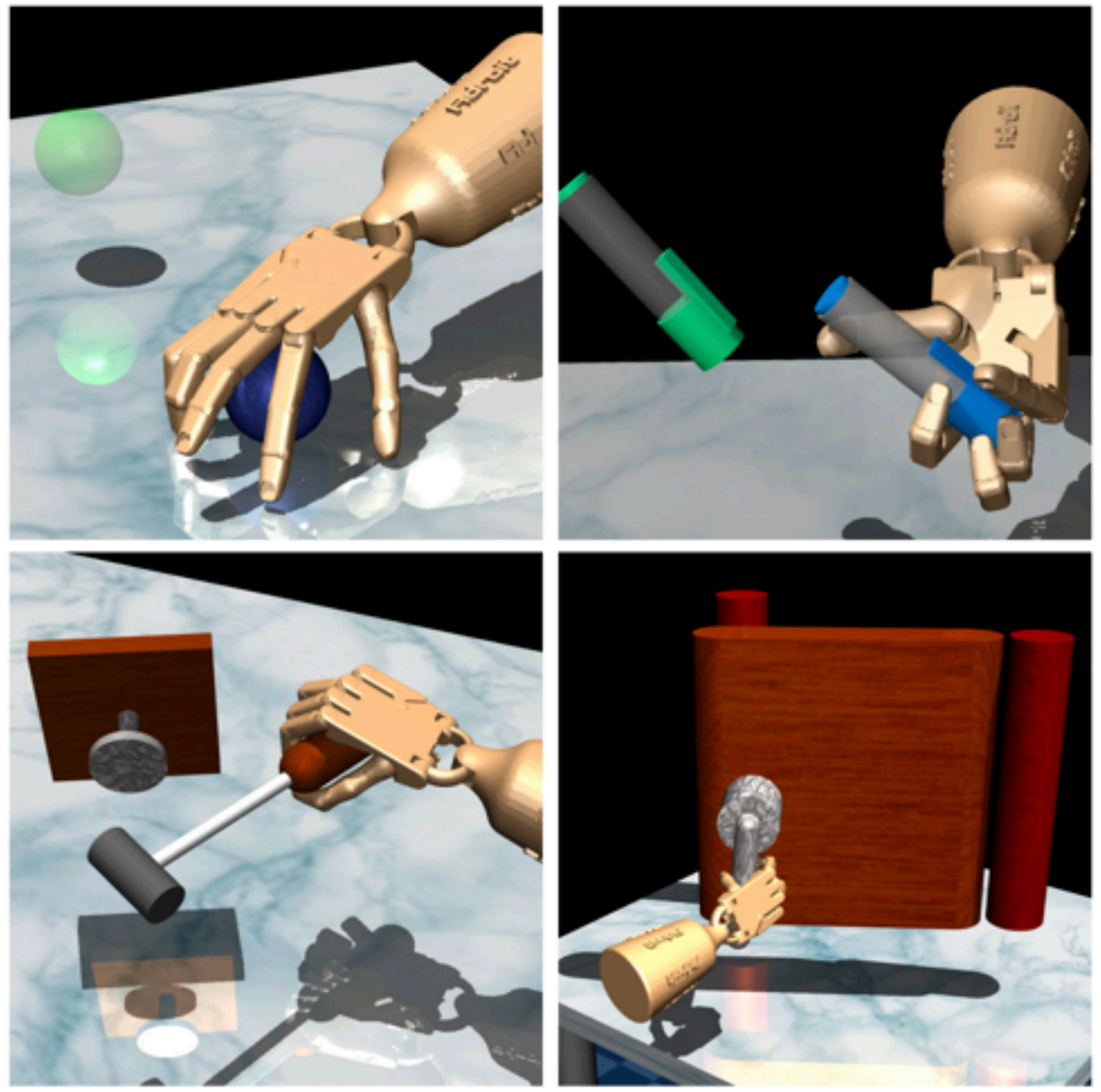


Realizable settings are easy ...

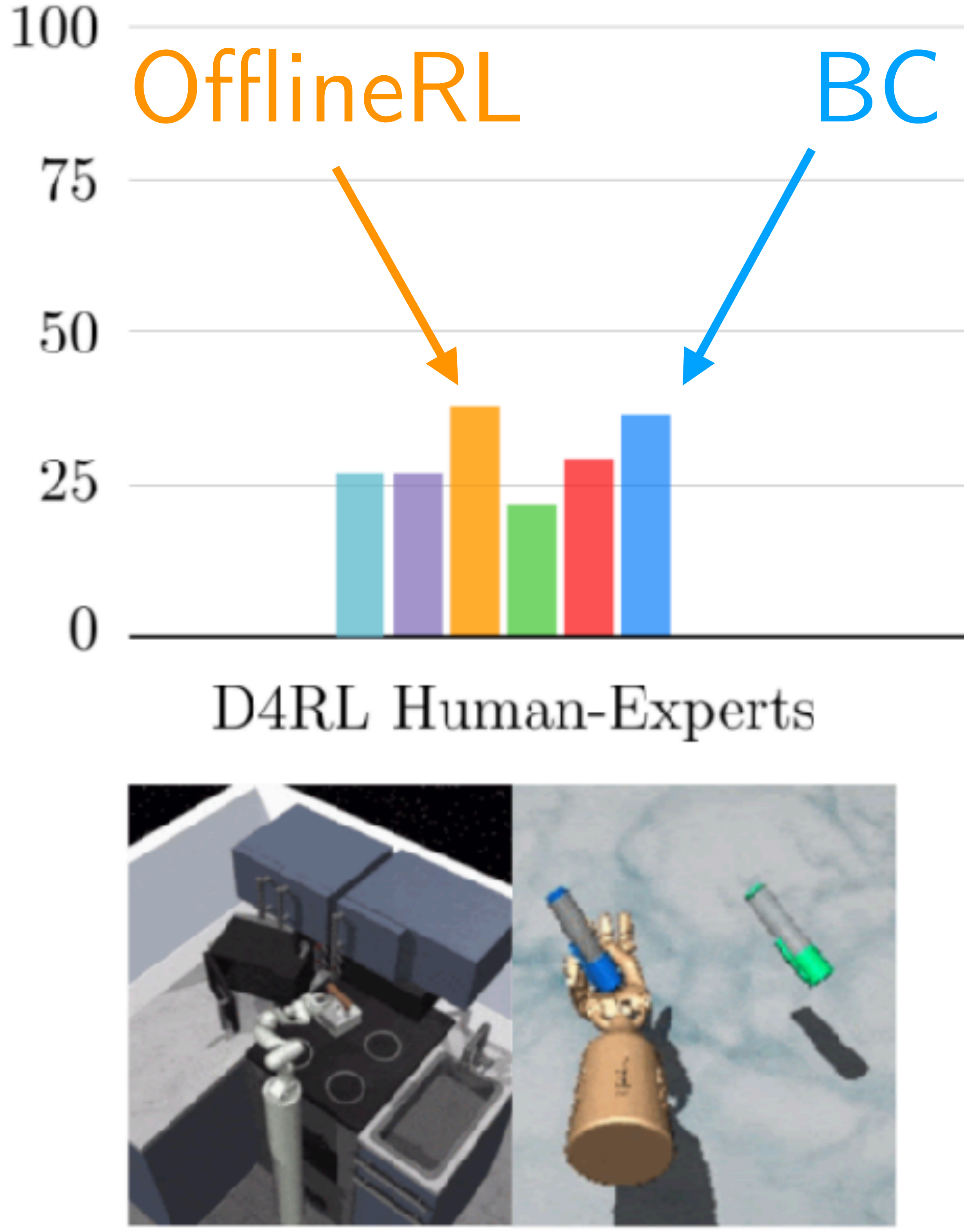
Why is the expert realizable here? (Easy)

Environment	Expert	BC
CartPole	500 ± 0	500 ± 0
Acrobot	-71.7 ± 11.5	-78.4 ± 14.2
MountainCar	-99.6 ± 10.9	-107.8 ± 16.4
Hopper	3554 ± 216	3258 ± 396
Walker2d	5496 ± 89	5349 ± 634
HalfCheetah	4487 ± 164	4605 ± 143
Ant	4186 ± 1081	3353 ± 1801

[SCV+ arXiv '21]



[Rajeswaran et al. '17]



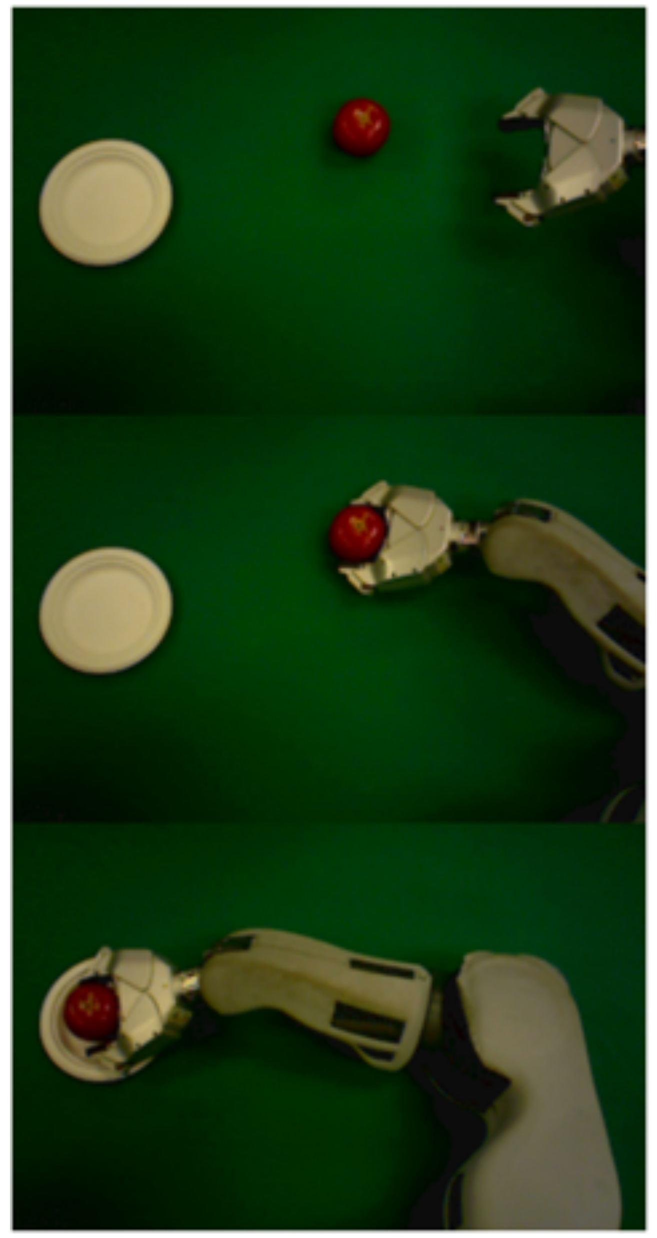
[Florence et al. '21]

Why is the expert realizable here? (Easy)

Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation

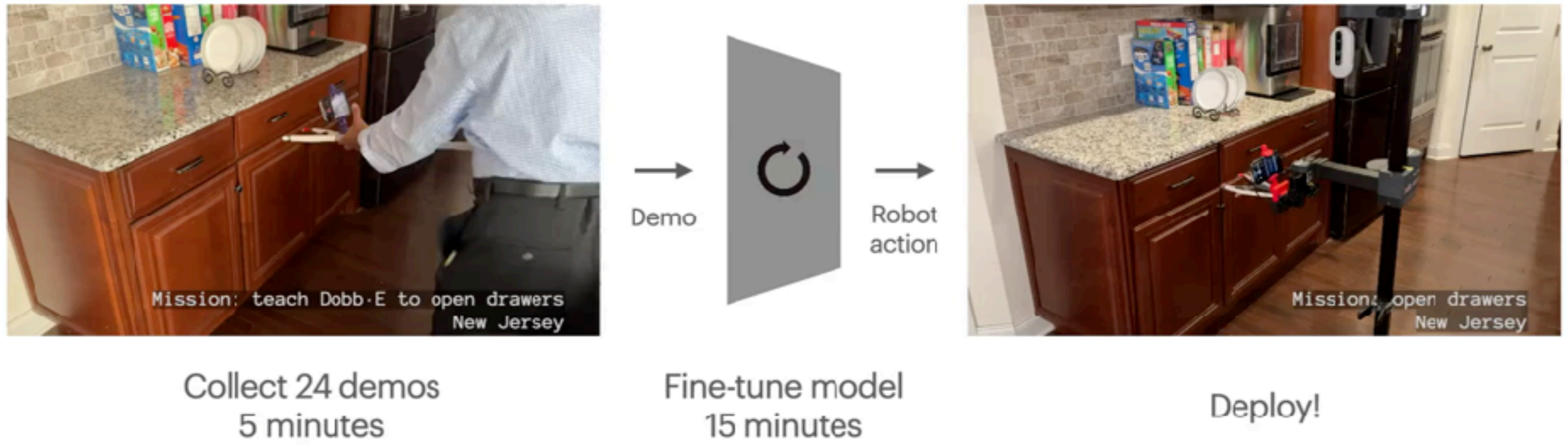
Tianhao Zhang*[12], Zoe McCarthy*[1], Owen Jow[1], Dennis Lee[1], Xi Chen[12], Ken Goldberg[1], Pieter Abbeel[1-4]

Fig. 1: Virtual Reality teleoperation in action



On Bringing Robots Home

Nur Muhammad (Mahi) Shafiullah*† NYU, Anant Rai* NYU, Haritheja Etukuru NYU, Yiqian Liu NYU
Ishan Misra Meta, Soumith Chintala Meta, Lerrel Pinto NYU



What is the hard case where $\epsilon > 0$?

Non-realizable Expert!

Poll

Give examples of a non-realizable expert

When poll is active respond at **PollEv.com/sc2582**

Hint: There are at least 3 categories!

Idea for a New Algorithm!

What if we
just queried the expert for the best action on states the learner visits?

Interactive Behavior Cloning

Initialize with a random policy π_1 # Can be BC

For $i = 1, \ldots, N$

Execute policy π_i in the real world and collect data

$$\mathcal{D}_i = \{s_0, a_0, s_1, a_1, \ldots\}$$

Also called a rollout

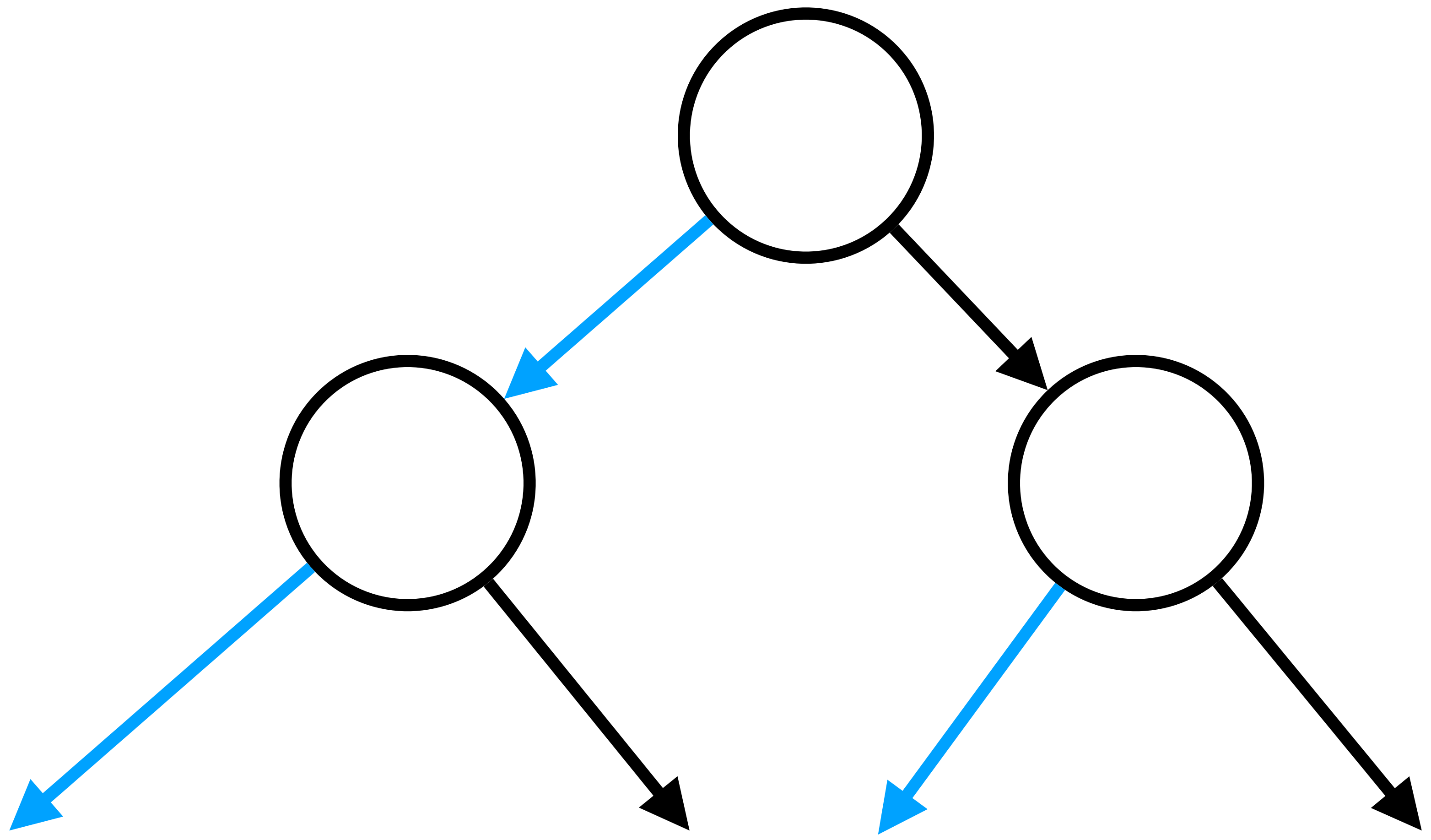
Query the expert for the optimal action on learner states

$$\mathcal{D}_i = \{s_0, \pi^\star(s_0), s_1, \pi^\star(s_1), \ldots\}$$

Train a new learner on this dataset

$$\pi_{i+1} \leftarrow \text{Train}(\mathcal{D}_i)$$

Does Interactive BC solve our Tree MDP?



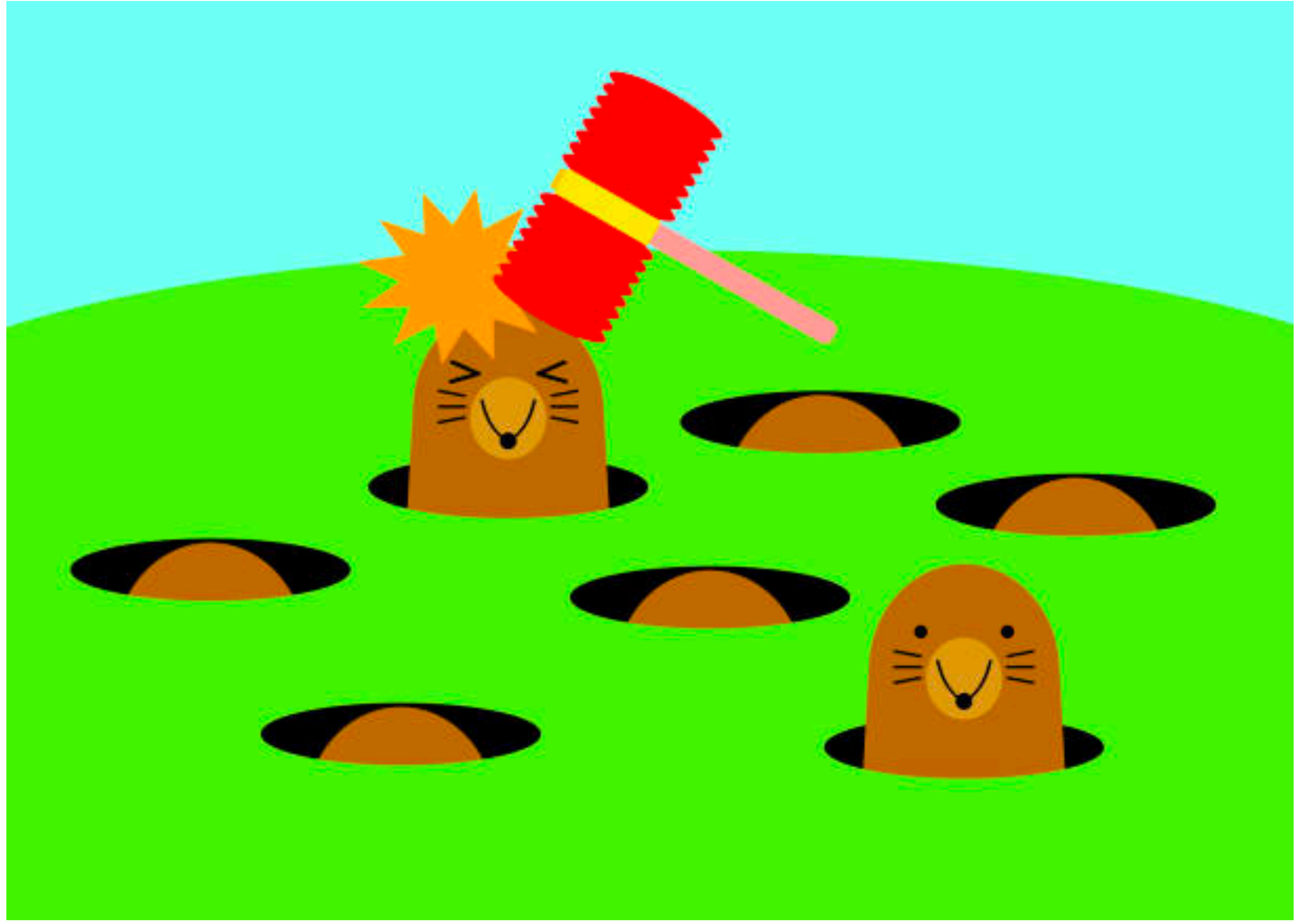
Let's assume depth 2

Expert always take left

Assume every iteration $\pi_{i+1} \leftarrow \text{Train}(\mathscr{D}_i)$
we can drive down loss to ϵ

Let's walk through how interactive BC does!

Interactive BC is also $O(\epsilon T^2)$!



$$\pi_{i+1} \leftarrow \text{Train}(\mathscr{D}_i)$$

π_{i+1} can have a totally different distribution than $\mathscr{D}_i$ generated by π_i

Instead of throwing out the old dataset, what if we aggregated data over iterations?

DAgger (Dataset Aggregation)

Initialize with a random policy π_1 # Can be BC

Initialize empty data buffer $\mathscr{D} \leftarrow \{\}$

For $i = 1, \ldots, N$

Execute policy π_i in the real world and collect data

$$\mathscr{D}_i = \{s_0, a_0, s_1, a_1, \ldots\}$$ # Also called a rollout

Query the expert for the optimal action on learner states

$$\mathscr{D}_i = \{s_0, \pi^\star(s_0), s_1, \pi^\star(s_1), \ldots\}$$

Aggregate data $\mathscr{D} \leftarrow \mathscr{D} \cup \mathscr{D}_i$

Train a new learner on this dataset $\pi_{i+1} \leftarrow \text{Train}(\mathscr{D})$

Select the best policy in $\pi_{1:N+1}$

Why does DAgger work?

Theory of Online Learning explains why (Next Lecture!)