Learning by Cheating

Drew Bagnell

Aurora

An Ode to Imitation Learning

[K. Mülling et al., 2013]

[M. Zucker et al., 2011]

[A. Coates et al., '08]

[D. Pomerleau, '89]

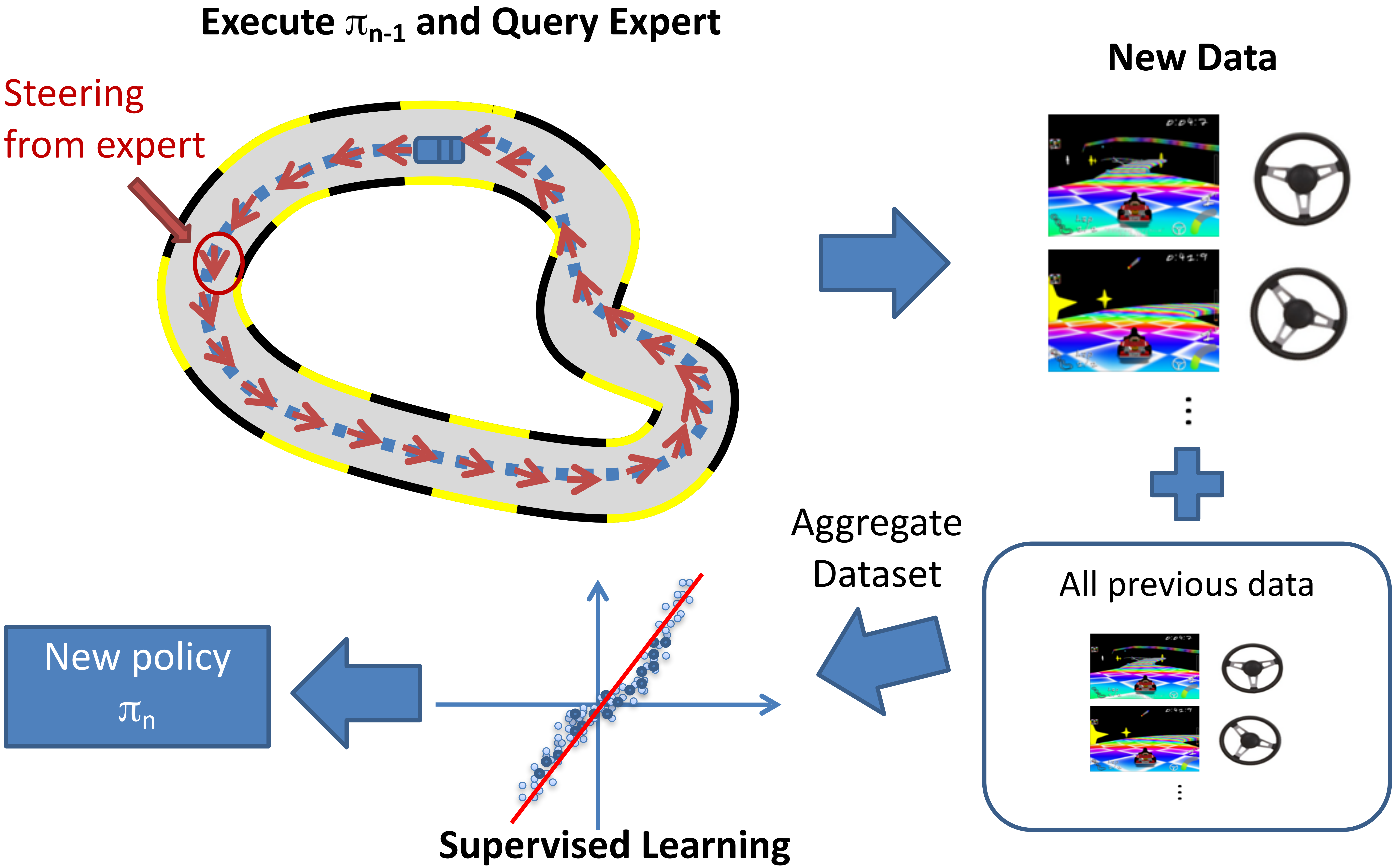
[N. Ratliff et al., 2006]

[J. A. Bagnell et al., 2010]

[Ross11a]

DAgger: Dataset Aggregation

n^{th} iteration



Why would having access to the Q's be better?

AGGREVATE: Expert provides values

Just like DAGGER

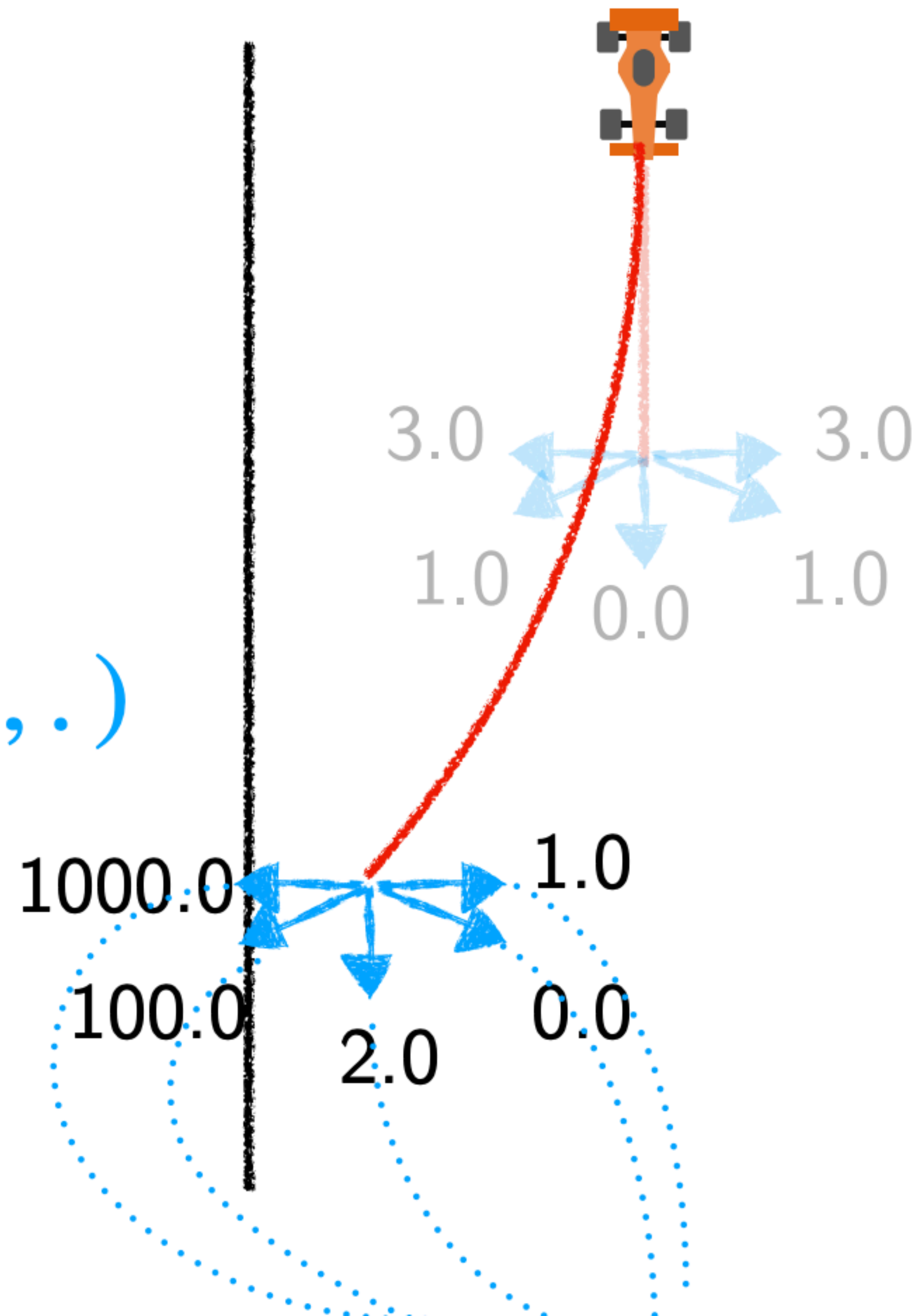
For i = 0 … N-1

Roll-in learner π_i to get $\{s \sim d_{\pi_i}\}$

Query expert for advantage vector $A^*(s, .)$

Aggregate data $\mathscr{D} \leftarrow \mathscr{D} \cup \{s, A^*(s, .)\}$

Train policy $\pi_{i+1} = \mathbb{E}_{s,A^* \sim \mathscr{D}}(A^*(s, \pi(s)))$



Imitation Great for Robotics!

But.... sometimes hard to get humans

1) to do the task well

2) generate enough data

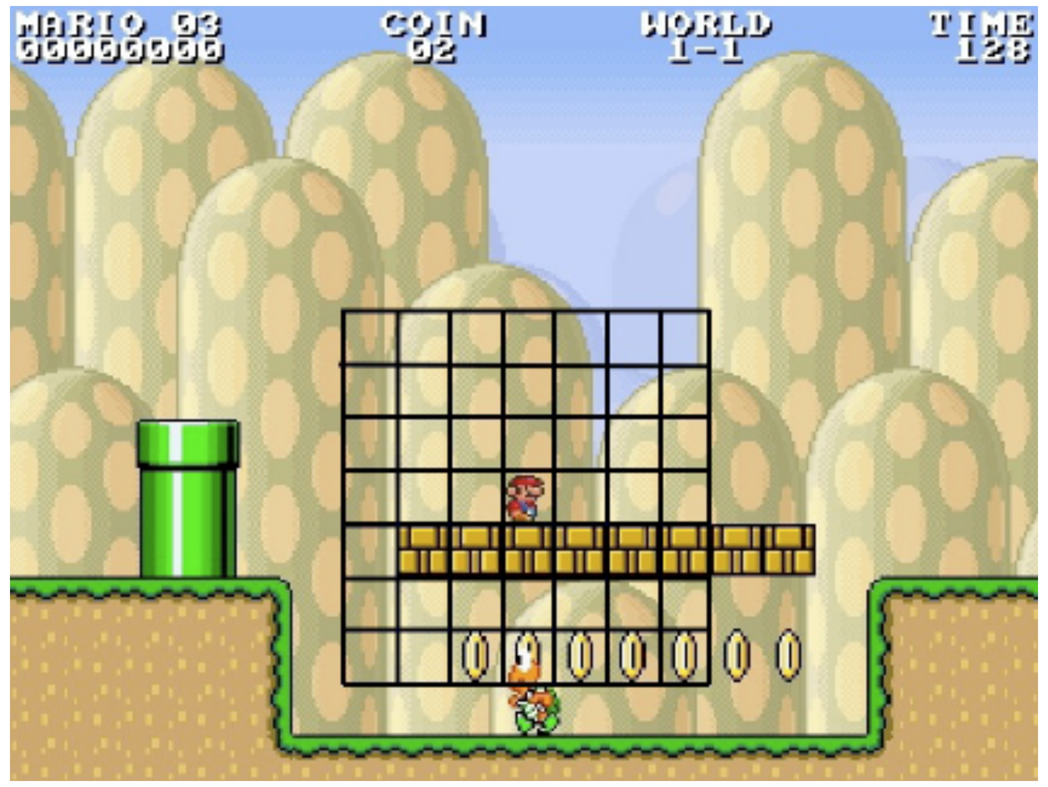
3) provide "critic" or Q-values

So let's just make the computer the teacher!

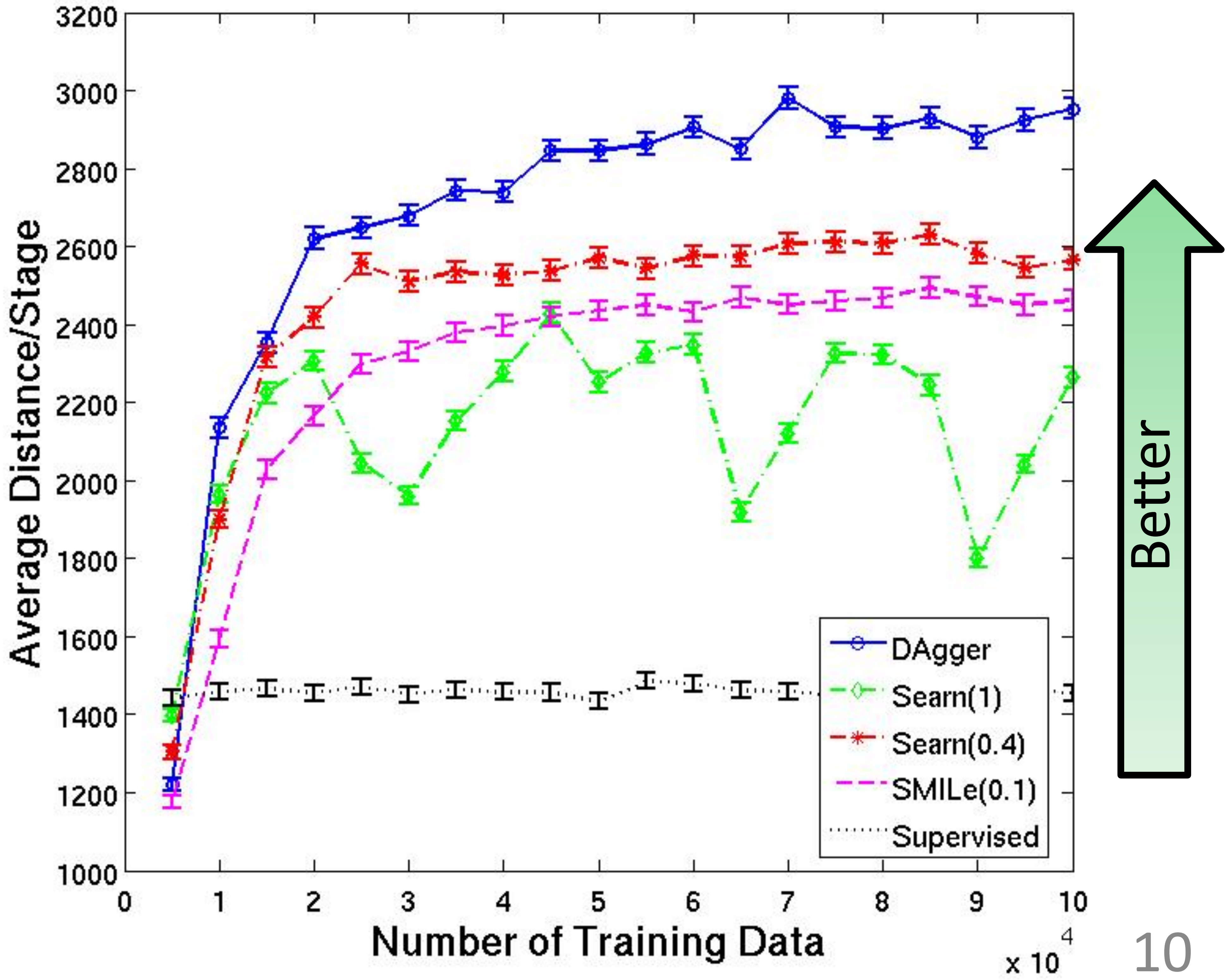
So let's just make the computer the teacher!

[Ross11a]

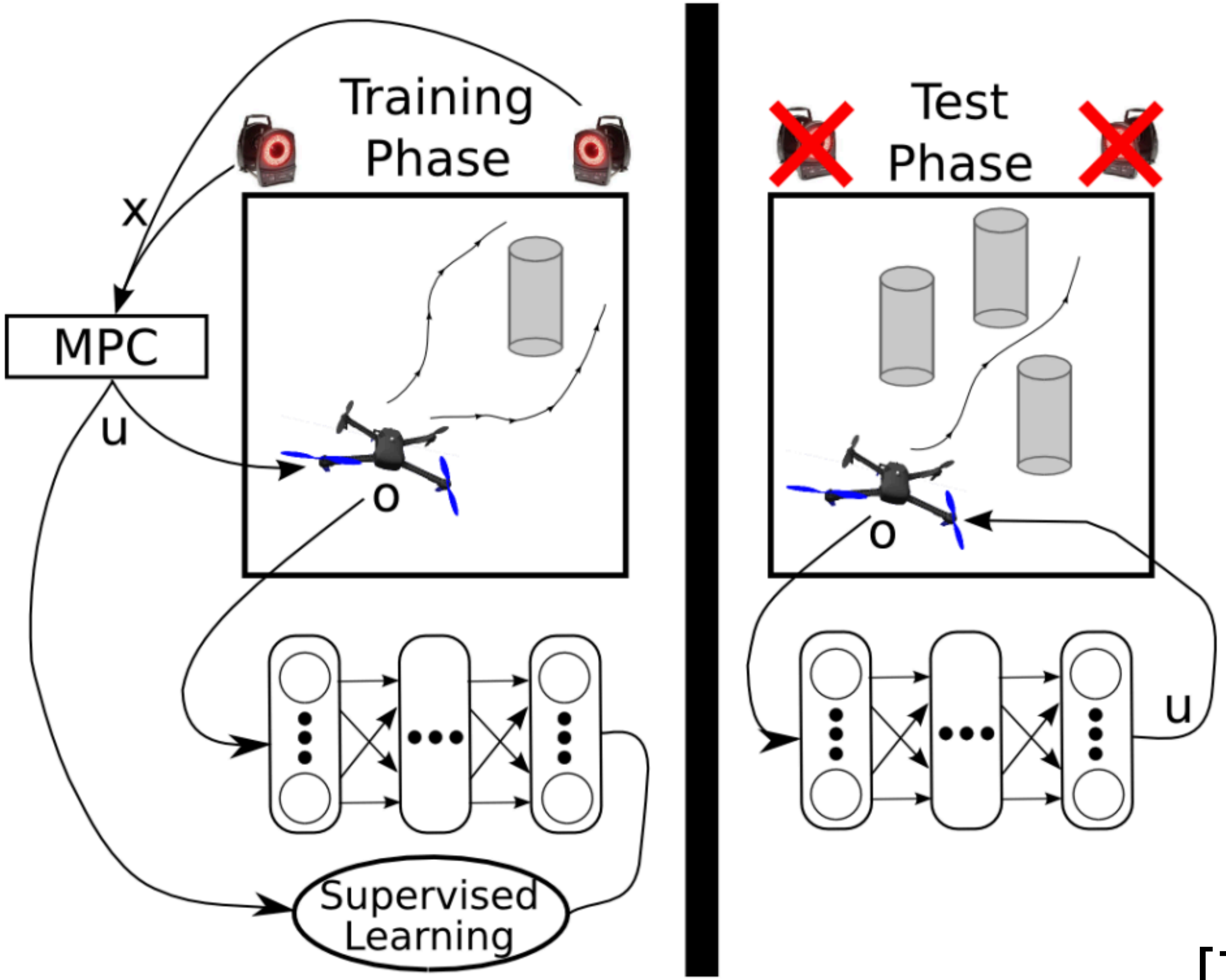
The OG: Super Mario Bros



- Improved Performance over Supervised and other state-of-the-art methods such as SMILE and SEARN.
- https://www.youtube.com/@aistats11anon



Privileged Information: UAV Navigation

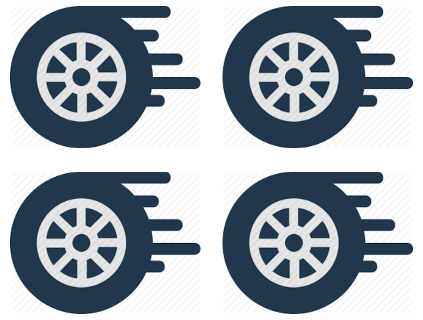


[Zhang et al. 2016]

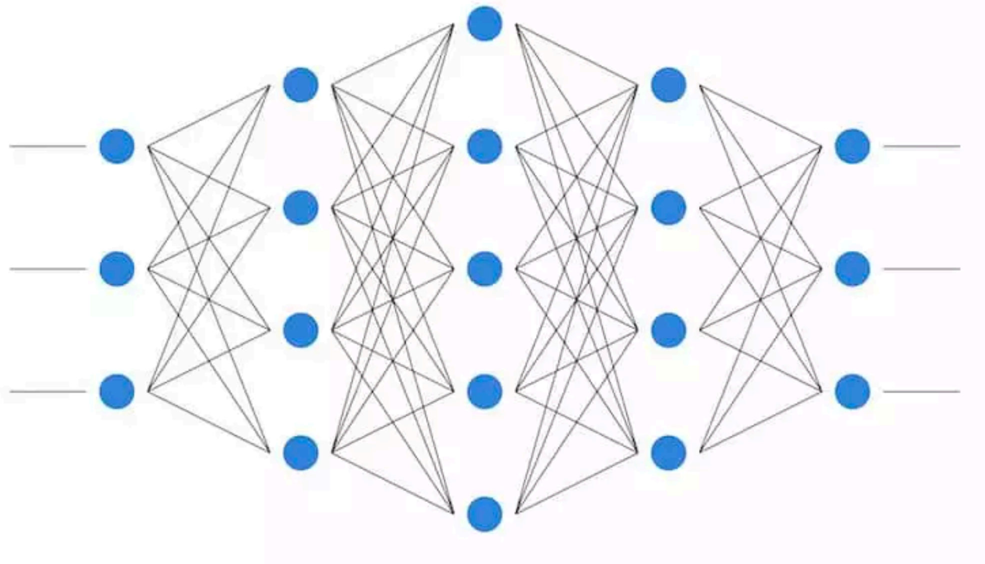
Example: Online Imitation Learning (DAgger)

Goal: learn a reactive policy to drive as fast as possible without crashing by mimicking an expert.

state → neural network policy → control command



RGB image wheel speed



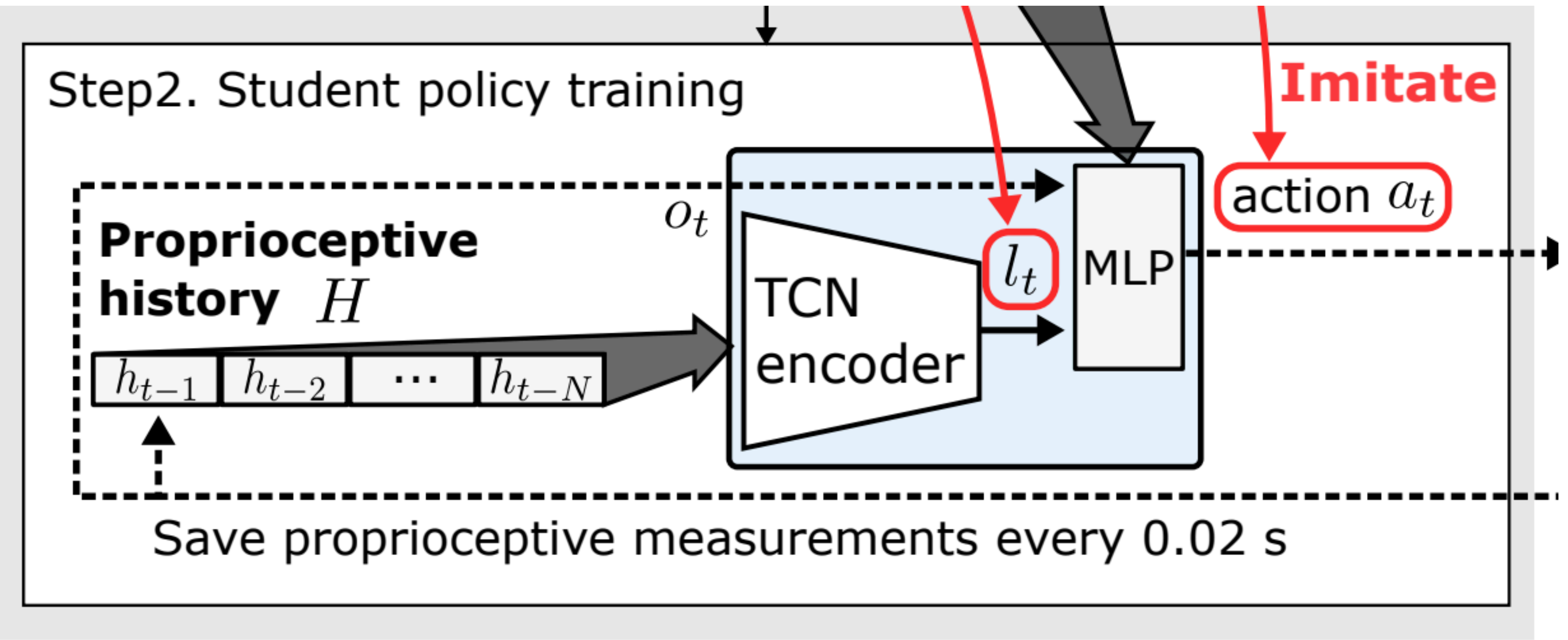
steering throttle

algorithmic expert: can be queried given any state

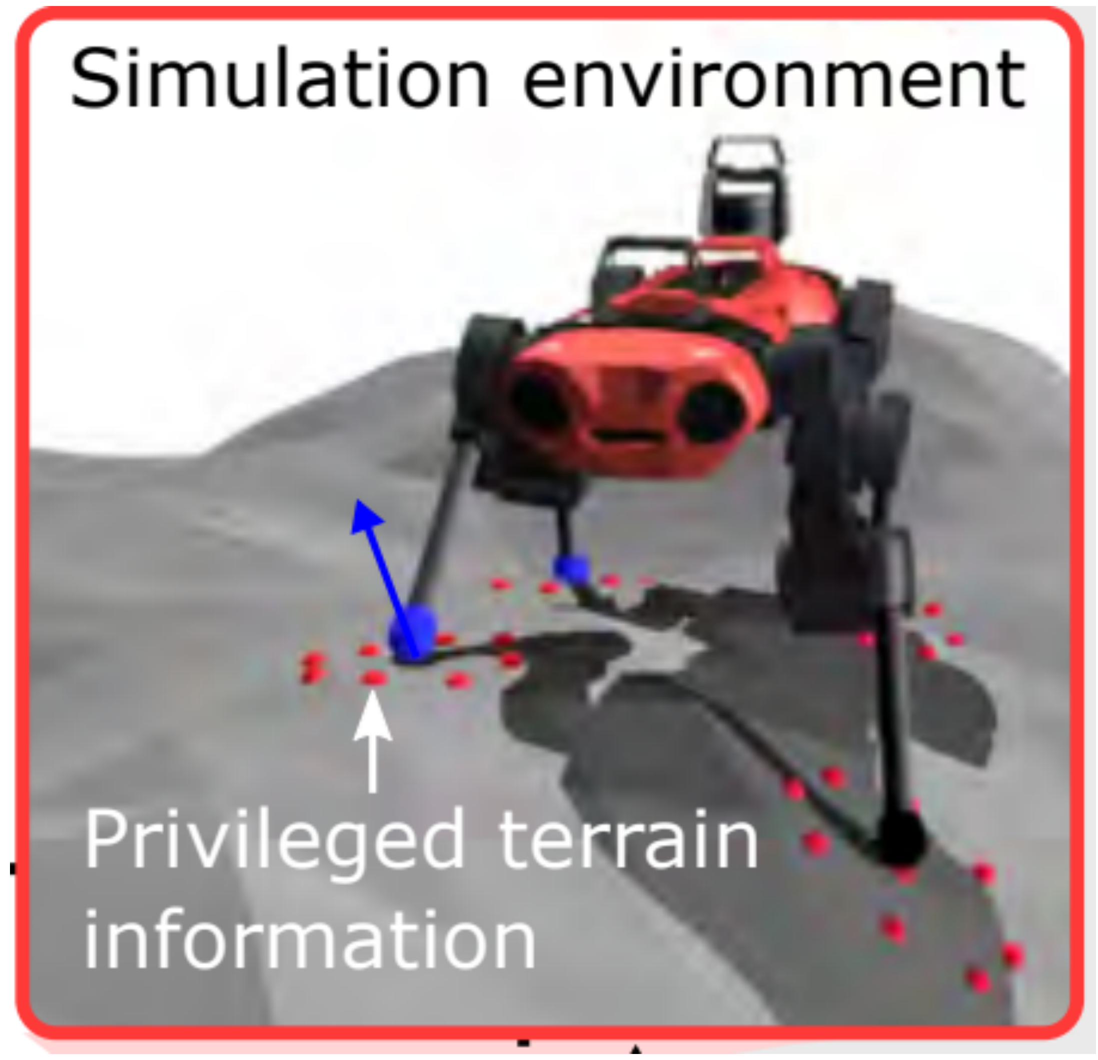
1. https://www.youtube.com/watch?v=hUoDNeZS4so
2. https://www.youtube.com/watch?v=FsRP4rEYiLI

Privileged Information: Legged Locomotion

Student Policy



Imitate



Teacher Policy

[Lee et al. 2020]

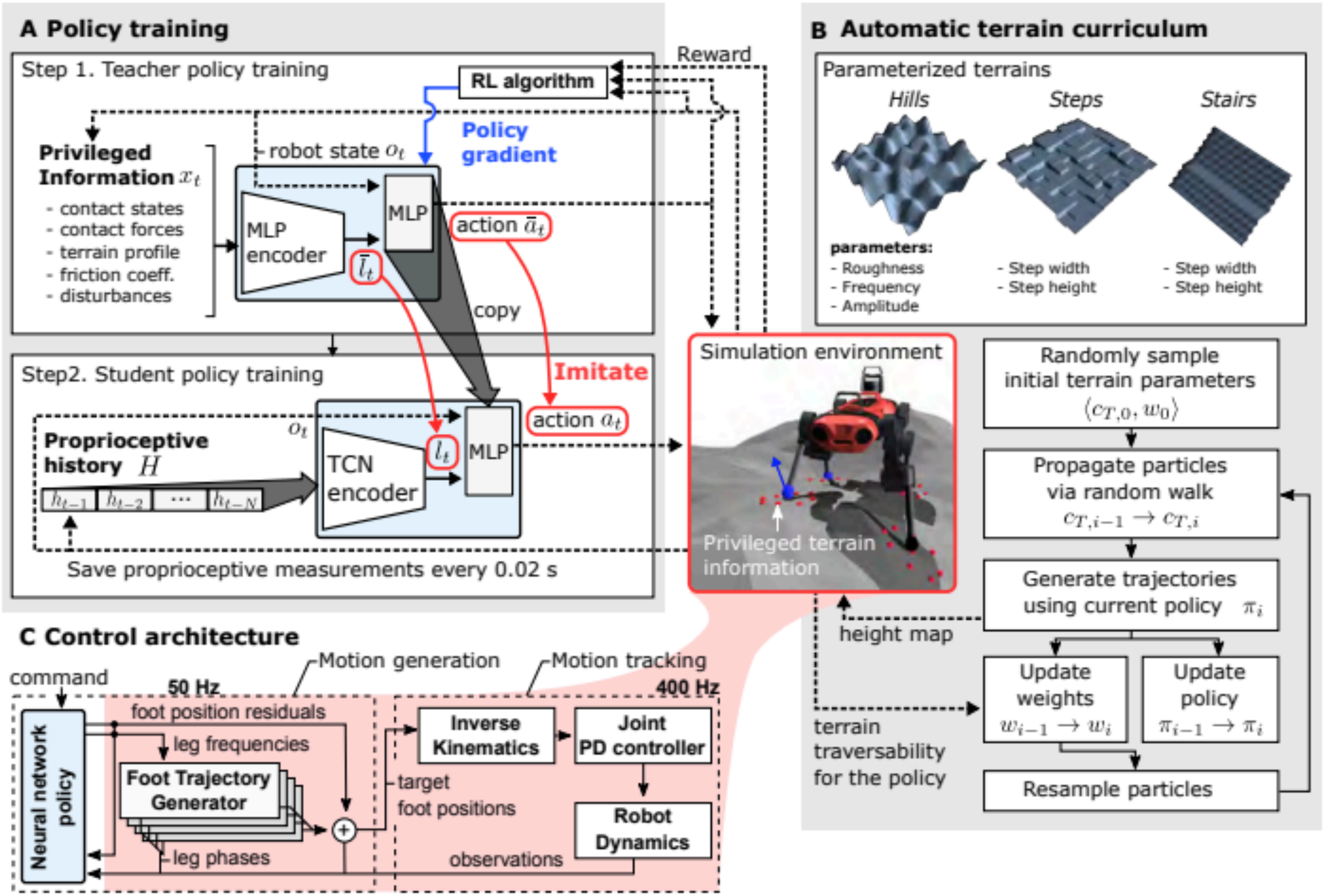
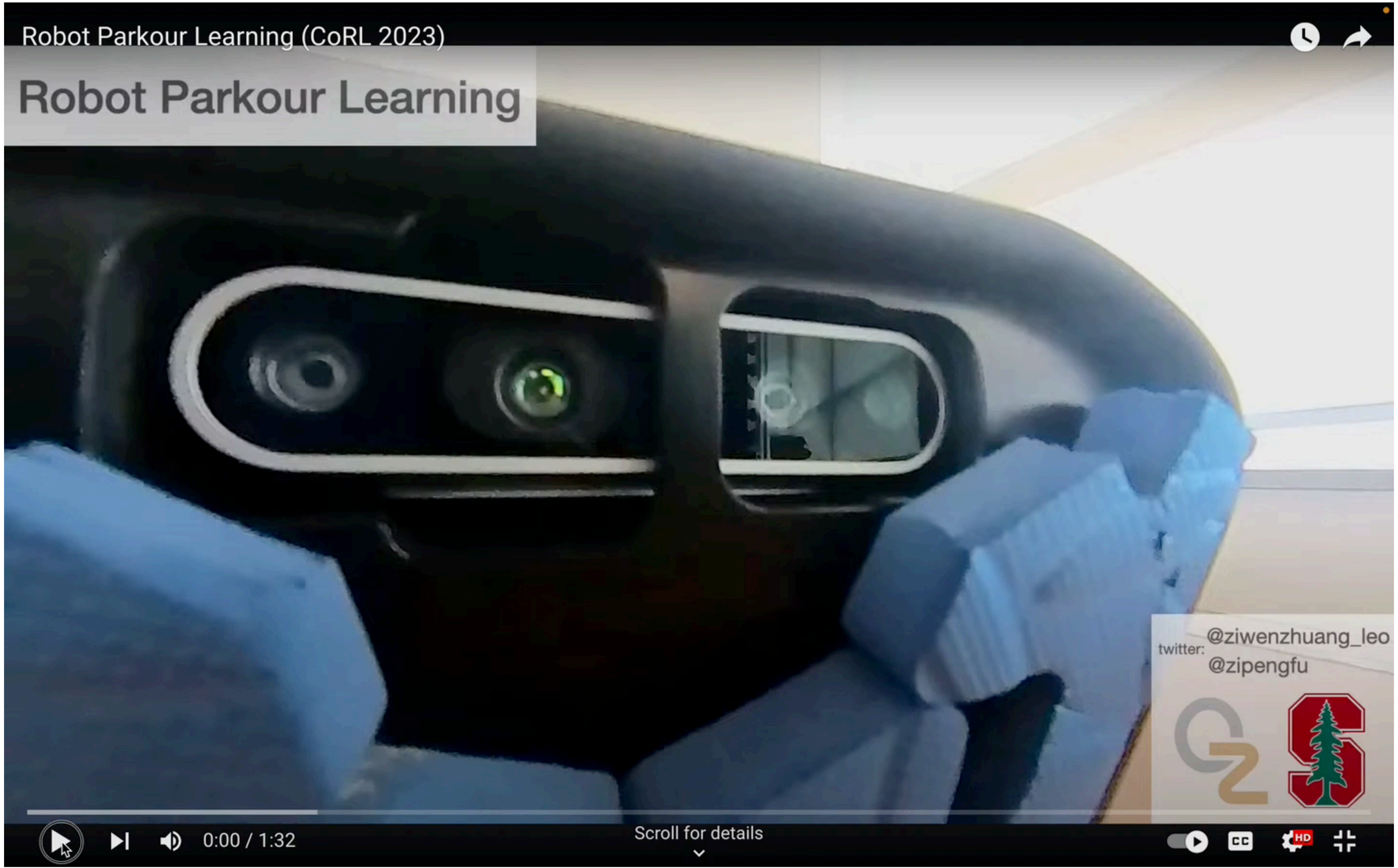


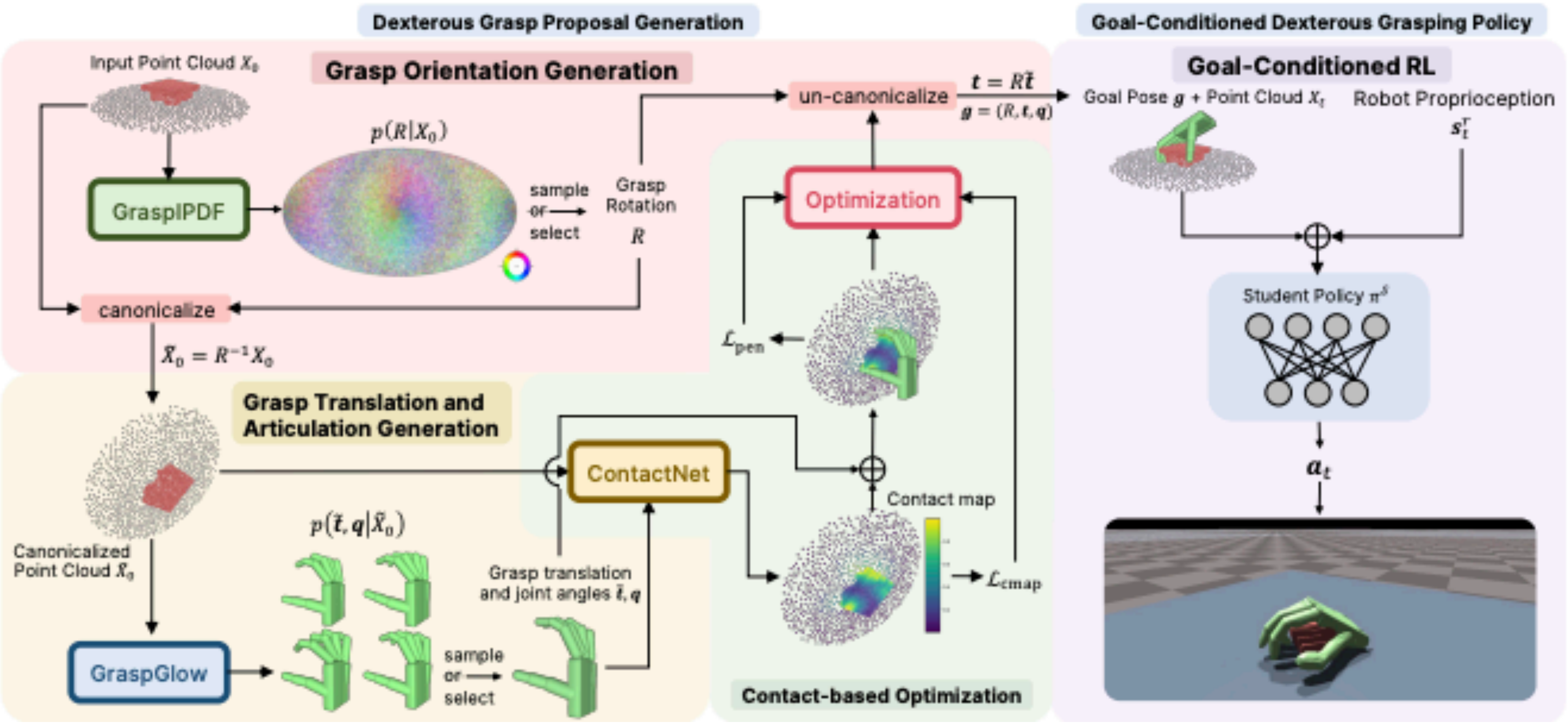
Fig. 4. Overview of the presented approach. (**A**) Two-stage training process. First, a teacher policy is trained using reinforcement learning in simulation. It has access to privileged information that is not available in the real world. Next, a proprioceptive student policy learns by imitating the teacher. The student policy acts on a stream of proprioceptive sensory input and does not use privileged information. (**B**) An adaptive terrain curriculum synthesizes terrains at an appropriate level of difficulty during the course of training. Particle filtering is used to maintain a distribution of terrain parameters that are challenging but traversable by the policy. (**C**) Architecture of the locomotion controller. The learned proprioceptive policy modulates motion primitives via kinematic residuals. An empirical model of the joint PD controller facilitates deployment on physical machines.

Adding vision in….



Unigrasp (and others): Distilling Grasping in simulation

Yinzhen Xu, et al. 2023



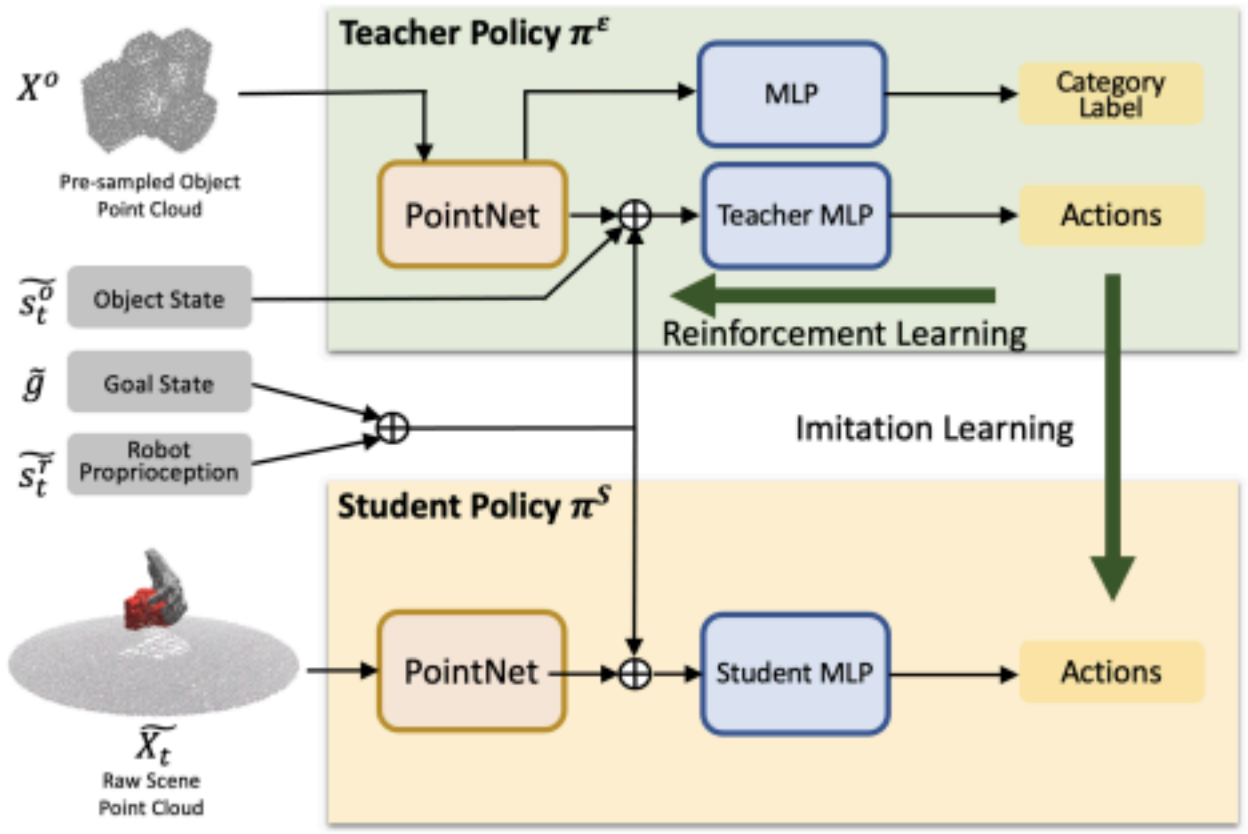
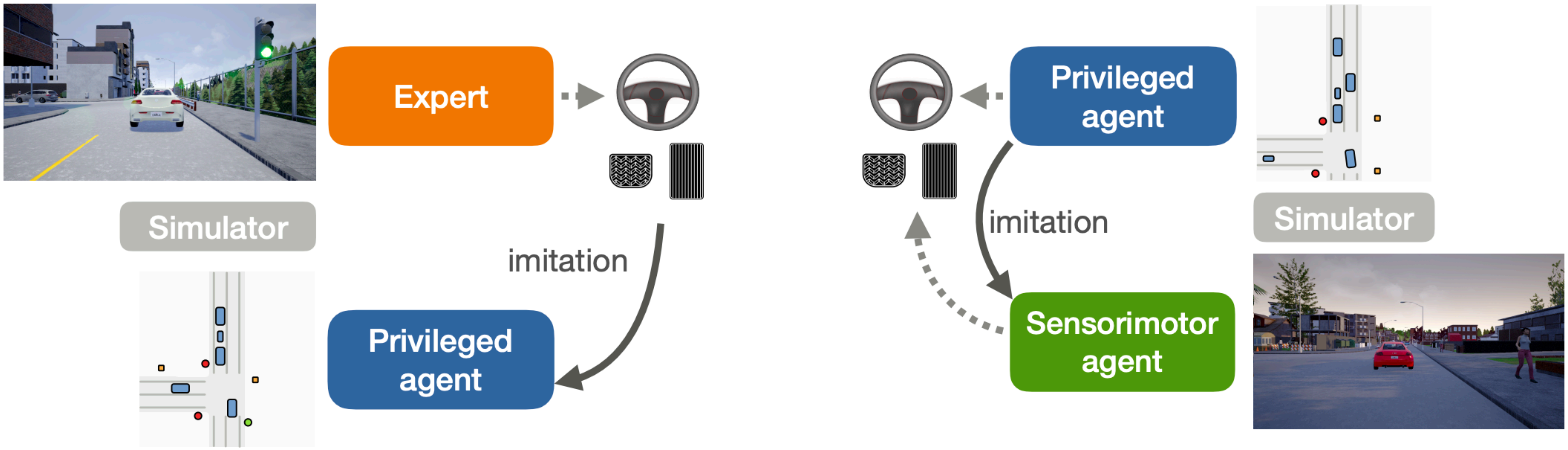


Figure 3. **The goal-conditioned dexterous grasping policy pipeline.** $\widetilde{S}_t^\mathcal{E} = (\widetilde{s}_t^r, \widetilde{s}_t^o, X^O, \widetilde{g})$ and $\widetilde{S}_t^S = (\widetilde{s}_t^r, \widetilde{X}_t, \widetilde{g})$ denote the input state of the teacher policy and student policy after state canonicalization, respectively; $\oplus$ denotes concatenation.

Privileged Information: Self-driving

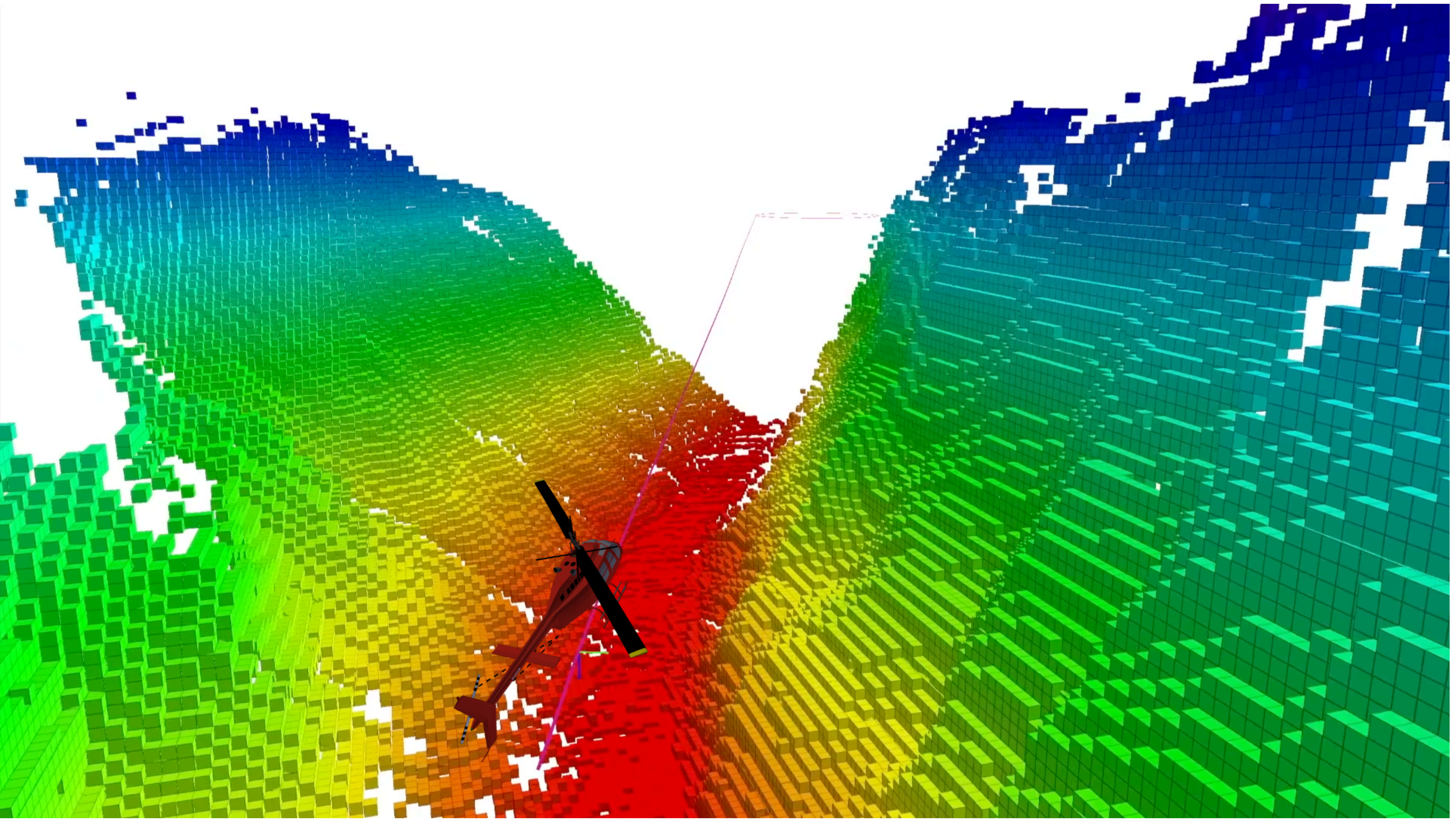


(a) Privileged agent imitates the expert

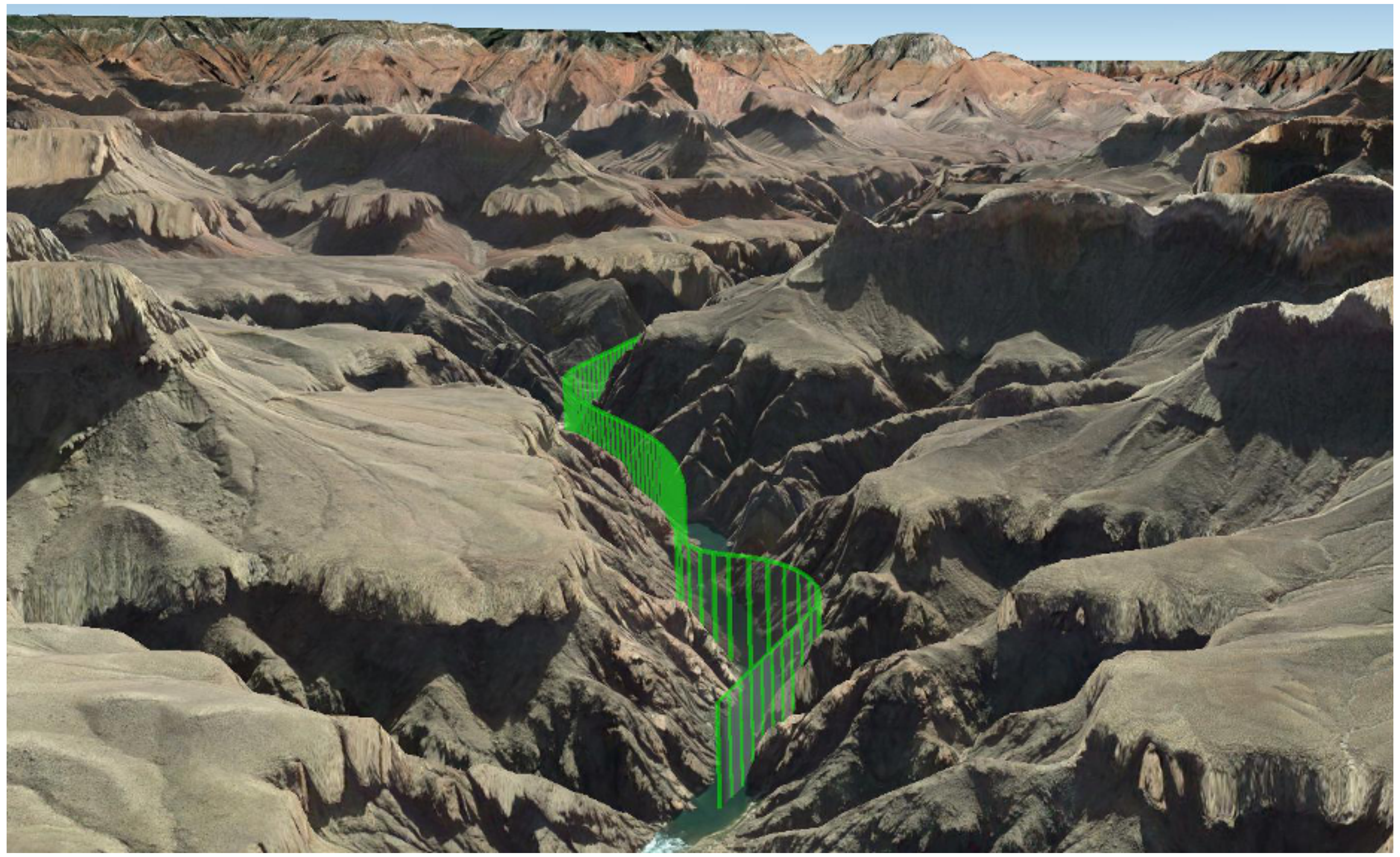
(b) Sensorimotor agent imitates the privileged agent

[Chen et al. 2020]

Privileged Information: Motion Planning



Imitate →

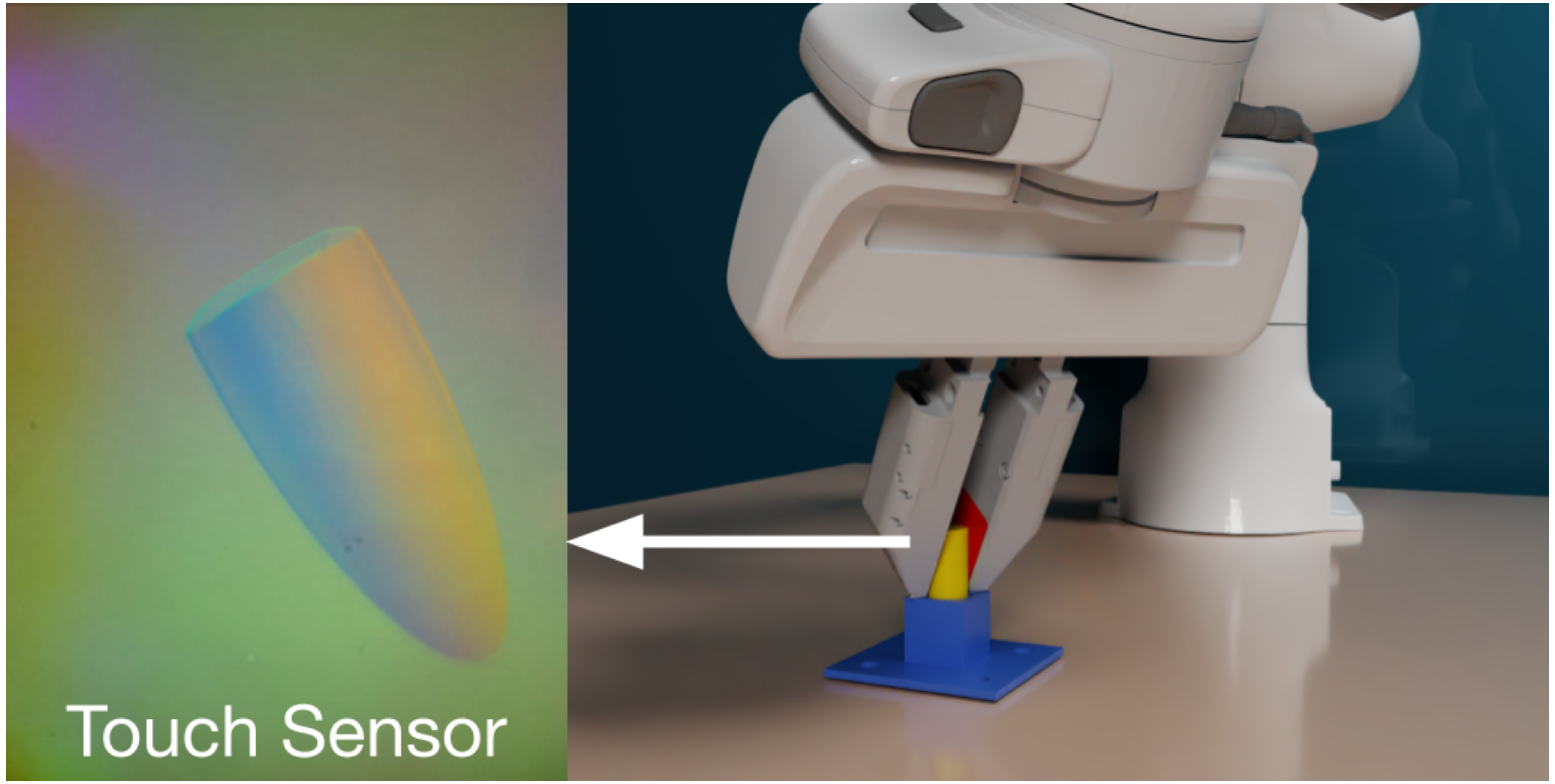


[Choudhury et al. '2018]

Part 3: Visuo-Tactile Simulation for Policy Learning

Key strategies:

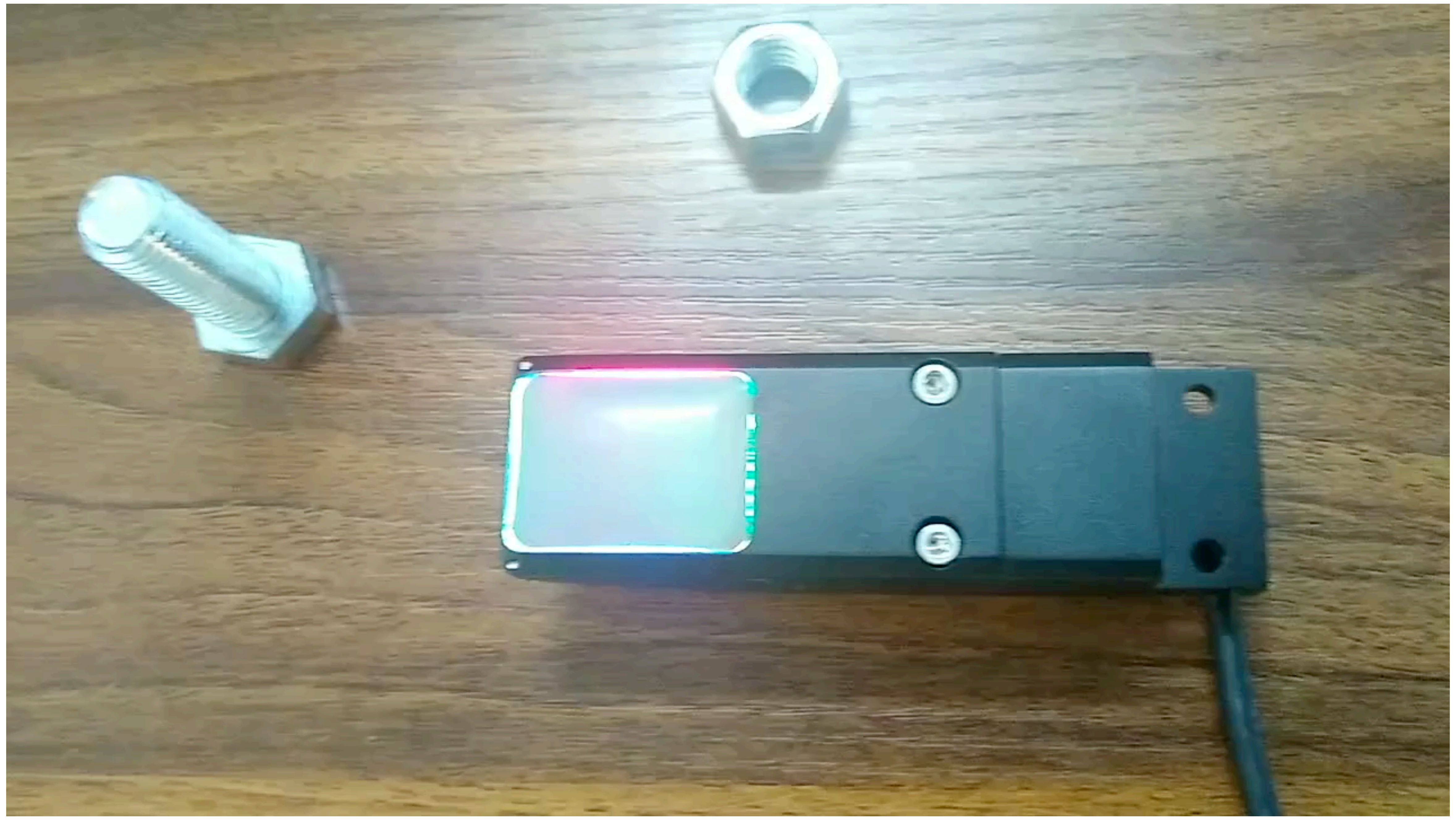
- Fast Tactile Simulation using compliant contact modeling
- Using pretrained critic with augmentation for sim2real transfer



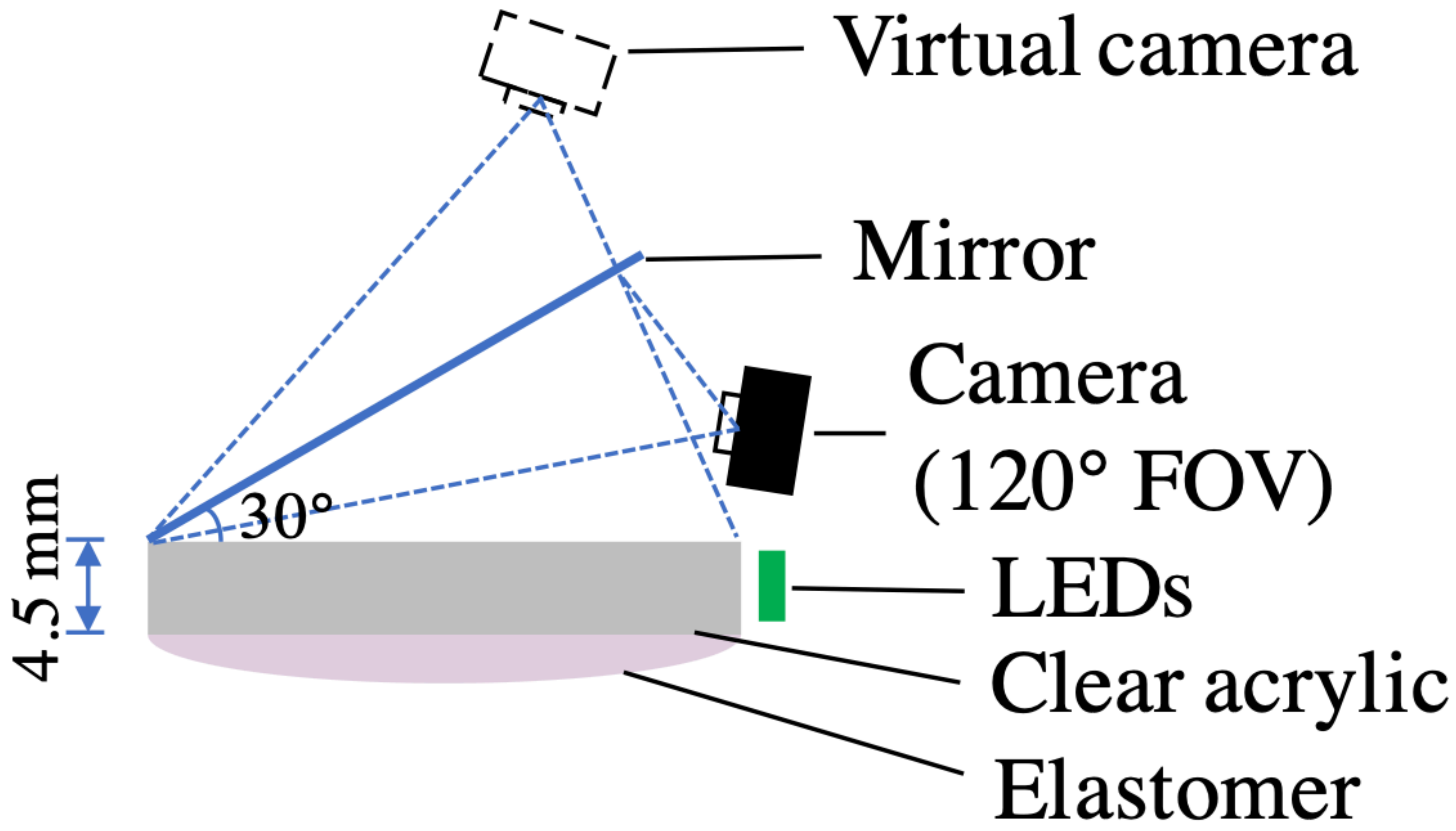
Akinola, Xu et al, TacSL: A Library for Visuotactile Sensor Simulation and Learning, T-RO 2025

Visuo-tactile sensors

High-resolution tactile sensing in Real



Real visuo-tactile sensor



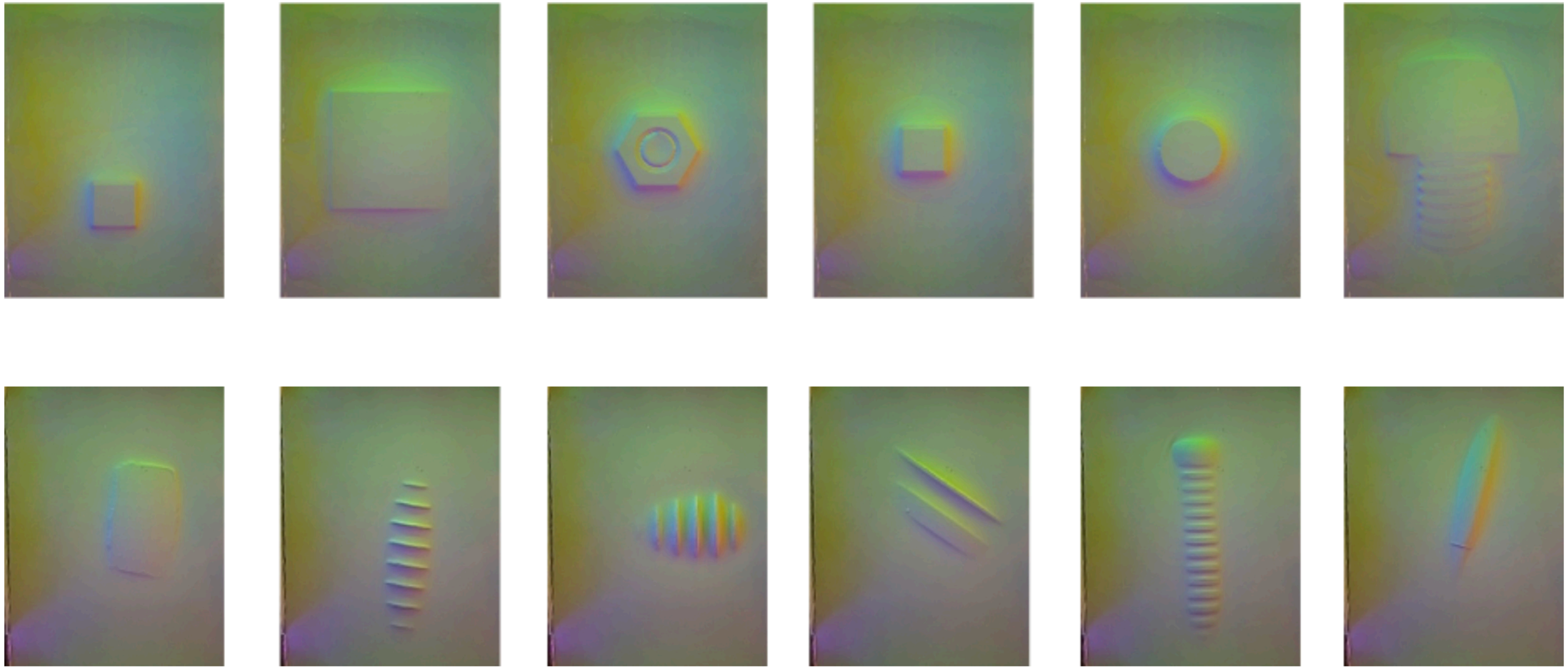
Schematic of Gelsight R1.5
Wang et al.

Simulating visuo-tactile sensors

Which row is real and which is simulated?

Simulated

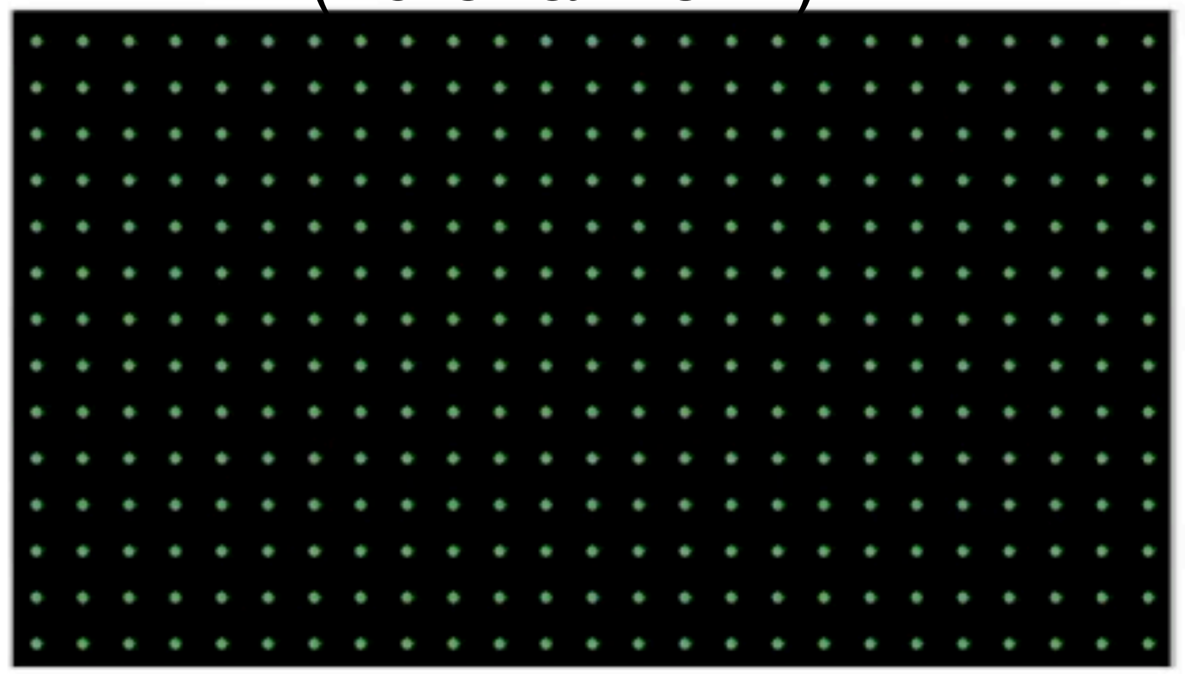
Real



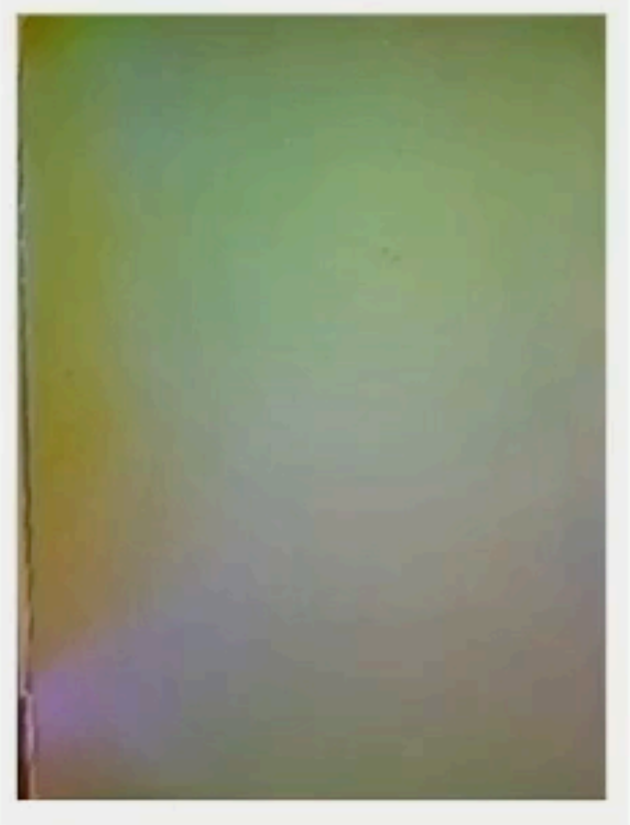
Visuo-tactile sensors

High-resolution tactile sensing in Simulation

Tactile Force Field
(Xu et al 2022)



>300x
speed up

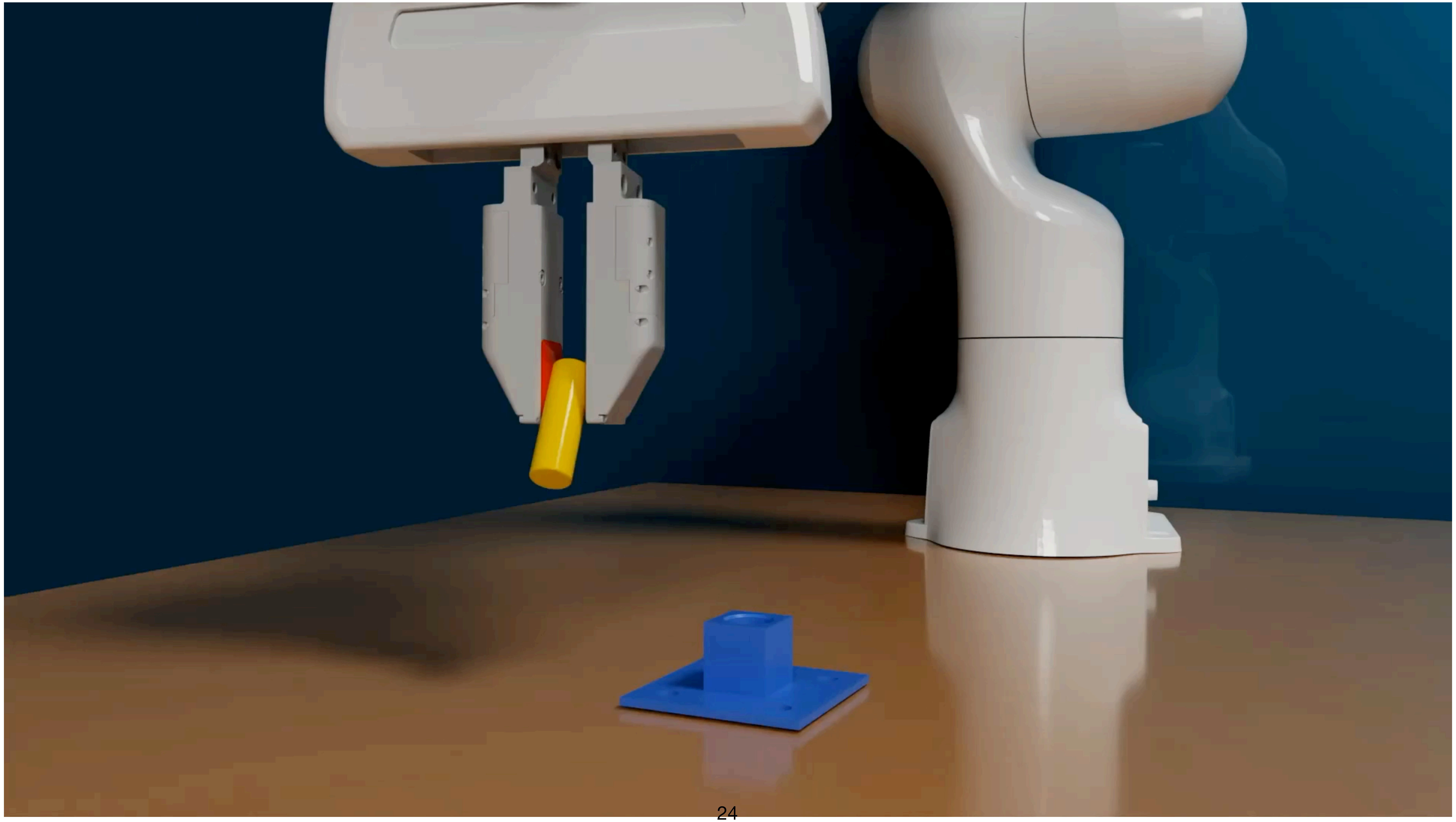


Tactile RGB (Si & Yuan 2021)

>200x
speed up

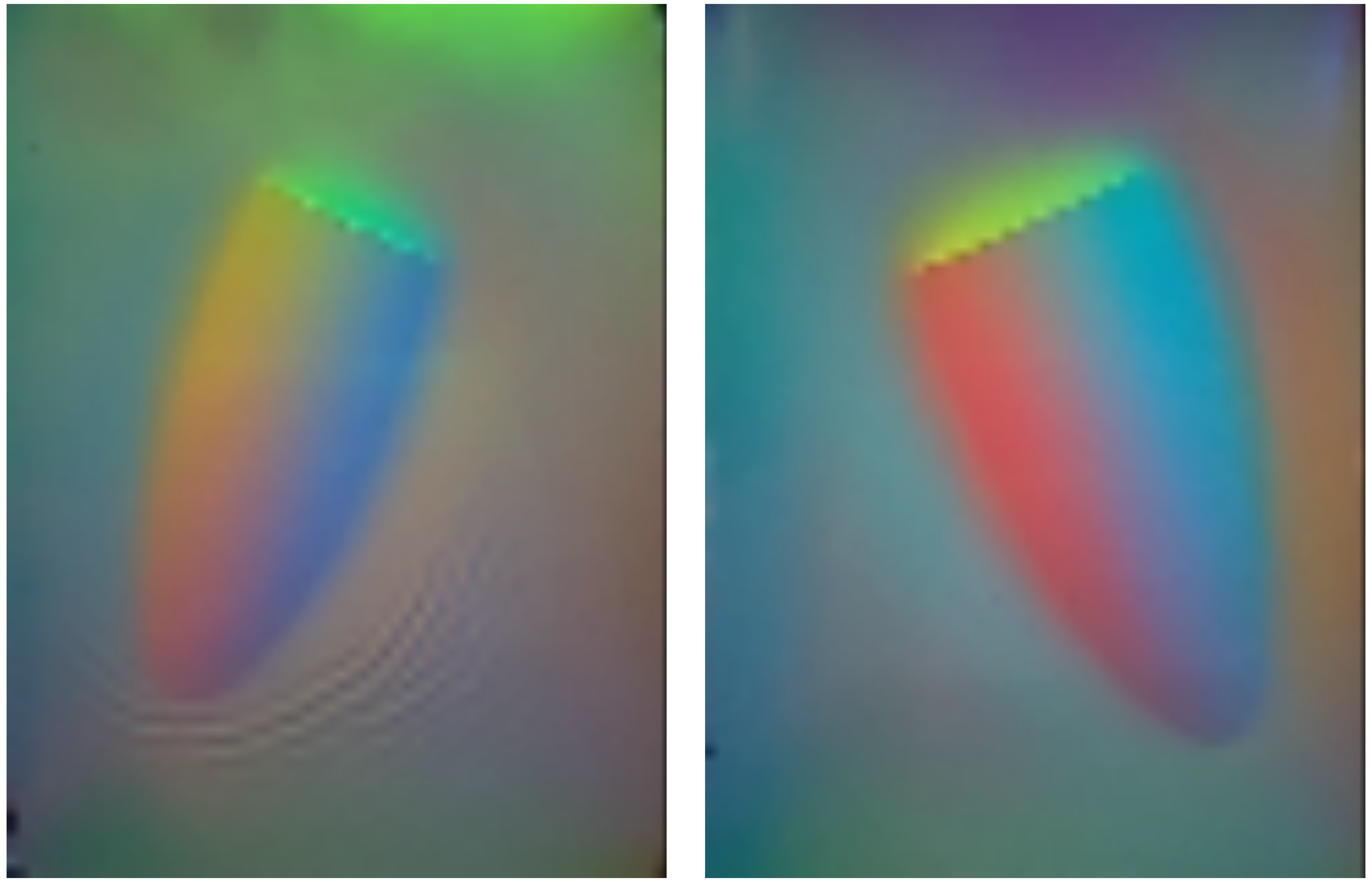
Tactile Policy Learning

Tactile policy learning in simulation

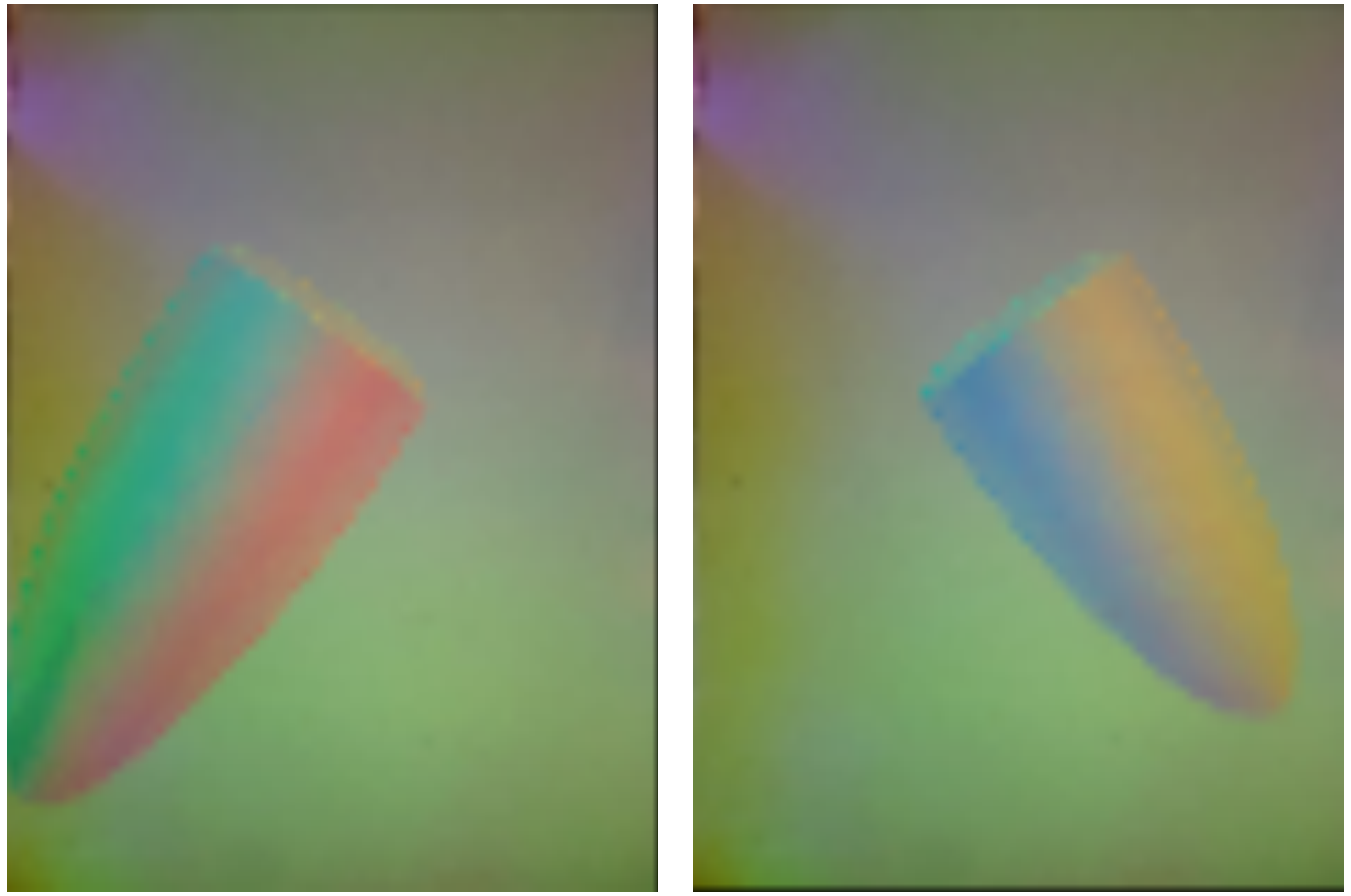


Transferring Visuo-tactile Policies from Sim to Real

Dealing with manufacturing sensor variations



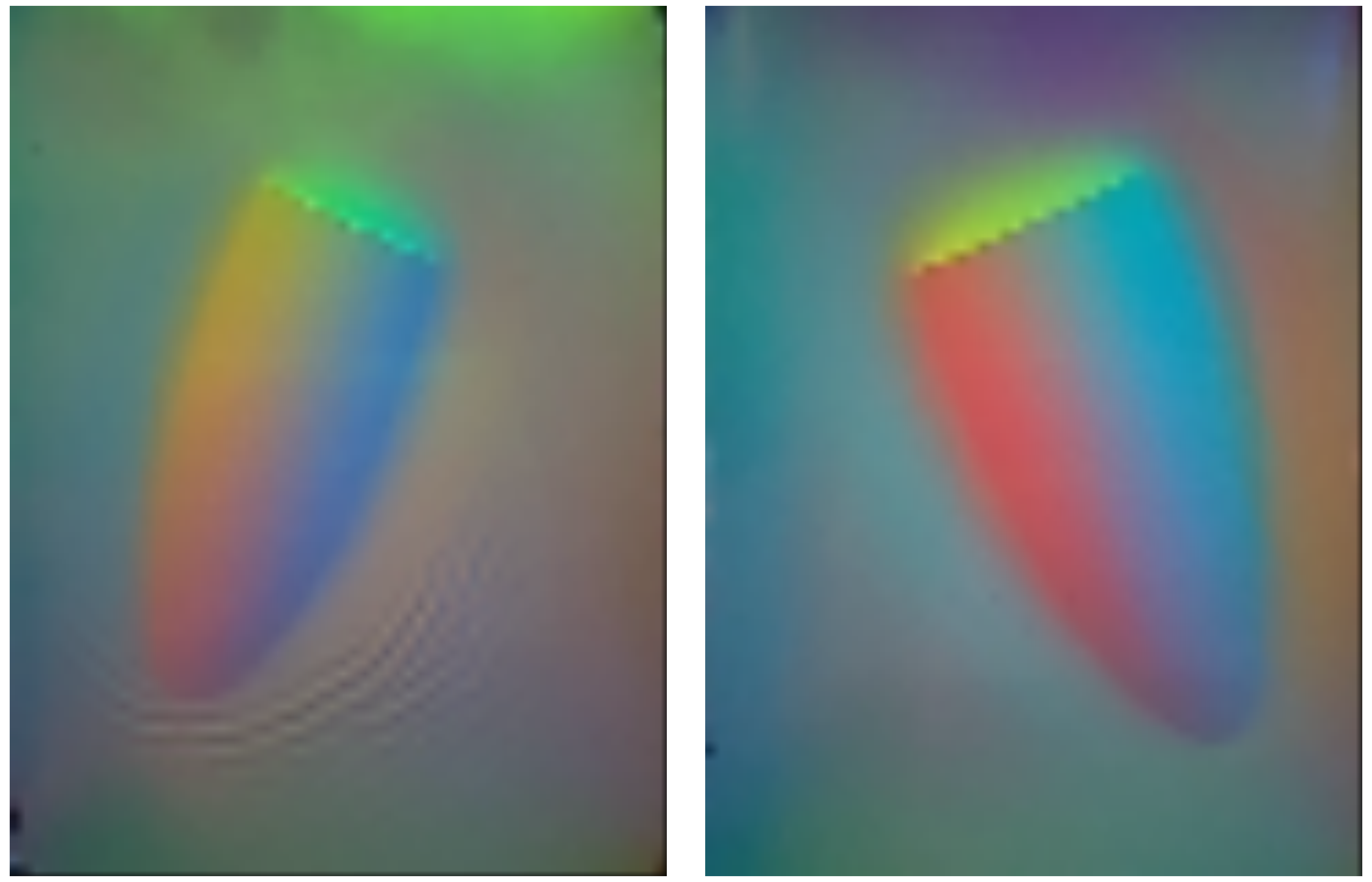
Real



Simulated

Transferring Visuo-tactile Policies from Sim to Real

Image augmentation of simulated readings during policy learning

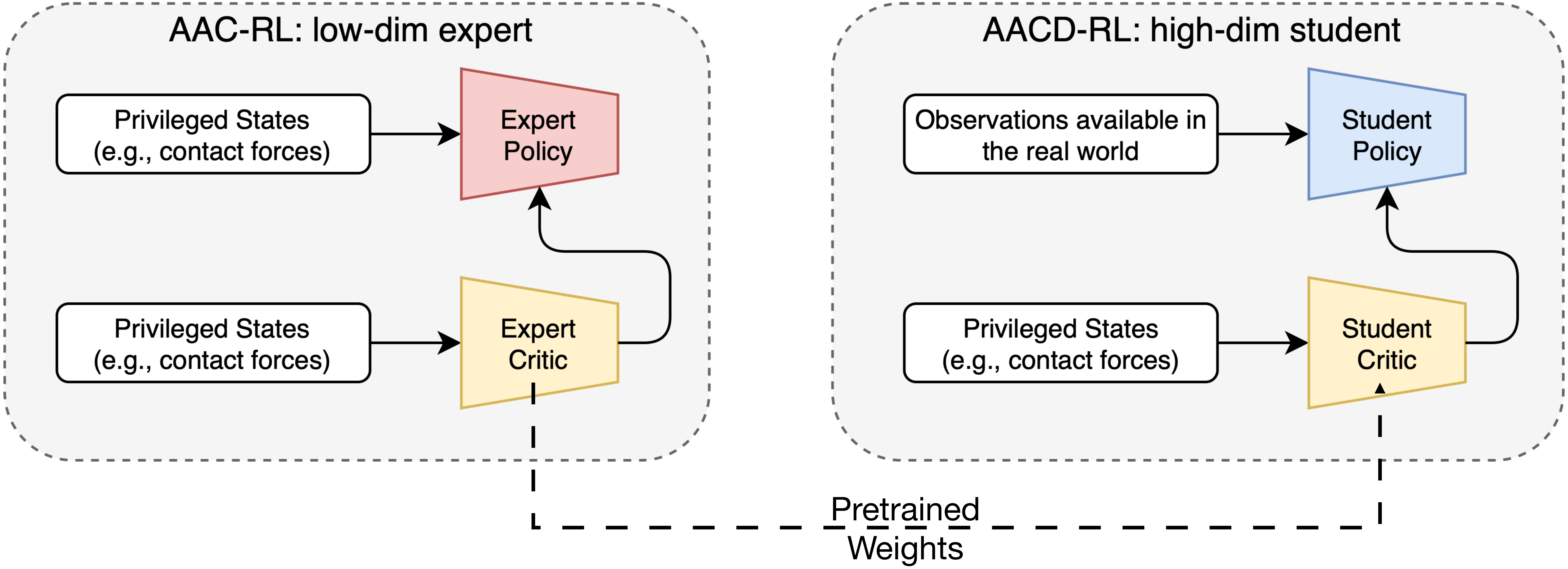


Real

Simulated

AACD Policy Learning Algorithm

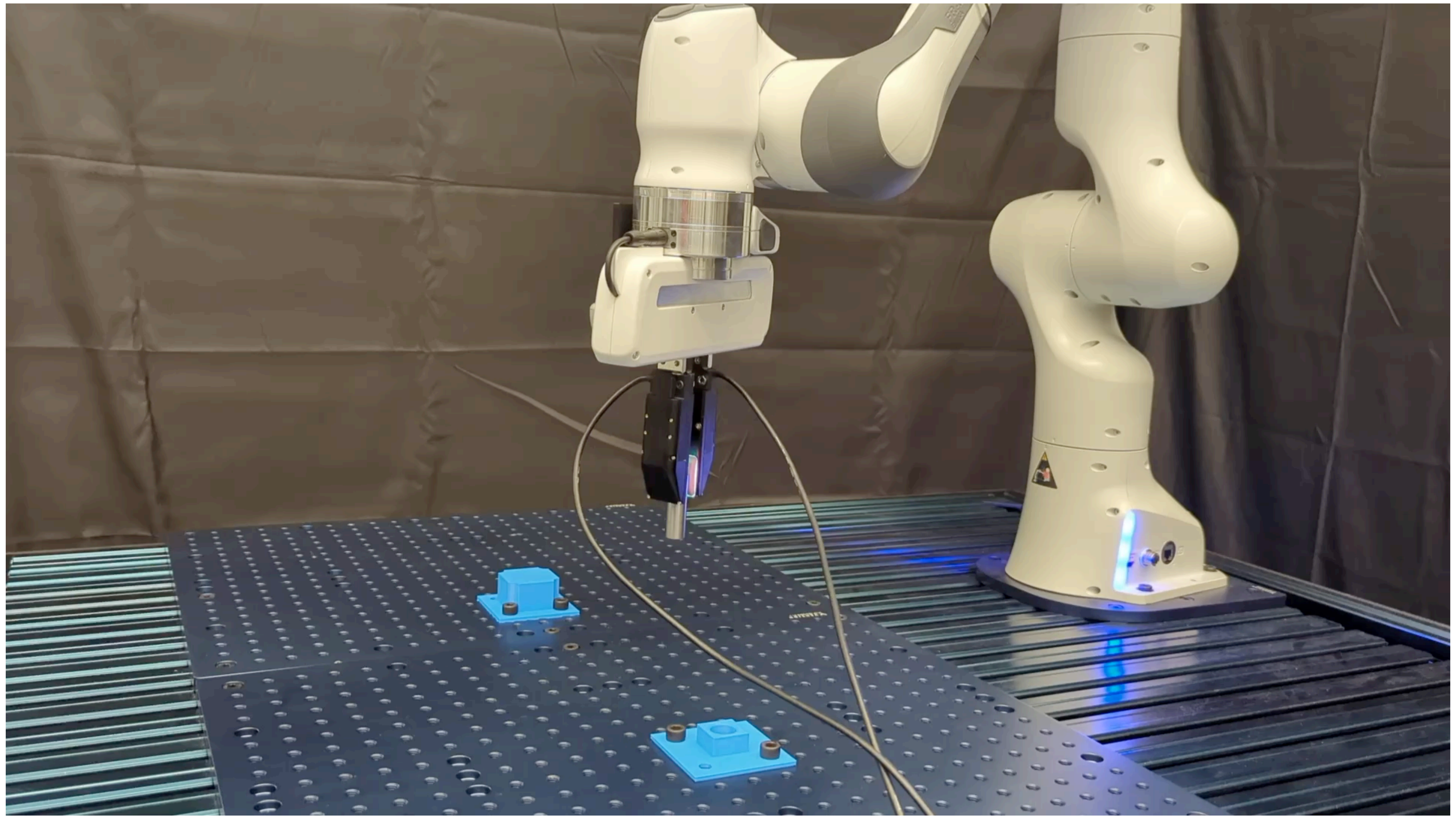
Reinforcement Learning with High-Dimensional Image Augmentation



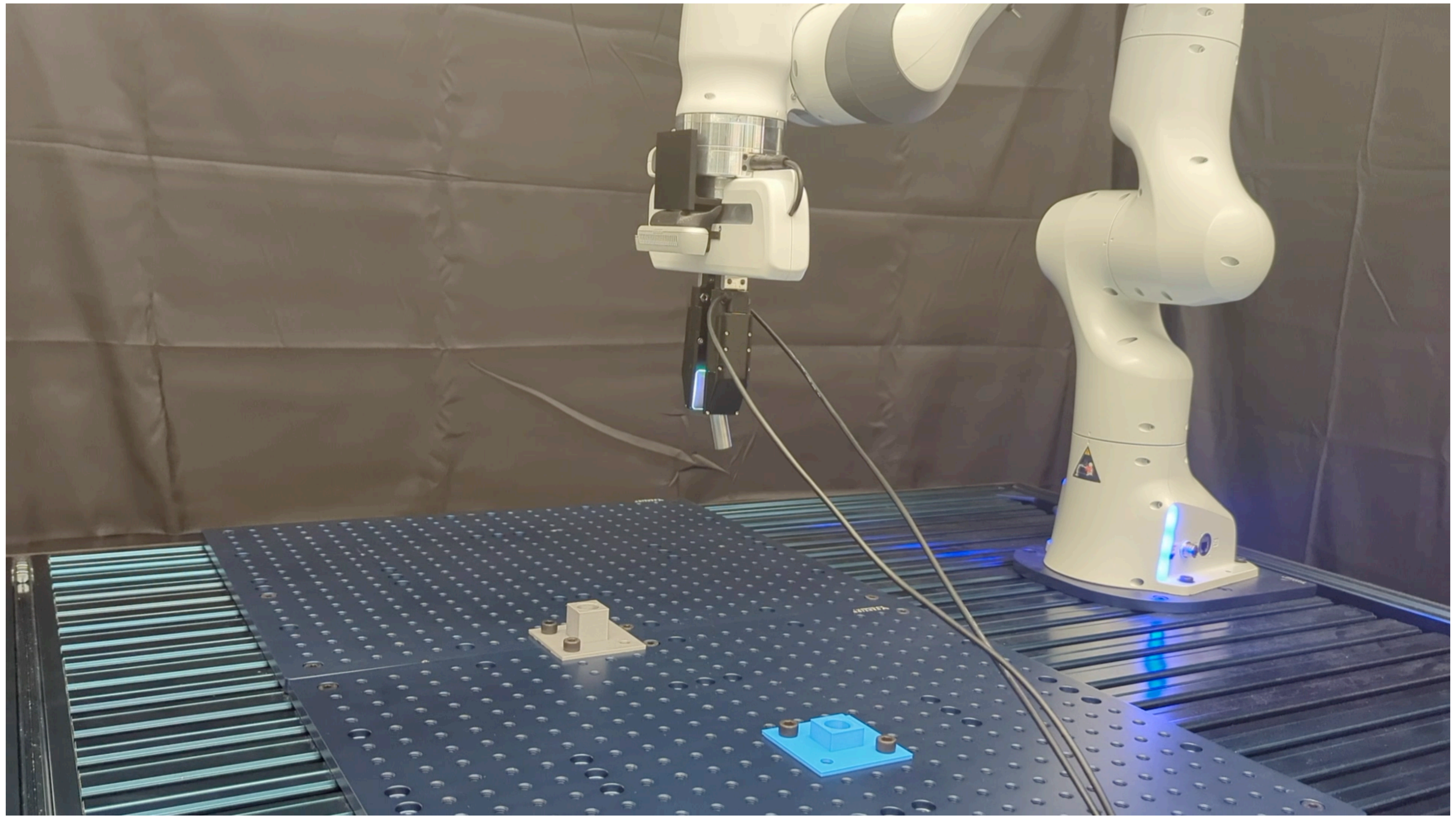
AACD leverages a pre-trained critic to guides high-dimensional RL

Tactile Policy Transfer to the real robot

Robustness to physical disturbances and acute illumination changes.



Peg-placement policy



Peg-insertion policy

We have a recipe of sorts

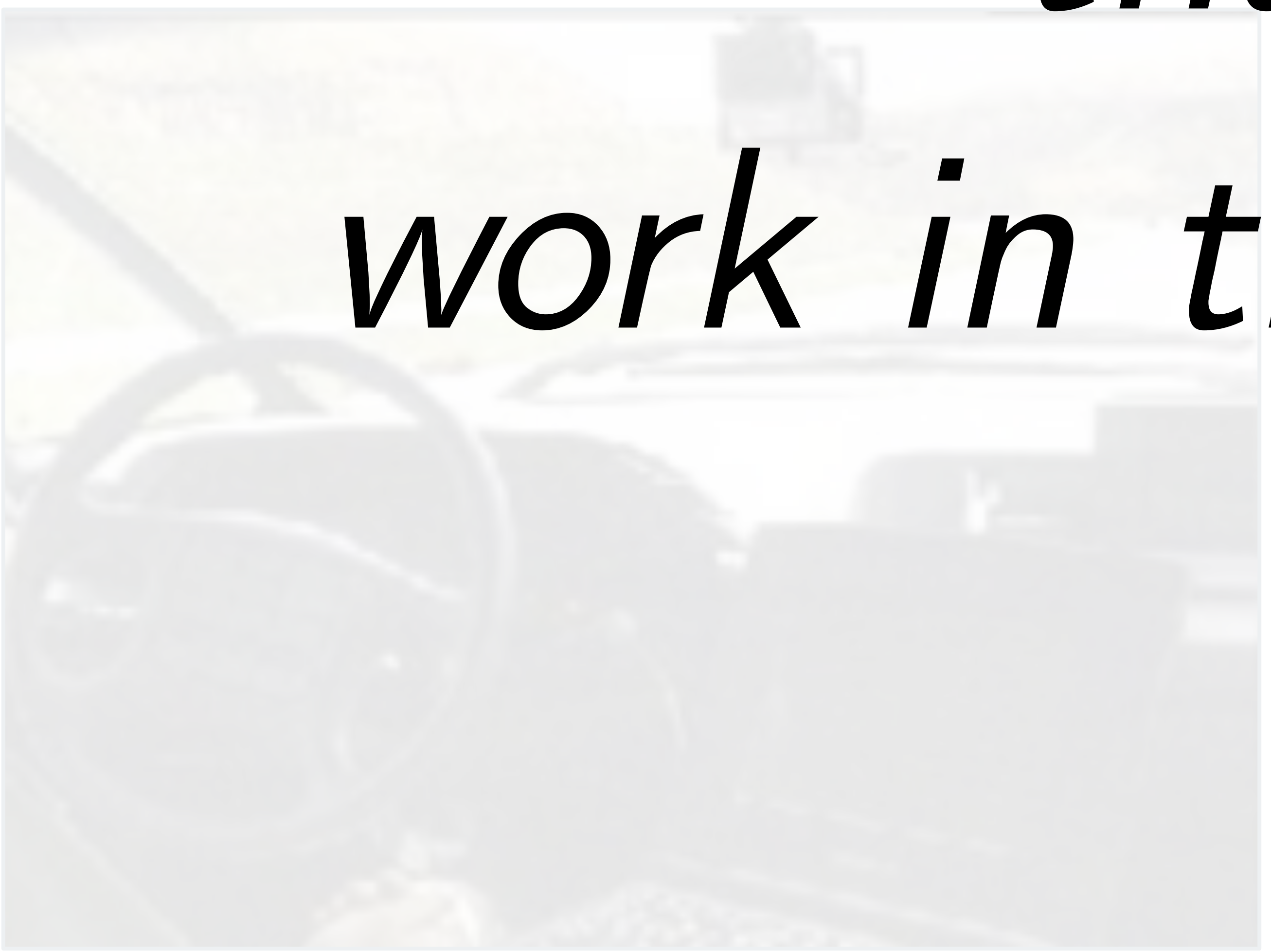
- 1) Build a simulator
- 2) Learn (or use planning) to solve for a teacher policy in simulator using whatever privileged information makes the problem easier
- Optional: learn policy that is “Bayesian Robust”/Domain Randomization
- 3) Train (on policy) a student policy that uses the modality of input the real world will provide (e.g. simulated camera images) with the teacher policy providing corrections
- Optional: use teacher critic instead of just actions
- 4) Use in real world
- Optional: RL fine tune in the real world

An Ode to Imitation Learning

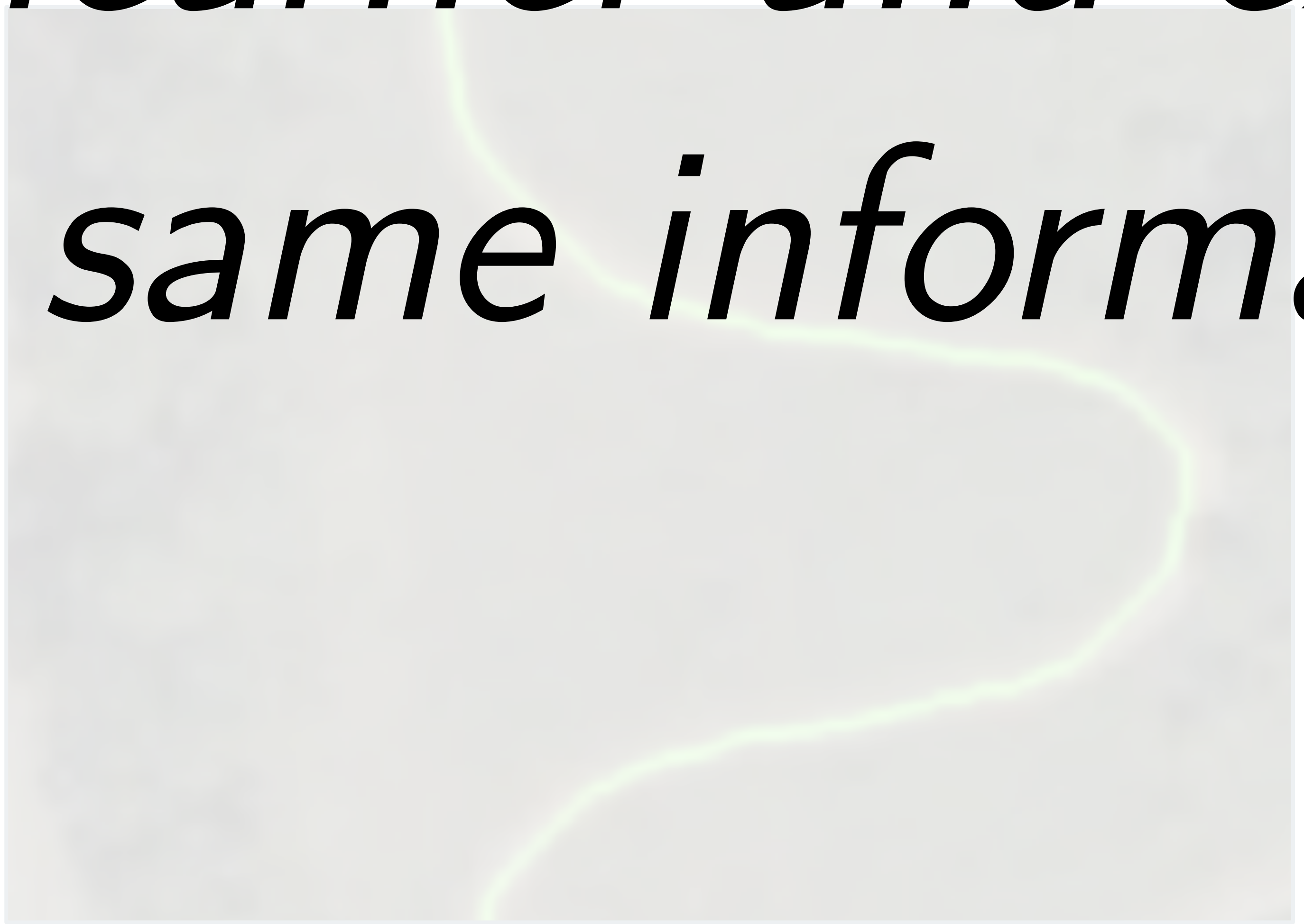
[K. Mülling et al., 2013]

[M. Zucker et al., 2011]

[A. Coates et al., '08]



[D. Pomerleau, '89]

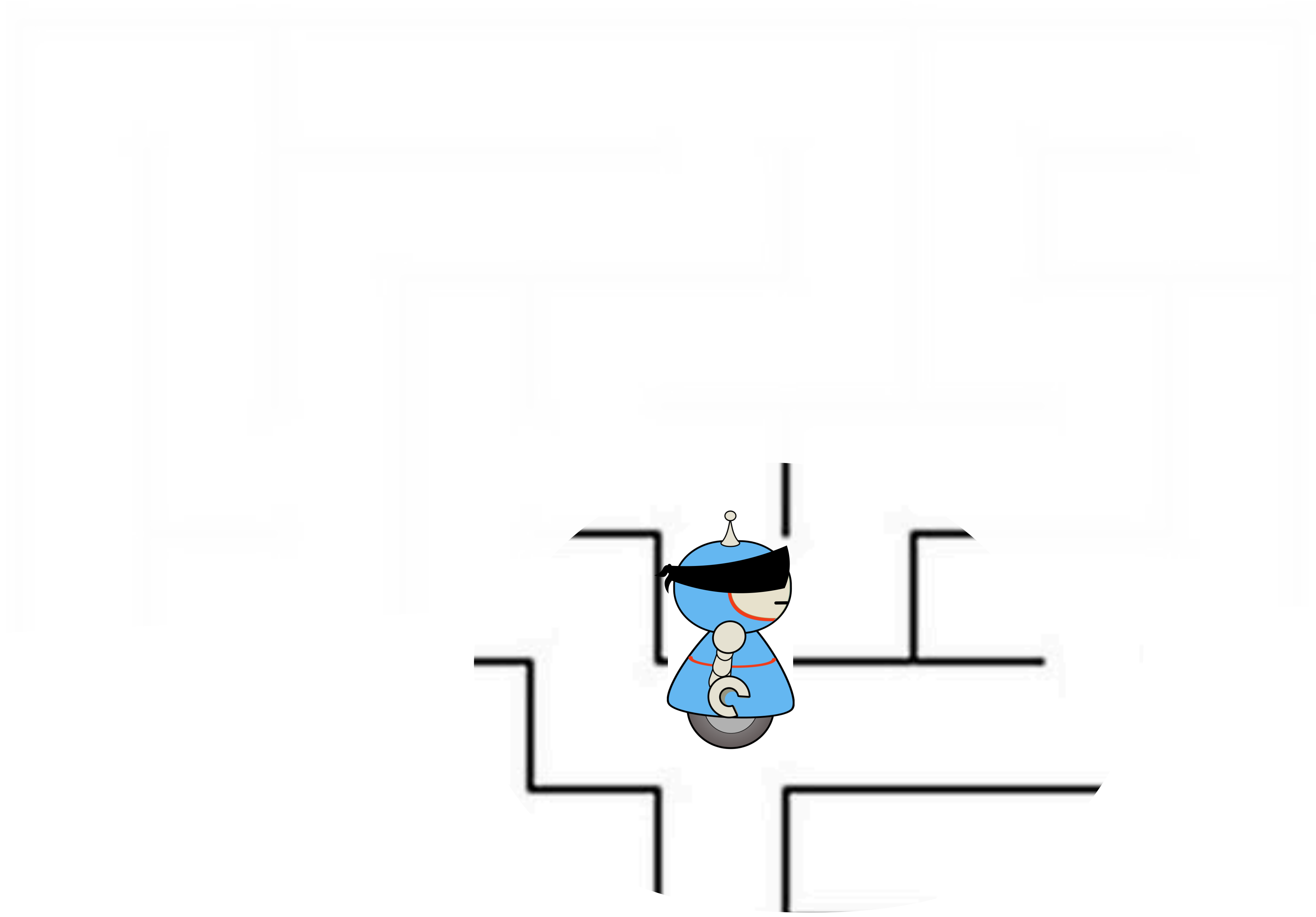


[N. Ratliff et al., 2006]

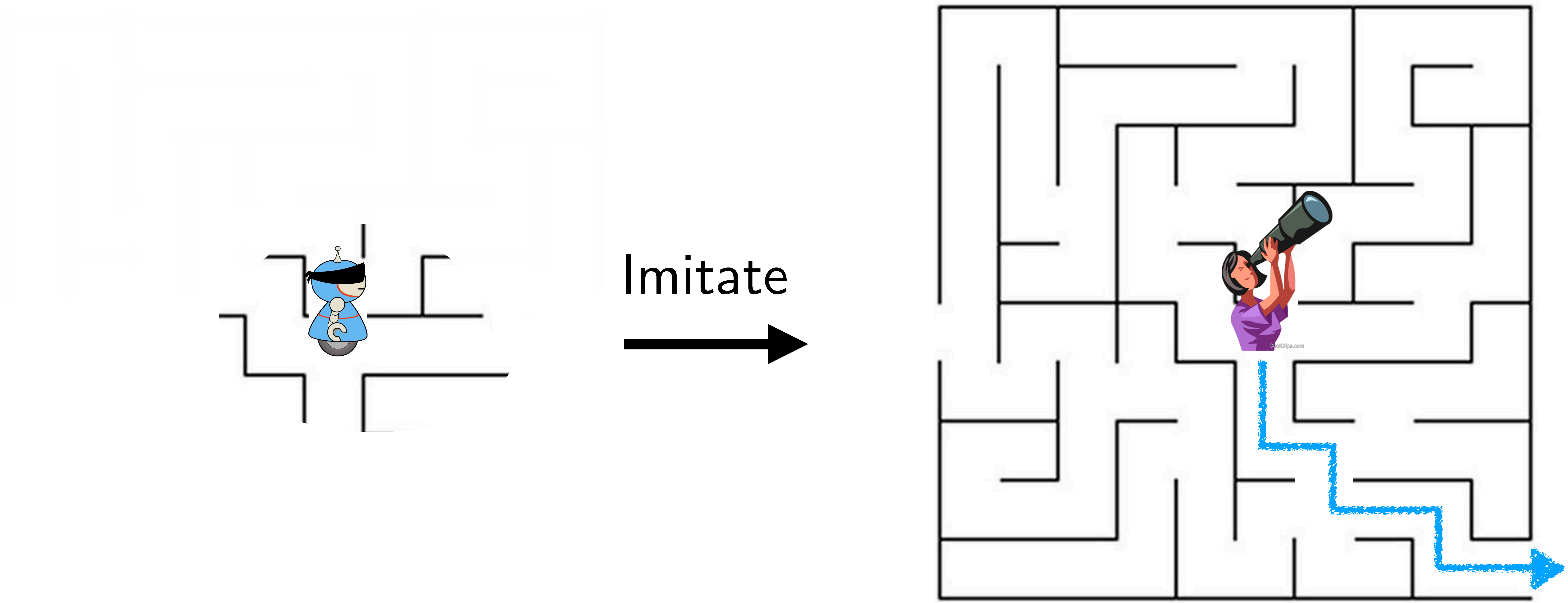
[J. A. Bagnell et al., 2010]

All of these approaches assumed that learner and expert work in the same information space

The notion of a POMDP



Imitating Experts with Privileged Information



Learner
w/ limited sensing

Expert
can see further

Contextual Markov Decision Process (MDP)

$$< S , A , \mathcal{C} , R , \mathcal{T} >$$

State Actions Context Rewards Transitions

At the beginning of each episode, a context is sampled from p(c) and is held fixed until the next reset

Context can affect both transitions and rewards

Expert sees context, but learner does not!

Just accumulate history
and do Behavior Cloning?

Just do Behavior Cloning!

1. Collect data from experts (who know the context)

$$s_0^*, a_0^*, s_1^*, a_1^*, \ldots, s_T^*$$

2. Train a policy that maps history to action

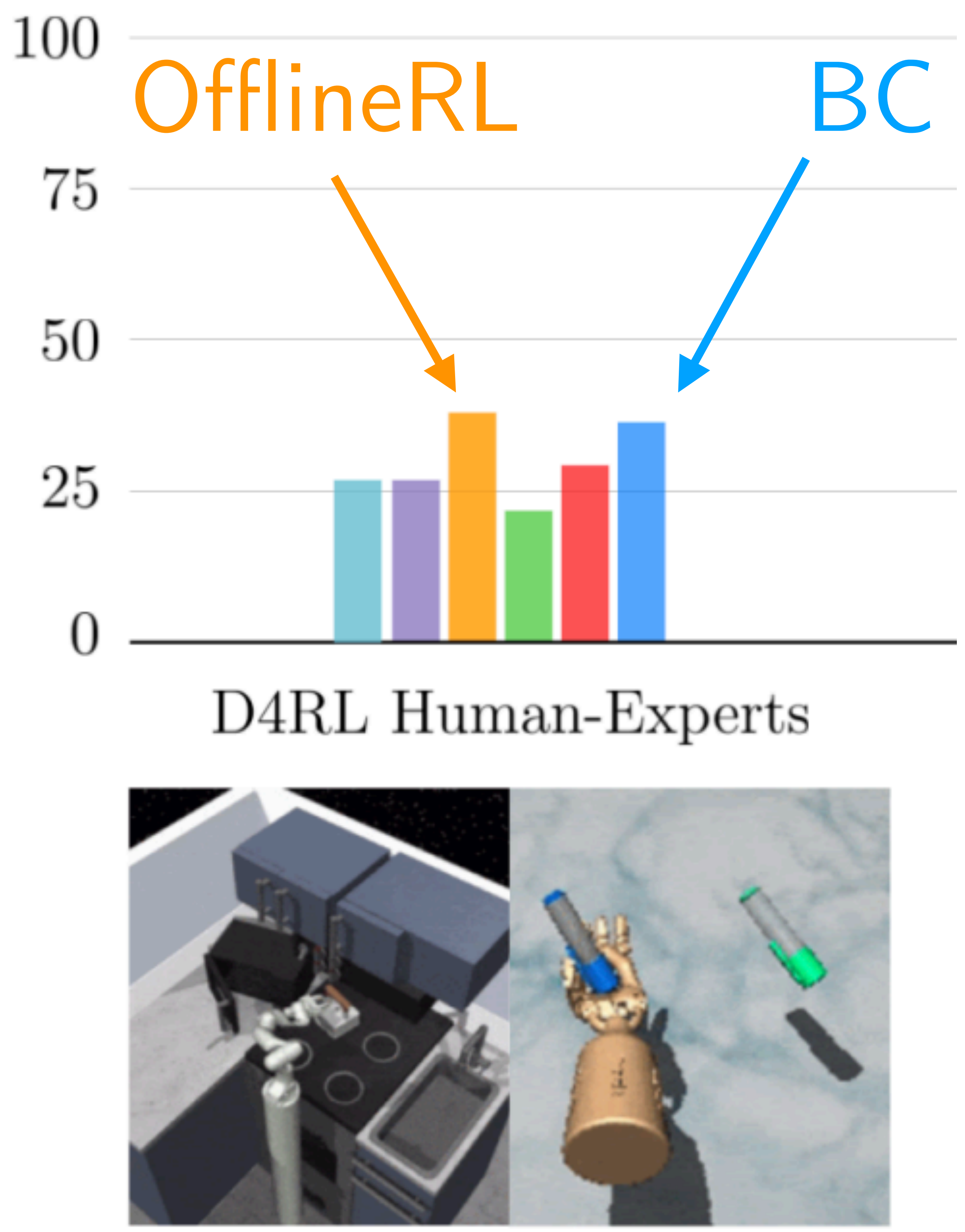
$$h_t^* = \{s_t^*, a_{t-1}^*, s_{t-1}^*, \ldots, s_{t-k}^*\} \qquad \pi : h_t^* \rightarrow a_t^*$$

Rationale: Sure we'll make errors in the beginning, but we will always be recoverable and asymptotically imitate the expert

Behavior cloning mostly works fine?

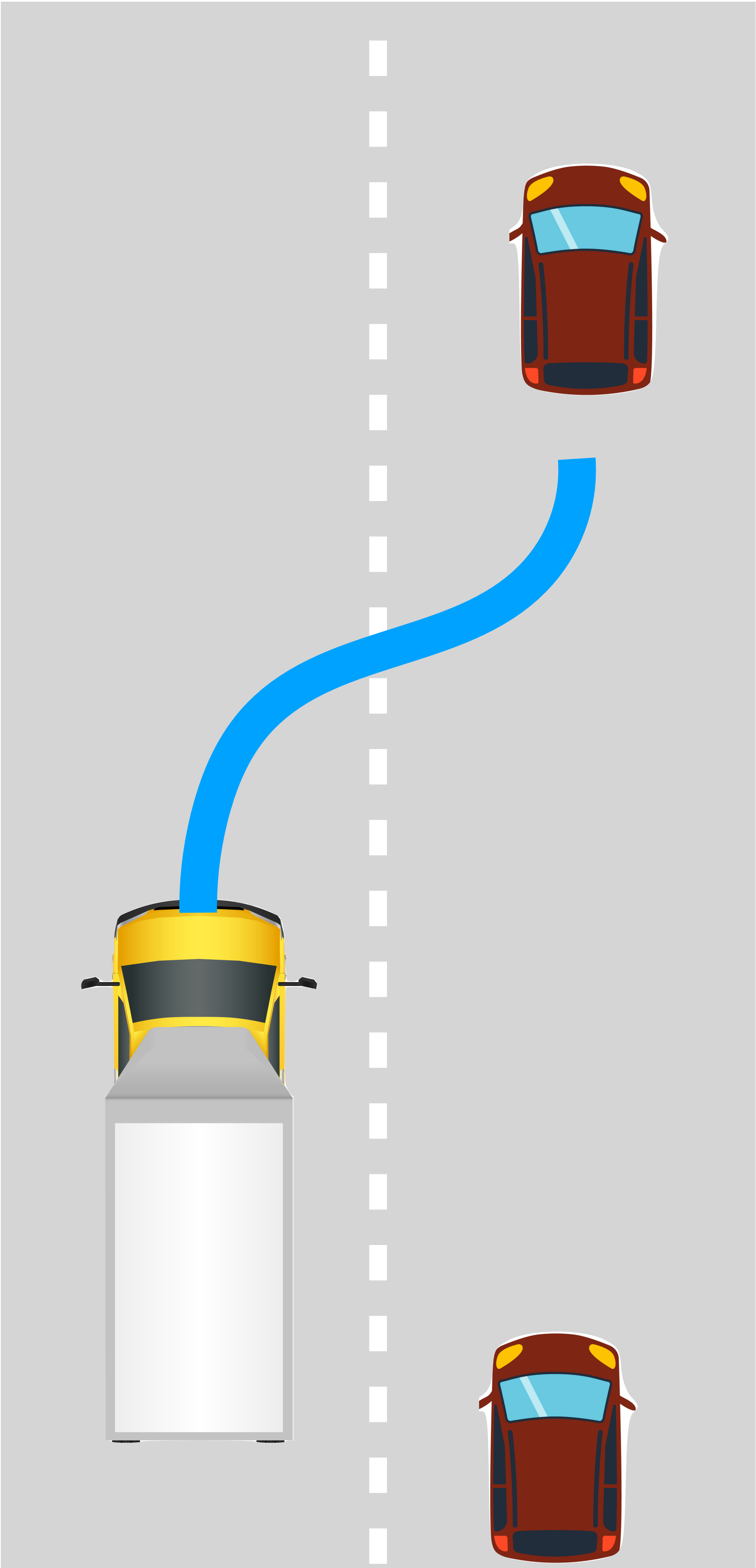
Environment	Expert	BC
CartPole	500 ± 0	500 ± 0
Acrobot	-71.7 ± 11.5	-78.4 ± 14.2
MountainCar	-99.6 ± 10.9	-107.8 ± 16.4
Hopper	3554 ± 216	3258 ± 396
Walker2d	5496 ± 89	5349 ± 634
HalfCheetah	4487 ± 164	4605 ± 143
Ant	4186 ± 1081	3353 ± 1801

[SCV+ arXiv '21]



[Florence et al. '21]

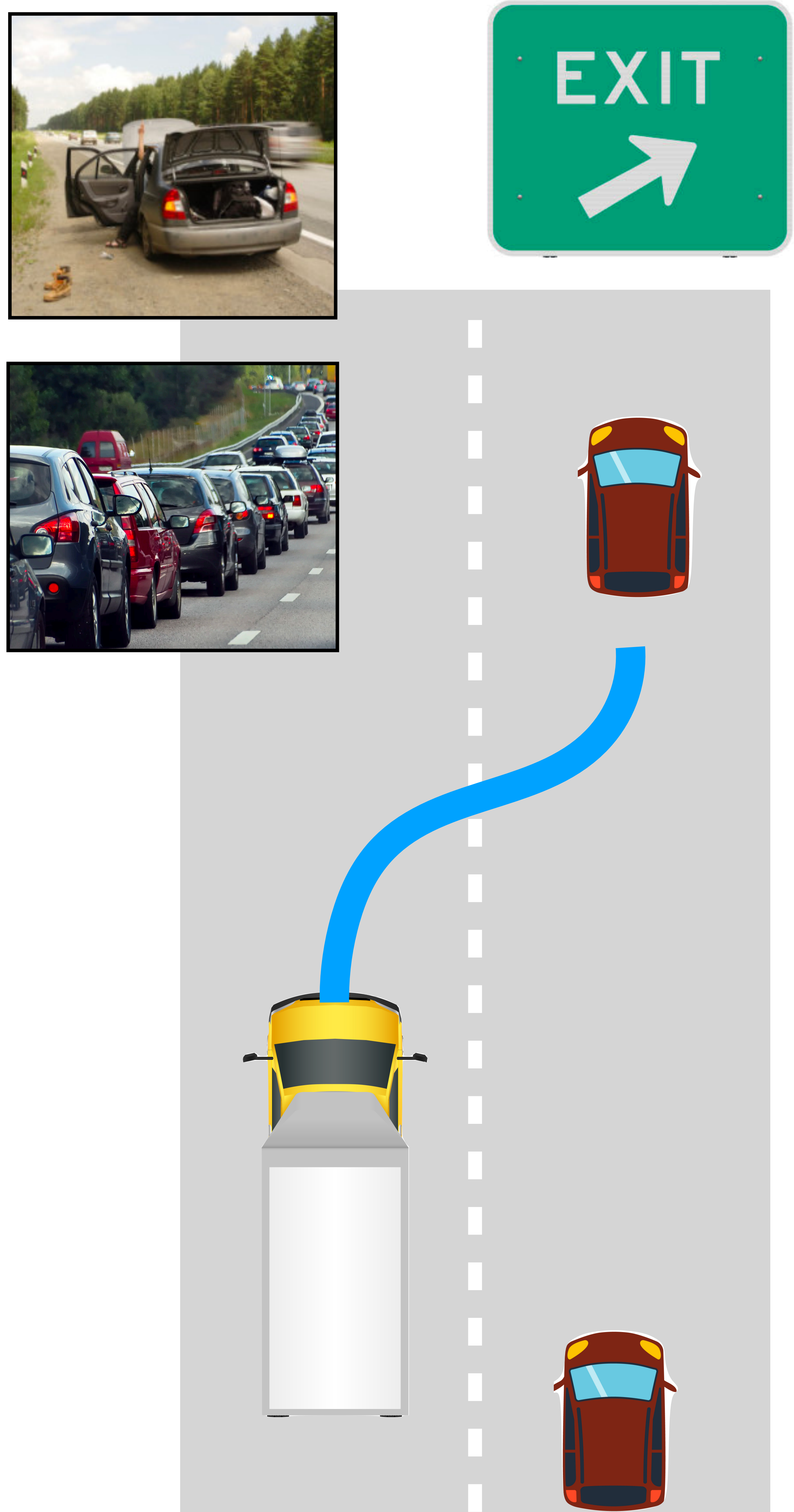
In NLP, standard practice is to do Teacher Forcing ...



Tales from the Road:

A curious case of belligerent lane changing

Example: Learning to Lane Change



Features

- Distance to exit
- Disabled vehicle on shoulder?
- Traffic congestion level?

⋮

- Past action (Y / N)

Action

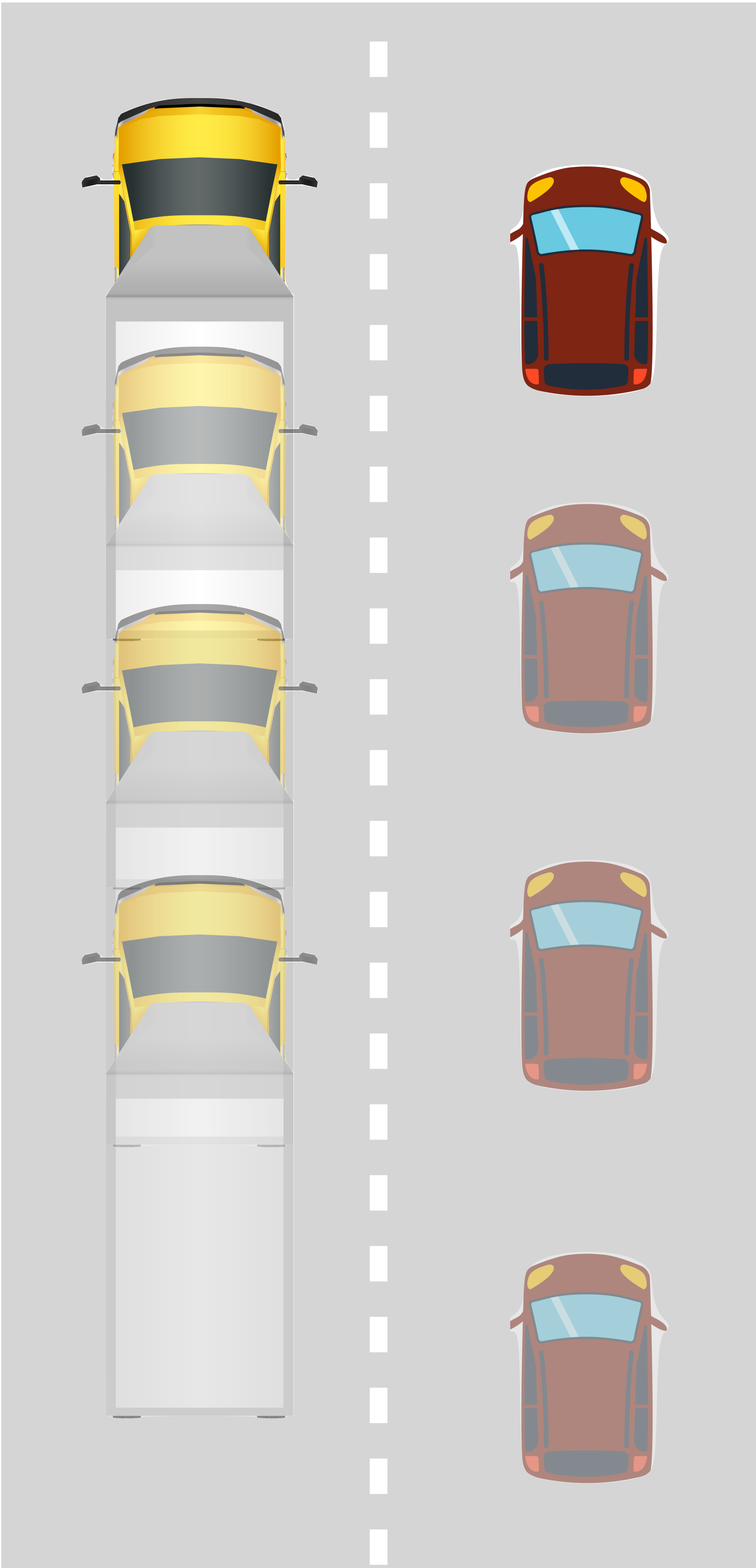
- Should I execute lane change? (Y / N)

Just do Behavior Cloning!

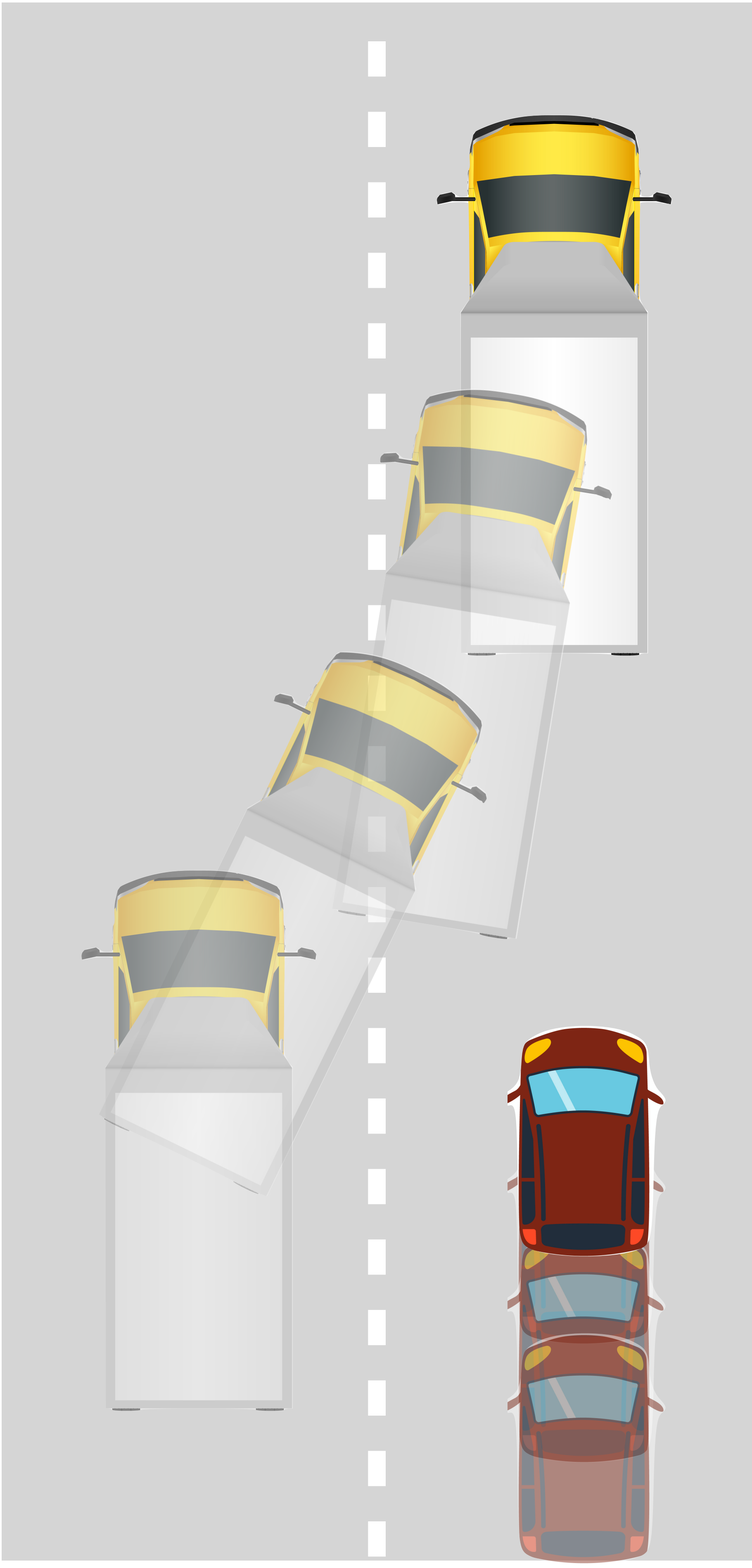
[Pomerleau'91]

Train Data (Human Demonstrations)

N
N
N
N



Y
Y
Y
Y



1. Collect data of humans lane changing

2. Train a classifier

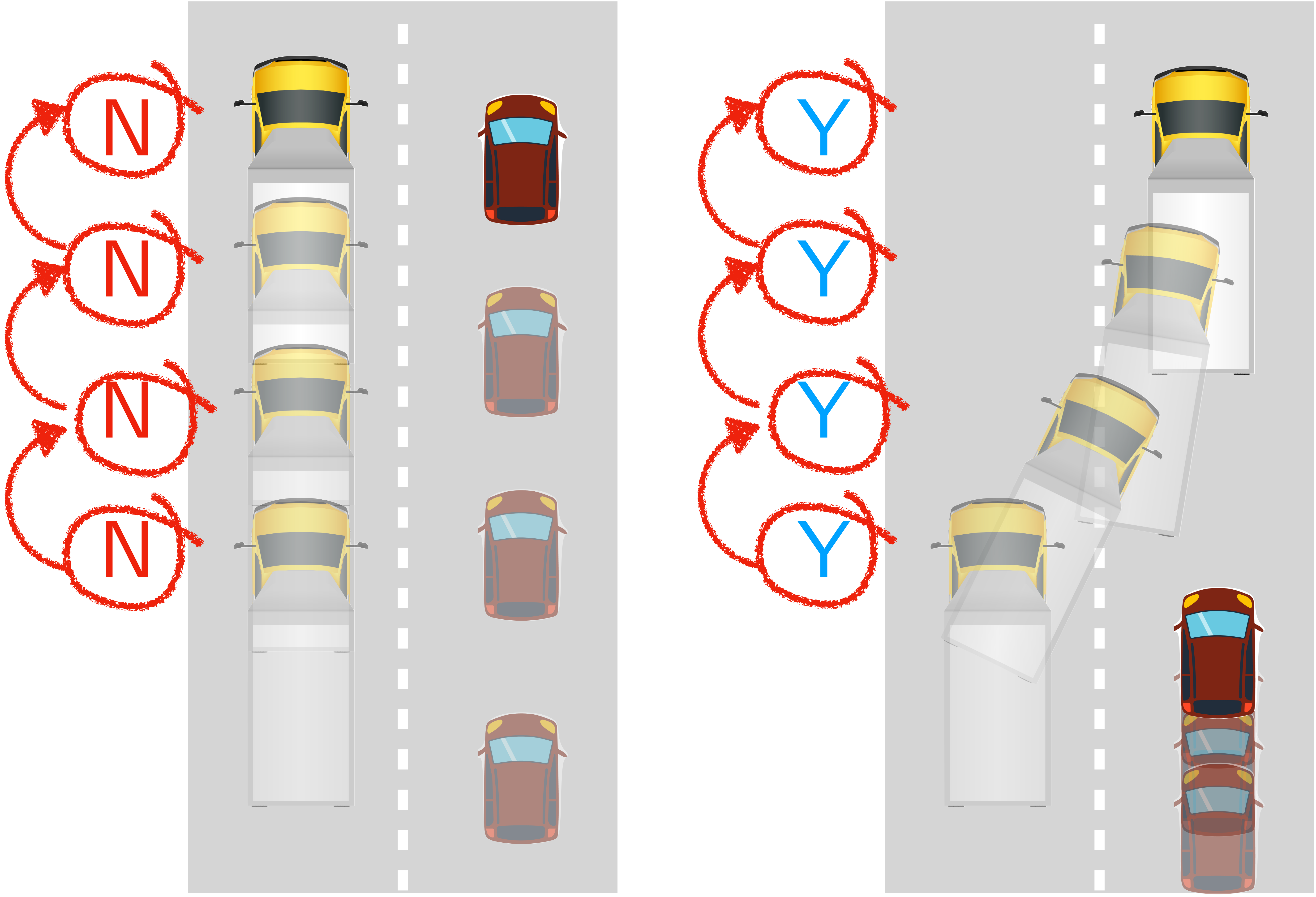
99% accuracy!!

What happens at test time ...

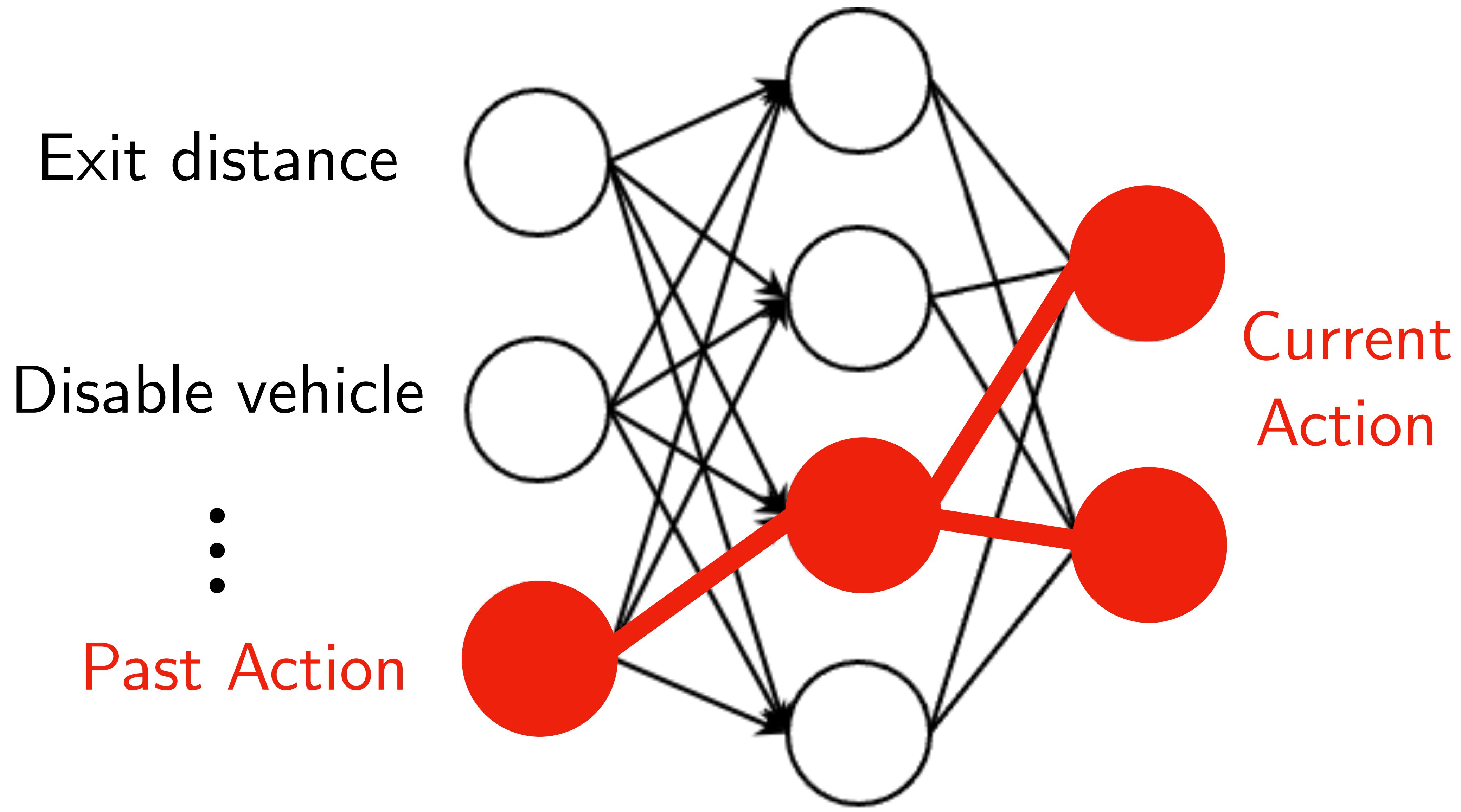
Why didn't we abort the lane change?

Train Distribution

(human driving)



Learnt Policy



"Do what I did in previous cycle"

Why didn't we abort the lane change?

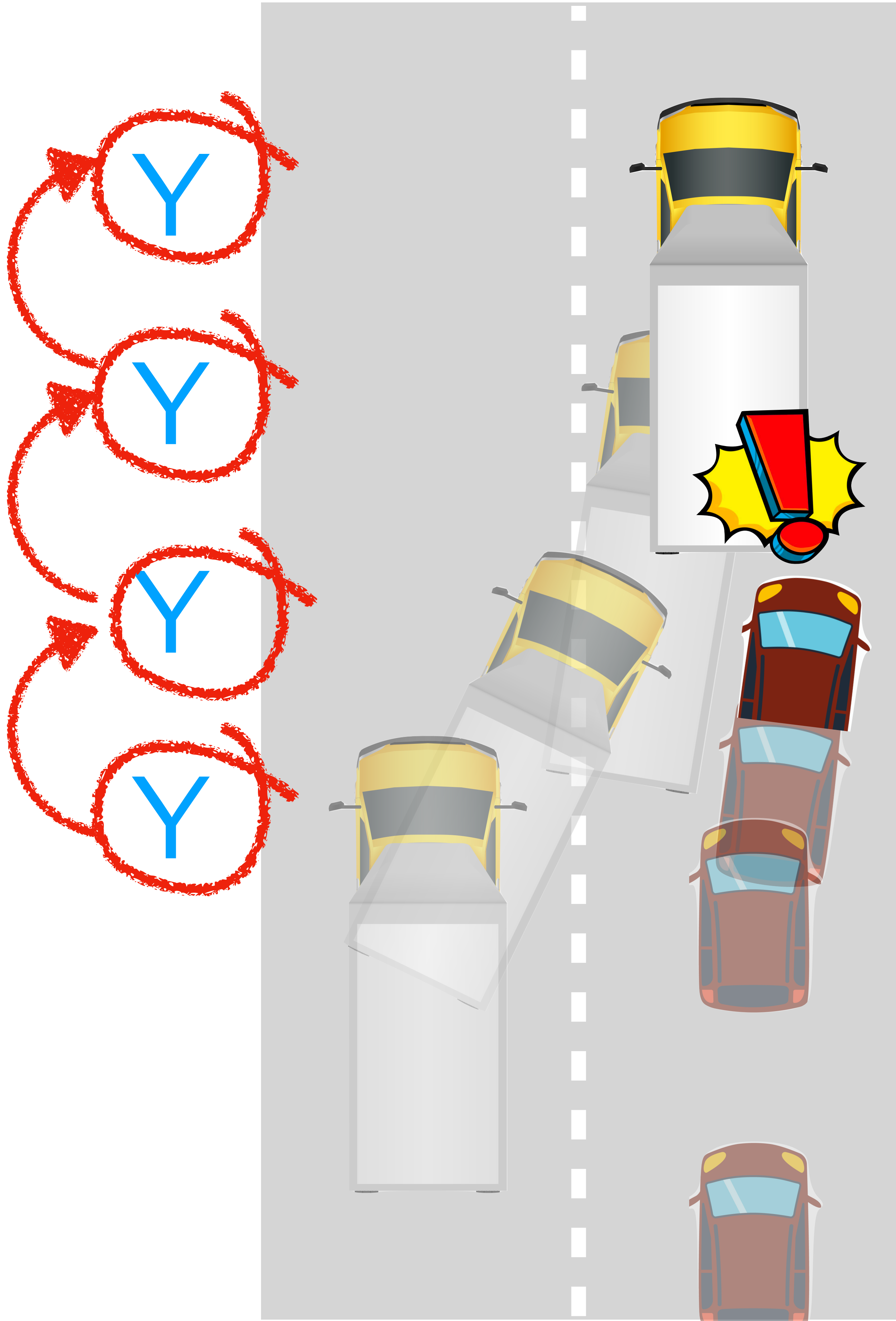
Test Distribution $\neq$ Train

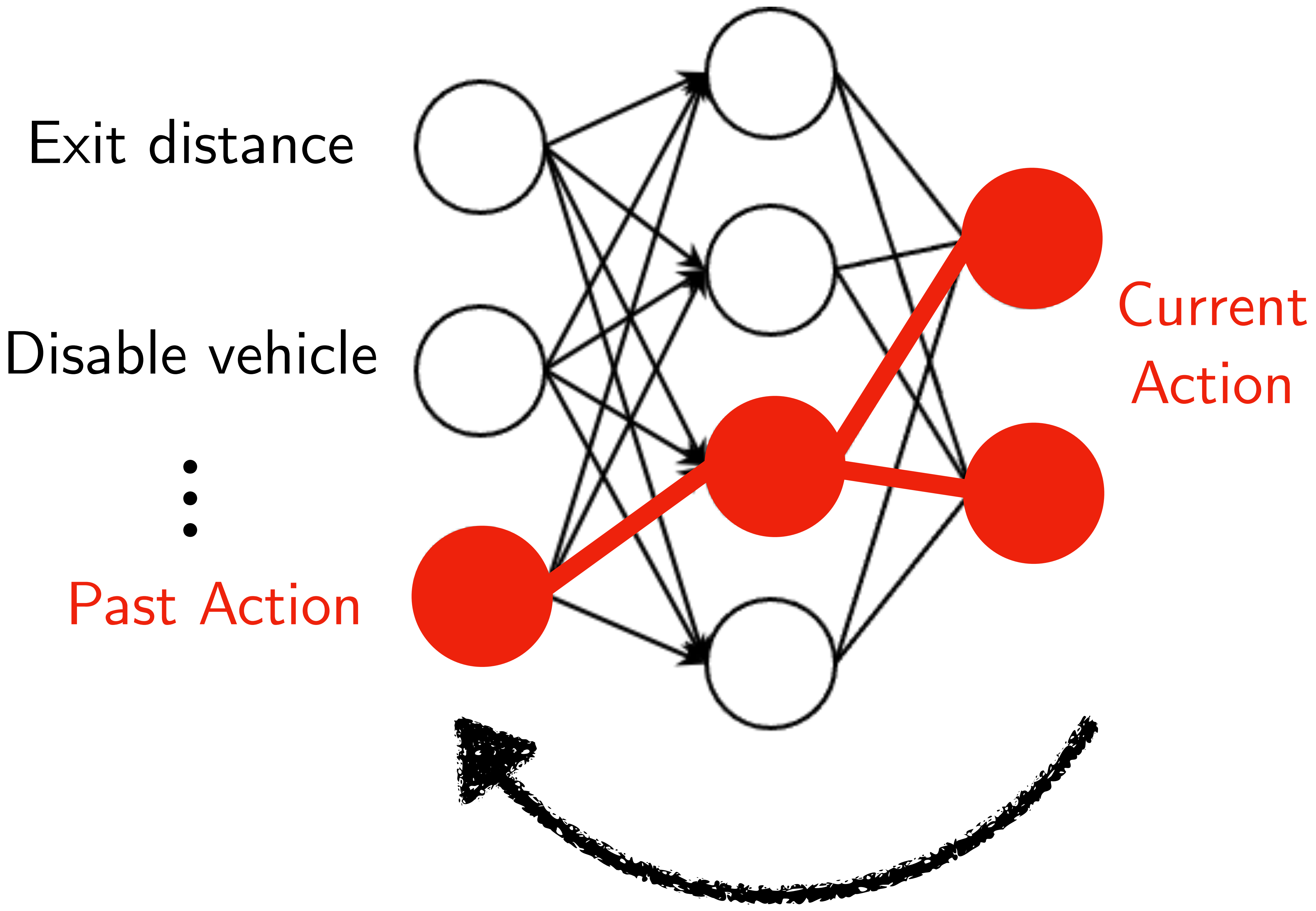
(robot driving)

Latching Effect

where the learner repeats past action

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





Feedback!

Feedback drives covariate shift

Creates a "Latching effect"

"Latching Effect" in self-driving

"... the inertia problem. When the ego vehicle is stopped (e.g., at a red traffic light), the probability it stays static is indeed overwhelming in the training data. This creates a spurious correlation between low speed and no acceleration, inducing excessive stopping and difficult restarting in the imitative policy ..."

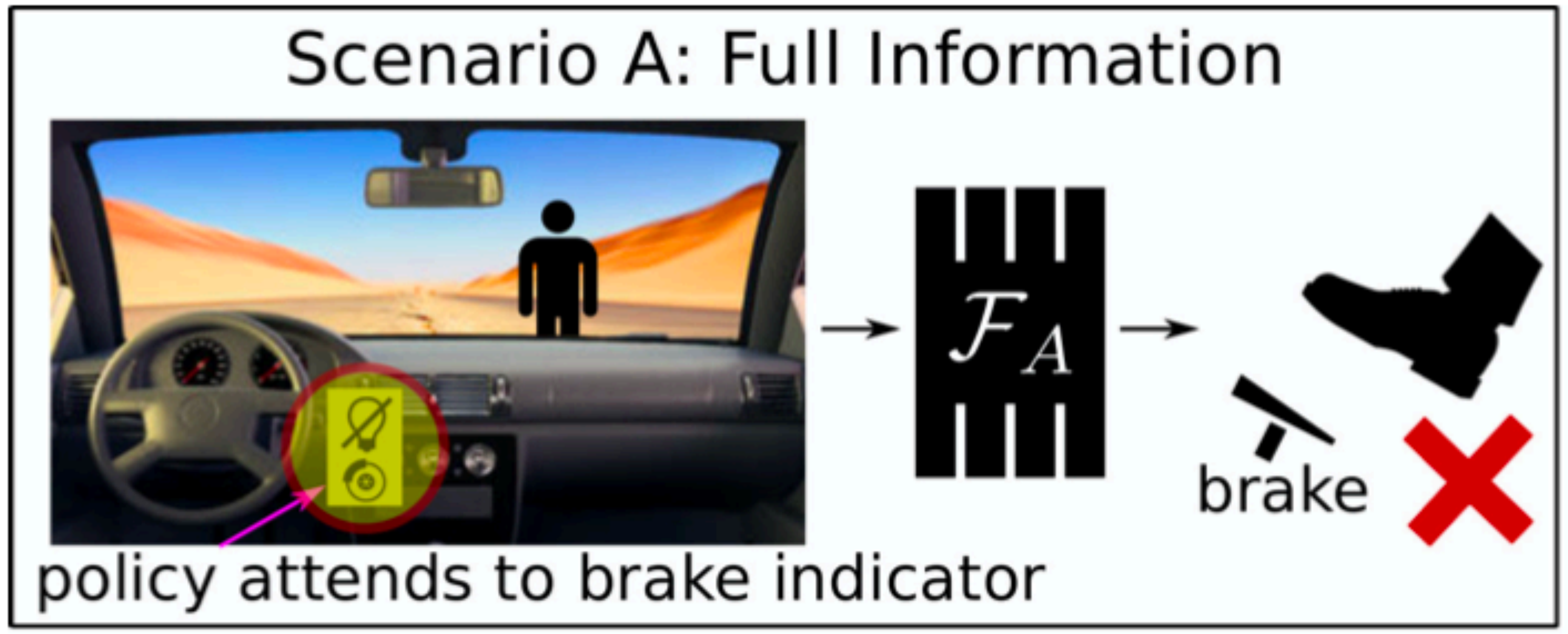
"Exploring the Limitations of Behavior Cloning for Autonomous Driving." F. Codevilla, E. Santana, A. M. Lopez, A. Gaidon. ICCV 2019

"... small errors in action predictions to compound over time, eventually leading to states that human drivers infrequently visit and are not adequately covered by the training data. Poorer predictions can cause a feedback cycle known as cascading errors ..."

"Imitating Driver Behavior with Generative Adversarial Networks". A. Kuefler, J. Morton, T. Wheeler, M. Kochenderfer, IV 2017

"... During closed-loop inference, this breaks down because the past history is from the net's own past predictions. For example, such a trained net may learn to only stop for a stop sign if it sees a deceleration in the past history, and will therefore never stop for a stop sign during closed-loop inference ..."

"ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst". M. Bansal, A. Krizhevsky, A. Ogale, Waymo 2018



"Causal Confusion in Imitation Learning". P. de Haan, D. Jayaraman, S. Levine, NeurIPS '19

An old problem in self-driving

"Using multiple successive frames as input would seem like a good idea since the multiple views resulting from ego-motion facilitates the segmentation and detection of nearby obstacles ... the current rate of turn is an excellent predictor of the next desired steering angle ... Hence, a system trained with multiple frames would merely predict a steering angle equal to the current rate of turn as observed through the camera. This would lead to catastrophic behavior in test mode. The robot would simply turn in circles."

"Off-Road Obstacle Avoidance through End-to-End Learning"
Y. LeCun, U. Muller, J. Ben, E.Cosatto, B.Flepp, NeurIPS 2005

Latching effect in NLP

Beam Search

...to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and...

"The probability of a repeated phrase increases with each repetition, creating a positive feedback loop"

The curious case of neural text de-generation Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2019).

"The main problem is that mistakes made early in the sequence generation process are fed as input to the model and can be quickly amplified because the model might be in a part of the state space it has never seen at training time."

"Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks." Bengio, S., Vinyals, O., Jaitly, N., & Shazeer, N. (2015).

Thus, the model trained with teacher forcing may over-rely on previously predicted words, which would exacerbate error propagation

"On exposure bias, hallucination and domain shift in neural machine translation." Wang, C., & Sennrich, R. (2020).

DeepMind

Technical Report

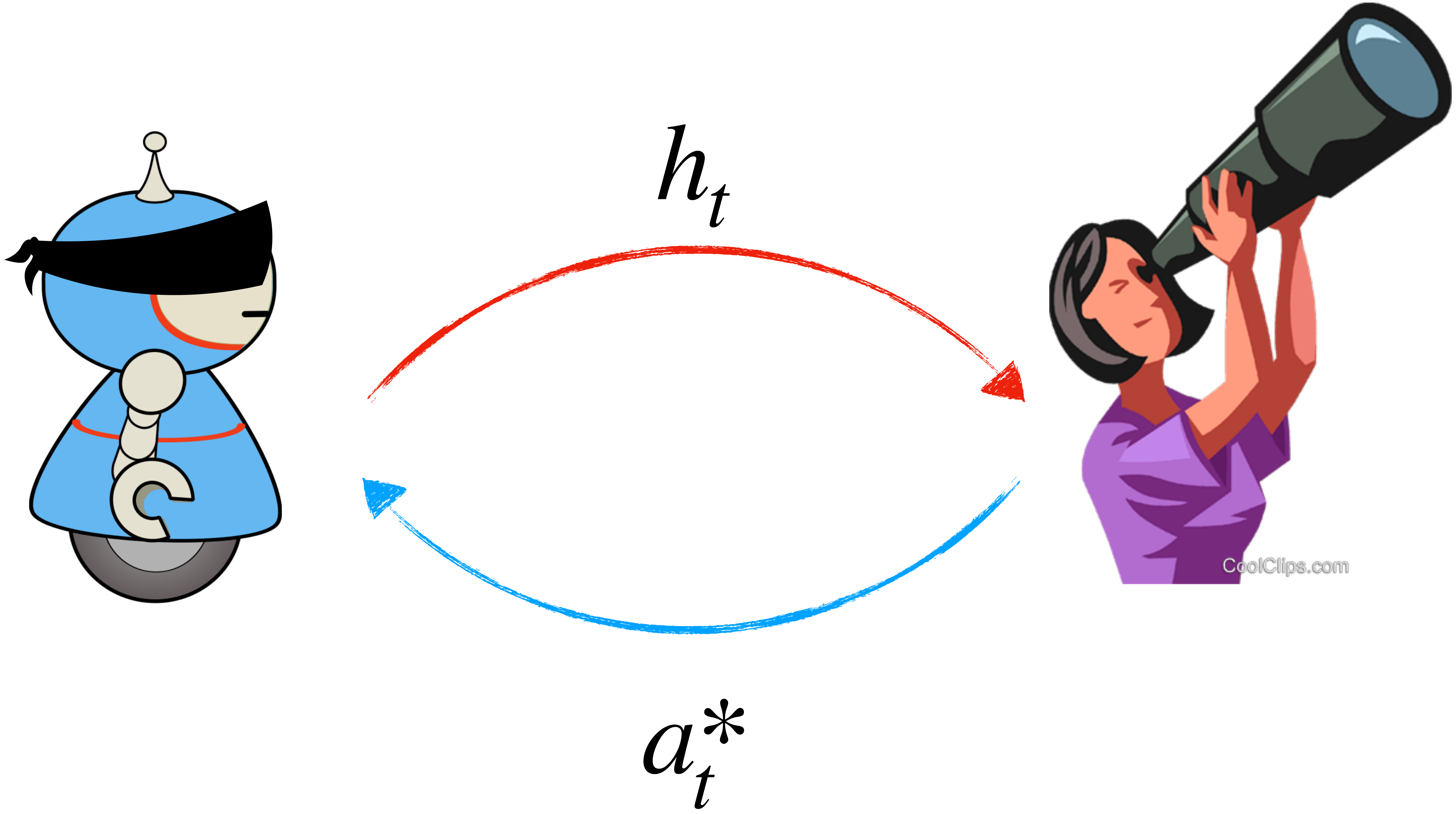
2021-10-22

Shaking the foundations: delusions in sequence models for interaction and control

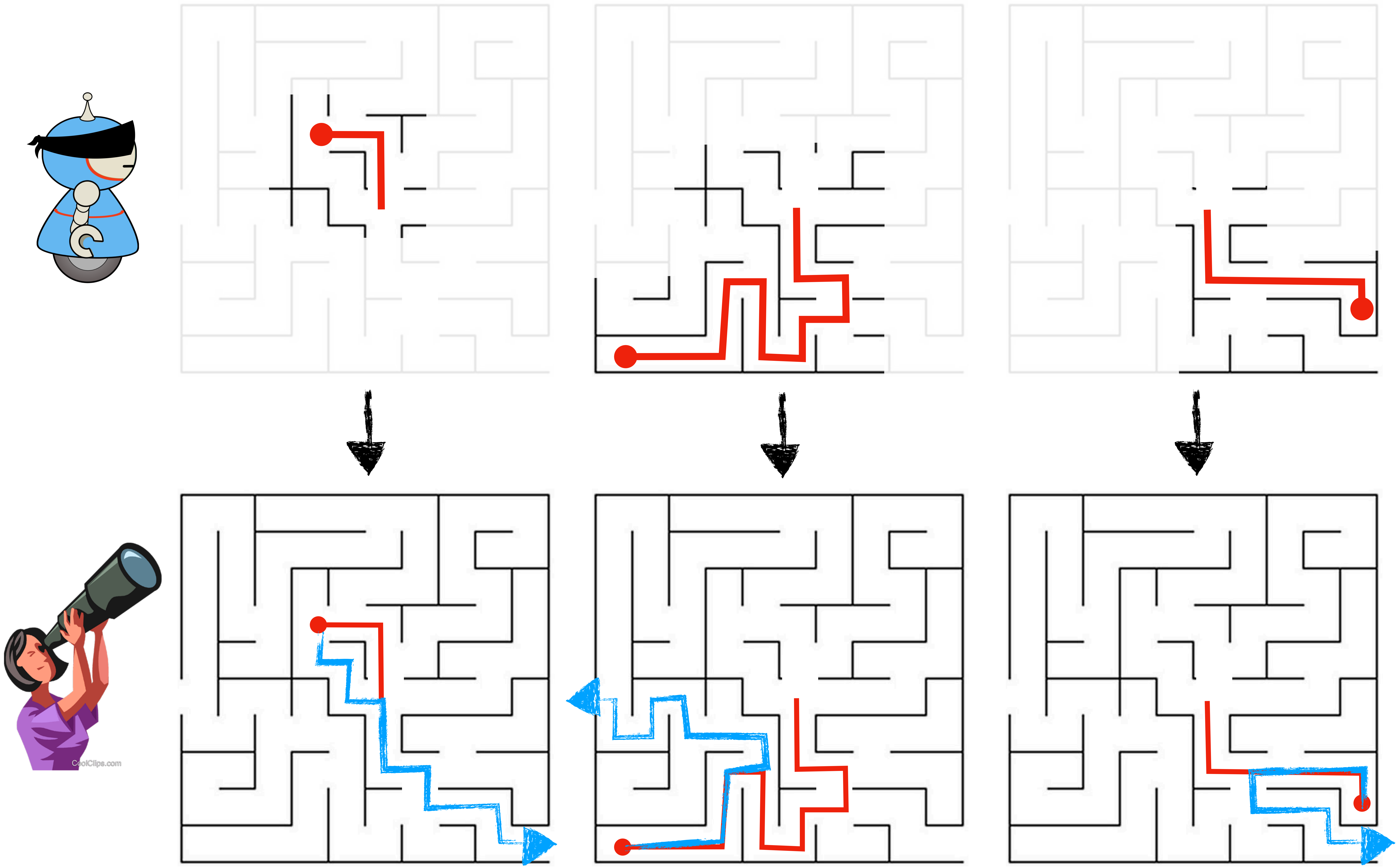
Pedro A. Ortega[*], Markus Kunesch[*], Grégoire Delétang[*], Tim Genewein[*], Jordi Grau-Moya[*], Joel Veness[1], Jonas Buchli[1], Jonas Degrave[1], Bilal Piot[1], Julien Perolat[1], Tom Everitt[1], Corentin Tallec[1], Emilio Parisotto[1], Tom Erez[1], Yutian Chen[1], Scott Reed[1], Marcus Hutter[1], Nando de Freitas[1] and Shane Legg[1]

[*]Deepmind Safety Analysis, [1]DeepMind

Solution: Interactively query expert



Solution: Interactively query expert



e.g DAGGER

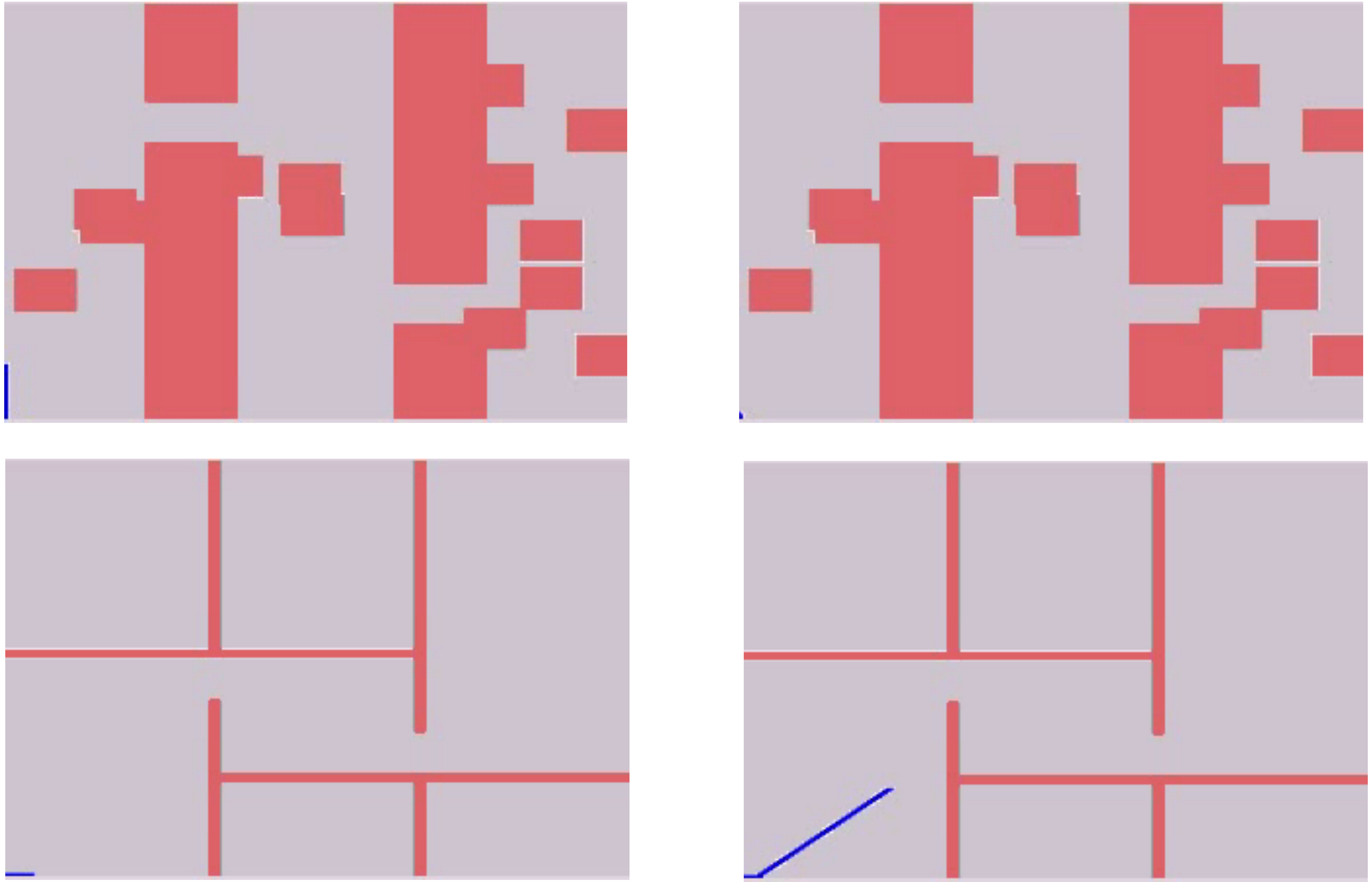
1. Roll out learner
2. Query Expert
3. Aggregate Data

and repeat!

[Choudhury 2018]

Example: Training search heuristics

On-policy (Aggrevate) | Behavior Cloning



Why / When does this work?

Proved that this approximates Hindsight Optimization / QMDP

Fails when you need to explicitly explore (i.e. asymptotic realizability not hold)

Wait ... isn't this the same old covariate shift problem?

Expert becomes realizable over time

	Easy	Medium	Hard
Setting	Expert is realizable $\pi^E \in \Pi$	Non-realizable expert but full expert support	Non-realizable expert + limited expert support
	As $N \to \infty$, drive down $\epsilon = 0$ (or Bayes error)	Even as $N \to \infty$, behavior cloning $O(\epsilon C T)$	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 /AGGREVATE) to provide labels $\Rightarrow O(\epsilon T)$

🤔 Why is behavior cloning so flaky?

In many cases it works just fine!

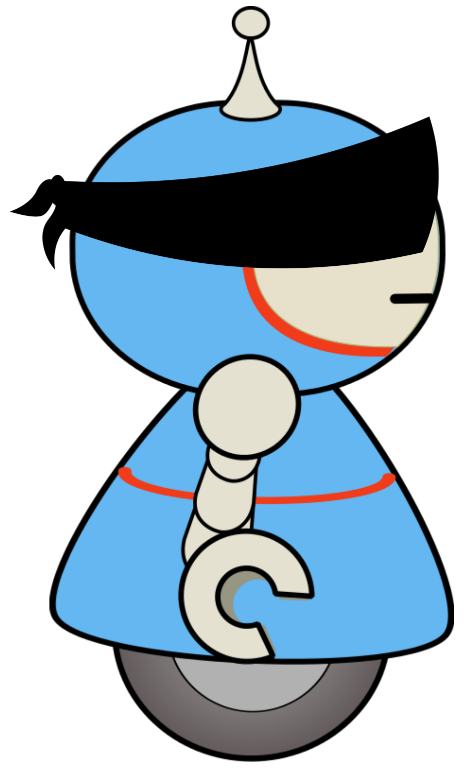
- matches state of the art in many offline RL problems
- standard practice in NLP (teacher forcing)

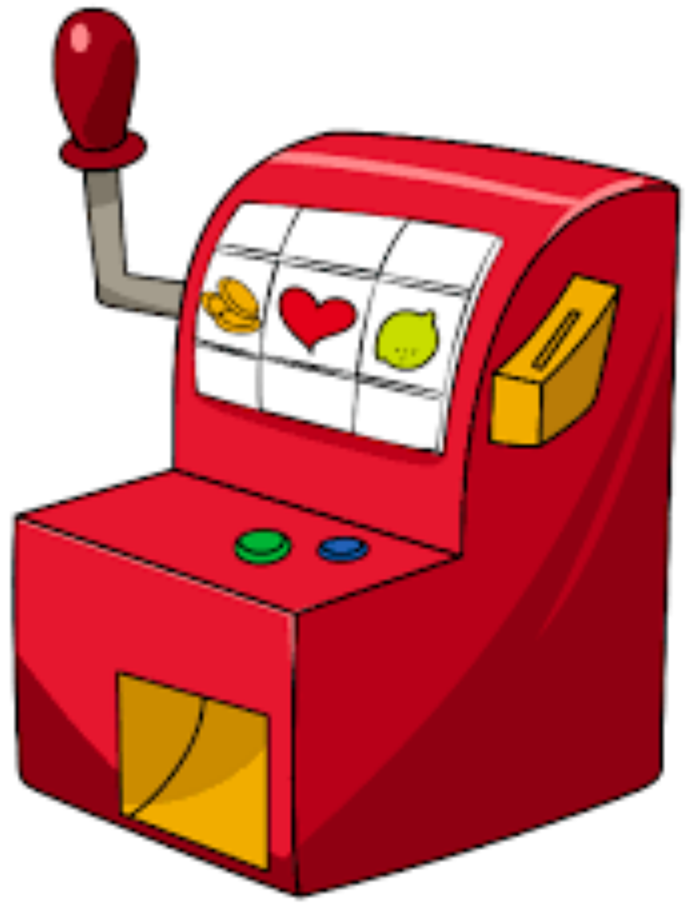
But often times it creates this undesirable latching effect

- extensively reported in self-driving, language models, etc

On-policy algorithms work consistently well

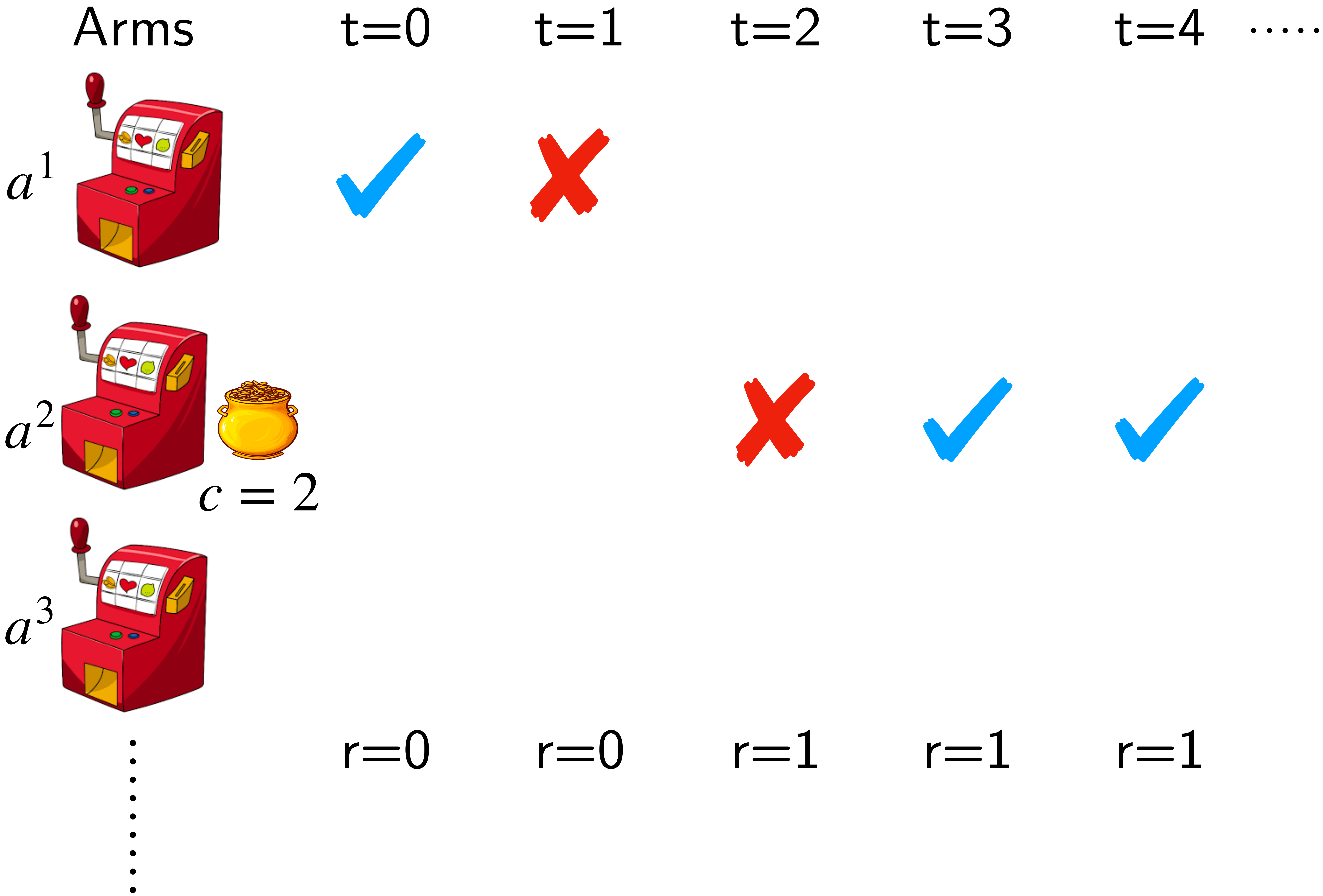
A Toy Bandit Example

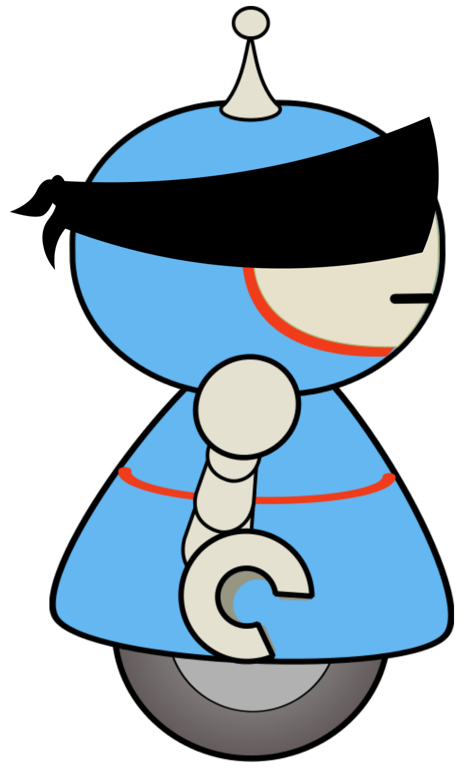




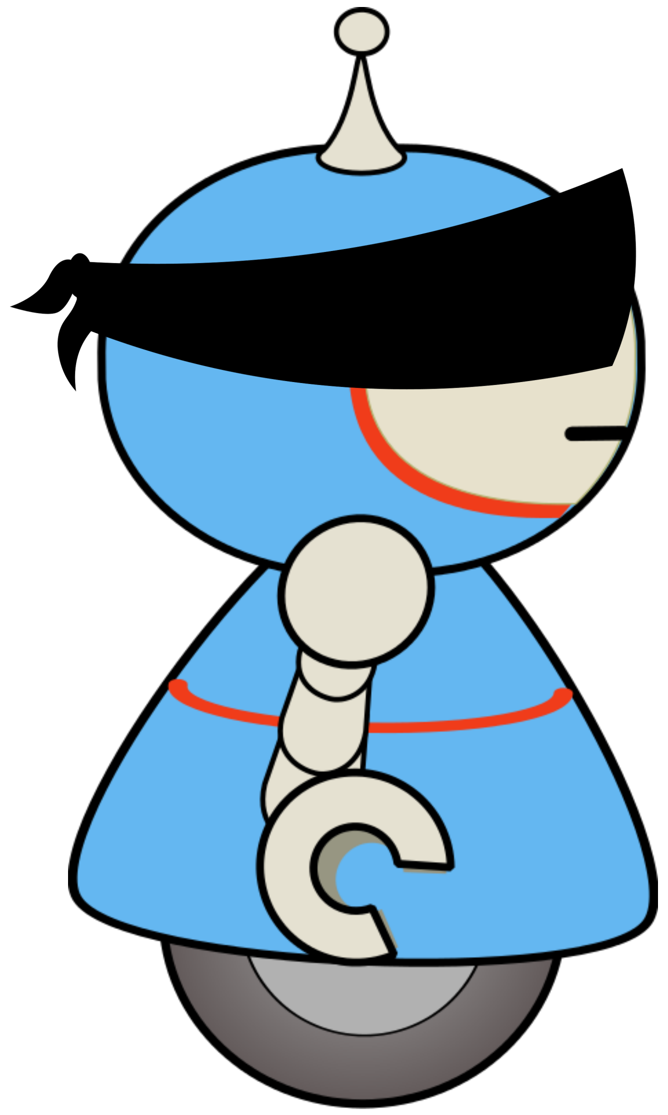
https://github.com/gkswamy98/sequence_model_il/blob/master/ConfoundedBandit.ipynb


Arms
t=0
t=1
t=2
t=3
t=4
a^1
a^2
$c = 2$
a^3
r=0
r=0
r=1
r=1
r=1
Learner only sees binary feedback



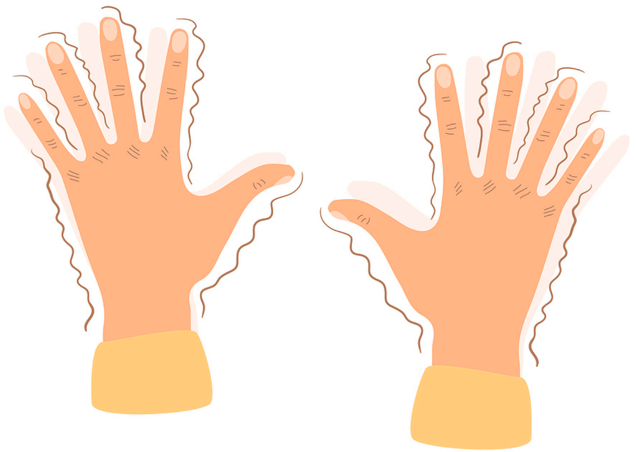


Feedback can be noisy! (ϵ_{obs})

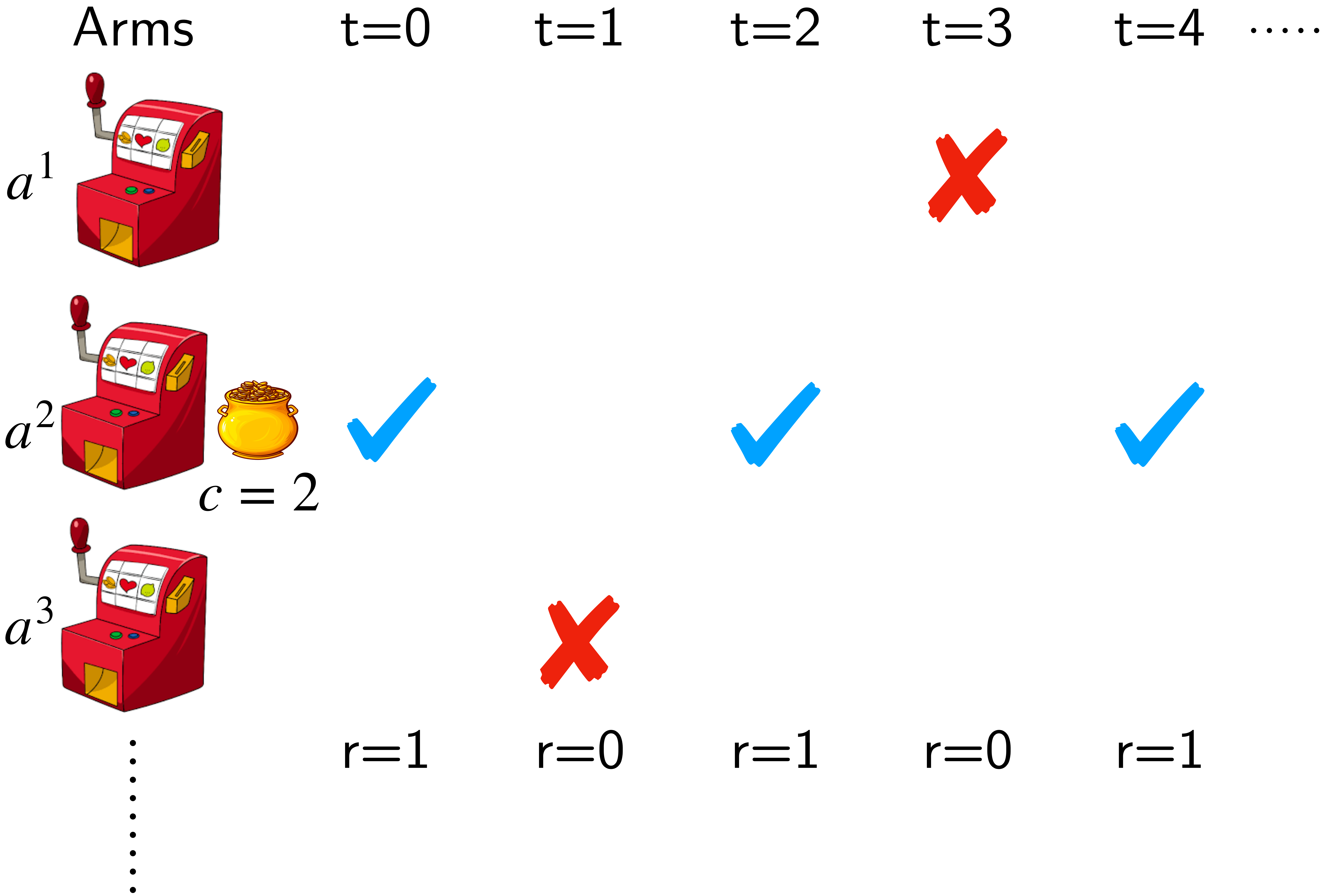


Imitate

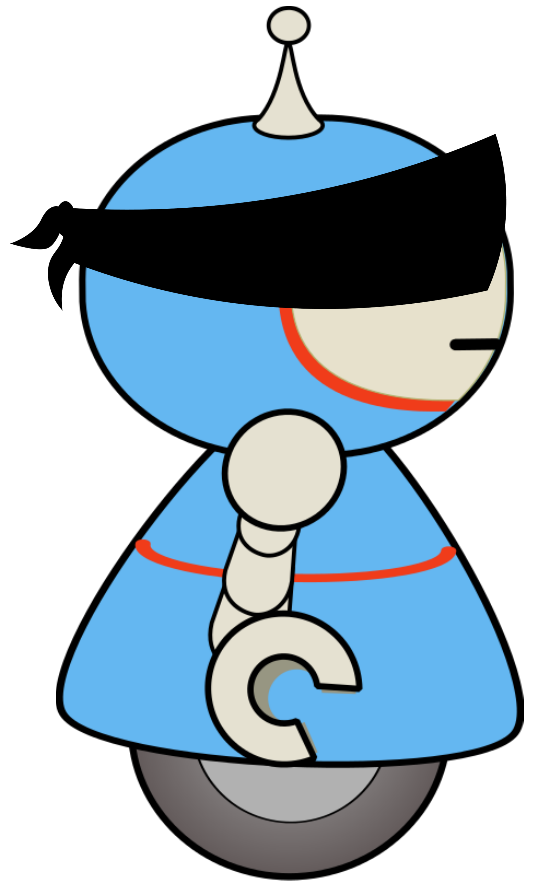
Human expert knows the context



Human expert can be noisy (ϵ_{exp})



Goal: Bound average performance difference



$$\frac{1}{T} J(\pi_E) - J(\pi)$$

$$\mathbb{E}_{\tau\sim\pi^E}[\frac{1}{T}\sum_{t=1}^{T} r(s_t, a_t, c)] - \mathbb{E}_{\tau\sim\pi}[\frac{1}{T}\sum_{t=1}^{T} r(s_t, a_t, c)].$$

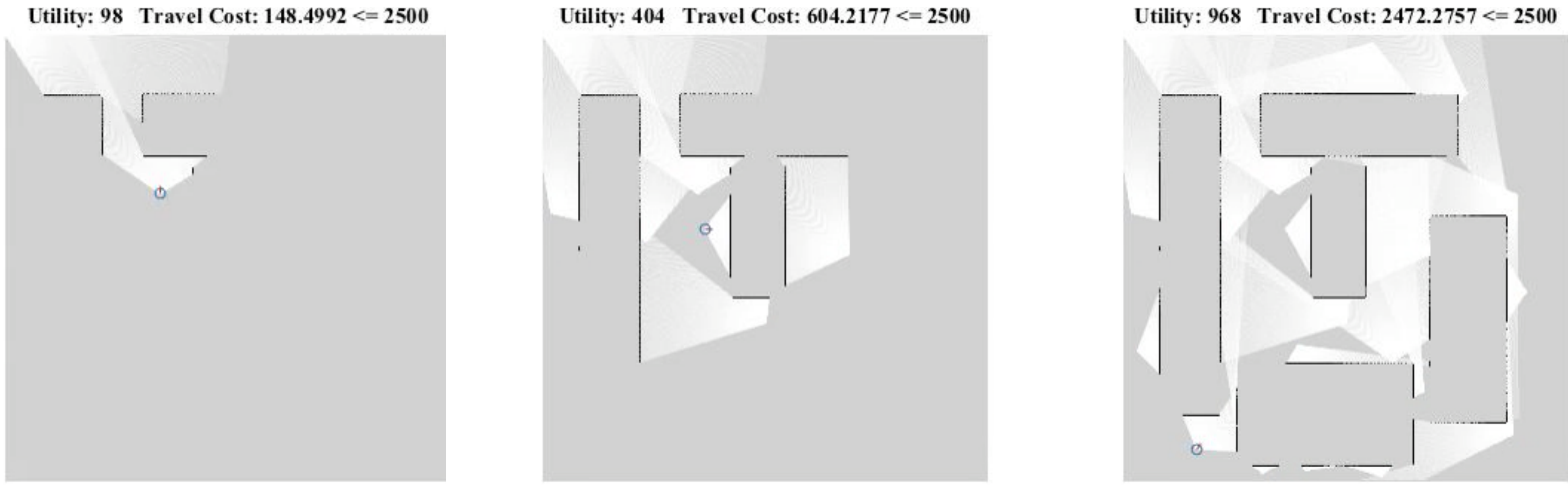
Assumptions!

1. Recoverability

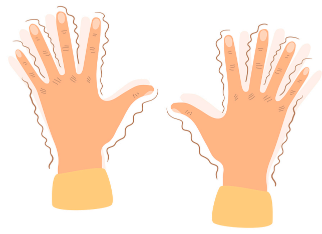
Bounds the total cost incurred for an expert to recover from an arbitrary mistake

2. Asymptotic Realizability

Learner performs as well as the expert after observing a long enough history



Trial 1: Behavior Cloning

Correct Door: 0 $\epsilon_{obs} = 0.0$ $\epsilon_{exp} = 0.0$ 

$$[1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\ 1.\\ 1.\ 1.\ 1.\ 1.]$$

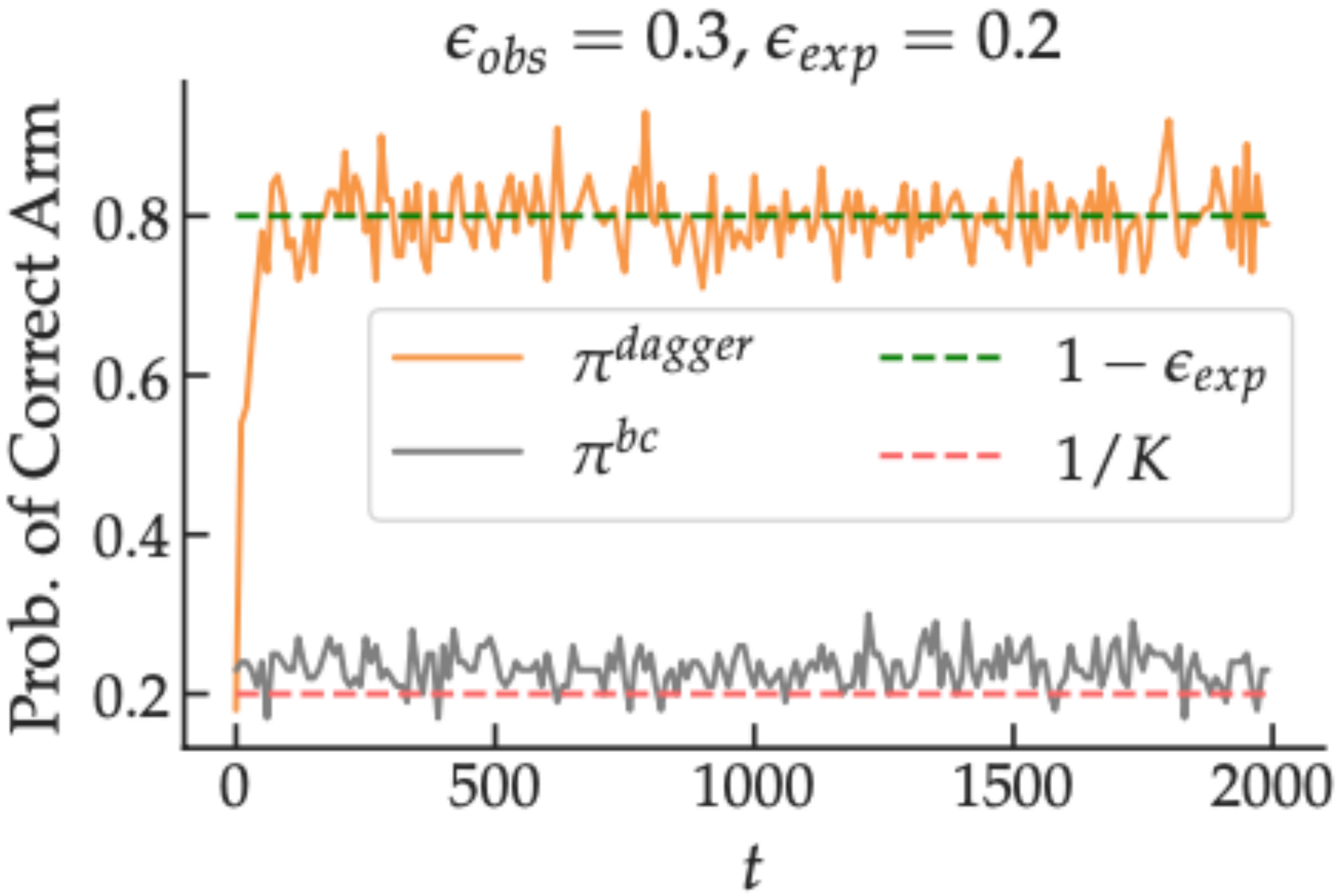
Trial 1: DAGGER

Correct Door: 4 ϵ_{obs} = 0.0 ϵ_{exp} = 0.0 🙌

[1. 0. 2. 3. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
4. 4. 4. 4.]

😄

BC performs similar to random actions!



Okay, so BC consistently fails and DAGGER consistently works?

Trial 1: Behavior Cloning

Correct Door: 0 $\epsilon_{obs} = 0.0$ $\epsilon_{exp} = 0.0$ 🙌

[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1.]

😱

Trial 2: Behavior Cloning

Correct Door: 0

$\epsilon_{obs} = 0.0$

$\epsilon_{exp} = 0.01$

[3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 1. 0. 0. 0.]

Trial 3: Behavior Cloning

Correct Door: 0 $\color{red}{\epsilon_{obs} = 0.05}$ $\epsilon_{exp} = 0.01$ 🙌

```
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 3. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1.]
```

😱

Trial 4: Behavior Cloning

Correct Door: 0 $\epsilon_{obs} = 0.05$ $\epsilon_{exp} = 0.2$ 🙌

[4. 0. 1. 2. 2. 3. 0. 0. 0. 0. 1. 0. 3. 0. 0. 0. 0. 0. 0. 0. 1. 3. 0. 0.
0. 0. 0. 4. 0. 0. 3. 0. 0. 2. 0. 0. 0. 0. 0. 3. 0. 1. 0. 0. 0. 1. 4. 0.
4. 0. 0. 0. 0. 0. 0. 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2. 0. 0. 0.
0. 0. 3. 0. 2. 3. 0. 0. 1. 1. 0. 0. 0. 0. 0. 3. 0. 0. 0. 4. 1. 0. 3. 0.
0. 0. 0. 2.]

😄

What about DAGGER?

Trial 2: DAGGER

Correct Door: 0 $\epsilon_{obs} = 0.0$ $\epsilon_{exp} = 0.01$ 🙌

[4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 4. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]

😄

Trial 3: DAGGER

Correct Door: 0 $\epsilon_{obs} = 0.05$ $\epsilon_{exp} = 0.01$ 🙌

[3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 4. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0.]

😱

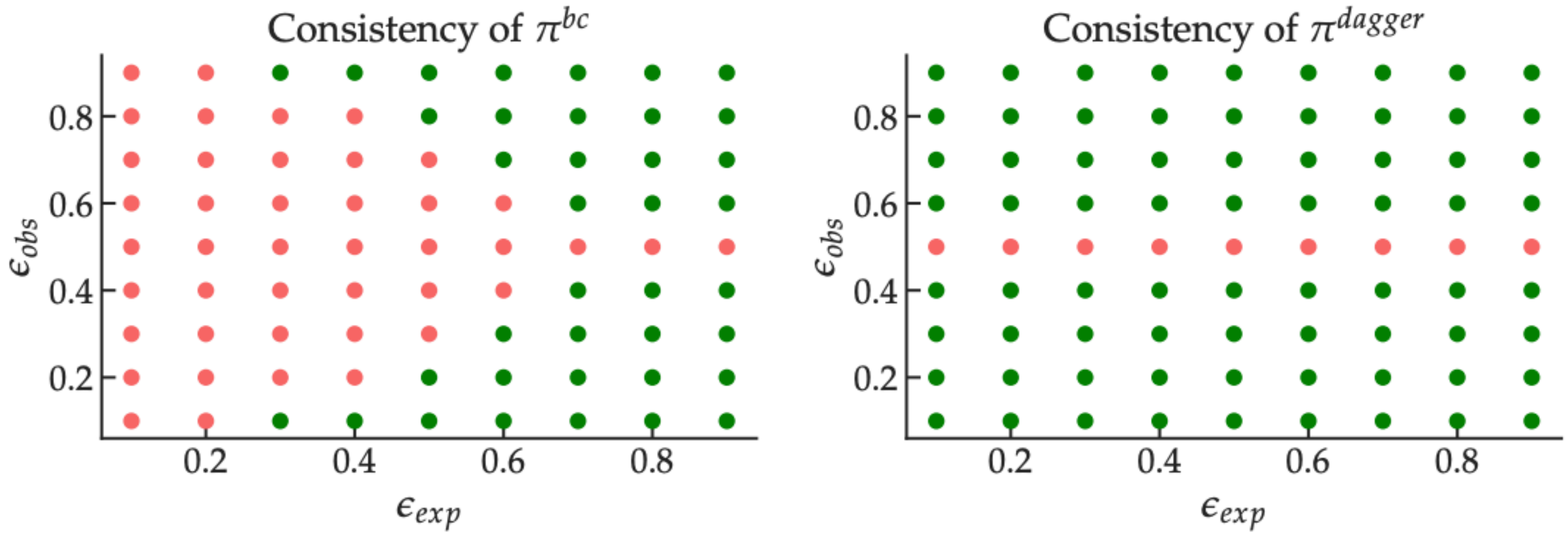
Trial 4: DAGGER

Correct Door: 0 $\epsilon_{obs} = 0.05$ $\epsilon_{exp} = 0.2$ 🙌

[3. 2. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 3. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
3. 4. 4. 0. 0. 0. 0. 0. 0. 0. 3. 0. 0. 4. 0. 2. 0. 0. 0. 0. 0. 3. 0. 0.
0. 0. 0. 0. 0. 0. 1. 0. 4. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 4.]

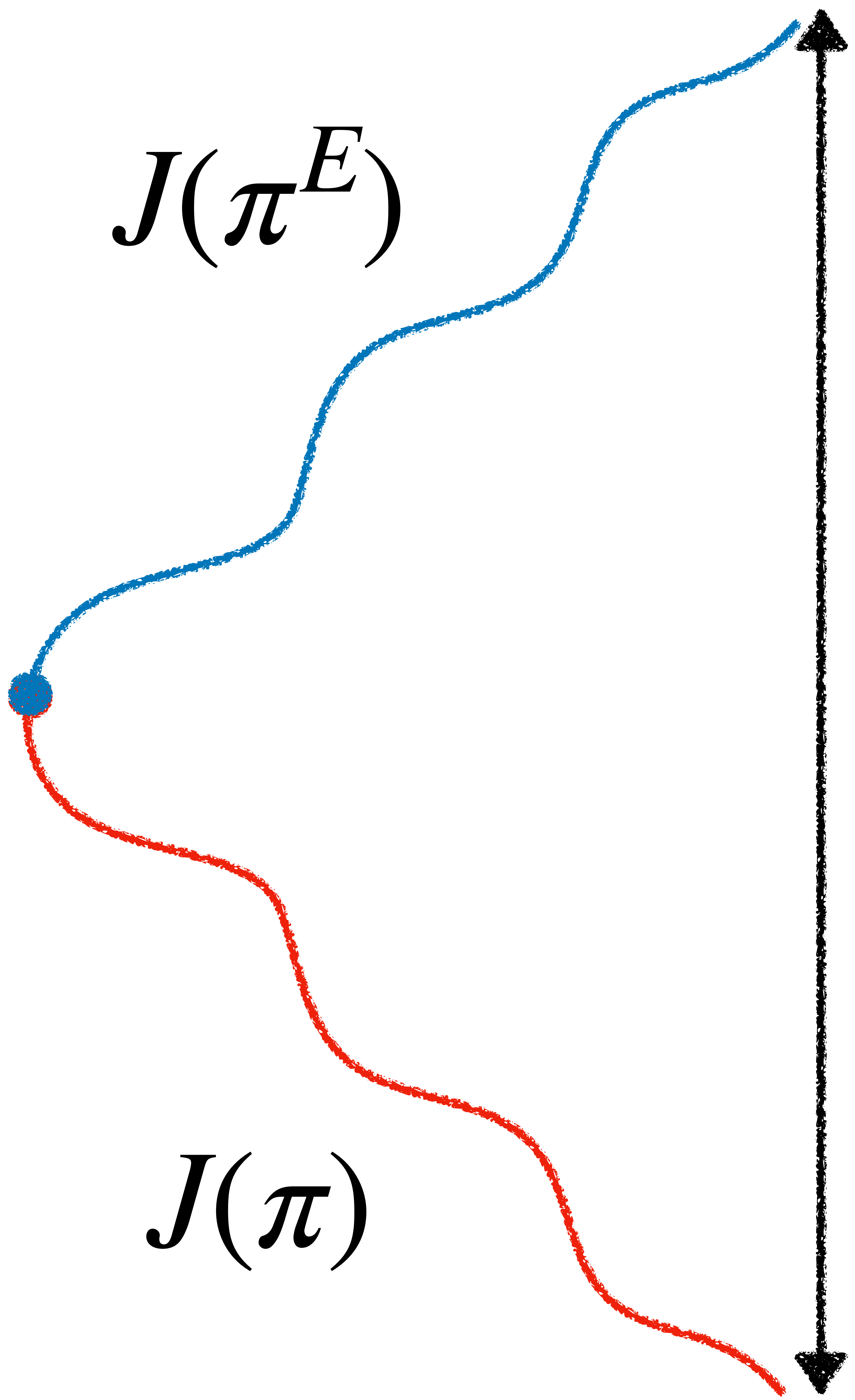
😄

Consistency of BC vs DAGGER

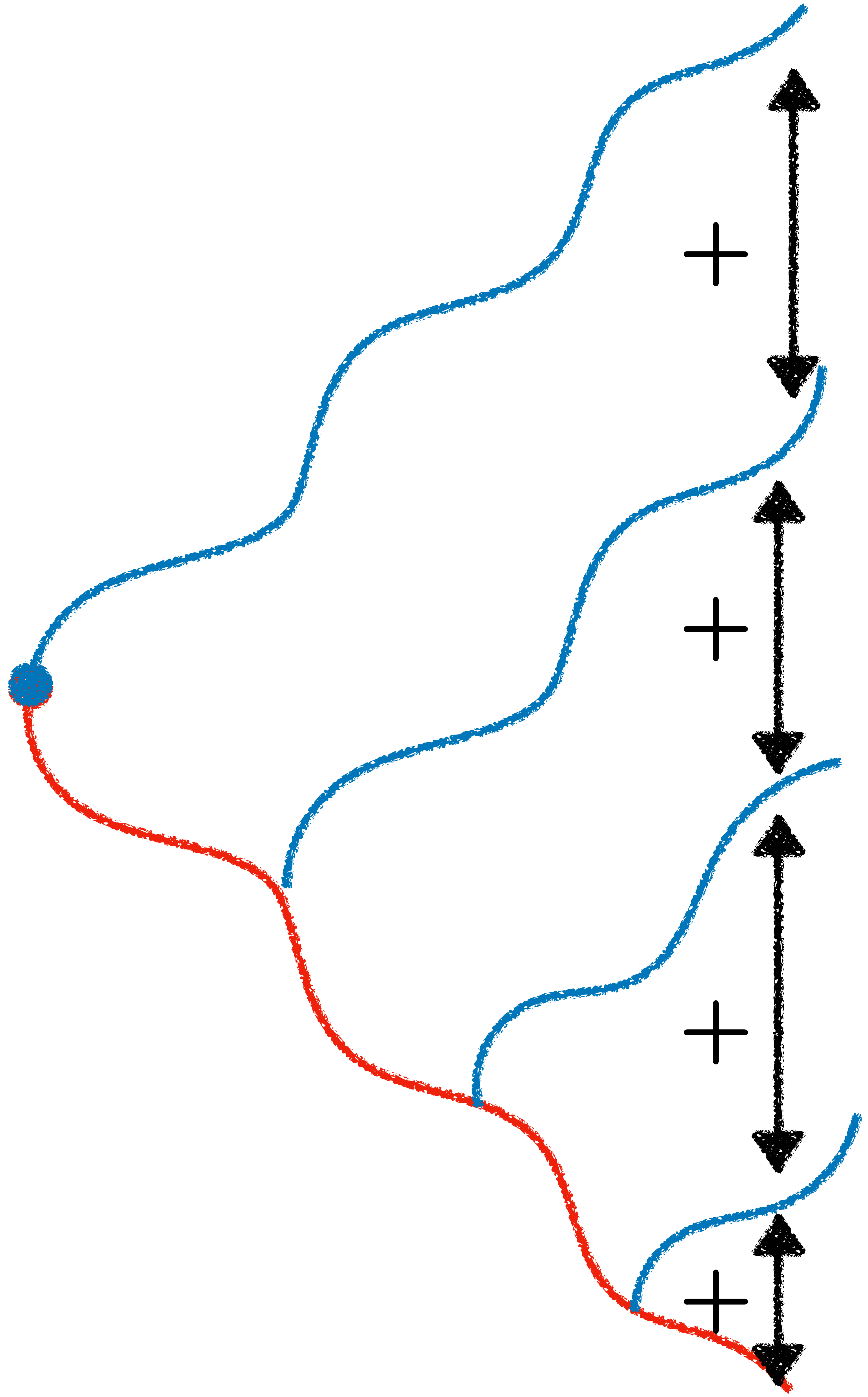


Green: After T=1000, learner picks the right arm

(more green is good)



$$\frac{1}{T}(J(\pi^E) - J(\pi))$$



$$\frac{1}{T}(J(\pi^E) - J(\pi))$$

$$= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{h_t \sim d_\pi^t}[Q^{\pi_E}(s_t, \pi_E(s_t, c)) - Q^{\pi_E}(s_t, \pi(h_t))]$$

$$\leq Q_{\max}\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{h_t \sim d_\pi^t}\mathbb{I}(\pi_E(s_t, c) \neq \pi(h_t))$$

$$\leq Q_{\max}\frac{1}{T}\sum_{t=1}^{T} \epsilon_{on}(t)$$

$$\frac{1}{T}(J(\pi^E) - J(\pi))$$

$$\leq \boxed{Q_{\max}} \boxed{\frac{1}{T}\sum_{t=1}^{T} \epsilon_{on}(t)}$$

Recoverability means this is small

Asymptotic Realizability means this goes to zero as $T \to \infty$

What happens with behavior cloning?

$$\frac{1}{T}(J(\pi^E) - J(\pi))$$

$$\leq Q_{\max} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{h_t \sim d_\pi^t} \mathbb{I}(\pi_E(s_t, c) \neq \pi(h_t))$$

$$\leq Q_{\max} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{h_t \sim d_{\pi^E}^t} \frac{d_\pi^t}{d_{\pi^E}^t} \mathbb{I}(\pi_E(s_t, c) \neq \pi(h_t))$$

Density ratio explodes!

$$\leq Q_{\max} \frac{1}{T} \sum_{t=1}^{T} \boxed{\| \frac{d_\pi^t}{d_{\pi^E}^t} \|_\infty} \epsilon_{off}(t)$$

On-policy

Make mistakes initially

Gets feedback on histories it generates

Asymptotic realizability ensures performance difference goes to zero

Off-policy

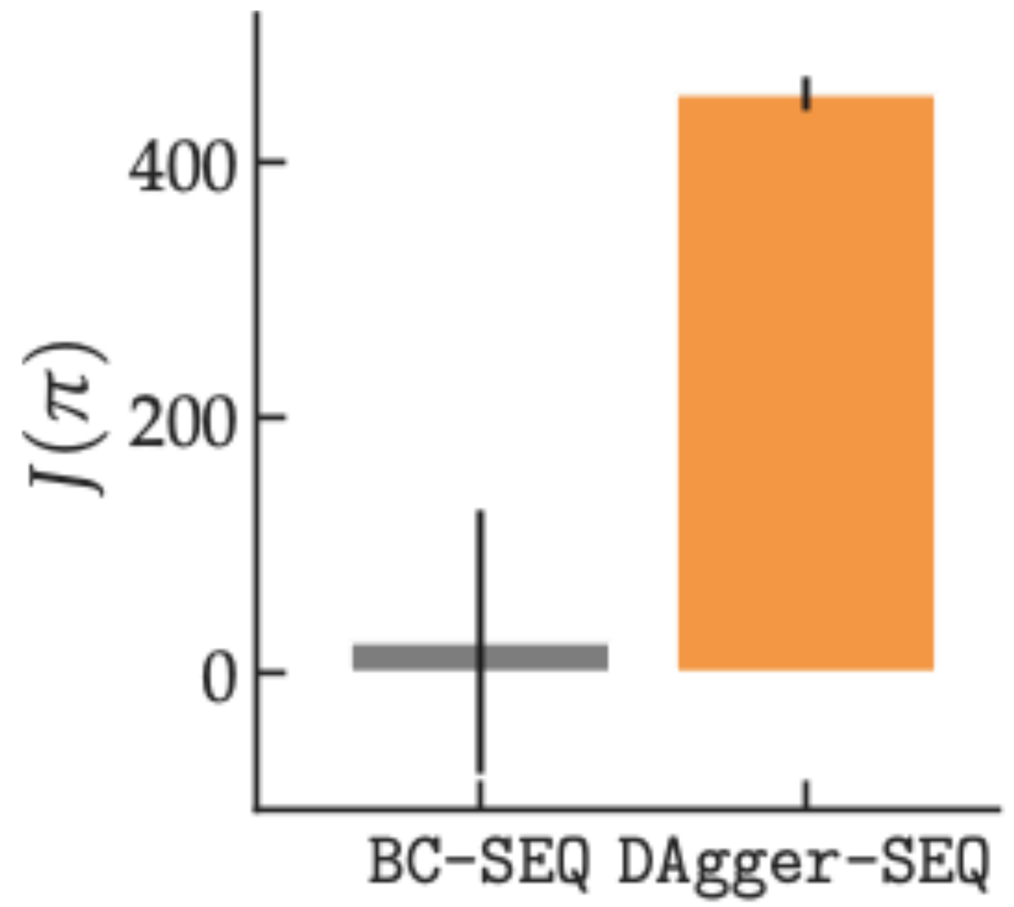
Make mistakes initially

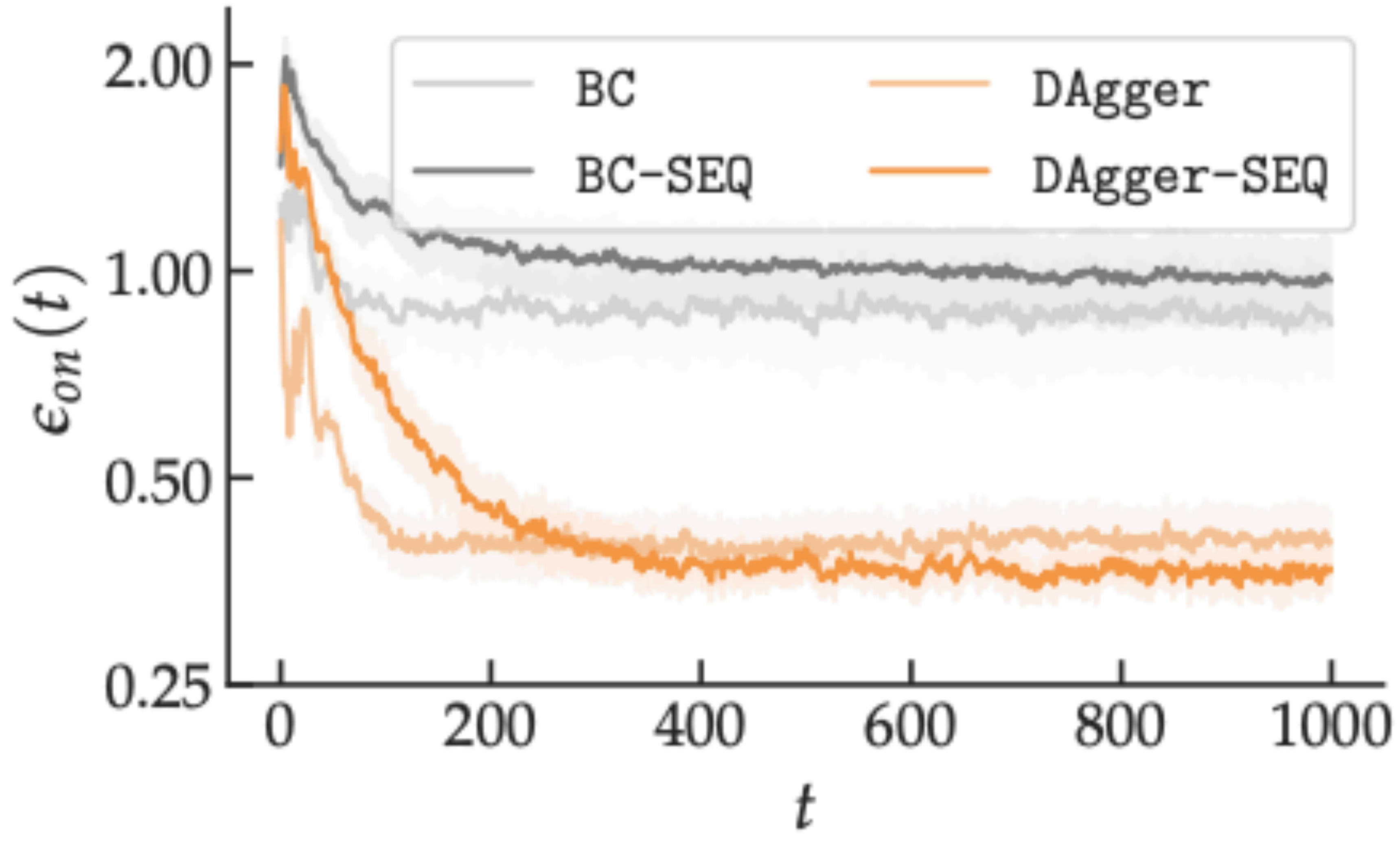
History diverges from expert history

As the density ratio blows up, performance difference blows up

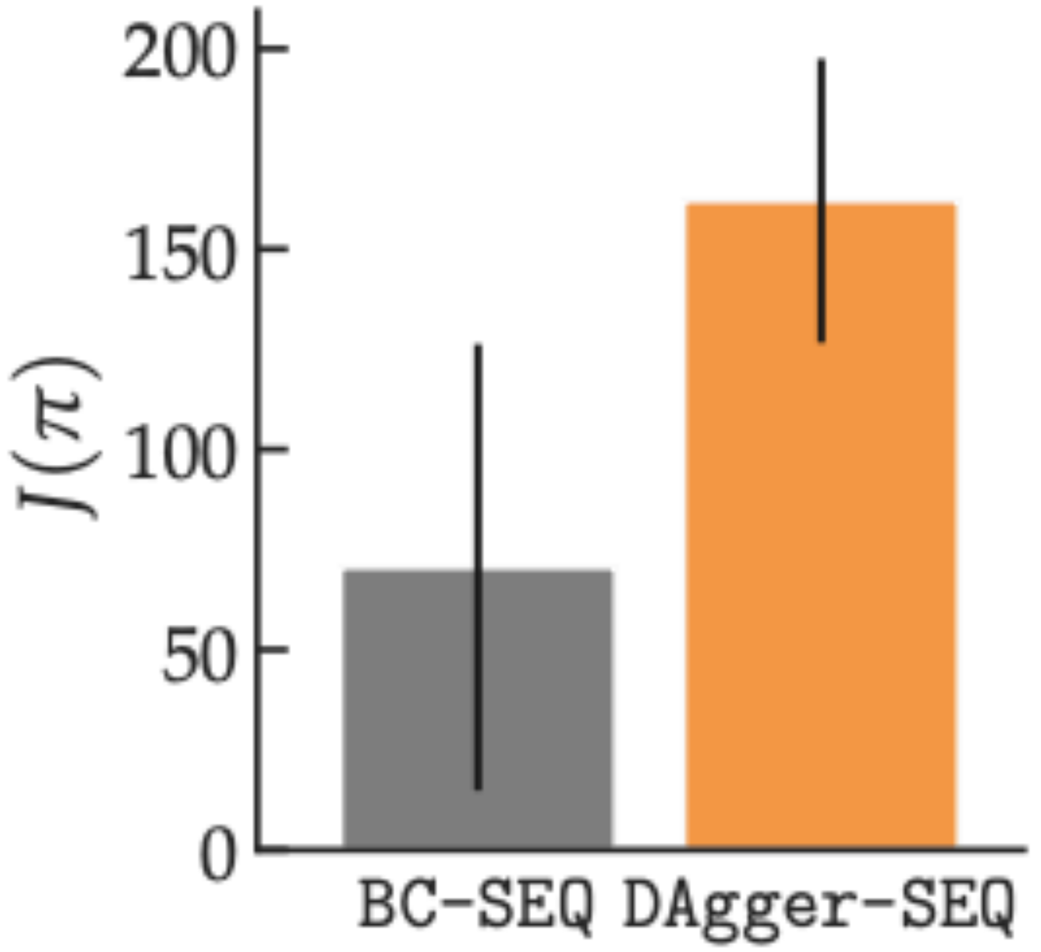
Results

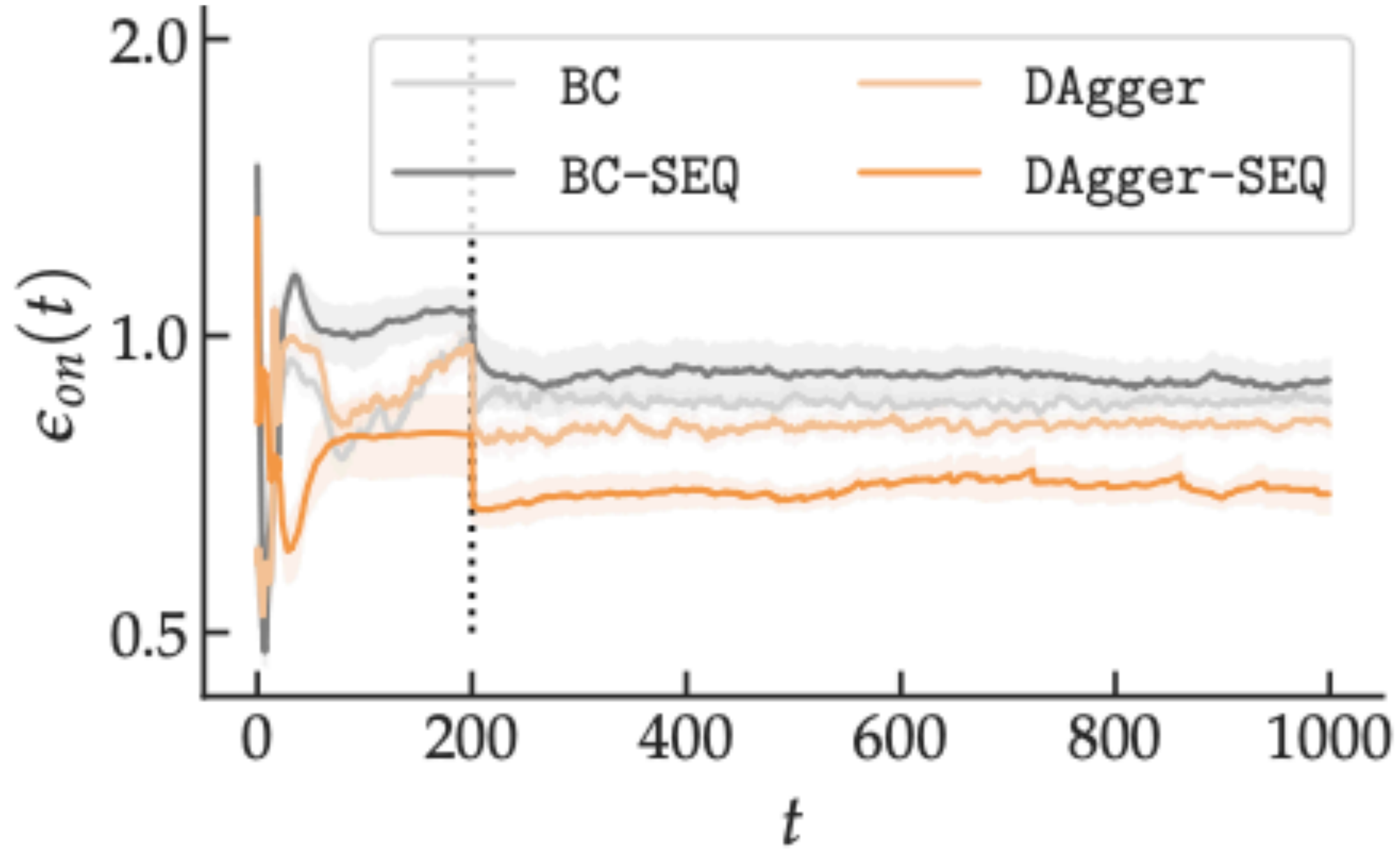
Half-Cheetah





Ant





Context (c) here is the latent speed that the robot should run at.

Expert sees context

Learner sees indicator feature 1(v>=c)

(From Finn et al. 2017)

Does training from the privilege expert lead to the optimal policy (for the student)?

What does it approximate?

The Q-MDP Approximation for POMDPs (Aka Hindsight Optimization)

QMDP

- Relax "partial" observability
  - The state of environment is fully observable after one action
    - After one action we solve MDP, i.e use Q-value of MDP
    - We are currently uncertain
      - Use expected Q-value

$$Q^{MDP}(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') V^{MDP}(s')$$

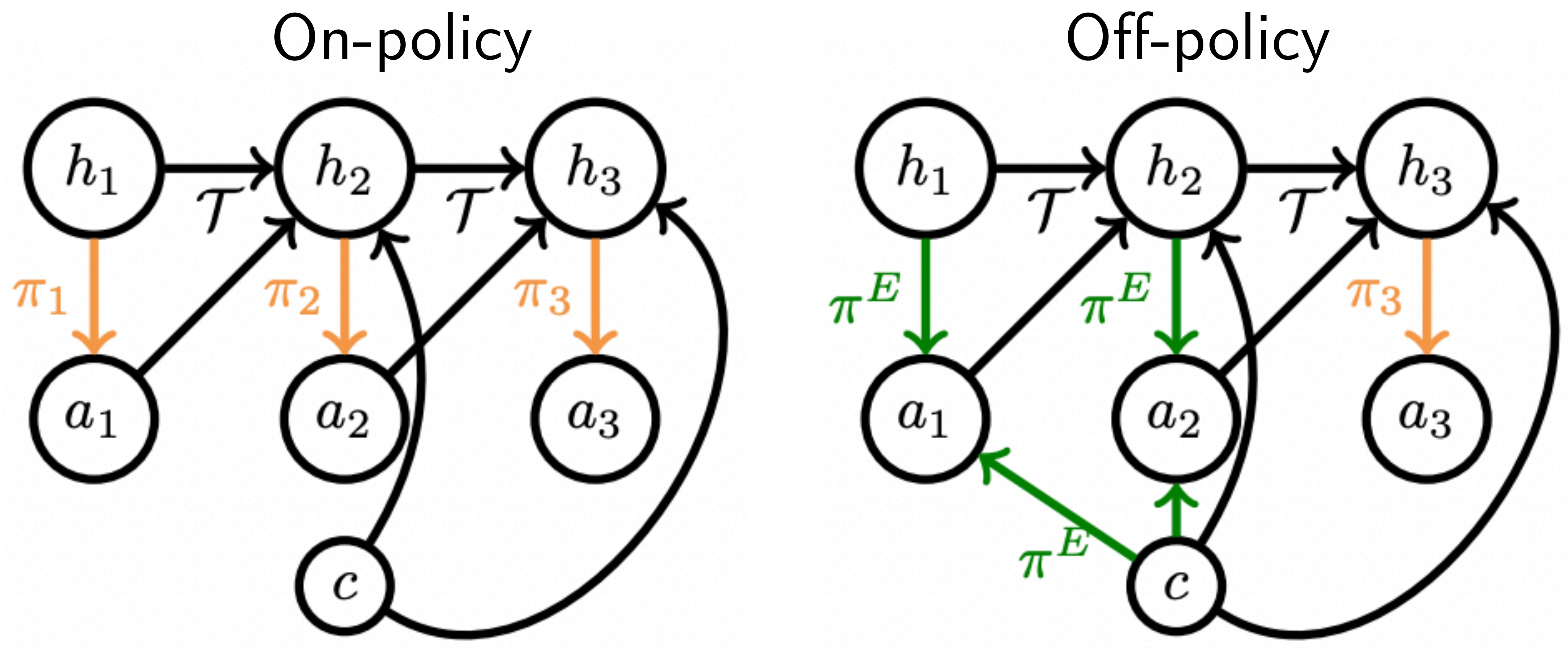
$$Q^{MDP}(b_t, a) = \sum_{s} b_t(s) Q^{MDP}(s, a)$$

$$V^{QMDP}(b_t) = argmax_a Q^{MDP}(b_t, a)$$

Sequence Model Imitation Learning with Unobserved Contexts

Swamy, G., Choudhury, S., Bagnell, J. A., & Wu, Z. S, (NeuRIPS 2022)

Structural Causal Model perspective

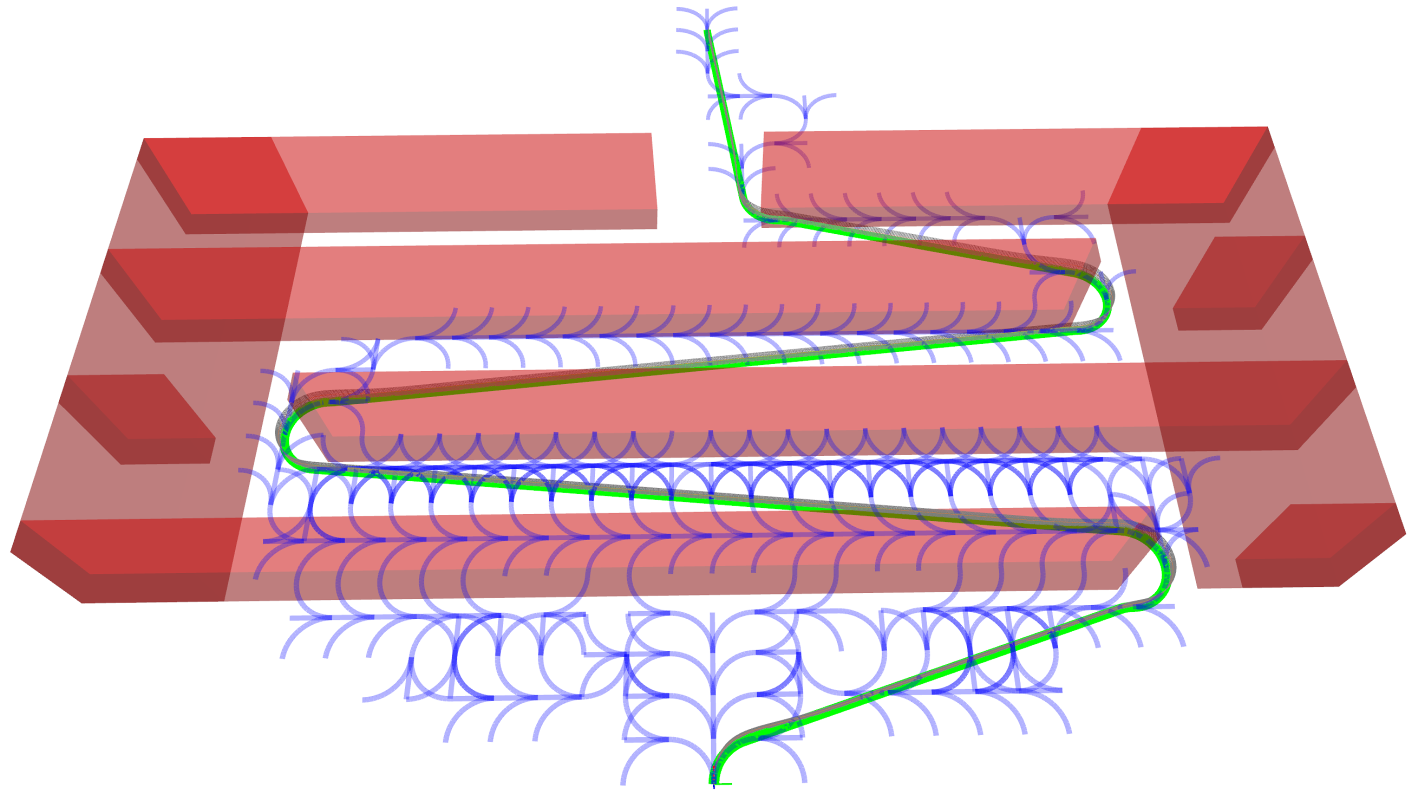


From 0-1 loss to Moment Matching

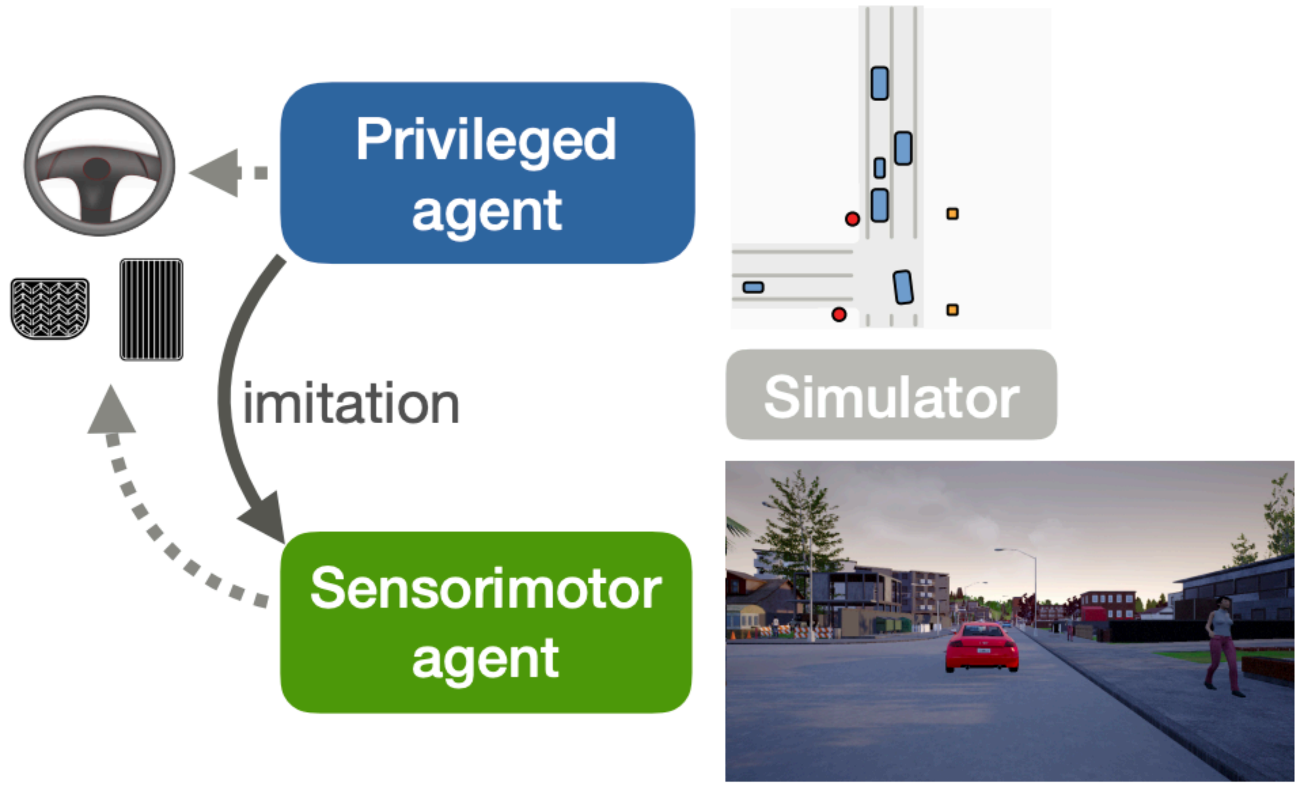
$$\epsilon_{\text{on}}(t) = \sup_{\tilde{f} \in \tilde{\mathcal{F}}_{\text{on}}} \mathbb{E}_{\tau \sim \pi}[\tilde{f}(h_t, a_t) - \mathbb{E}_{a' \sim \pi^E(s_t, c)}[\tilde{f}(h_t, a')]],$$

$$\epsilon_{\text{off}}(t) = \sup_{\tilde{f} \in \tilde{\mathcal{F}}_{\text{off}}} \mathbb{E}_{\tau \sim \pi^E}[\tilde{f}(h_t, a_t) - \mathbb{E}_{a' \sim \pi(h_t)}[\tilde{f}(h_t, a')]],$$

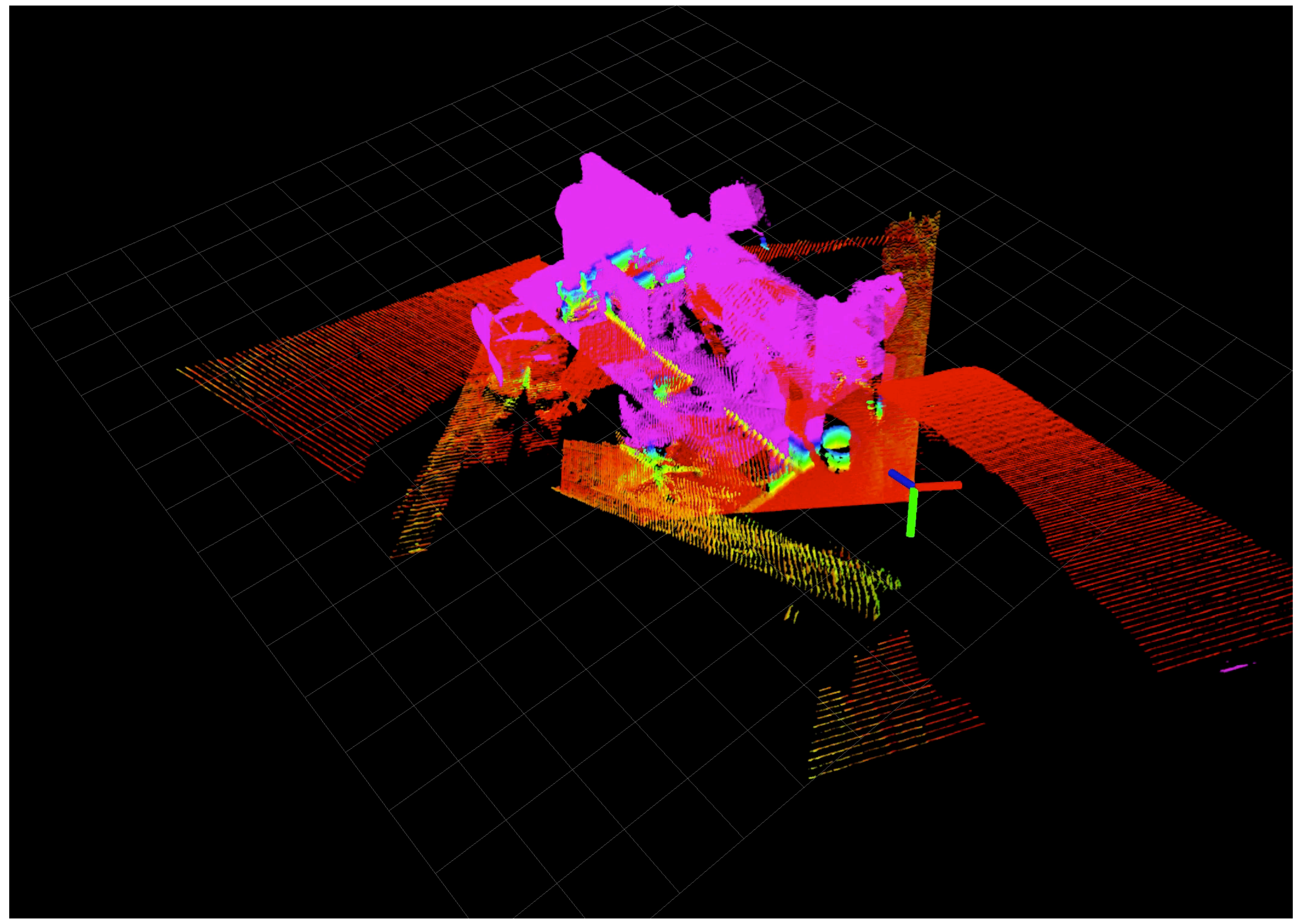
Learn by Cheating!



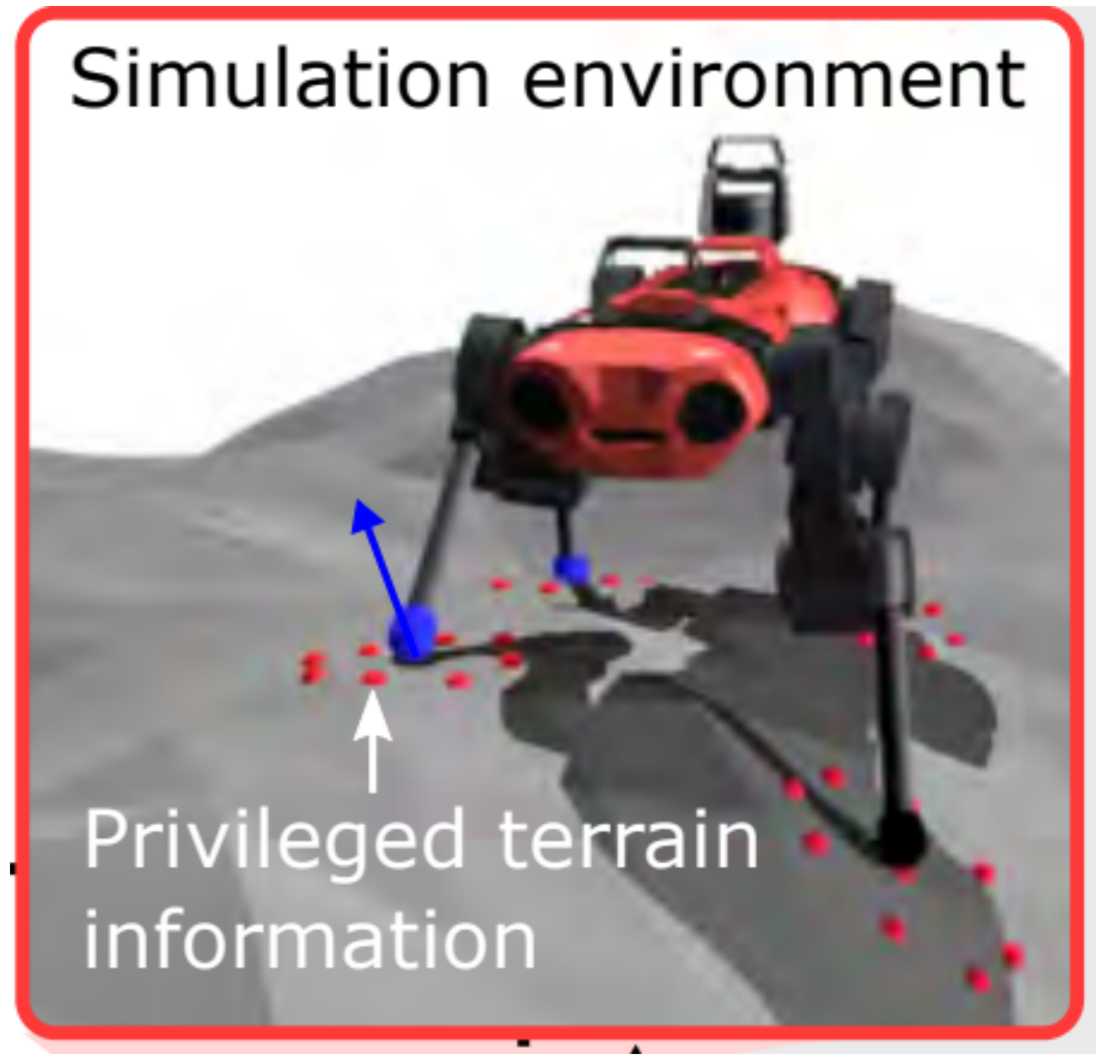
Search



Navigation



Mapping



Legged Locomotion