Imitation Learning from Privileged Information in Sim2Real

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Today's class

- Sim2Real: The double-edged sword

  Case study: OpenAI Dactyl Hand

- Teacher->Student distillation

  Case study: Visual Dexterity

- Imitation Learning with Privileged Information

Sim2Real: Double-edged sword

The Good

We can run reinforcement learning to compute optimal policies!

(1) Exploration is safe
(2) Leverage privileged information

The Danger

Mismatch between simulation and reality

(1) Observation mismatch
(2) Transition mismatch

Today's Robot: Dextrous Manipulation

How babies learn to manipulate

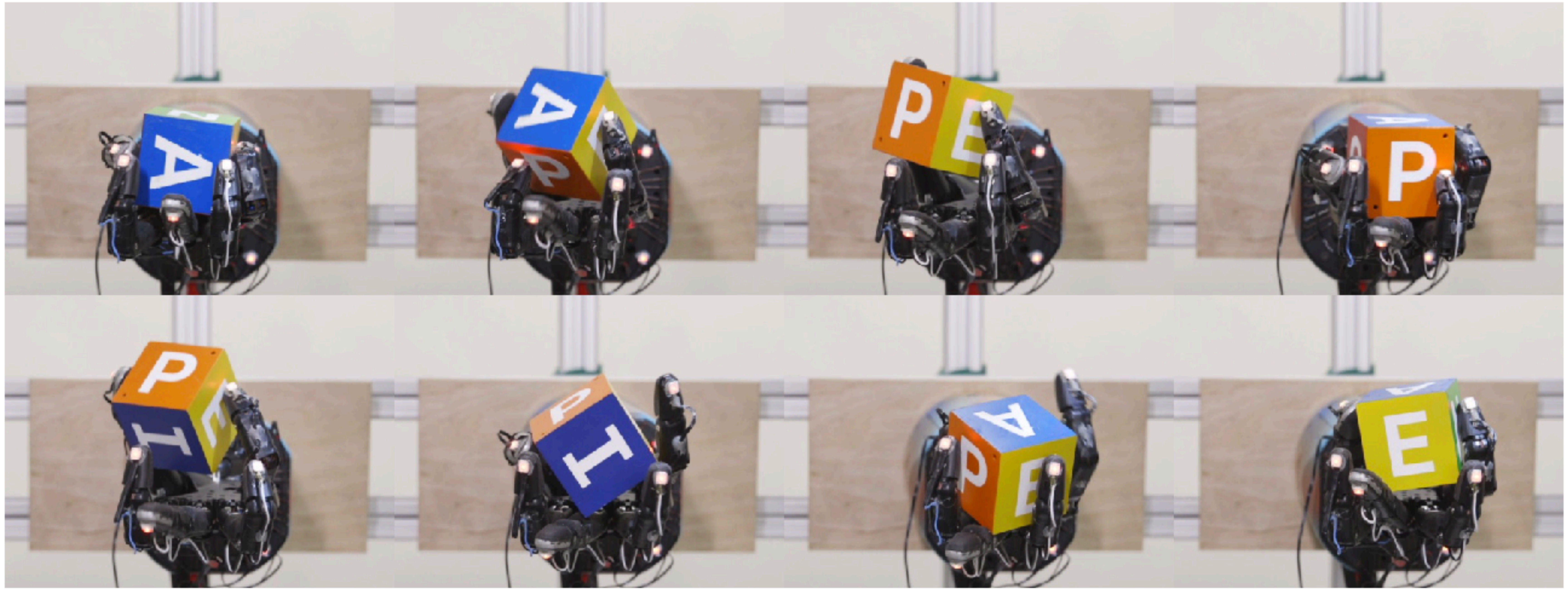
Most robots today!

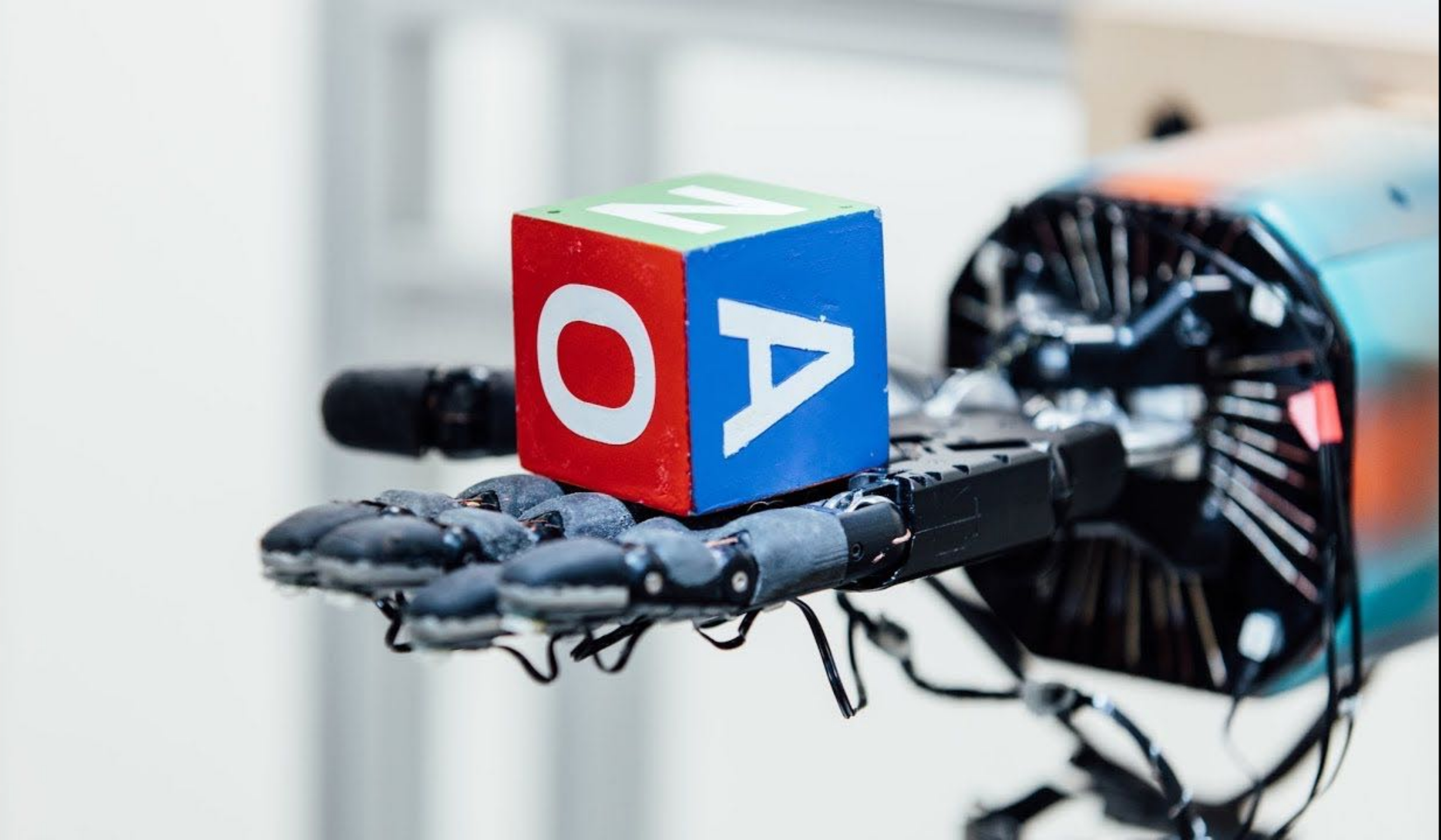
Learning Dexterity

(Open AI)

Learning Dexterous In-Hand Manipulation

OpenAI,* Marcin Andrychowicz, Bowen Baker, Maciek Chociej,
Rafał Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron,
Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor,
Josh Tobin, Peter Welinder, Lilian Weng, Wojciech Zaremba




O
A

Sim

Train a policy in simulation (RL)

Real

Test in real world

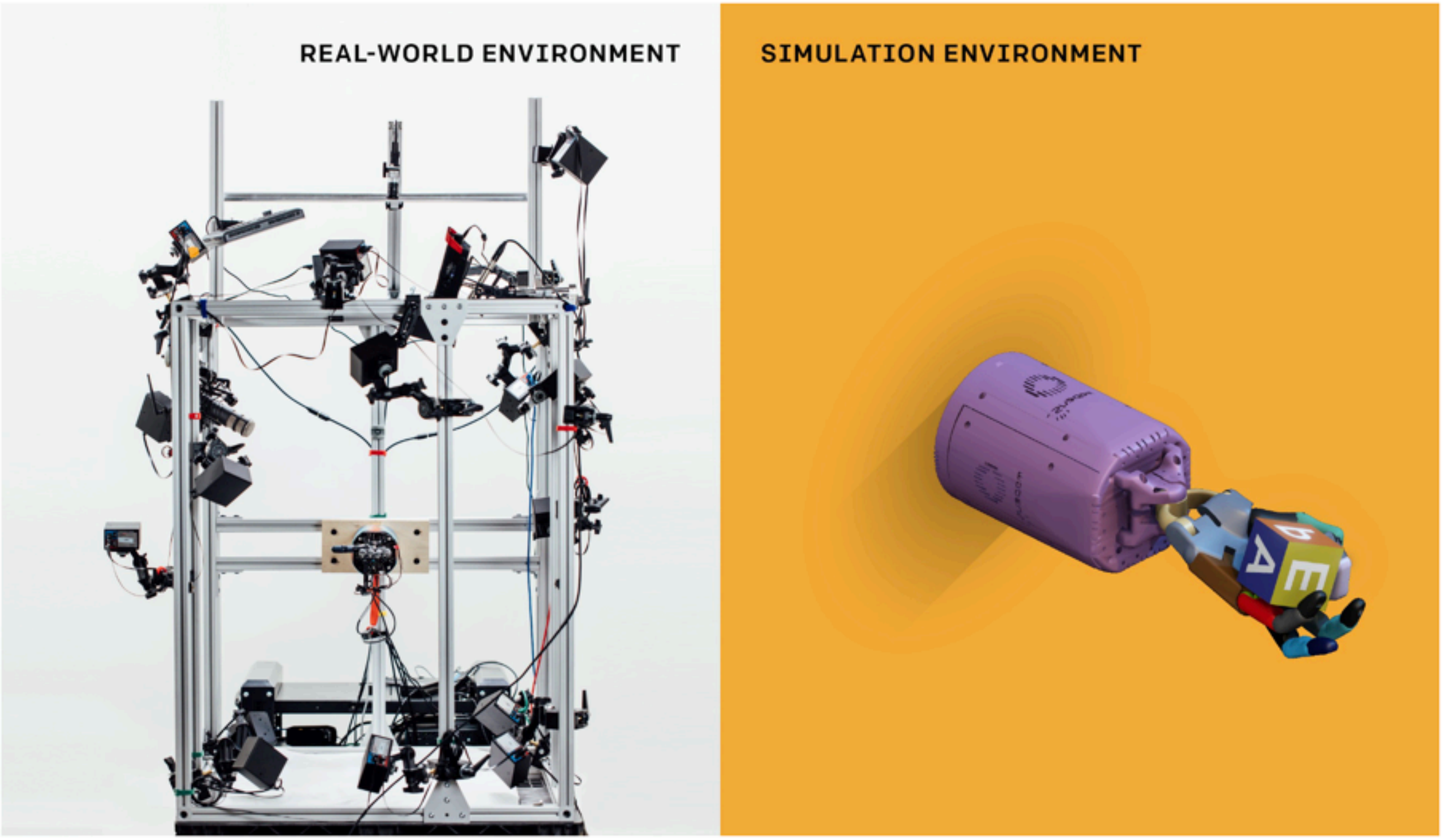
Activity!

Think-Pair-Share!

Think (30 sec): How will you train a policy in sim? (Input/output/method). What is a challenge in transferring it to real?

Pair: Find a partner

Share (45 sec): Partners exchange ideas

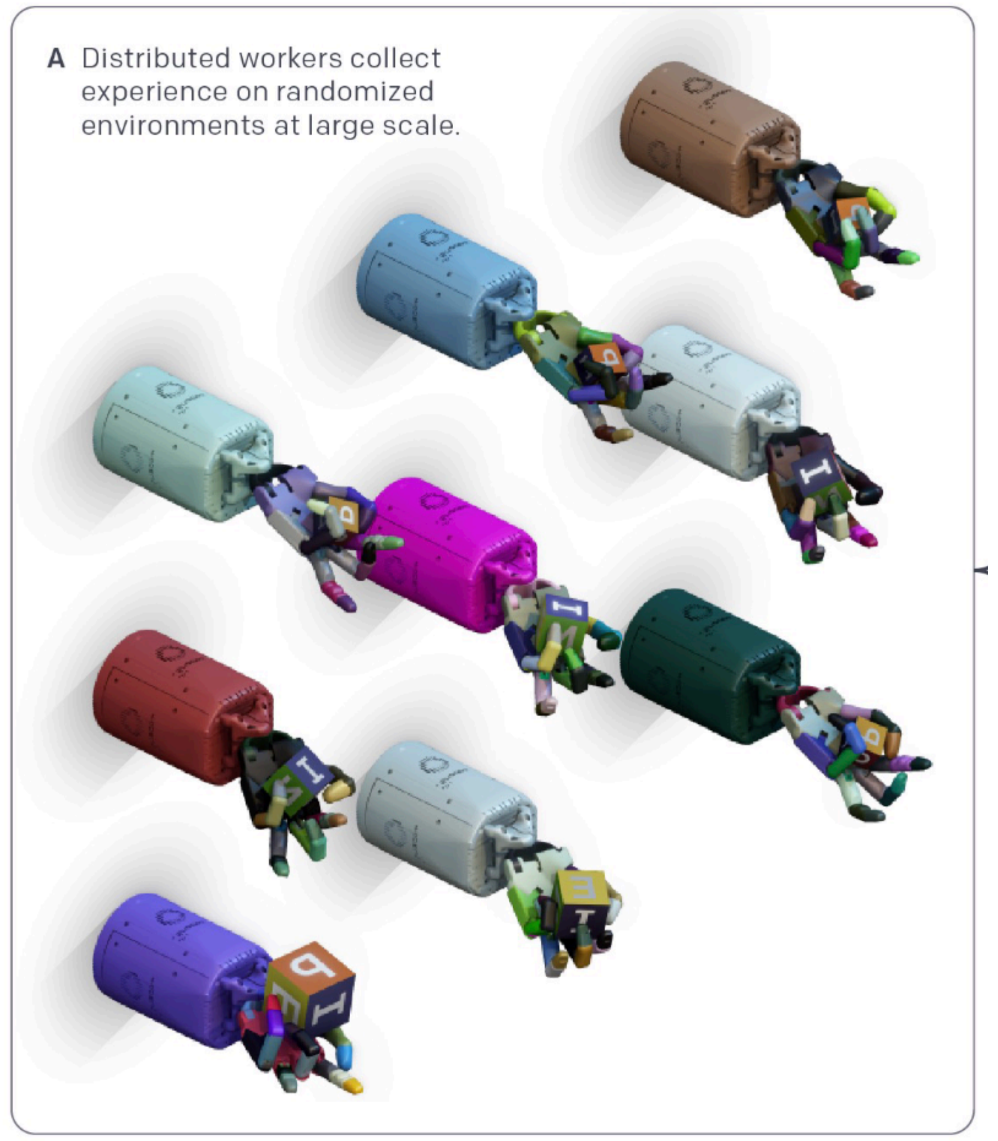


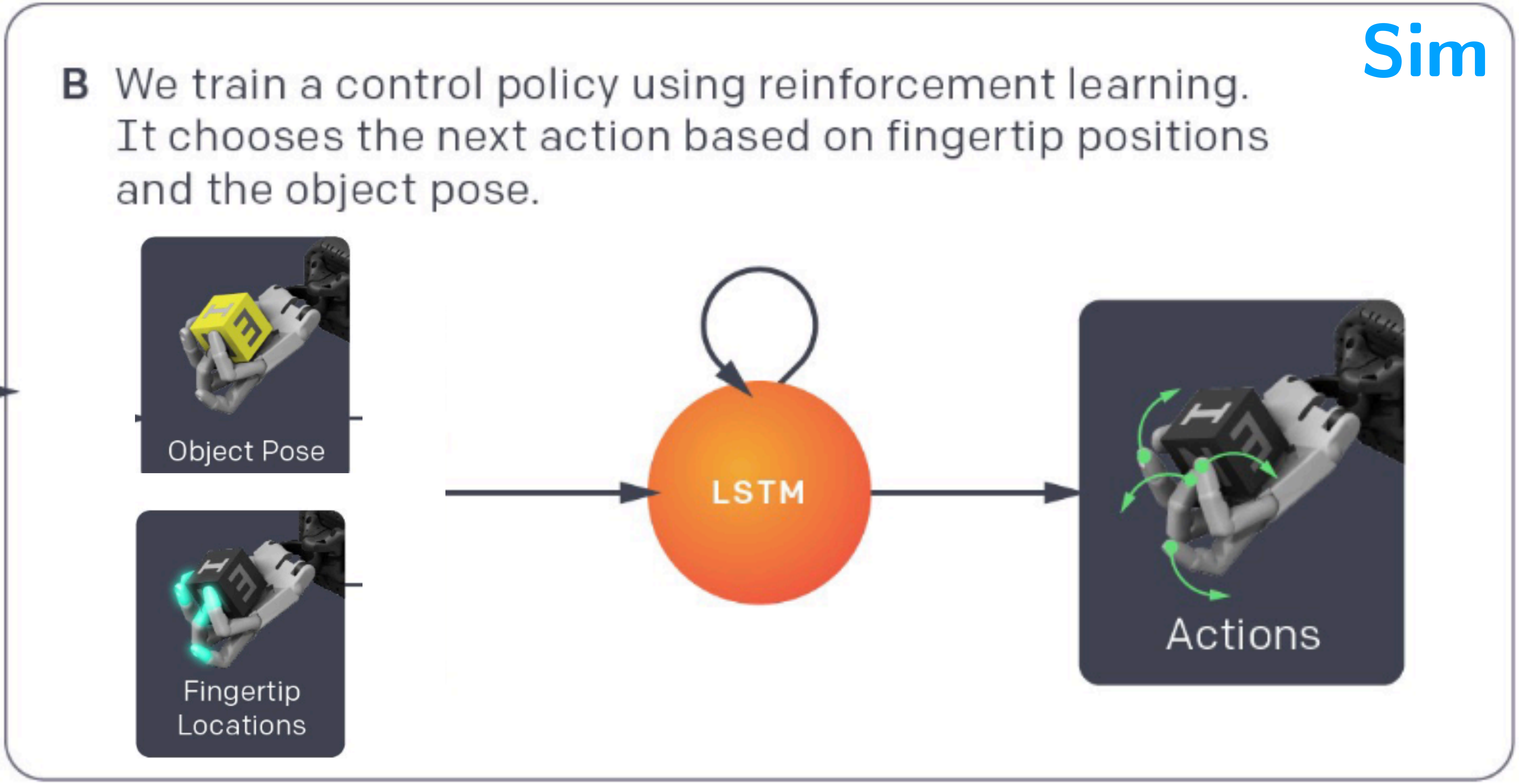
Lets see what OpenAI did!

Step 1

Sim

Setup parallelized simulator





Step 2

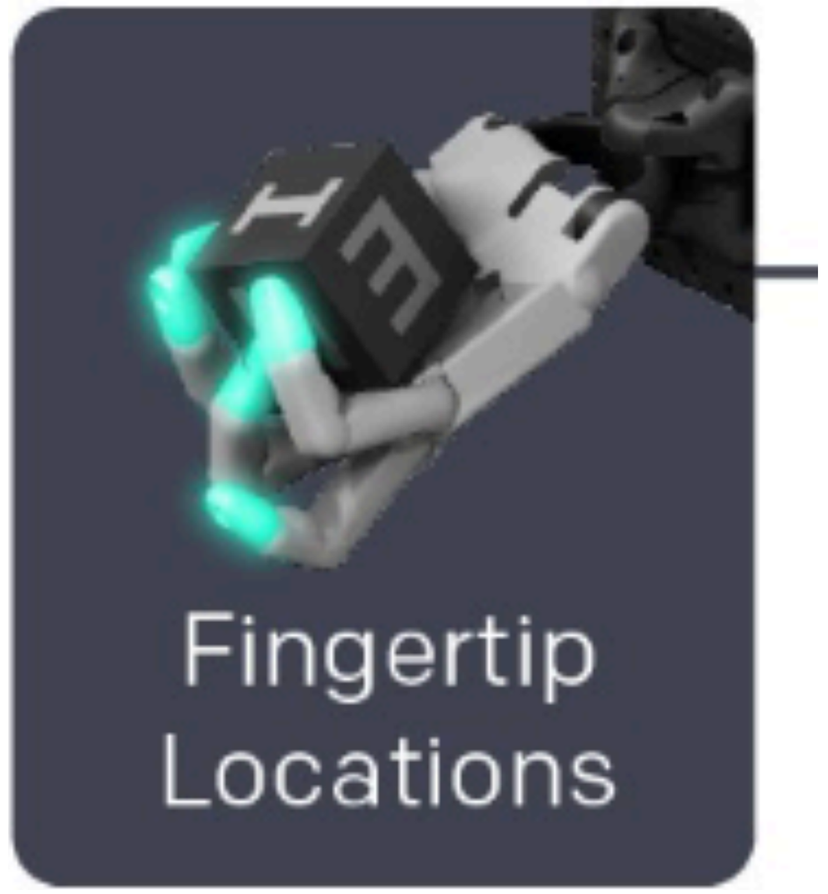
Train RL policy in sim

Why this input? Why LSTM?

$$S, A, R, \mathcal{T}$$

Let's setup the MDP for the problem!

$$\boxed{S}, A, R, \mathcal{T}$$

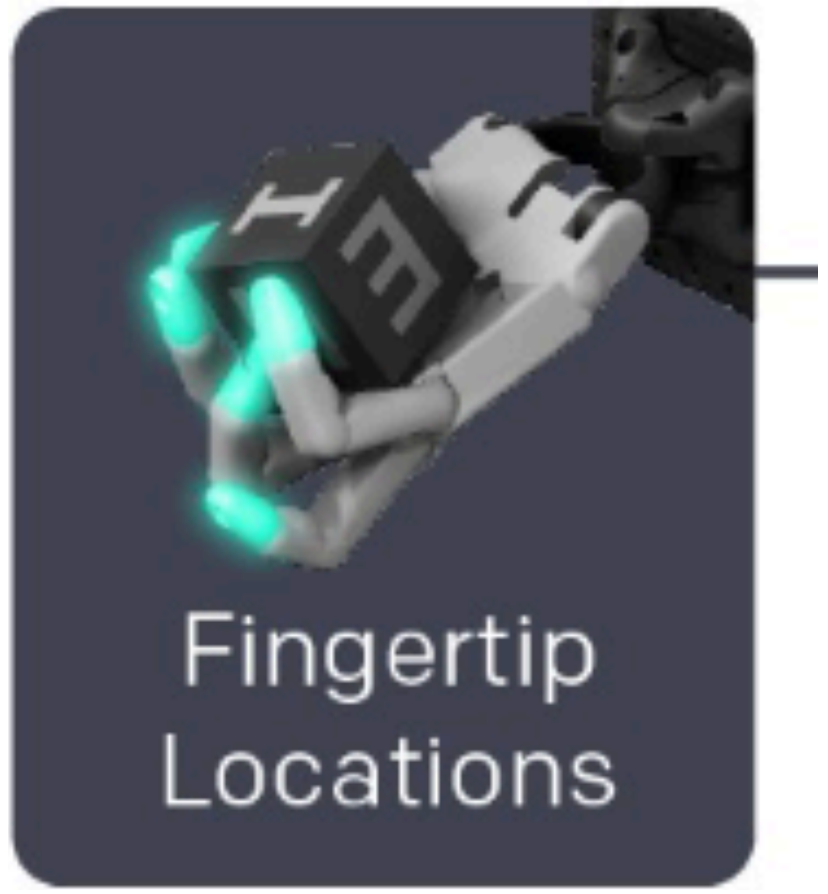


Question: Is the current object pose and fingertip location sufficient to capture state?

$$\boxed{S}, A, R, \mathcal{T}$$

No!

This is merely the current observation of a POMDP



Need to keep a HISTORY

E.g. History of observations can reveal the weight of the object or how fast the index finger can move.

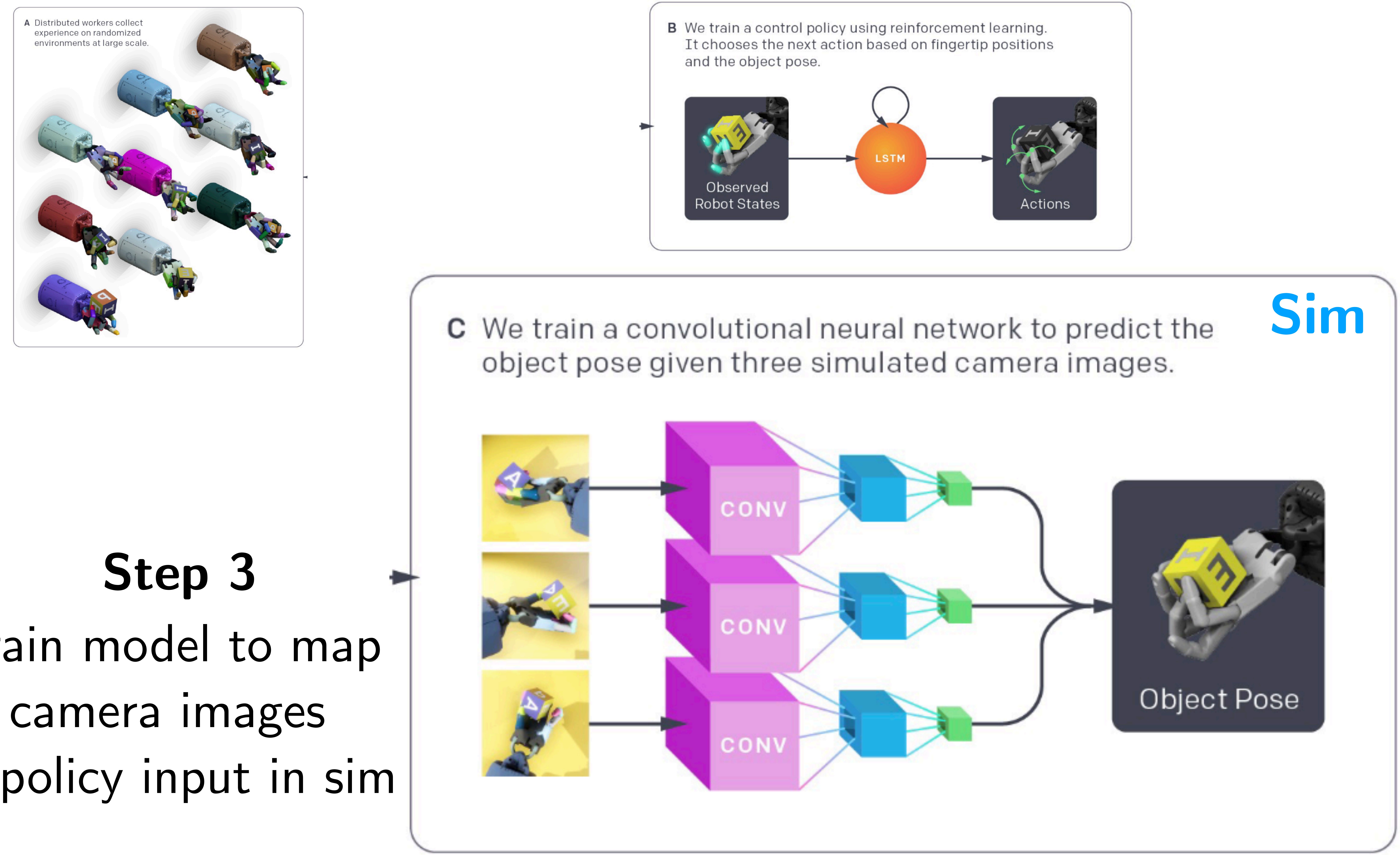
$$S\ ,\ \boxed{A}\ ,\ R\ ,\ \mathcal{T}$$

20 dimensional

Policy actions correspond to desired joints angles relative to the current ones[5] (e.g. rotate this joint by 10 degrees). While PPO can handle both continuous and discrete action spaces, we noticed that discrete action spaces work much better. This may be because a discrete probability distribution is more expressive than a multivariate Gaussian or because discretization of actions makes learning a good advantage function potentially simpler. We discretize each action coordinate into 11 bins.

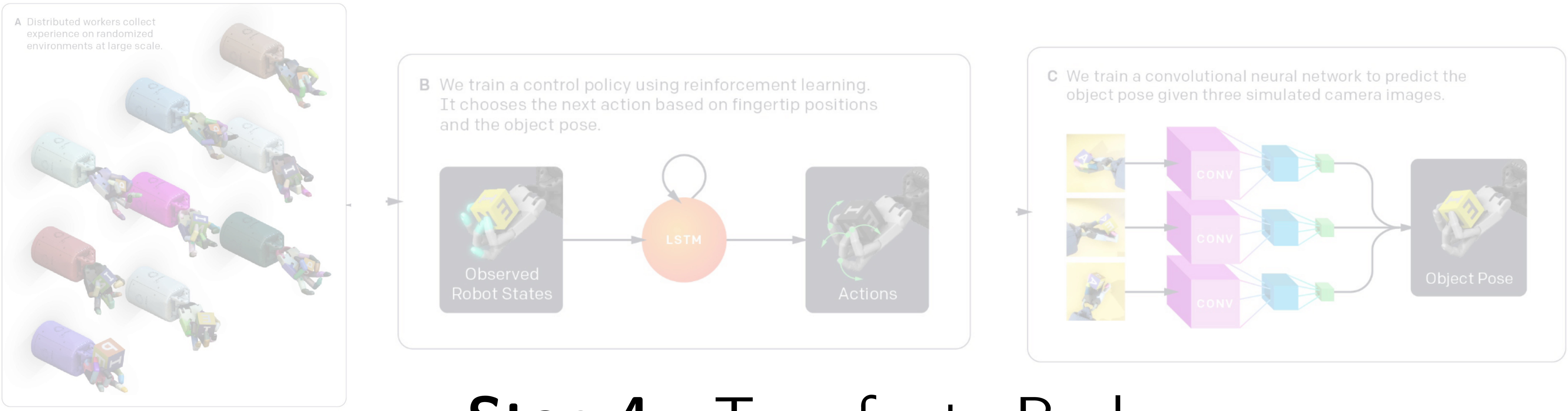
$$S\ ,\ A\ ,\ \boxed{R}\ ,\ \mathcal{T}$$

The reward given at timestep t is $r_t = d_t - d_{t+1}$, where d_t and d_{t+1} are the rotation angles between the desired and current object orientations before and after the transition, respectively. We give an additional reward of 5 whenever a goal is achieved and a reward of -20 (a penalty) whenever the object is dropped. More information about the simulation environment can be found in Appendix C.1.

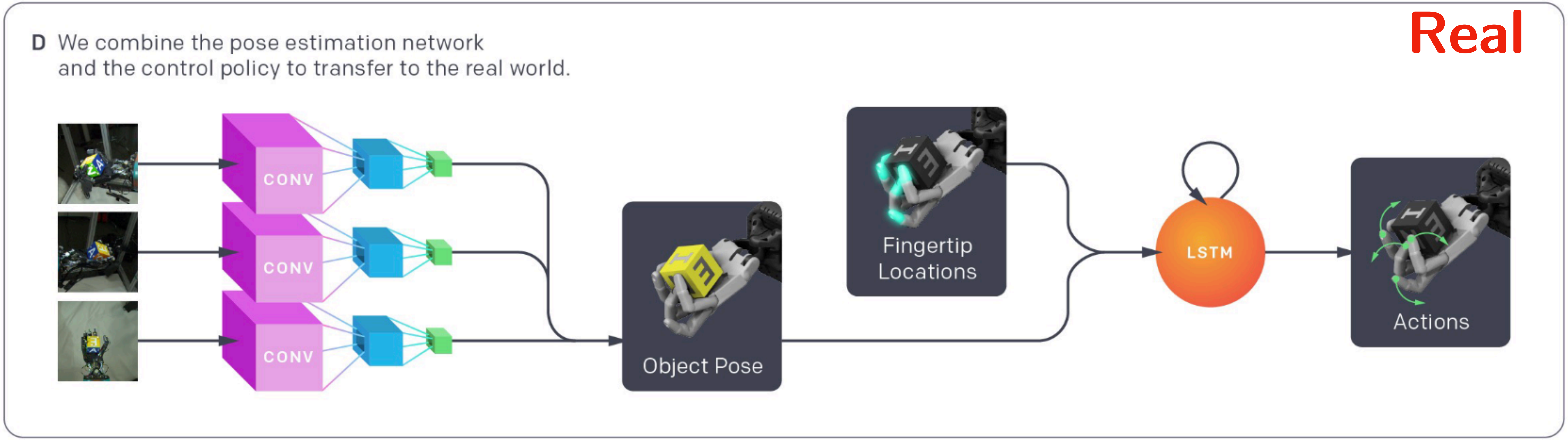


Step 3

Train model to map camera images to policy input in sim



Step 4: Transfer to Real



Sim2Real as Transferring MDPs

$$\hat{S}, A, R, \hat{\mathcal{T}} \longrightarrow S, A, R, \mathcal{T}$$

Sim

Real

There will be a **mismatch** in **state** representations and **transition**

Our policy needs to be **robust** to this mismatch

Key Idea: Add in Randomization in Sim

1. Randomize the observation

Observation noise. To better mimic the kind of noise we expect to experience in reality, we add Gaussian noise to policy observations. In particular, we apply a correlated noise which is sampled once per episode as well as an uncorrelated noise sampled at every timestep.

Key Idea: Add in Randomization in Sim

1. Randomize the observation

2. Randomize the physics

Physics randomizations. Physical parameters like friction are randomized at the beginning of every episode and held fixed. Many parameters are centered on values found during model calibration in an effort to make the simulation distribution match reality more closely. Table 1 lists all physics parameters that are randomized.

Key Idea: Add in Randomization in Sim

1. Randomize the observation
2. Randomize the physics
3. Unmodeled effects

Unmodeled effects. The physical robot experiences many effects that are not modeled by our simulation. To account for imperfect actuation, we use a simple model of motor backlash and introduce action delays and action noise before applying them in simulation. Our motion capture setup sometimes loses track of a marker temporarily, which we model by freezing the position of a simulated marker with low probability for a short period of time in simulation. We also simulate marker occlusion by freezing its simulated position whenever it is close to another marker or the object. To handle additional unmodeled dynamics, we apply small random forces to the object. Details on the concrete implementation are available in Appendix C.2.

Key Idea: Add in Randomization in Sim

Visual appearance randomizations. We randomize the following aspects of the rendered scene: camera positions and intrinsics, lighting conditions, the pose of the hand and object, and the materials and textures for all objects in the scene. Figure 4 depicts some examples of these randomized environments. Details on the randomized properties and their ranges are available in Appendix C.2.

1. Randomize the observation
2. Randomize the physics
3. Unmodeled effects
4. Visual randomization

Today's class

- [x] Sim2Real: The double-edged sword

  Case study: OpenAI Dactyl Hand

- [ ] Teacher->Student distillation

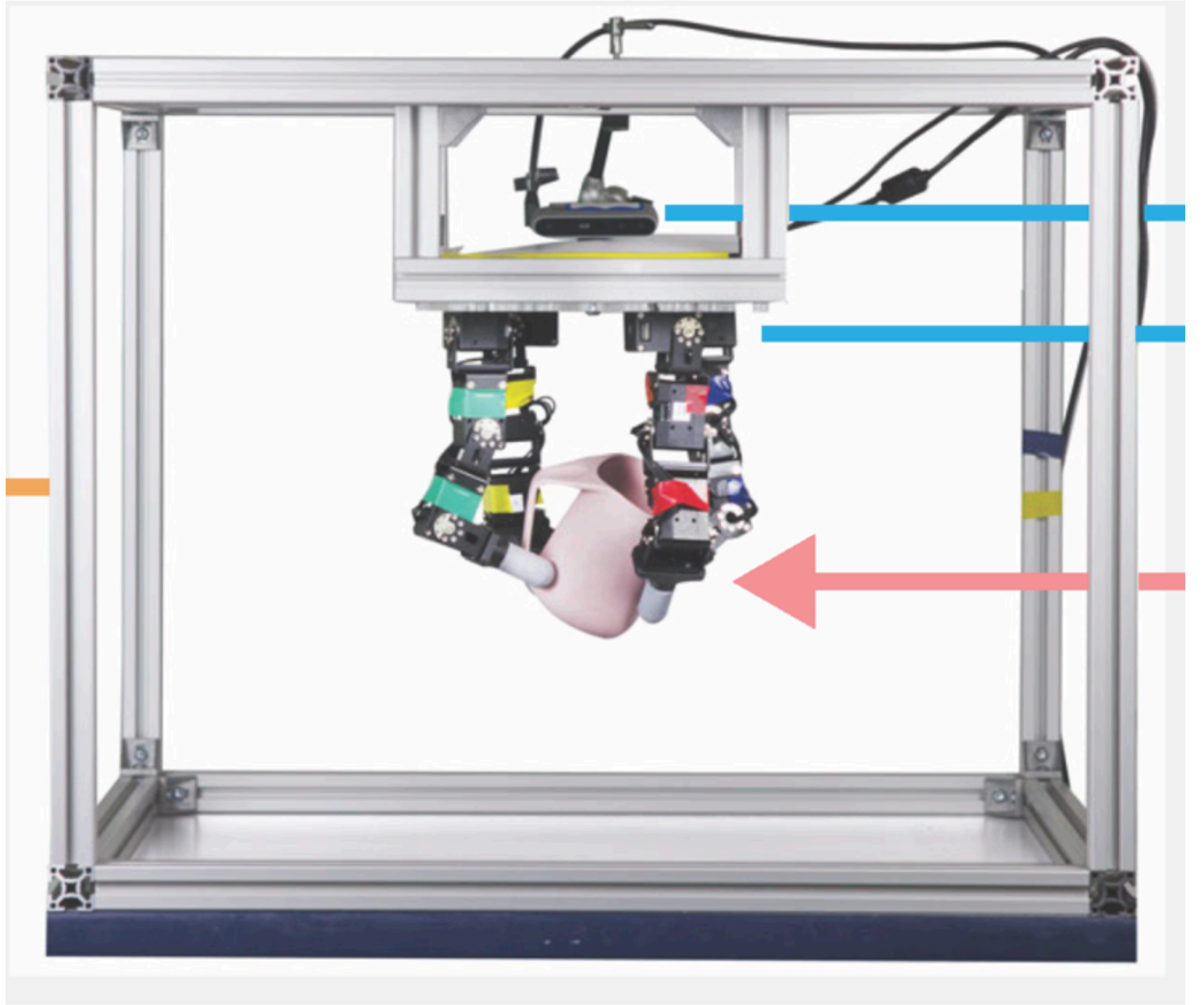
  Case study: Visual Dexterity

- [ ] Imitation Learning with Privileged Information

What if we made the problem much much harder?

Visual Dexterity: In-Hand Reorientation of Novel and Complex Object Shapes

Tao Chen[1,2], Megha Tippur[2], Siyang Wu[3], Vikash Kumar[4], Edward Adelson[2], Pulkit Agrawal*[1,2,5]



Upside down object manipulation

From 12 cameras to 1 camera

Generalize to lots of different objects


Goal orientation

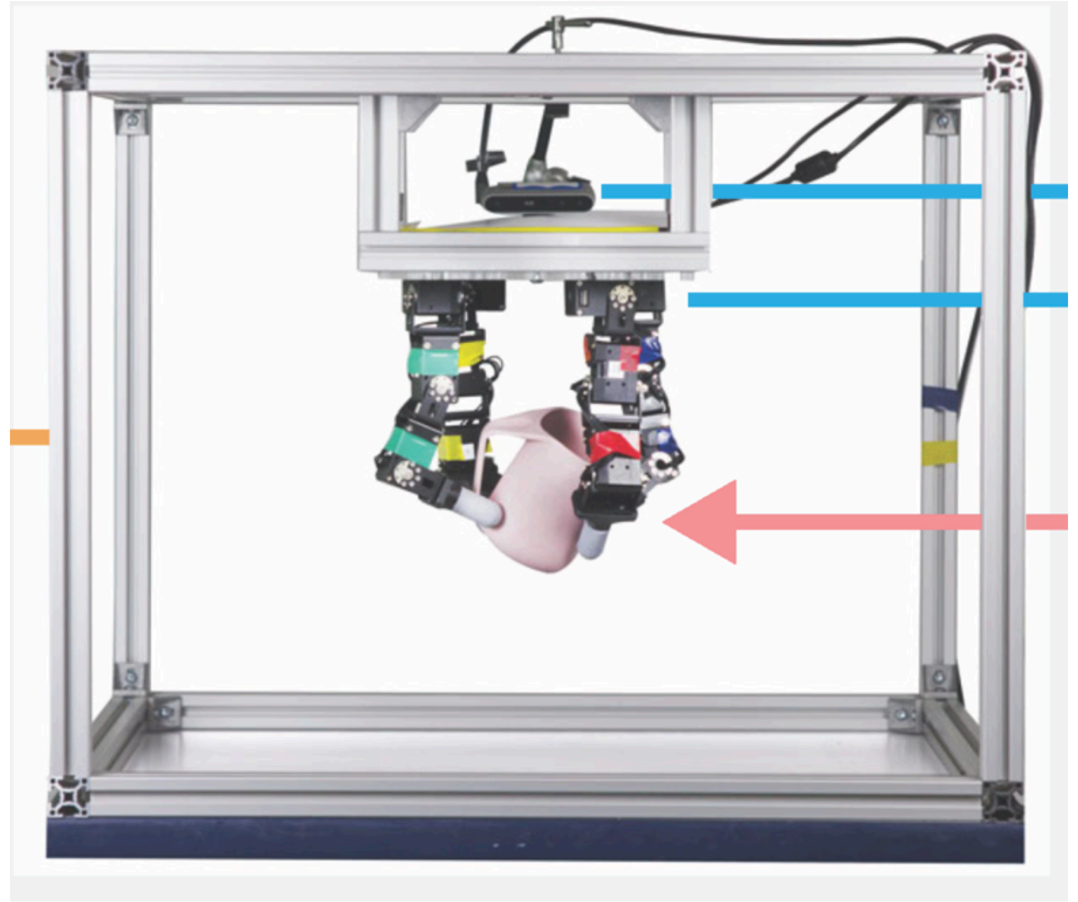
Activity!

Think-Pair-Share!

Think (30 sec): Why can't we apply OpenAI strategy to this setting? What are the challenges?

Pair: Find a partner

Share (45 sec): Partners exchange ideas



The Challenge

Doing RL purely based on observation data (point clouds) is very challenging

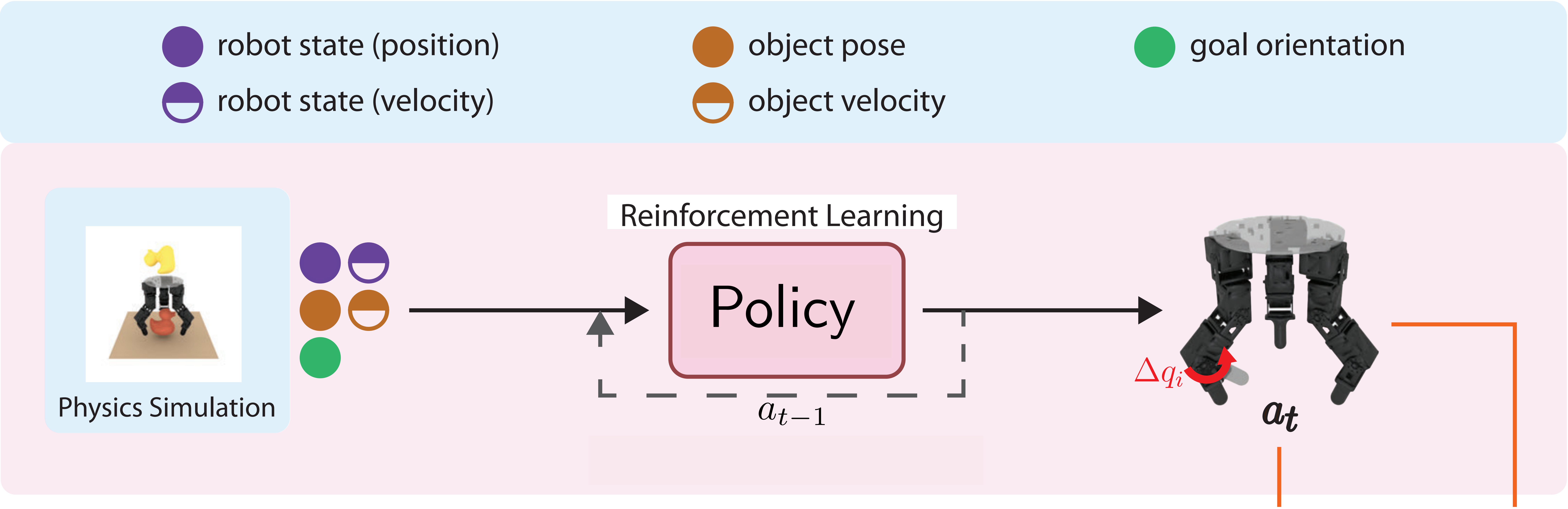
The policy needs to learn 2 things simultaneosly:

1. What are good visual features?

2. What are good actions?

Can we train the RL using privileged information that is present in sim during training?

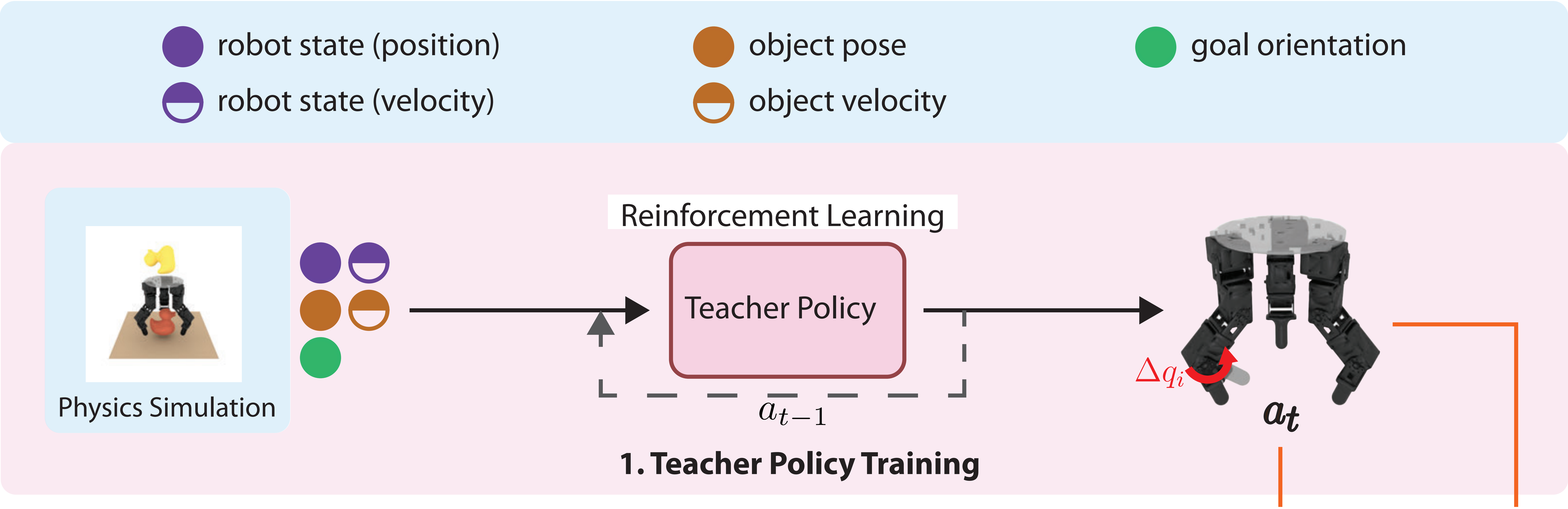
RL with provileged information

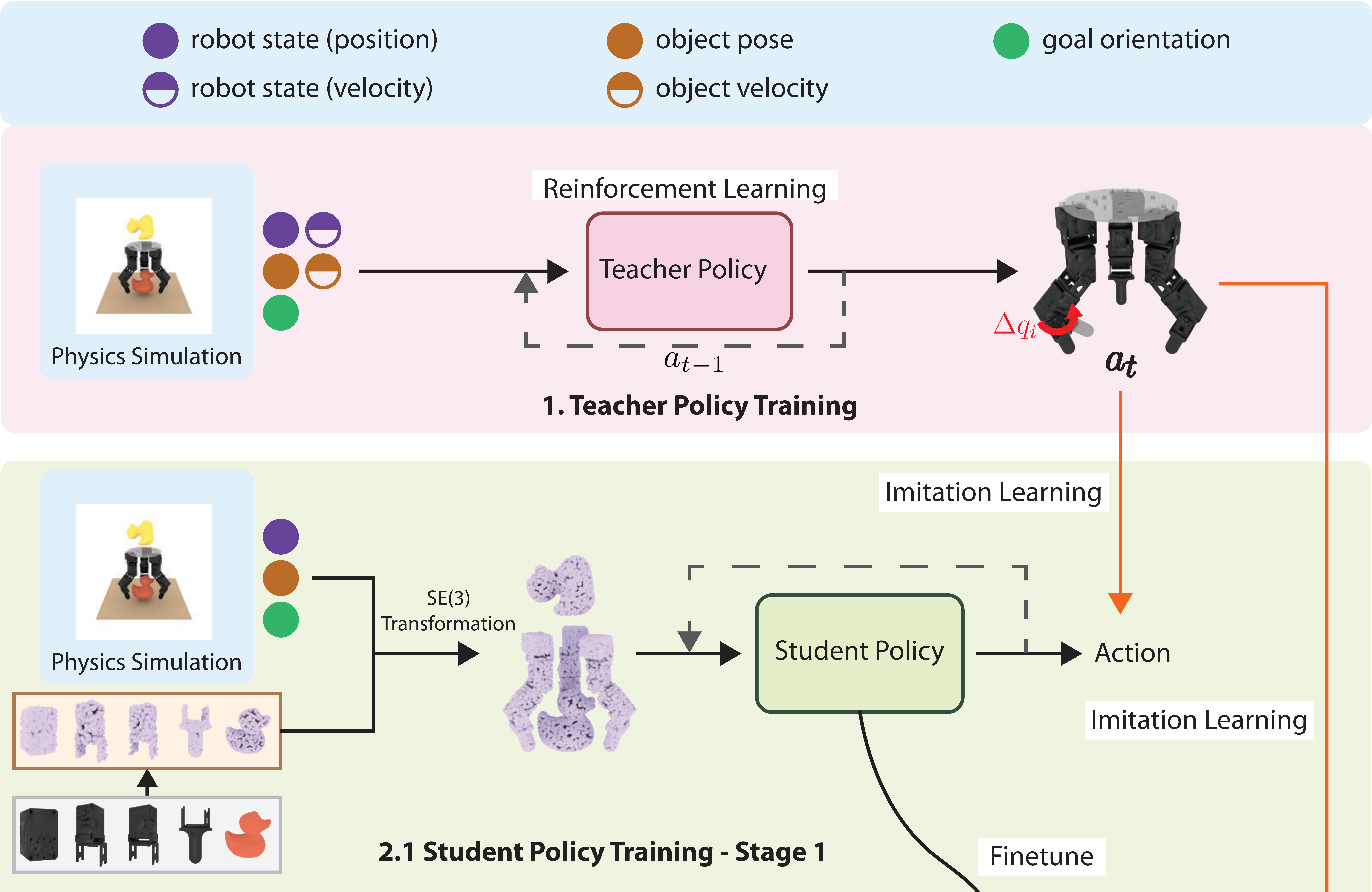


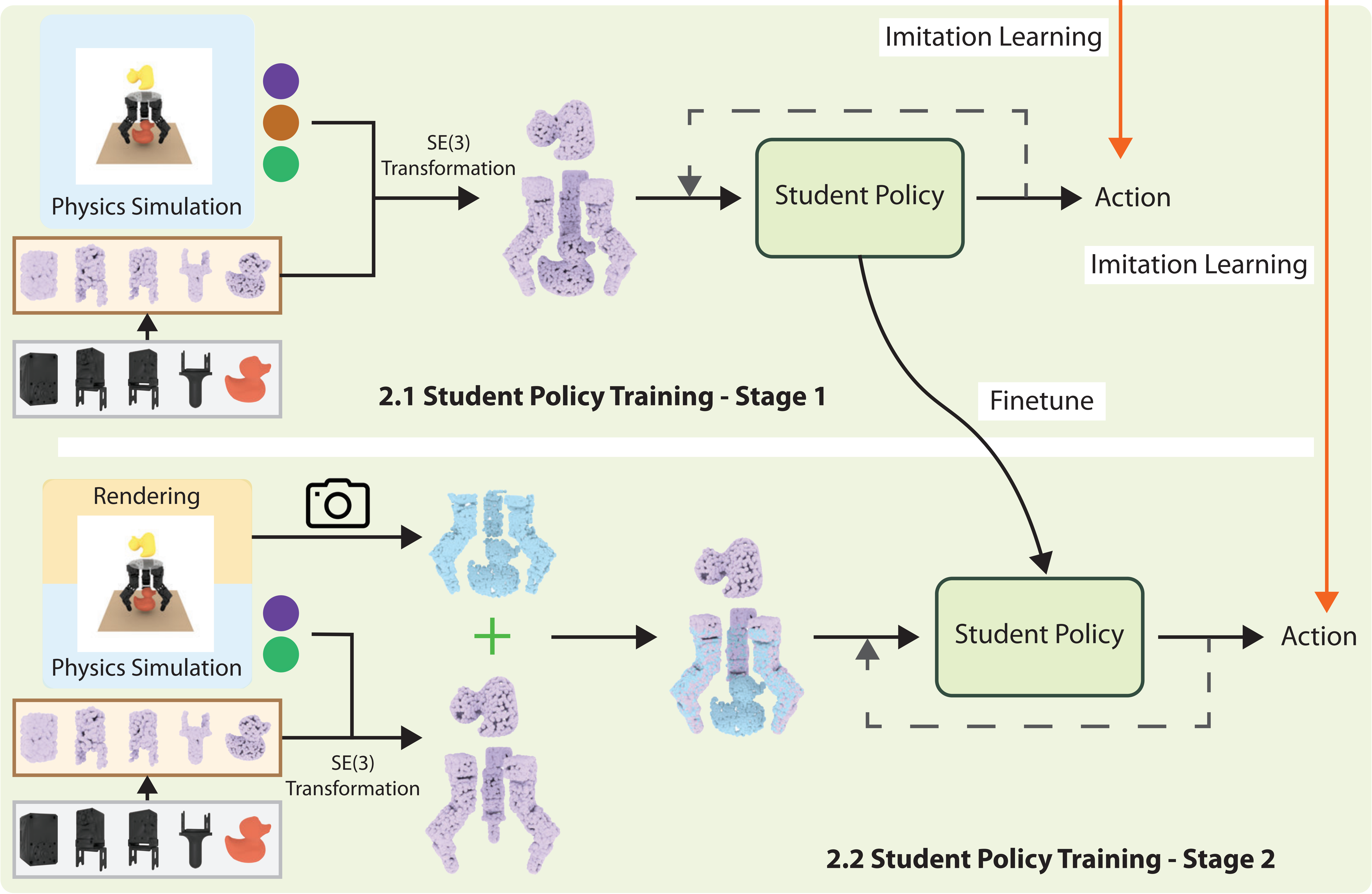
But if we train a policy using privileged information in sim, how will we run it in real where we don't have privileged information?

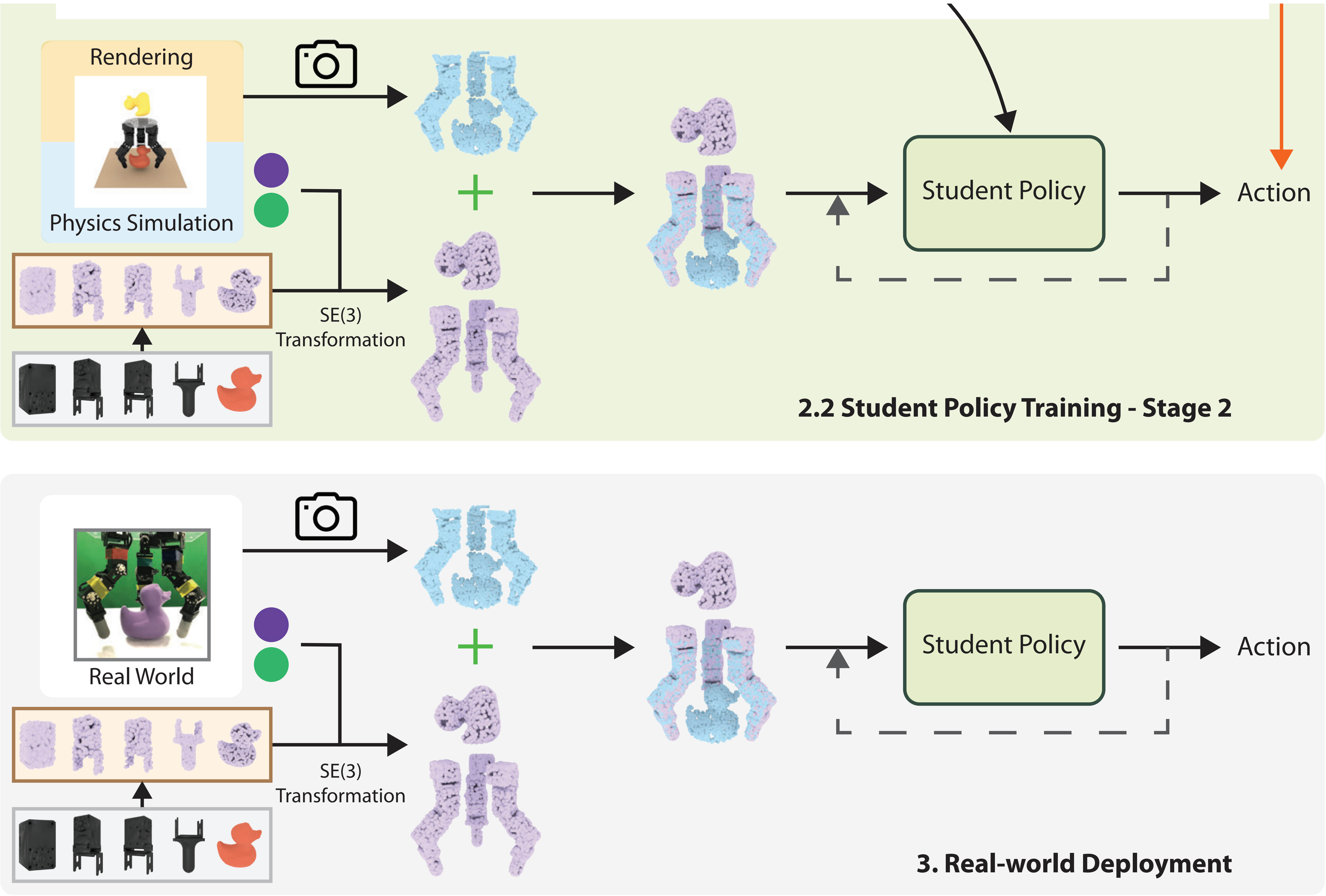
Can we train the RL using privileged information that is present in sim during training?

Can we imitate the RL policy with a policy that only has access to real sensor information?


robot state (position)
robot state (velocity)
object pose
object velocity
goal orientation
Physics Simulation
Reinforcement Learning
Teacher Policy
a_{t-1}
Δq_i
a_t
1. Teacher Policy Training


robot state (position)
robot state (velocity)
object pose
object velocity
goal orientation
Physics Simulation
Reinforcement Learning
Teacher Policy
a_{t-1}
Δq_i
a_t
1. Teacher Policy Training
Imitation Learning
Physics Simulation
SE(3)
Transformation
Student Policy
Action
Imitation Learning
2.1 Student Policy Training - Stage 1
Finetune


Imitation Learning
Physics Simulation
SE(3)
Transformation
Student Policy
Action
Imitation Learning
Finetune
2.1 Student Policy Training - Stage 1
Rendering
Physics Simulation
SE(3)
Transformation
Student Policy
Action
2.2 Student Policy Training - Stage 2



Rendering
Physics Simulation
SE(3)
Transformation
Student Policy
Action
2.2 Student Policy Training - Stage 2
Real World
SE(3)
Transformation
Student Policy
Action
3. Real-world Deployment

Today's class

- ☑ Sim2Real: The double-edged sword

  Case study: OpenAI Dactyl Hand

- ☑ Teacher->Student distillation

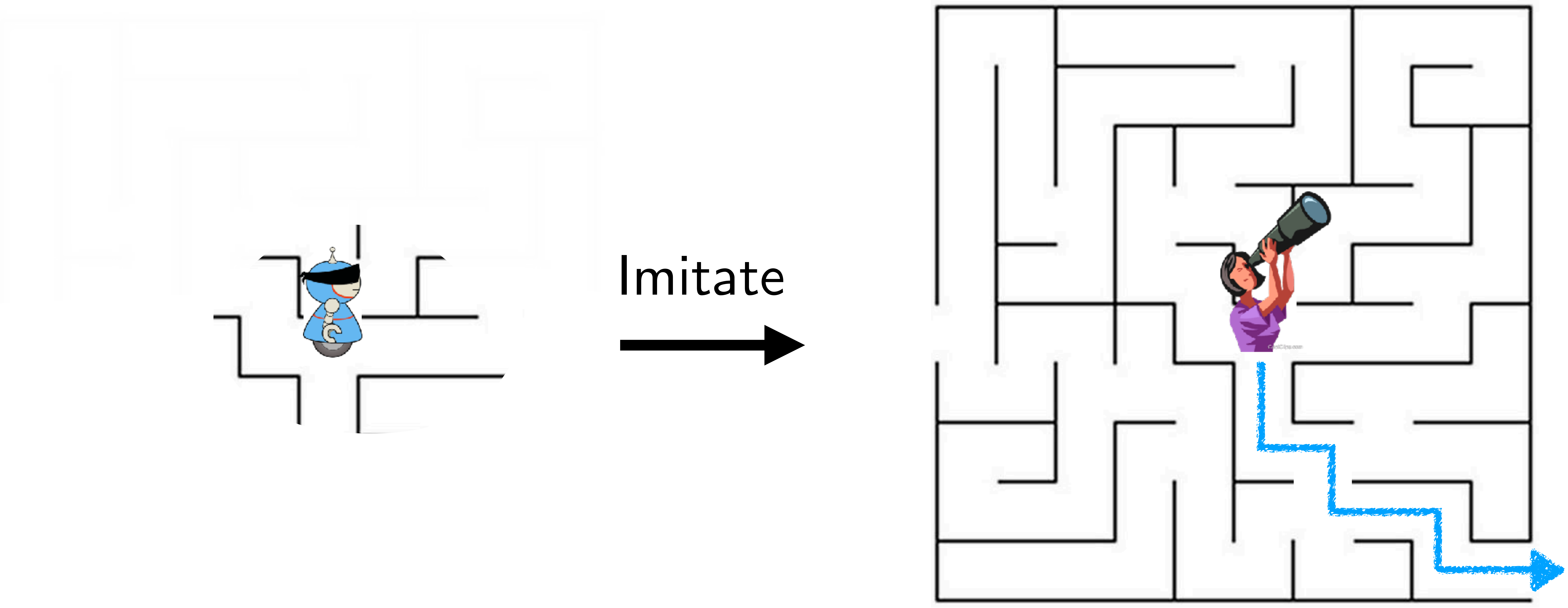
  Case study: Visual Dexterity

- ☐ Imitation Learning with Privileged Information

How should we imitate experts that have privileged information?


PRIVILEGED

Imitating Experts with Privileged Information



Learner
w/ limited sensing

Expert
can see further

Just do Behavior Cloning?

1. Collect data from experts (who have privileged information)

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

2. Train a policy that maps history to action

$$h_t^* = \{o_t^*, a_{t-1}^*, o_{t-1}^*, \ldots, o_{t-k}^*\} \qquad \pi : h_t^* \to a_t^*$$

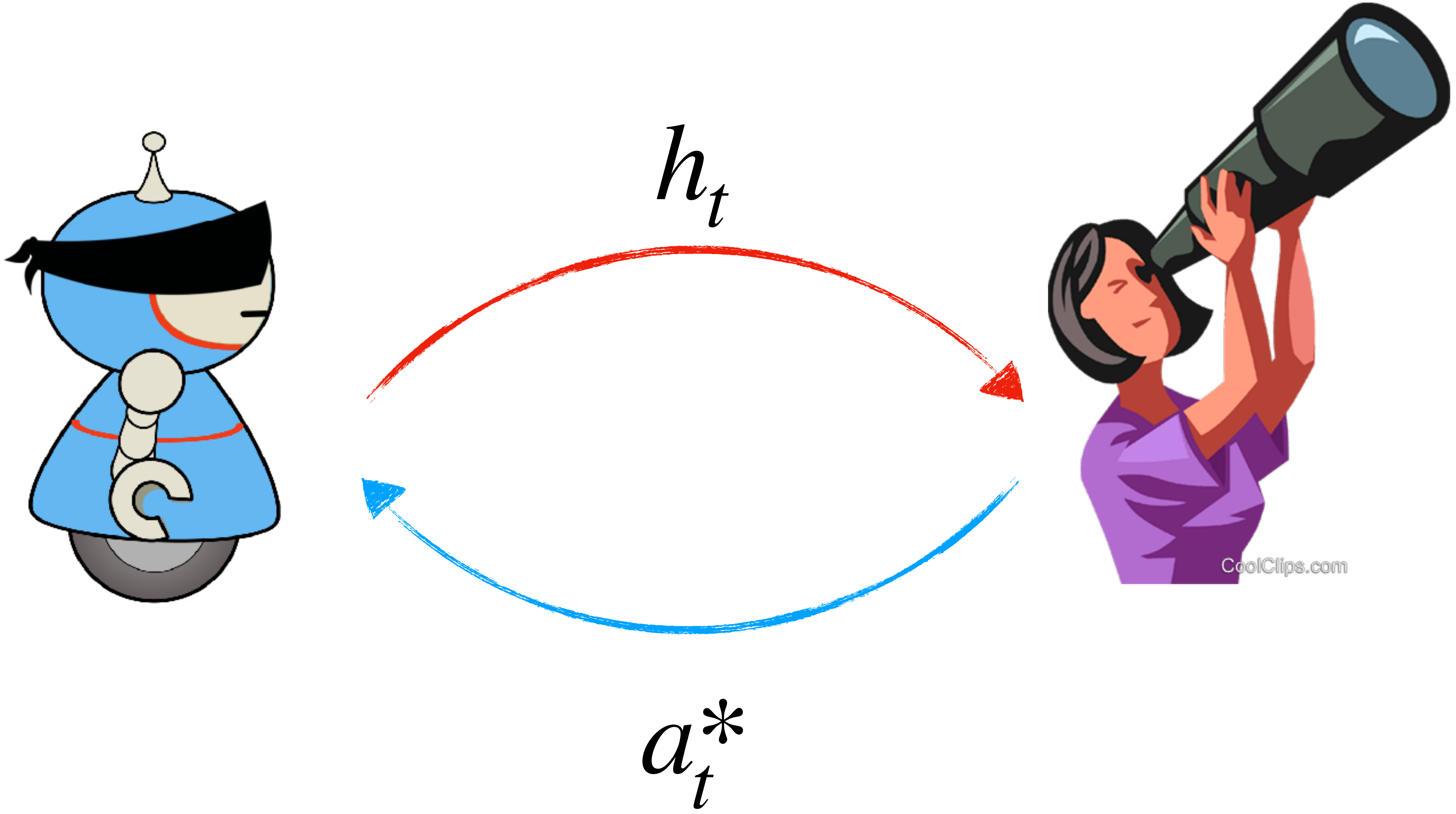
Why history?

Quiz!

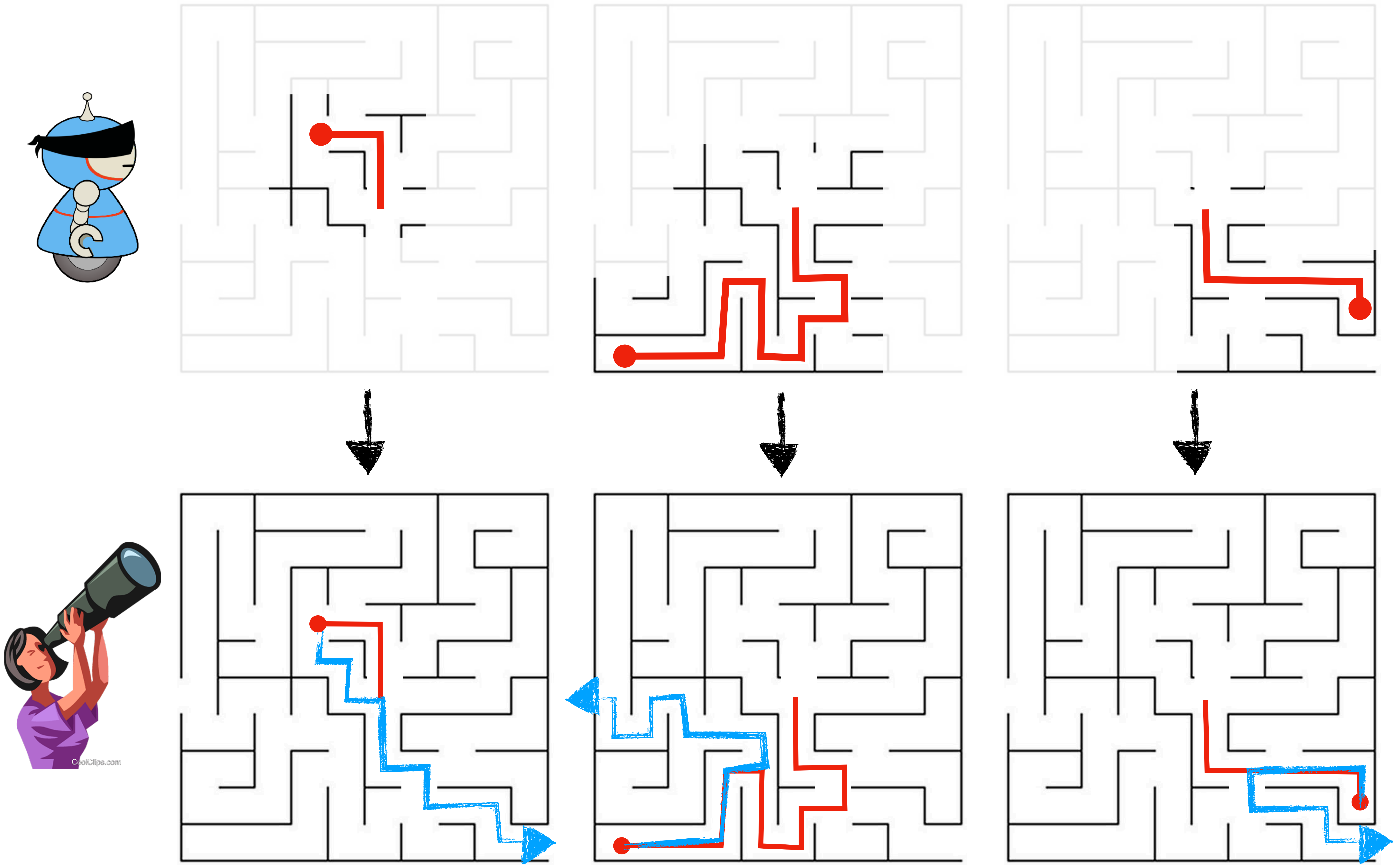
When poll is active respond at **PollEv.com/sc2582**

Send **sc2582** to **22333**

Solution: Interactively query expert



Solution: Interactively query expert



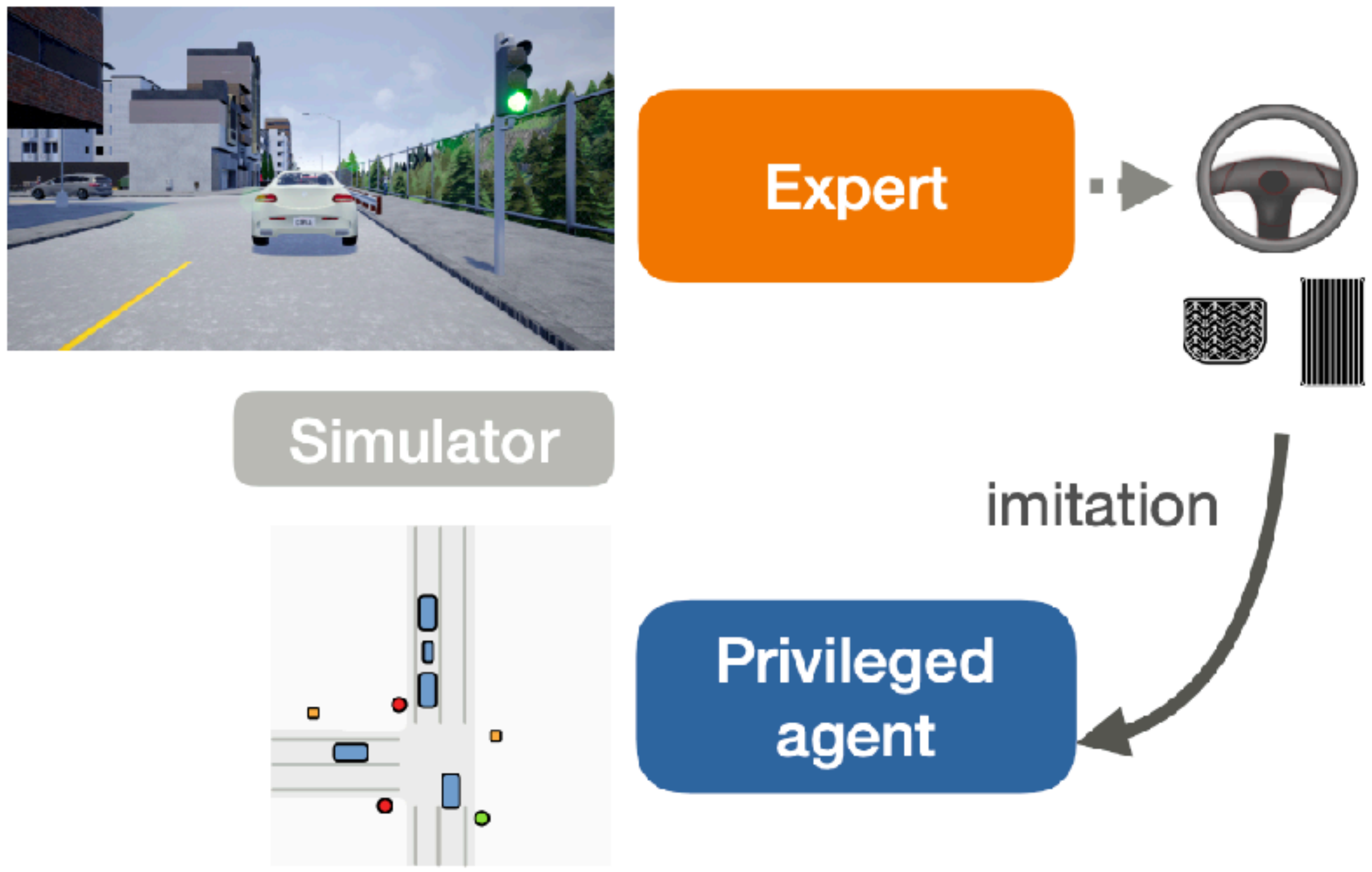
e.g DAGGER

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

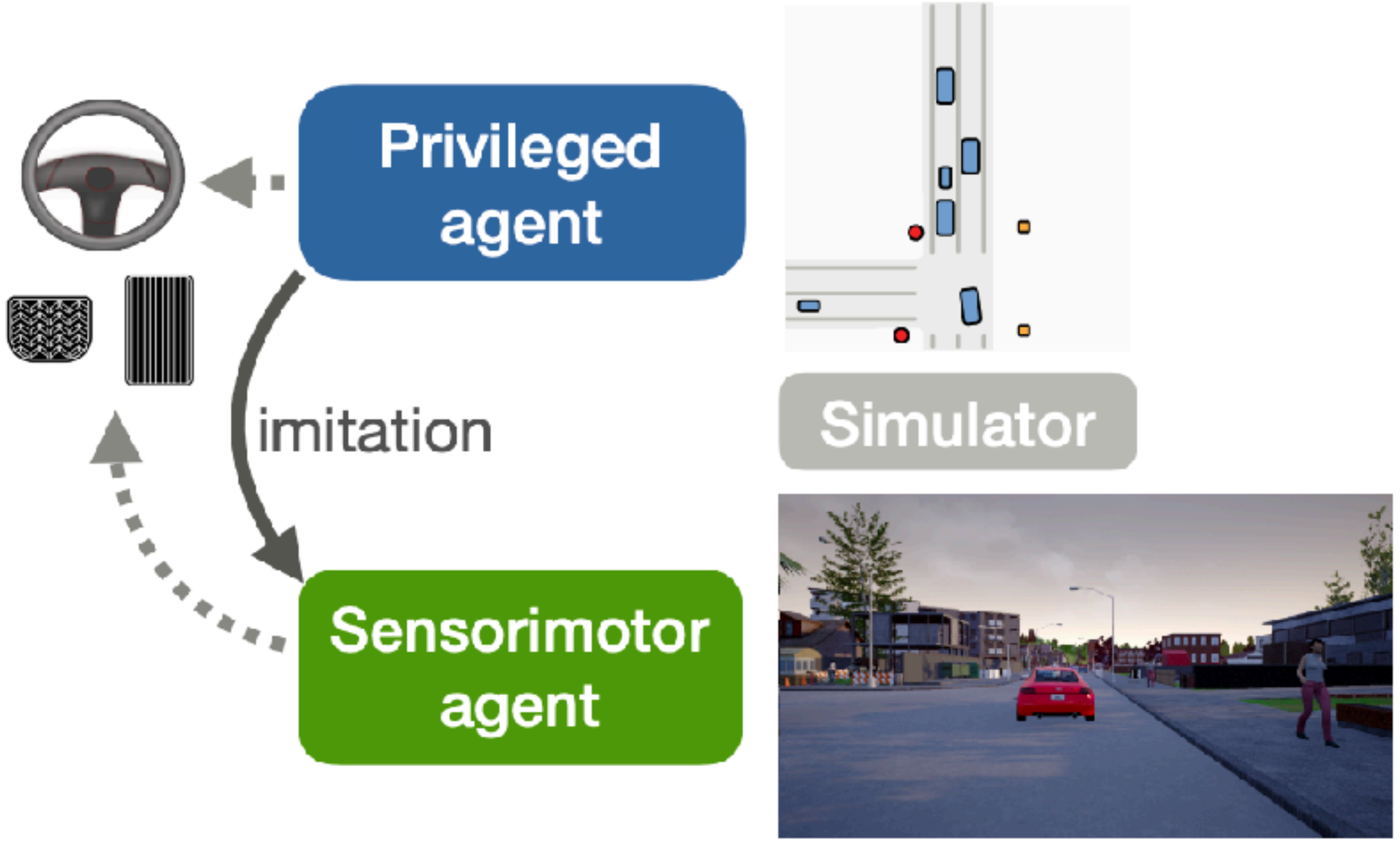
and repeat!

Incredibly successful idea that has worked across a lot of application!

Privileged Information: Self-driving



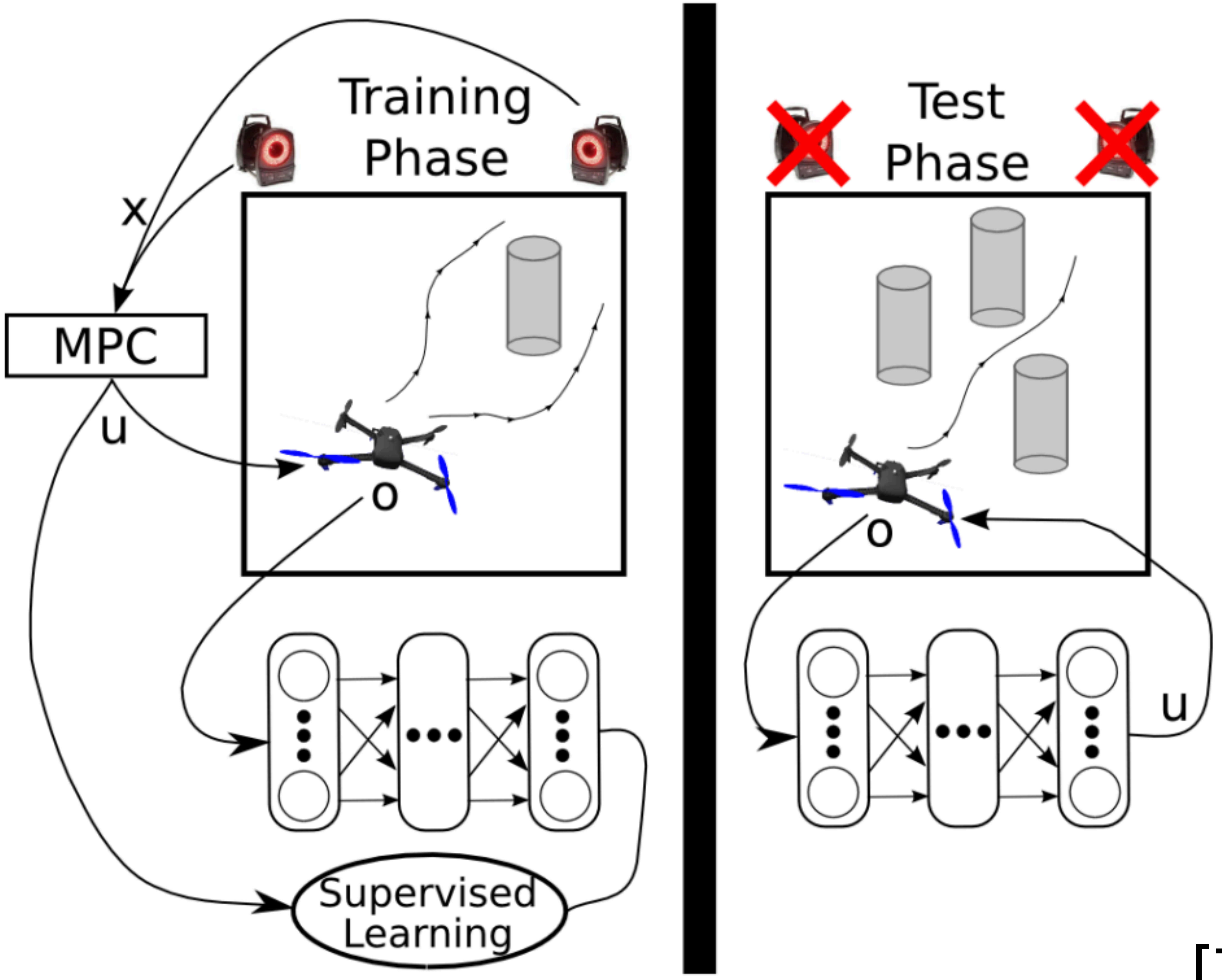
(a) Privileged agent imitates the expert



(b) Sensorimotor agent imitates the privileged agent

[Chen et al. 2020]

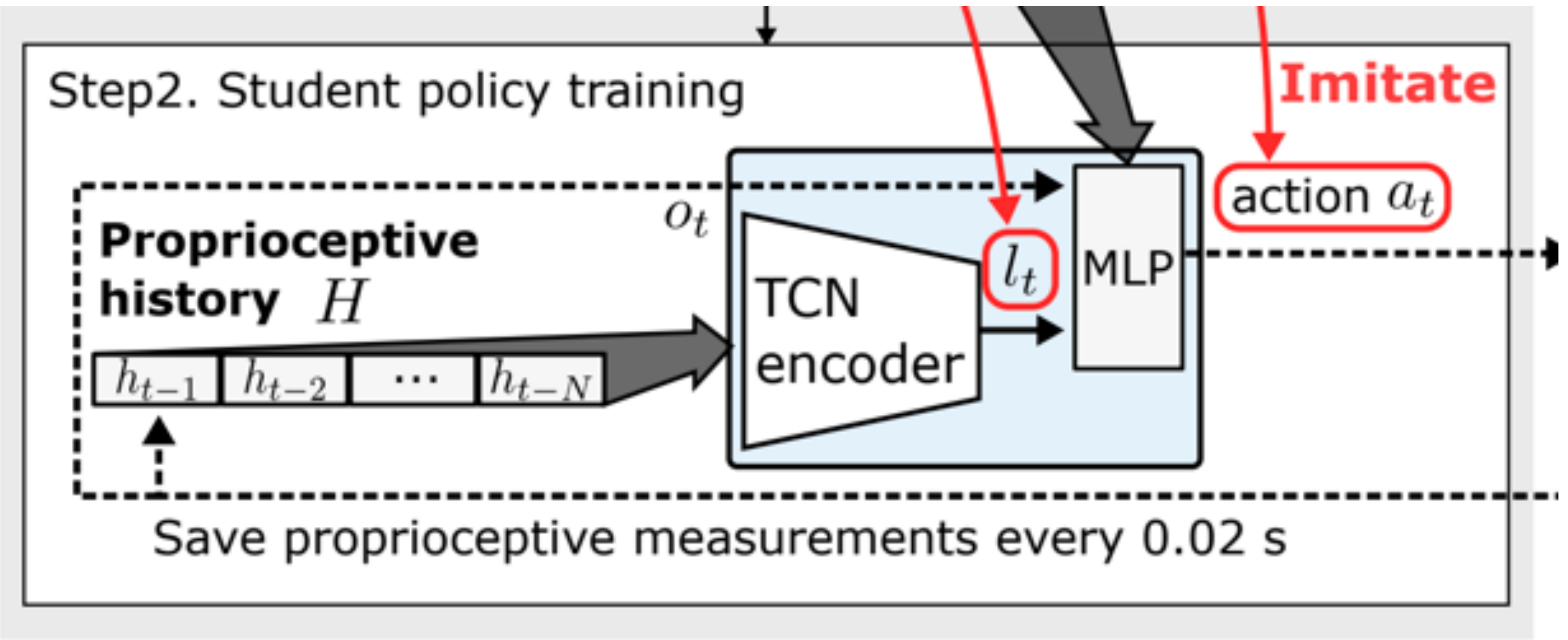
Privileged Information: UAV Navigation



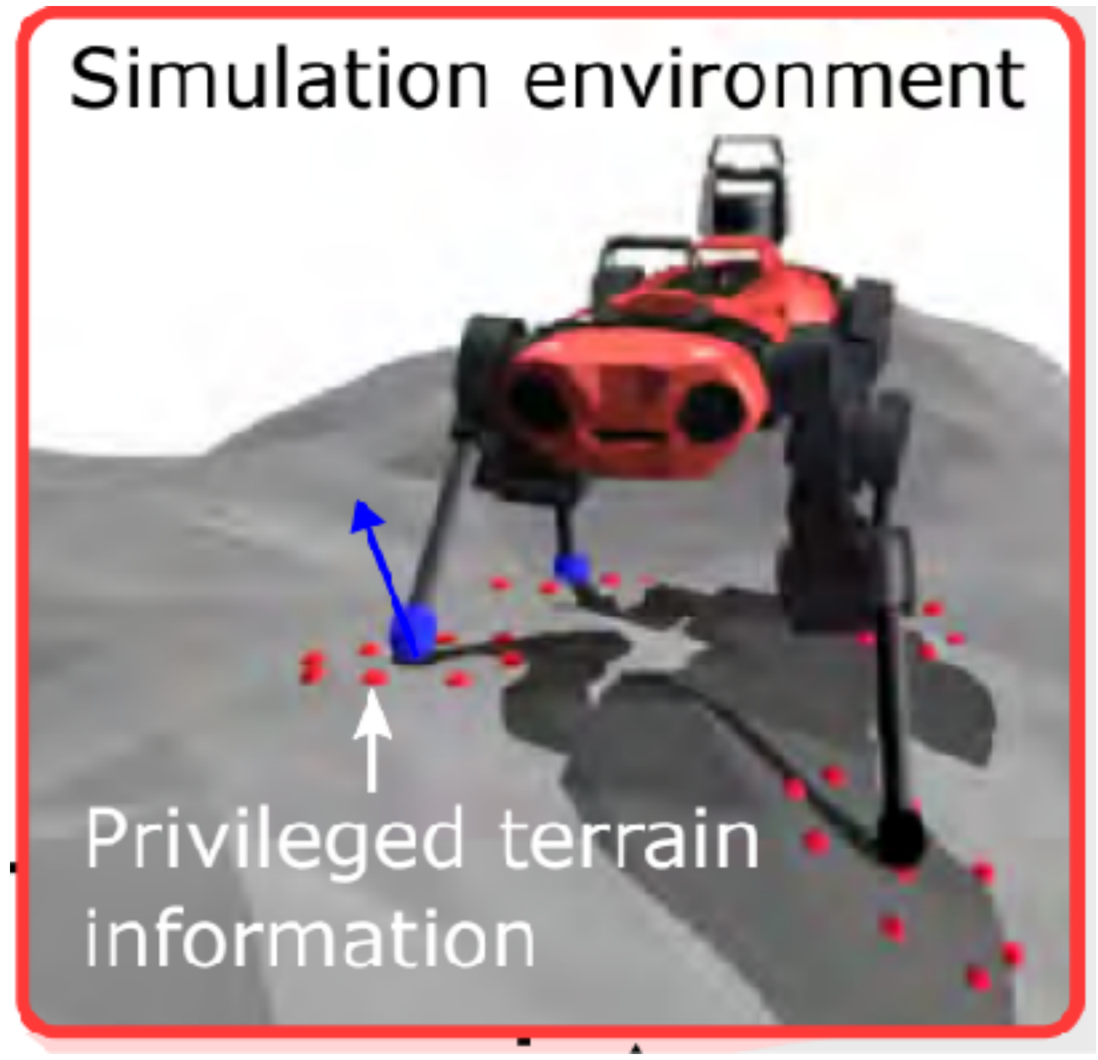
[Zhang et al. 2016]

Privileged Information: Legged Locomotion

Student Policy



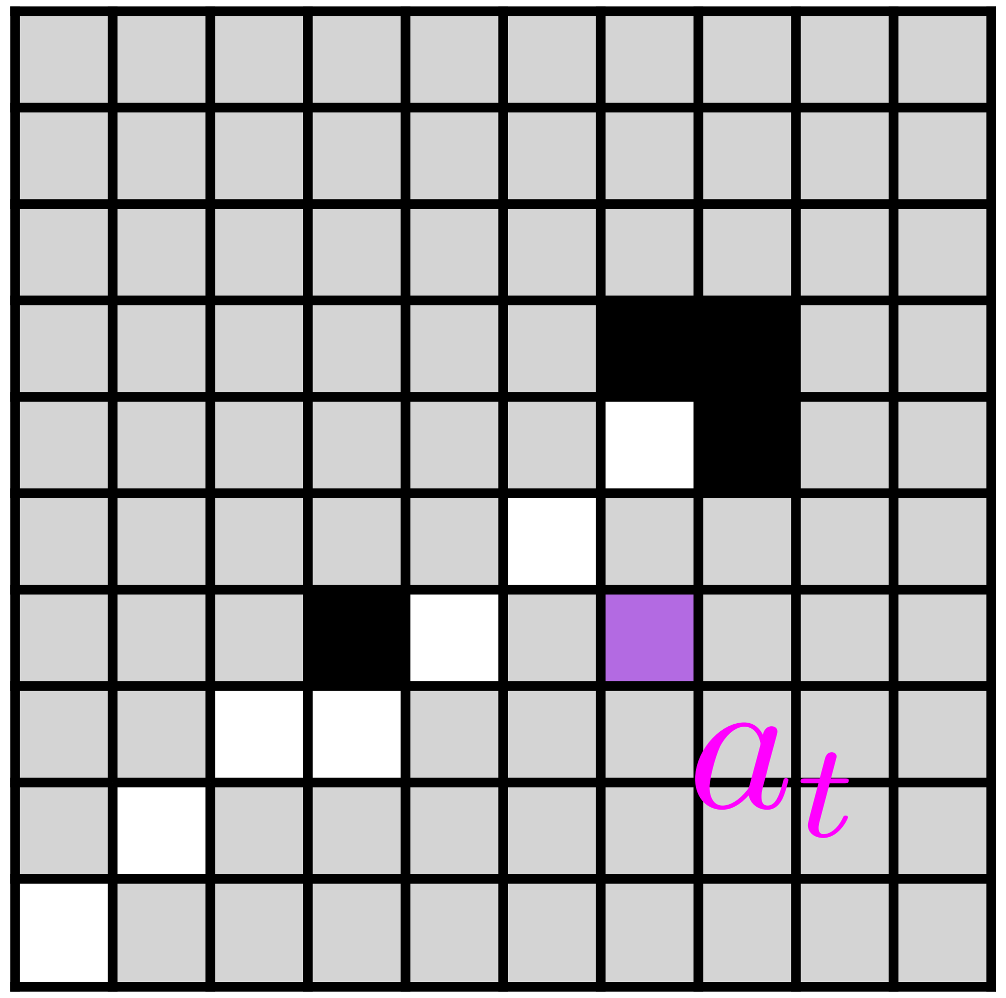
Imitate



Teacher Policy

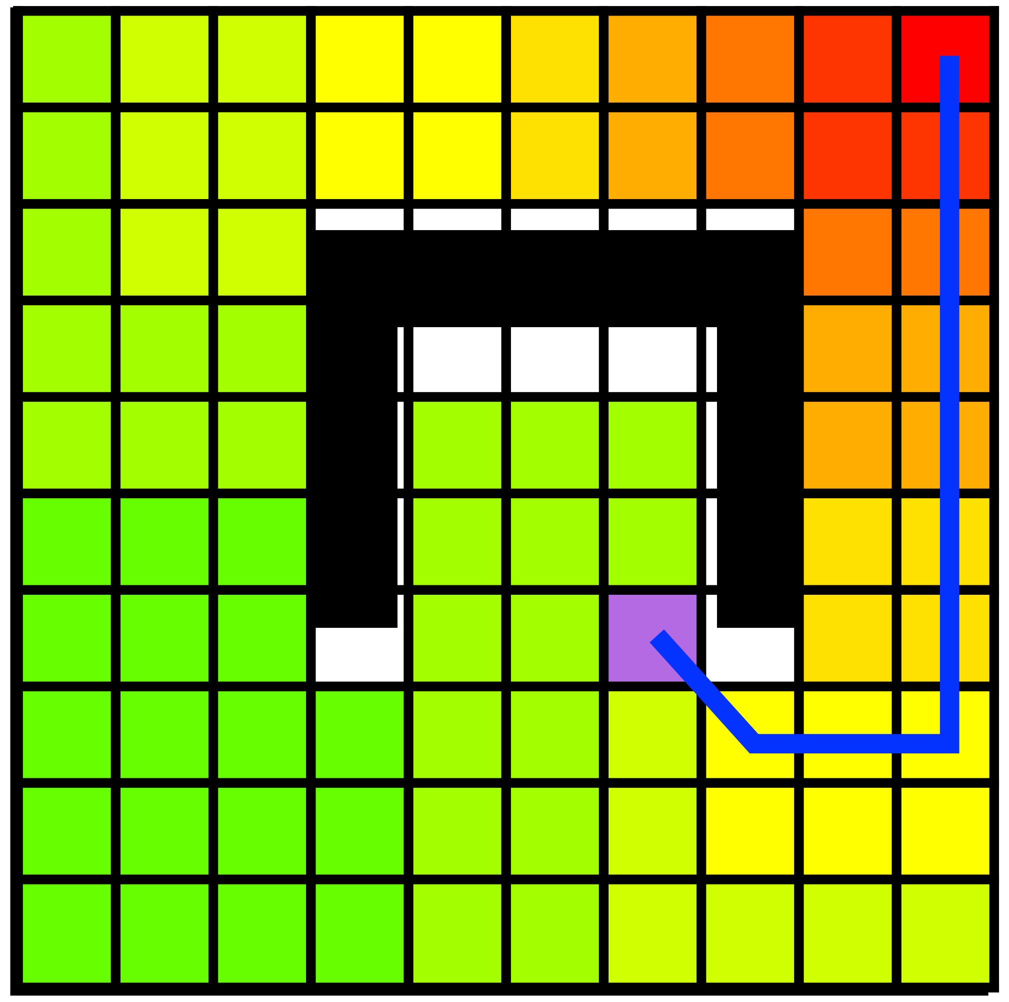
[Lee et al. 2020]

Privileged Information: Motion Planning

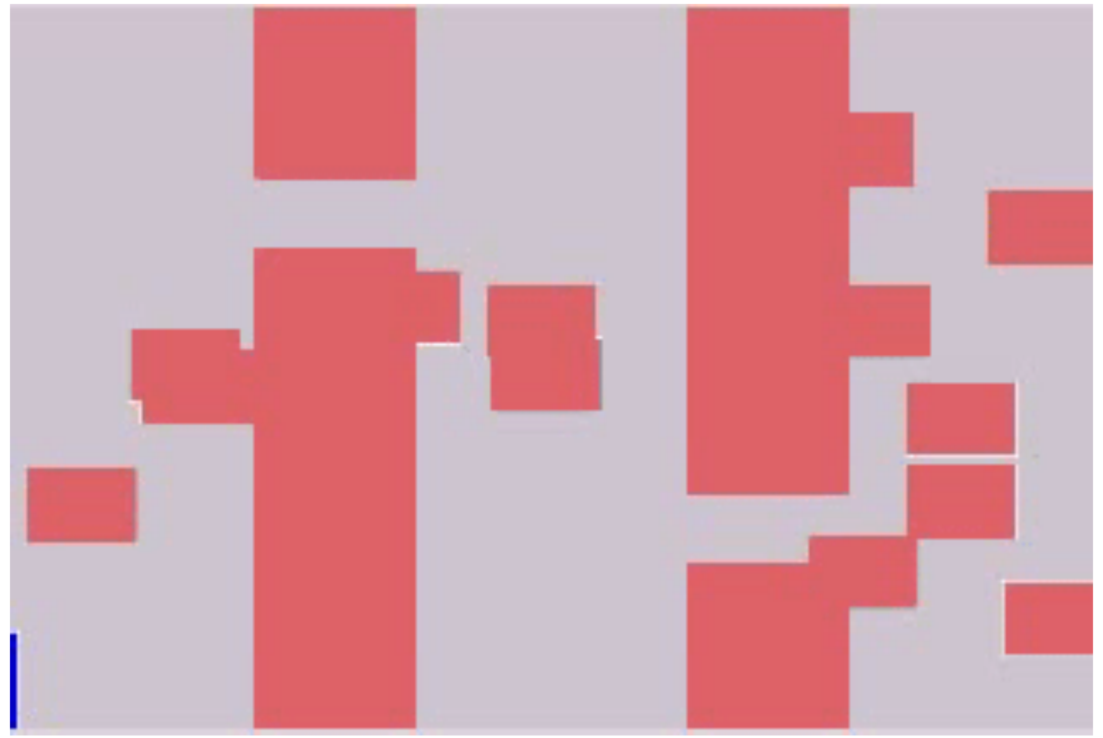


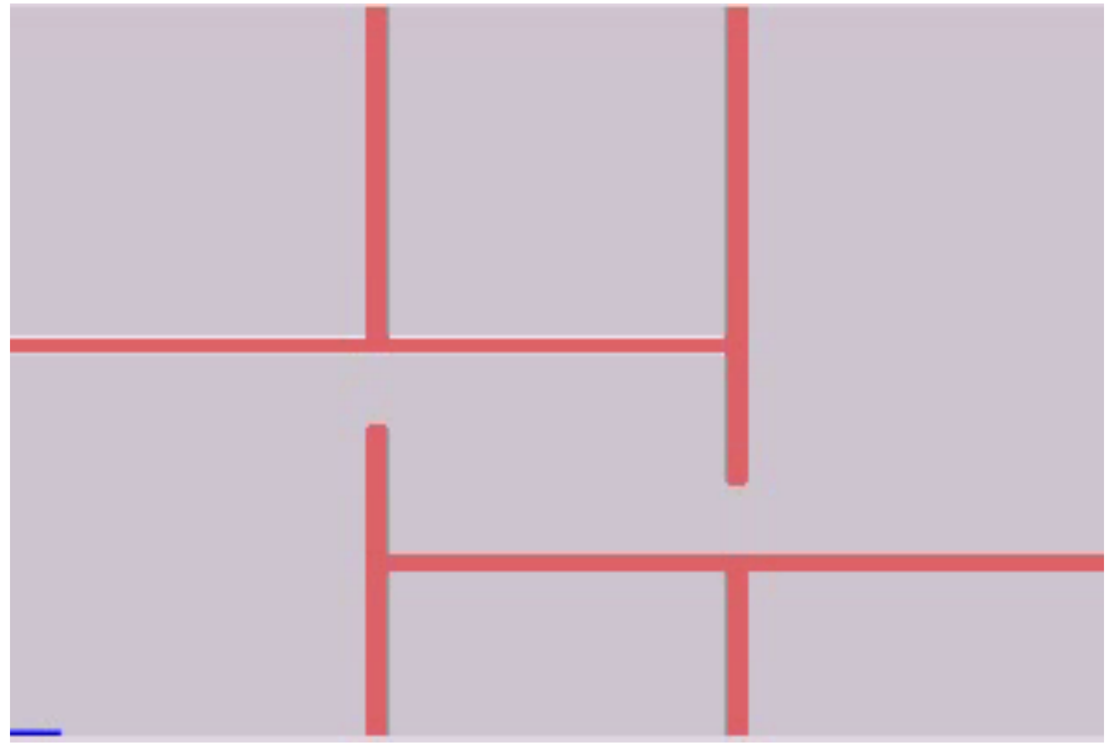
Learned
Search Heuristic

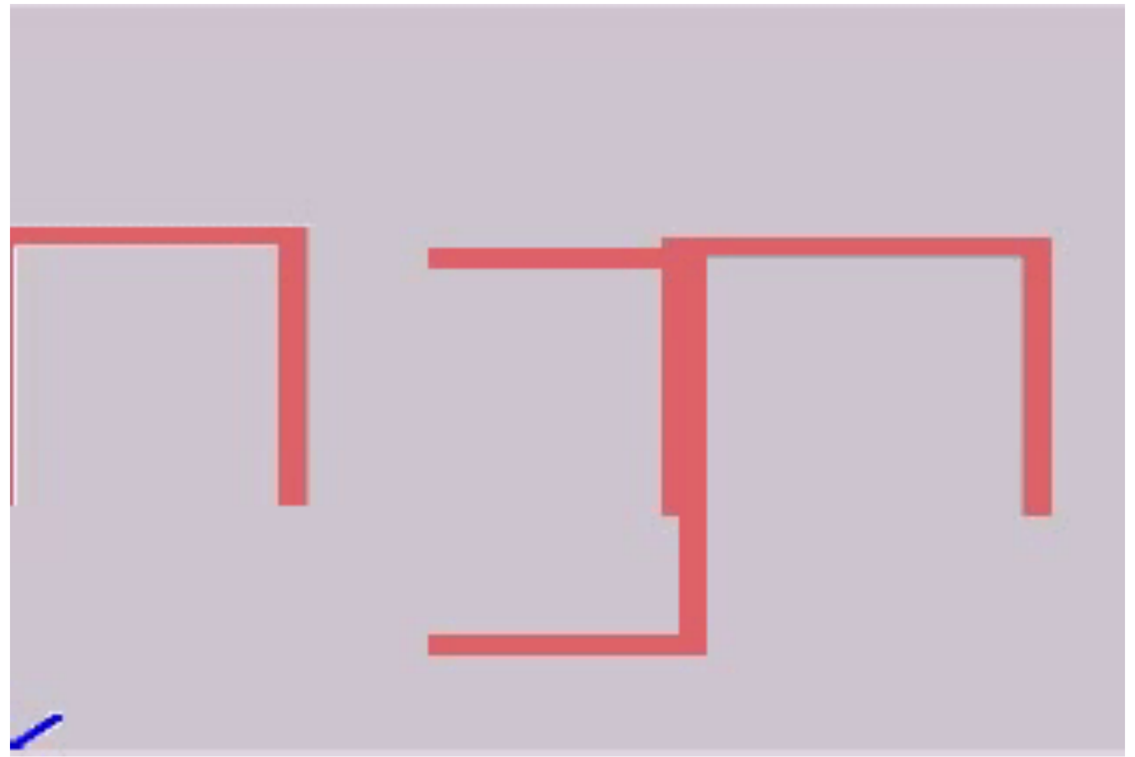
Imitate



Optimal
Value Function







[Choudhury et al. ‘2018]

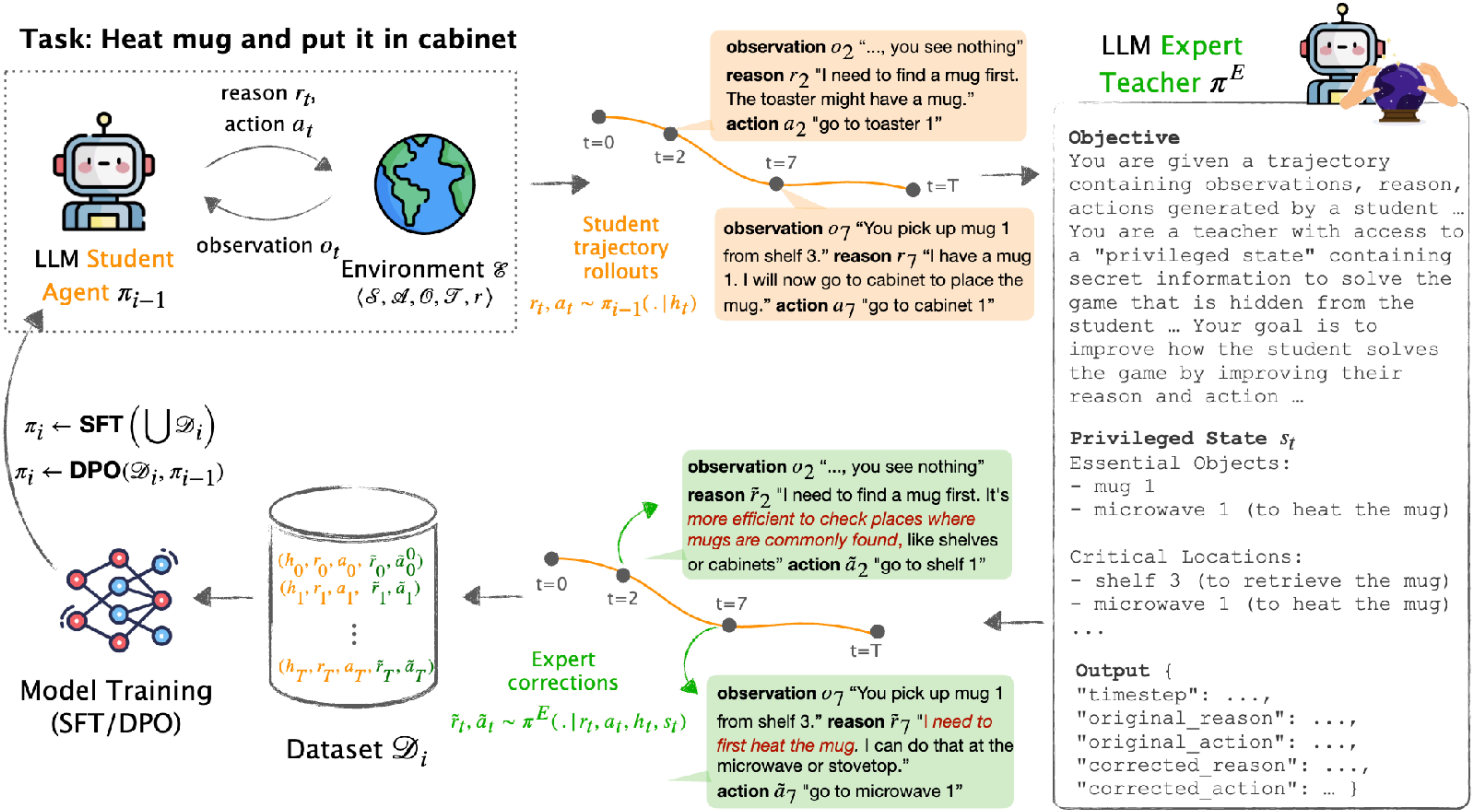
Privileged Information: **LLM Agents**

Better than Your Teacher: LLM Agents that learn from Privileged AI Feedback

Sanjiban Choudhury[1,*], **Paloma Sodhi**[2,*]
[1]Cornell University, NY, USA, [2]ASAPP Research, NY, USA
sanjibanc@cornell.edu, paloma.sodhi@gmail.com

Train weak student models (LLAMA-8B) to beat strong teachers (GPT-4!)

Privileged Information: **LLM Agents**



Today's class

- ☑ What are the challenges with sim2real?

  Case study: OpenAI Dactyl Hand

- ☑ Teacher->Student distillation

  Case study: Visual Dexterity

- ☑ Imitation Learning with Privileged Information

Counter Example to DAGGER w/ privileged information

You have a student agent in a dark room
with a door and a security lock

The passcode for the security lock is
written in a blackboard on the wall. There is also a light switch.

Teacher knows the passcode (privileged information)

What is the optimal student policy? What will DAGGER learn?