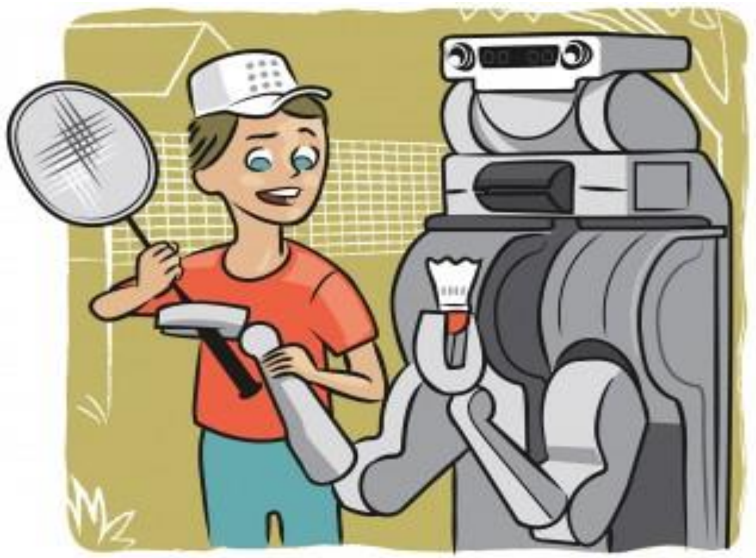
Imitation Learning and Inverse RL



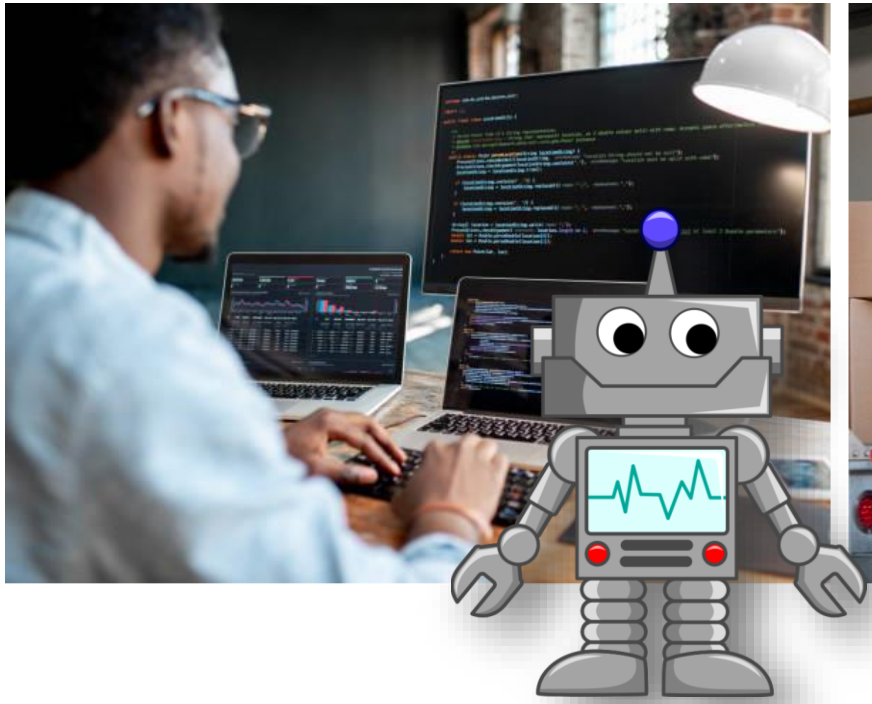
Instructor: Daniel Brown

[Some slides adapted from Sergey Levine (CS 285) and Alina Vereshchaka (CSE4/510), and Ryan Hoque]

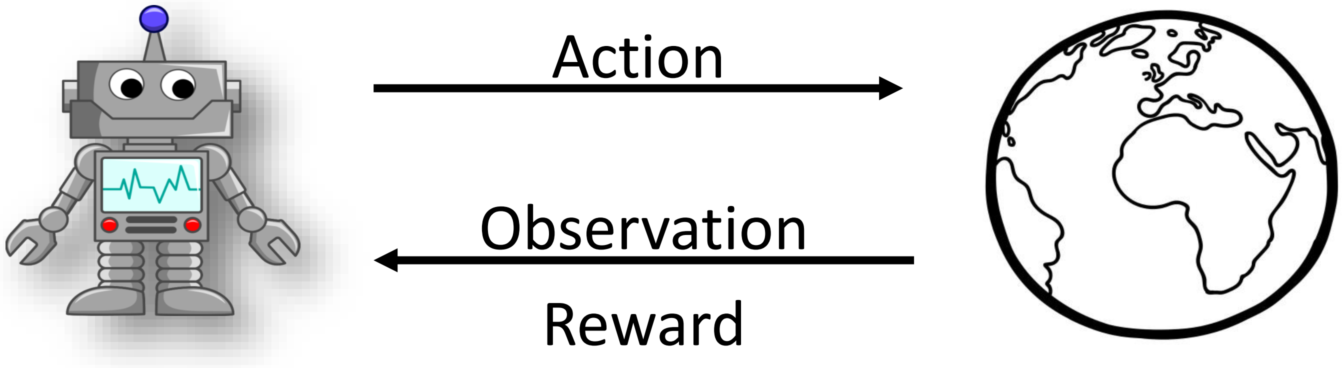
Announcements

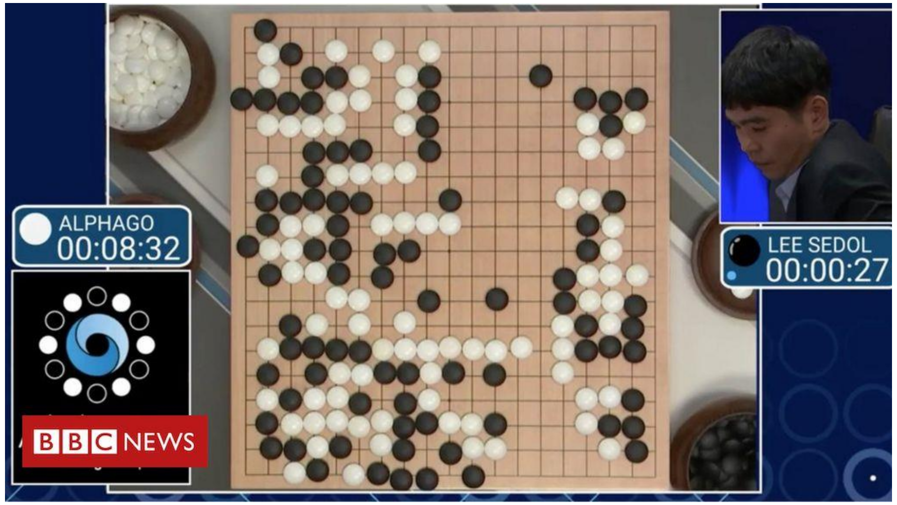
- HW 8 due tomorrow!
- Get started on Project 5.
- Check grades on gradescope and canvas to make sure they are consistent and show up on canvas. If not, visit a TA during office hours or make a post on Piazza.



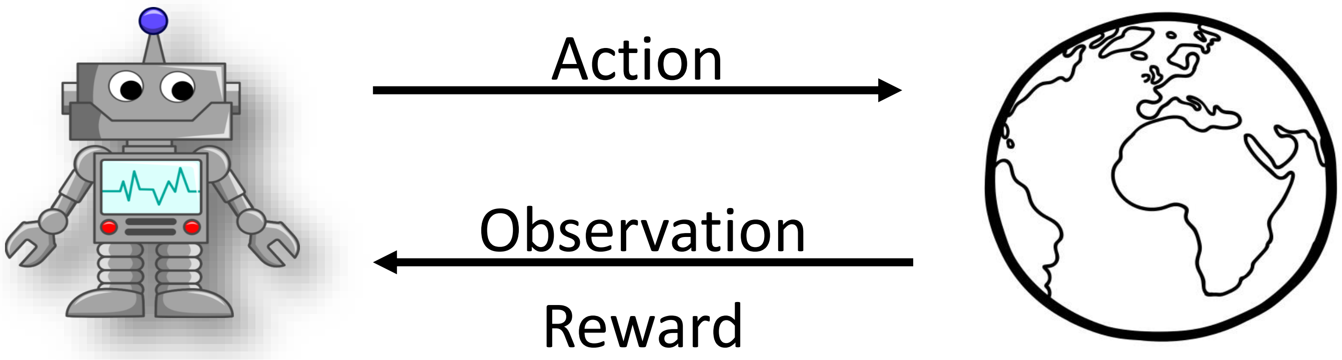



Reinforcement Learning

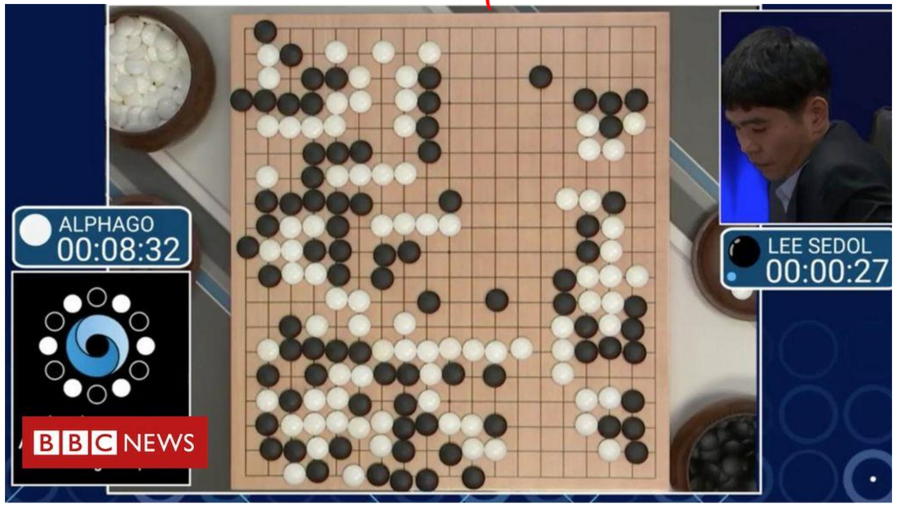




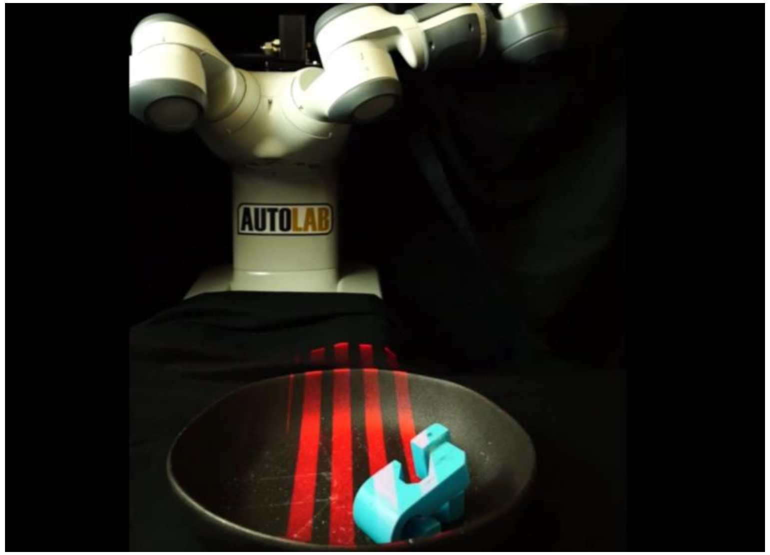
Reinforcement Learning



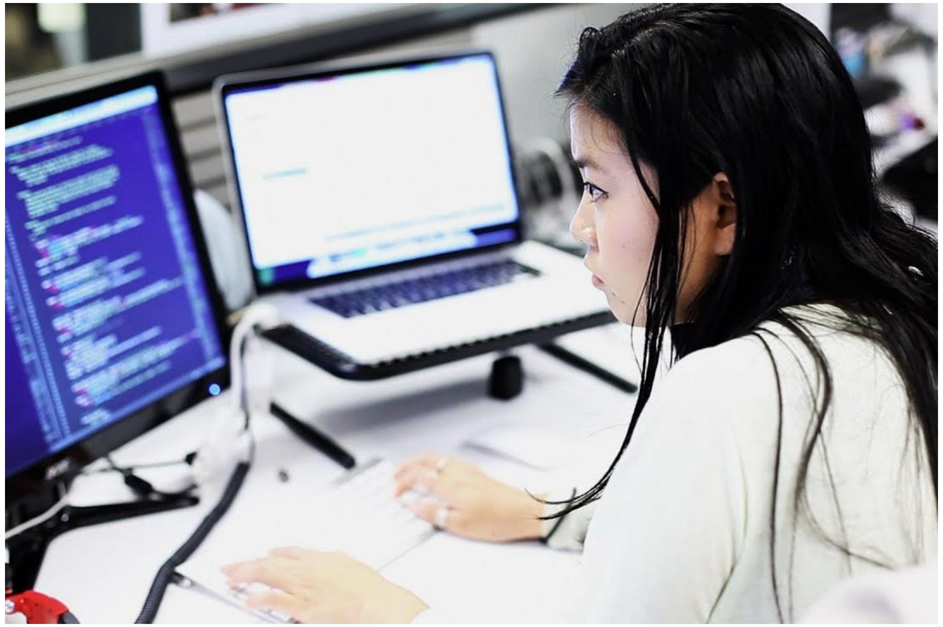
+1 / -1

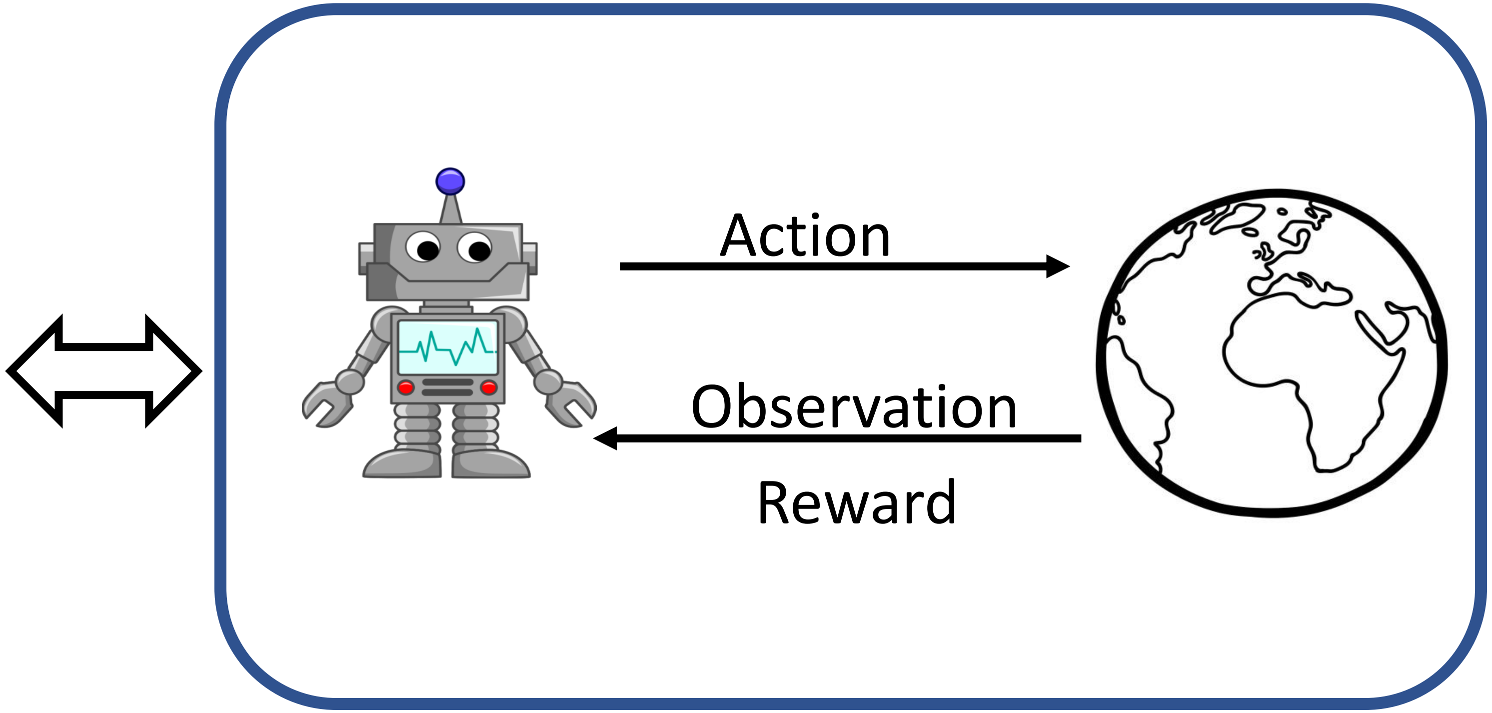


+1 / -1

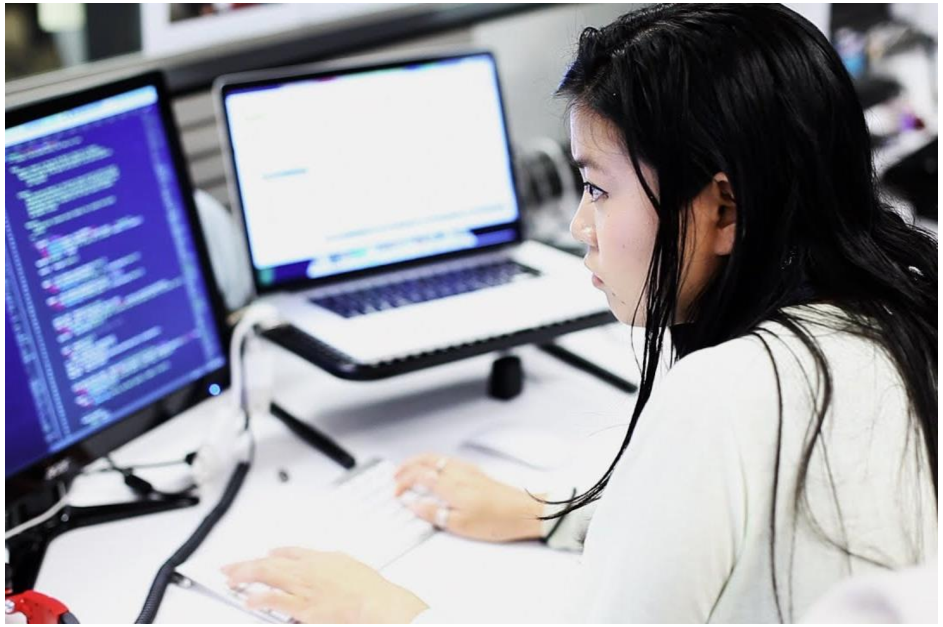


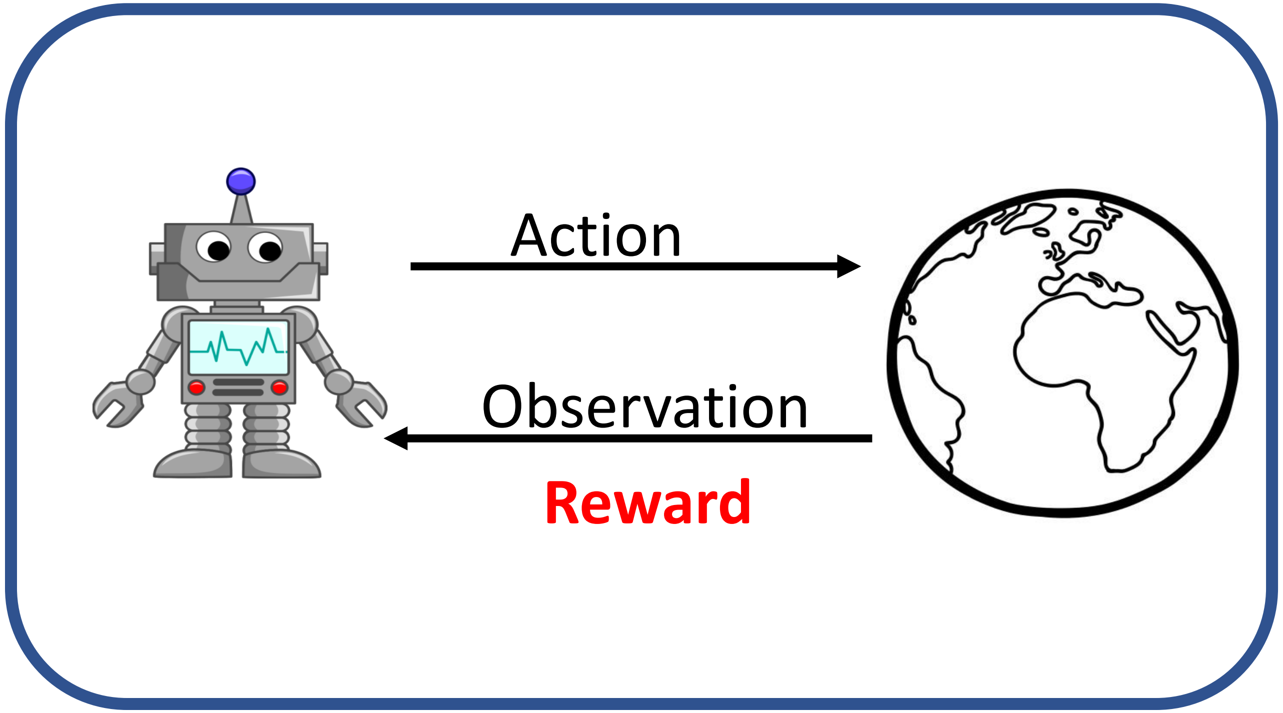
Reward engineering is hard!



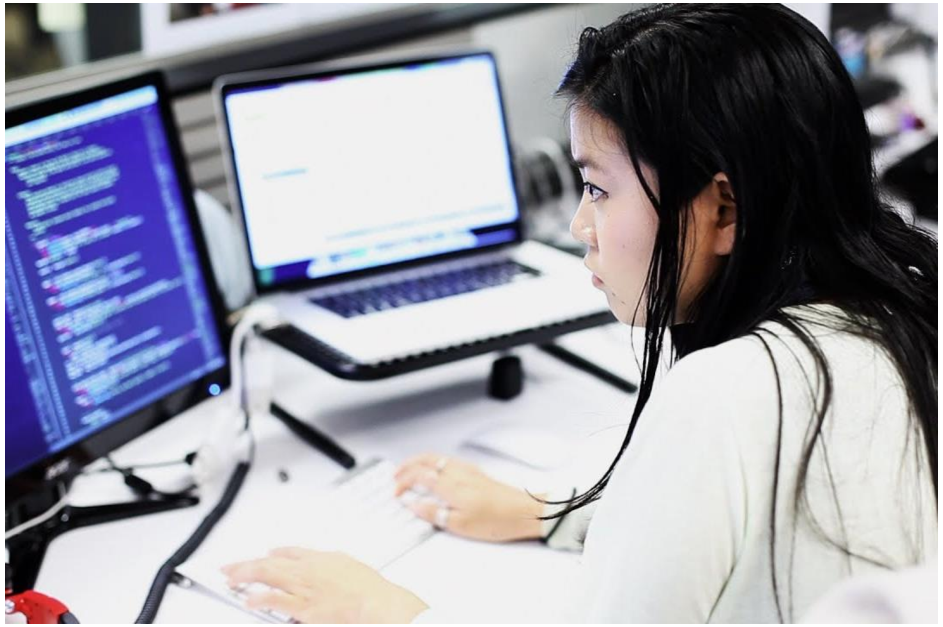


Reward engineering is hard!

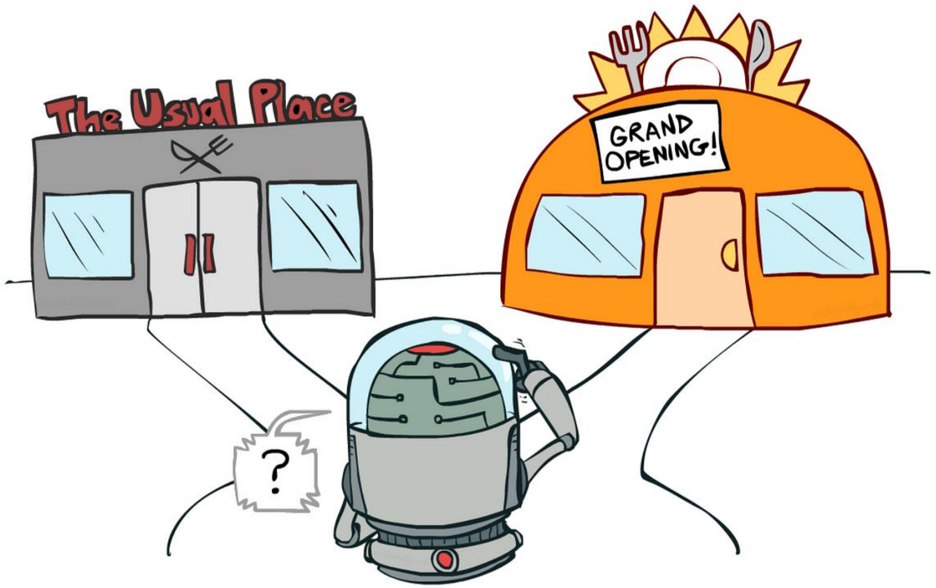




Reward engineering is hard!



Reinforcement learning is hard…even with a reward function!



Imitation Learning:

Learn a policy from examples of good behavior.

- Often showing is easier than telling.
- Alleviates problem of exploration.

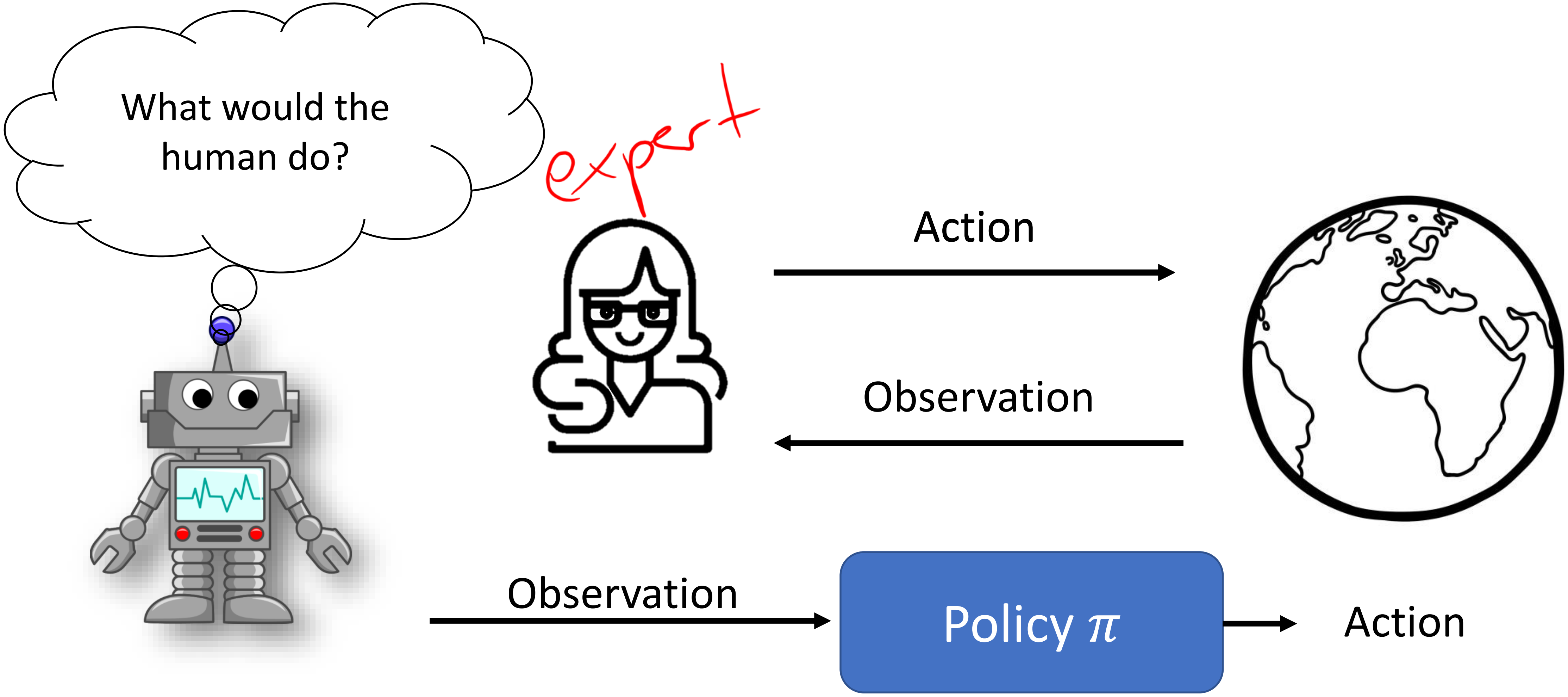

RIBA II

Learning From Demonstration (**LfD**)

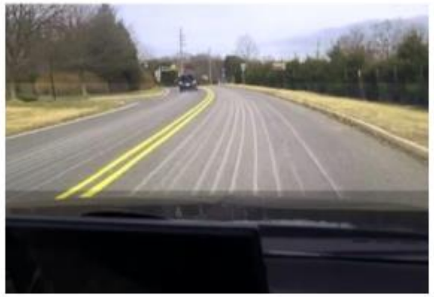
Behavioral Cloning

obs action

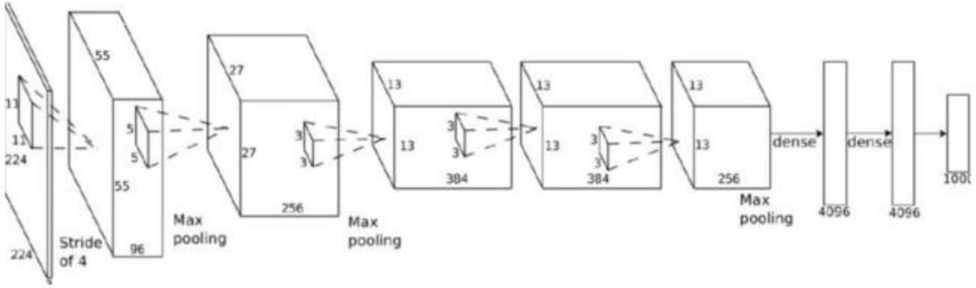
x -> y



Imitation Learning via Behavioral Cloning



$\mathbf{o}_t$



$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

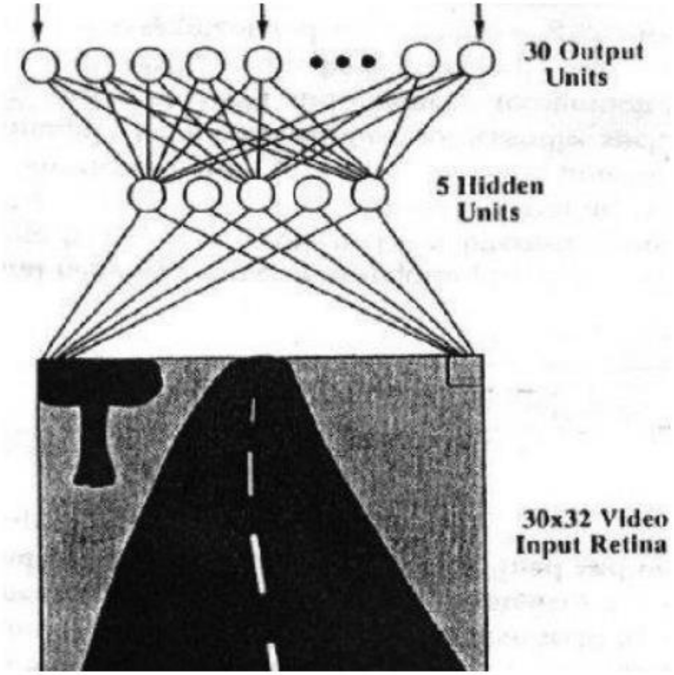
$\mathbf{a}_t$

$\mathbf{o}_t$
$\mathbf{a}_t$

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

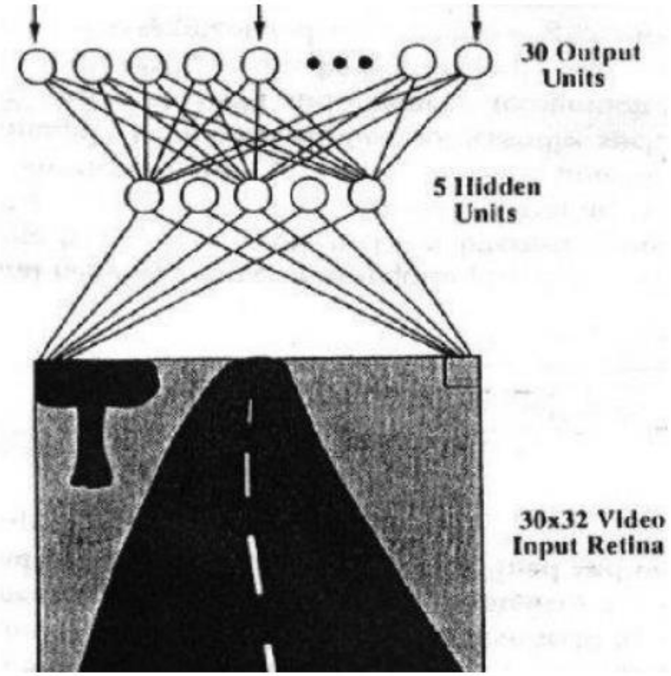
ALVINN: One of the first imitation learning systems

ALVINN: **A**utonomous **L**and **V**ehicle **I**n a **N**eural **N**etwork
1989



ALVINN: One of the first imitation learning systems

ALVINN: **A**utonomous **L**and **V**ehicle **I**n a **N**eural **N**etwork
1989



What could go wrong?

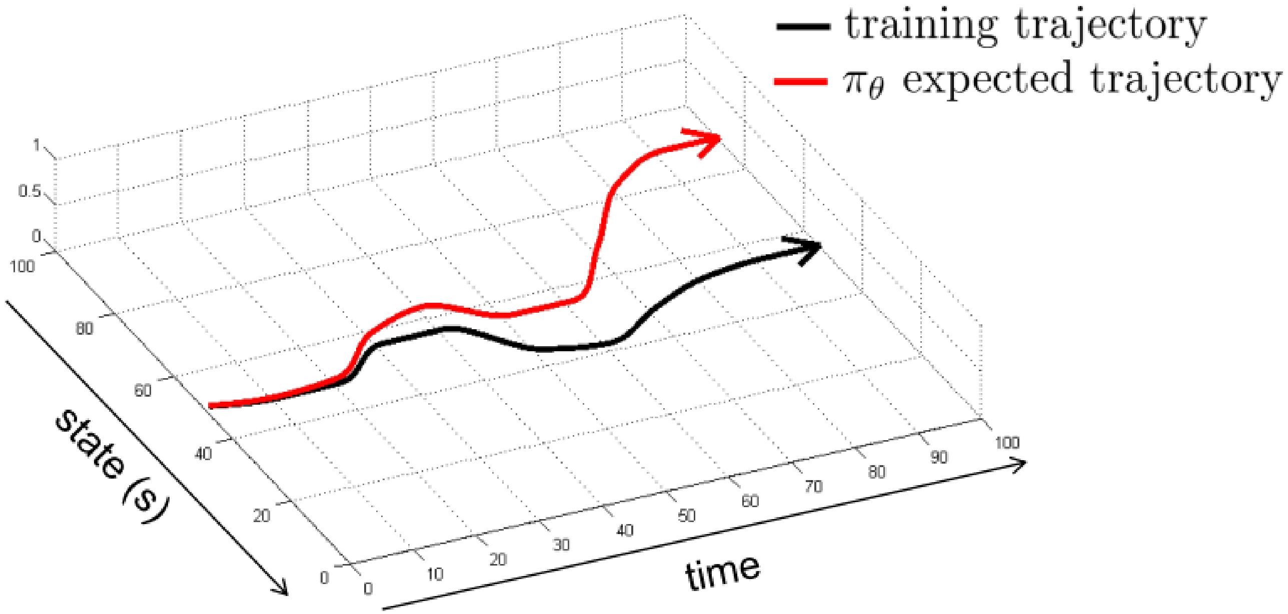
$\mathbf{o}_t$

$\mathbf{a}_t$

training data

supervised learning

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$



Distribution Shift

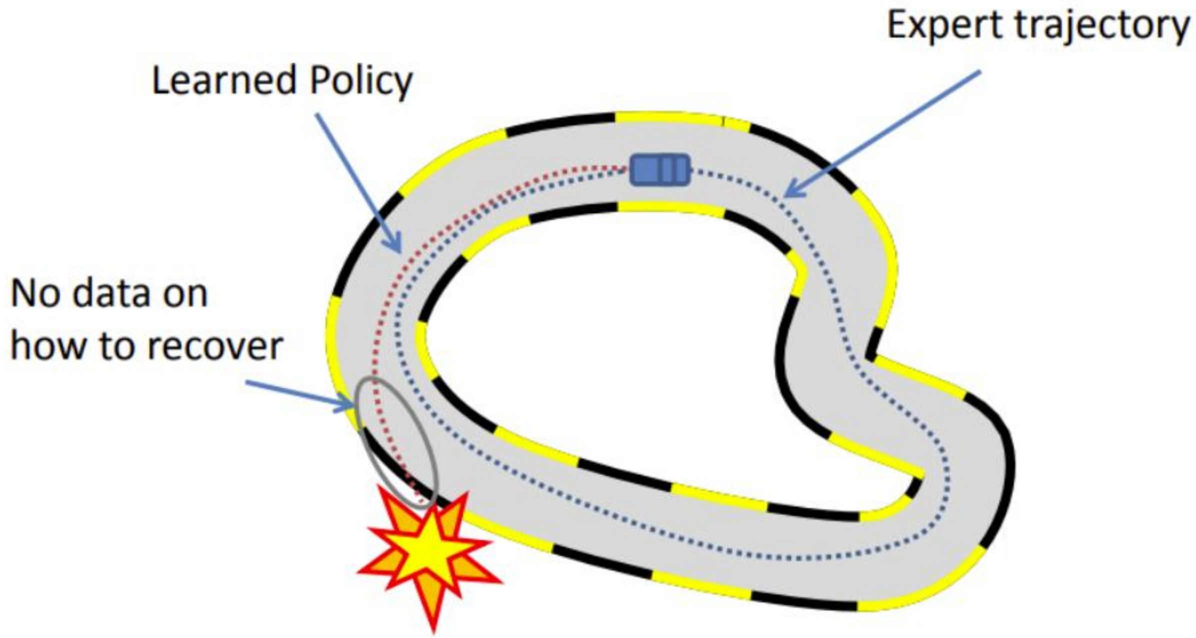
Demo learned

$$p_{\pi^*}(o_t) \neq p_{\pi_\theta}(o_t)$$

$(s, \pi^*(s))$

$\shortparallel$

(s, a)

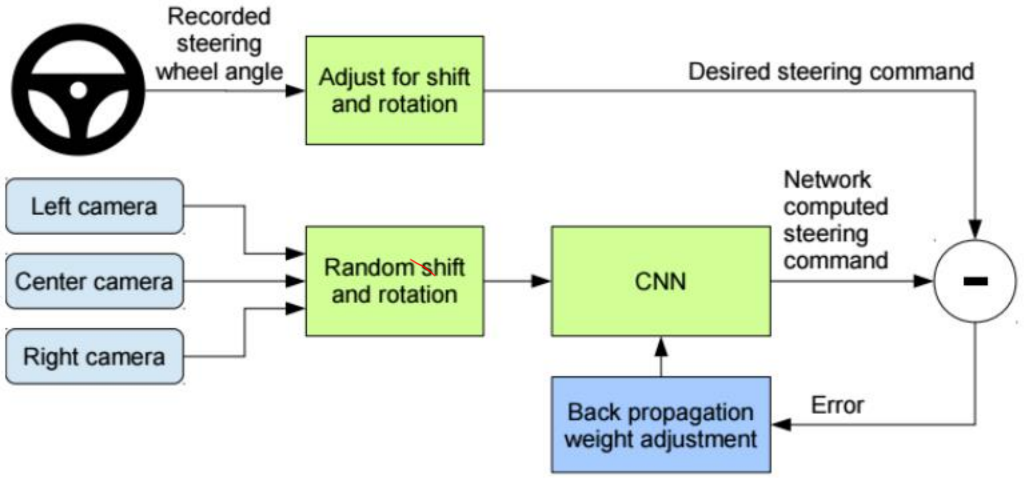


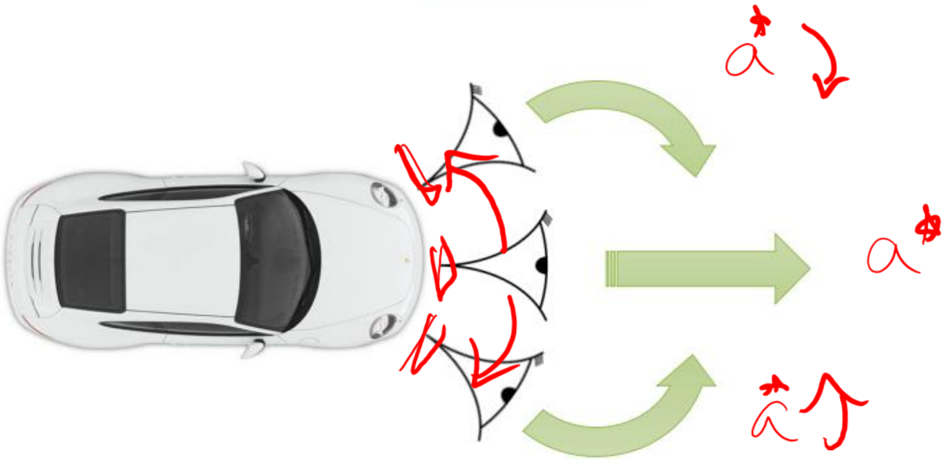
	Supervised Learning	Supervised Learning + Control
Train	$(x, y) \sim D$	$s \sim P(\cdot \mid s, \pi^*(s))$
Test	$(x, y) \sim D$	$s \sim P(\cdot \mid s, \pi_\theta(s))$

But it still can work in practice…

Bojarski et al. '16, NVIDIA

How?

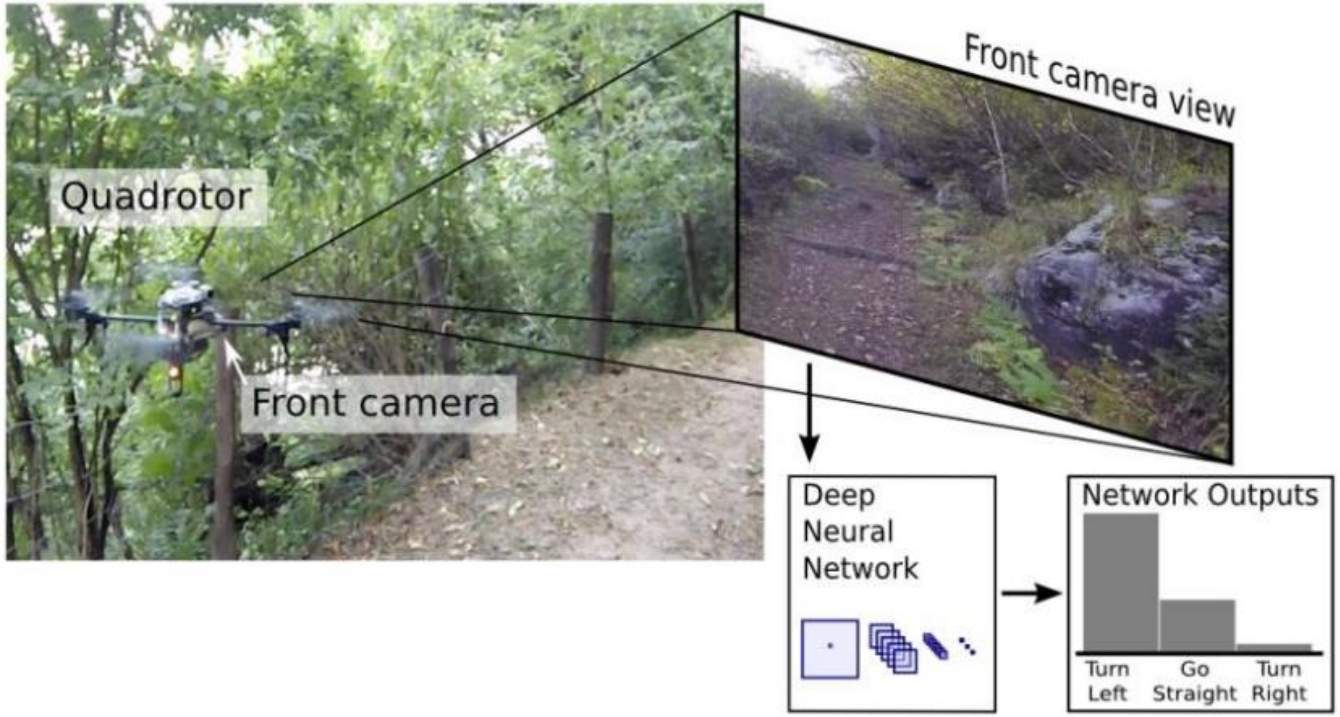




Bojarski et al. '16, NVIDIA

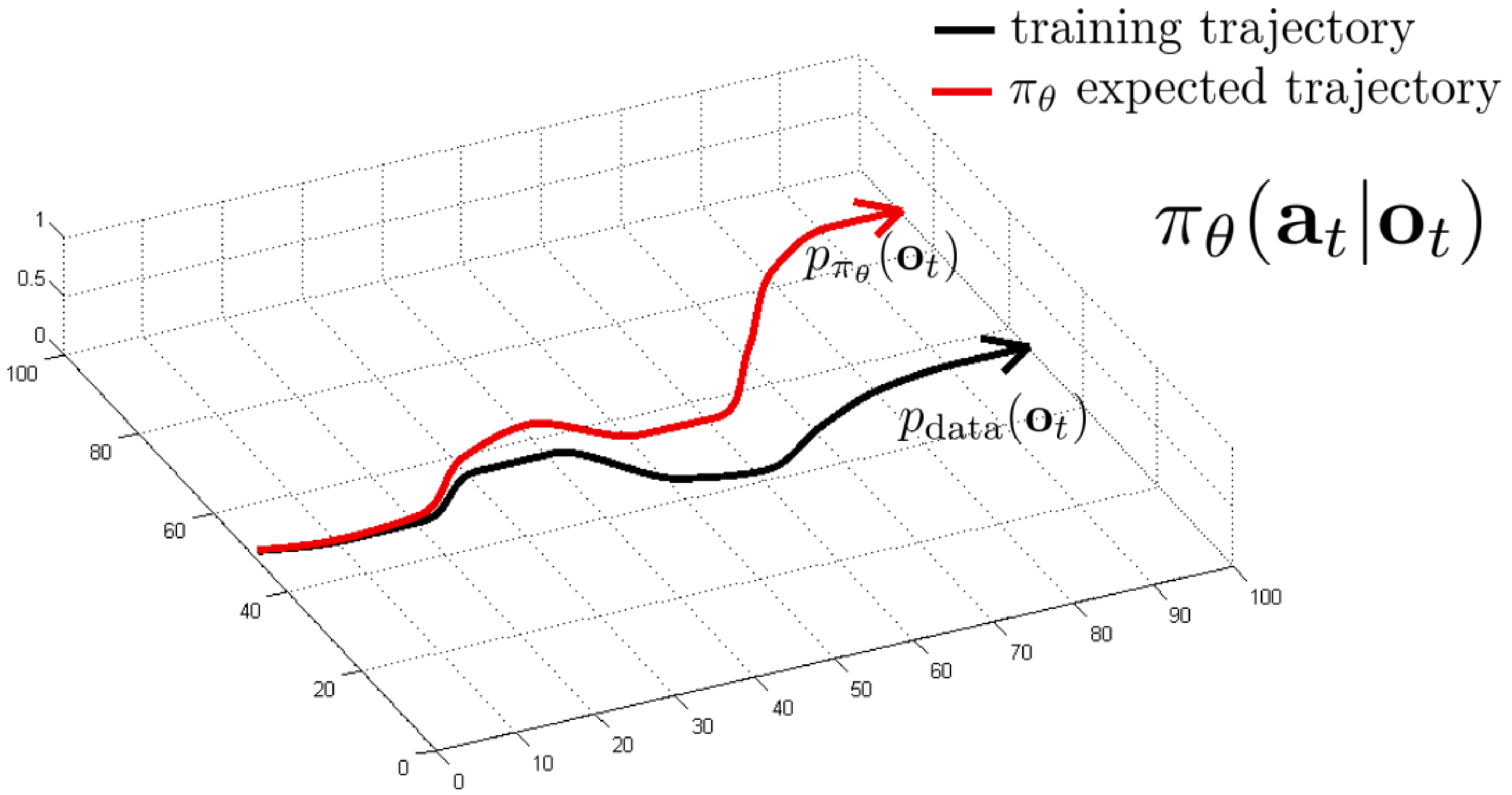
A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots

Alessandro Giusti[1], Jérôme Guzzi[1], Dan C. Cireşan[1], Fang-Lin He[1], Juan P. Rodríguez[1]
Flavio Fontana[2], Matthias Faessler[2], Christian Forster[2]
Jürgen Schmidhuber[1], Gianni Di Caro[1], Davide Scaramuzza[2], Luca M. Gambardella[1]



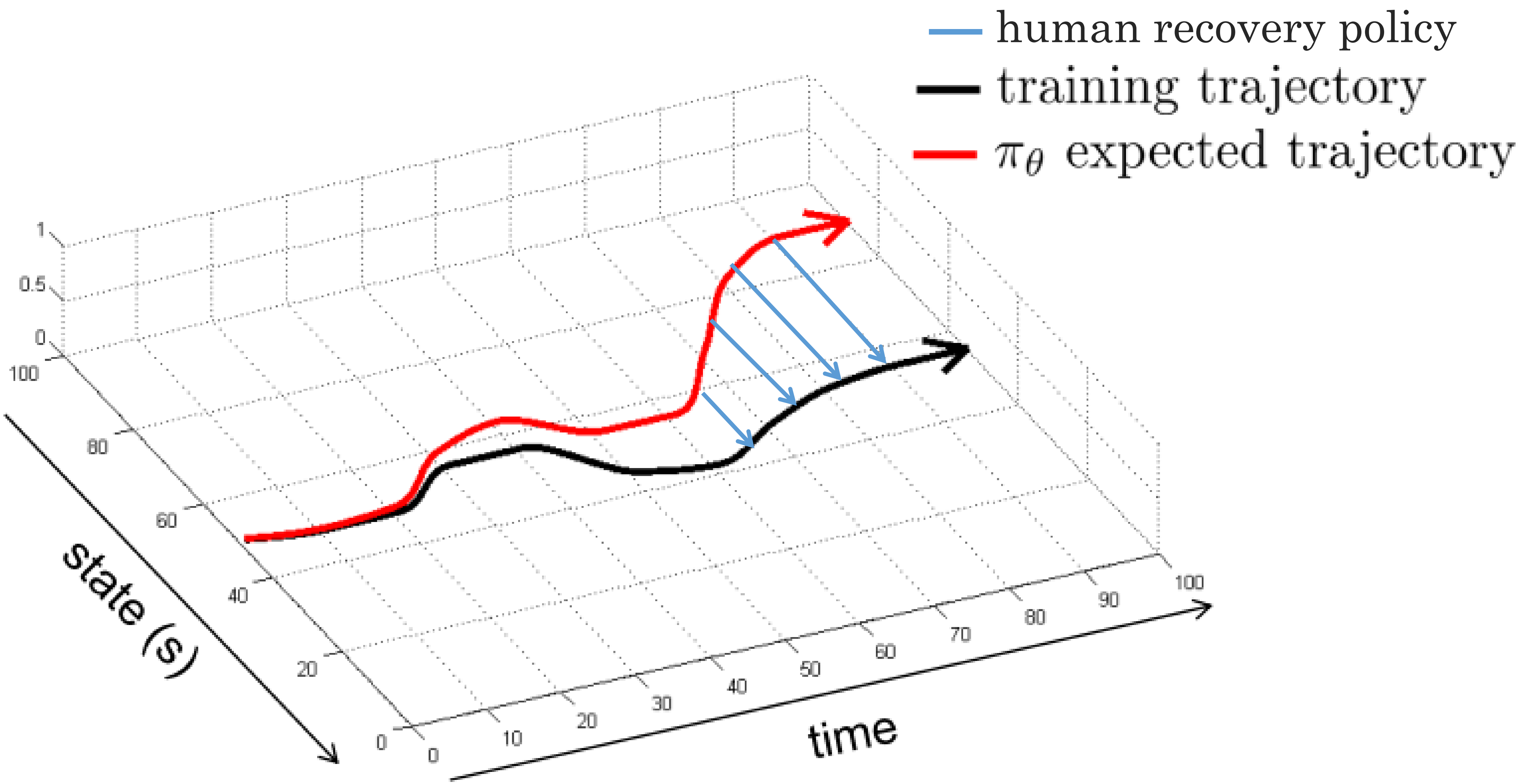

Deep Neural Network
Control Signal

Can we make it work more often?



$$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$$

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?



human recovery policy
training trajectory
π_θ expected trajectory
1
0.5
0
100
80
60
40
20
0
state (s)
0
10
20
30
40
50
60
70
80
90
100
time

DAgger

Dataset Aggregation

can we make $p_{\text{data}}(\mathbf{o}_t) = p_{\pi_\theta}(\mathbf{o}_t)$?

idea: instead of being clever about $p_{\pi_\theta}(\mathbf{o}_t)$, be clever about $p_{\text{data}}(\mathbf{o}_t)$!

DAgger: Dataset Aggregation

goal: collect training data from $p_{\pi_\theta}(\mathbf{o}_t)$ instead of $p_{\text{data}}(\mathbf{o}_t)$

how? just run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

but need labels $\mathbf{a}_t$!

1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{a}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Ross et al. '11

DAgger has very nice theoretical guarantees.

Why might it be **hard** to implement in practice?

DAgger: Dataset Aggregation

goal: collect training data from $p_{\pi_\theta}(\mathbf{o}_t)$ instead of $p_{\text{data}}(\mathbf{o}_t)$

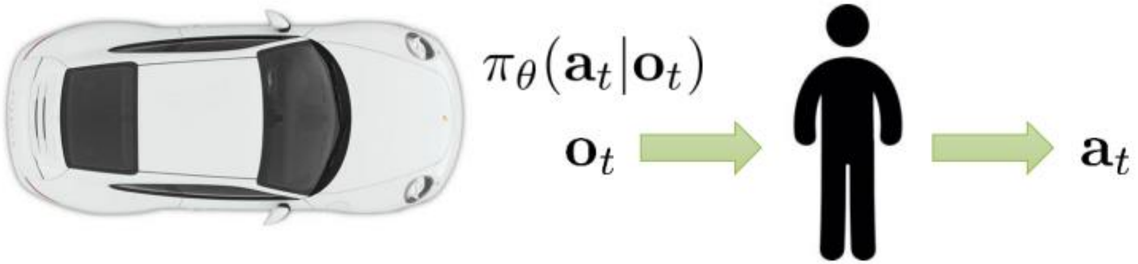
how? just run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

but need labels $\mathbf{a}_t$!

1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{a}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Ross et al. '11

1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask human to label $\mathcal{D}_\pi$ with actions $\mathbf{a}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$



DAgger has very nice theoretical guarantees.

Why might it be **easy** to implement in practice?

DAgger: Dataset **Aggregation**

goal: collect training data from $p_{\pi_\theta}(\mathbf{o}_t)$ instead of $p_{\text{data}}(\mathbf{o}_t)$

how? just run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

but need labels $\mathbf{a}_t$!

1. train $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ from human data $\mathcal{D} = \{\mathbf{o}_1, \mathbf{a}_1, \ldots, \mathbf{o}_N, \mathbf{a}_N\}$
2. run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ to get dataset $\mathcal{D}_\pi = \{\mathbf{o}_1, \ldots, \mathbf{o}_M\}$
3. Ask ~~human~~ expert/oracle to label $\mathcal{D}_\pi$ with actions $\mathbf{a}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

Ross et al. '11

Learn from an Algorithmic Supervisor!

Seita et al. 2020. “Deep Imitation Learning of Sequential Fabric Smoothing From an Algorithmic Supervisor”

But we don't always have access to an algorithmic supervisor…

Can we make DAgger more practical when dealing with real human labeling?

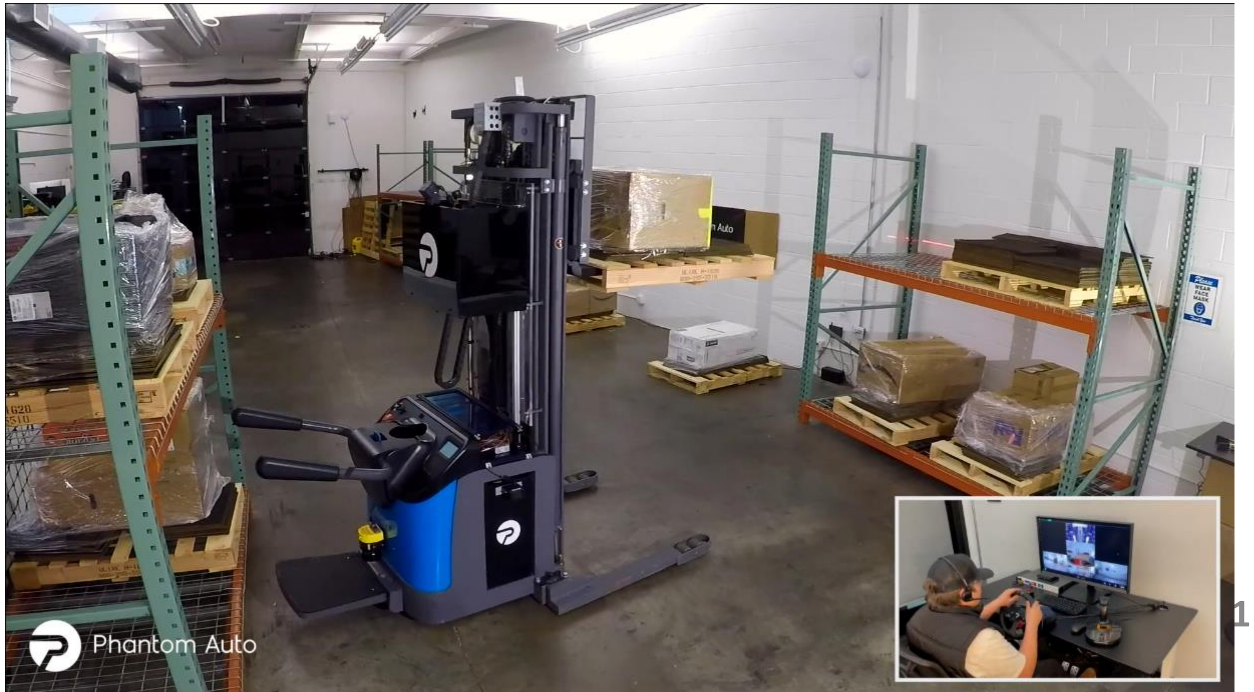
PLUS ONE ROBOTICS

WAYMO

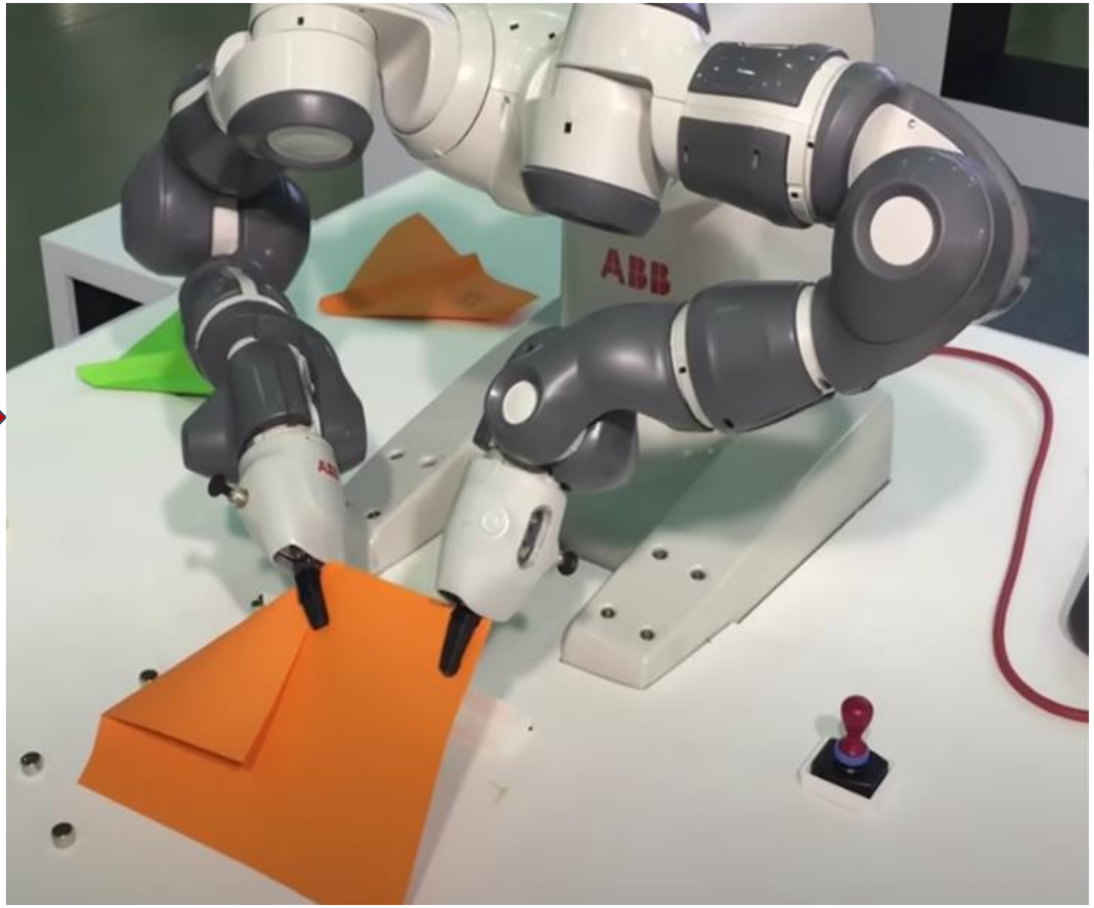
ZOOX

nimble

Phantom Auto



Interactive IL

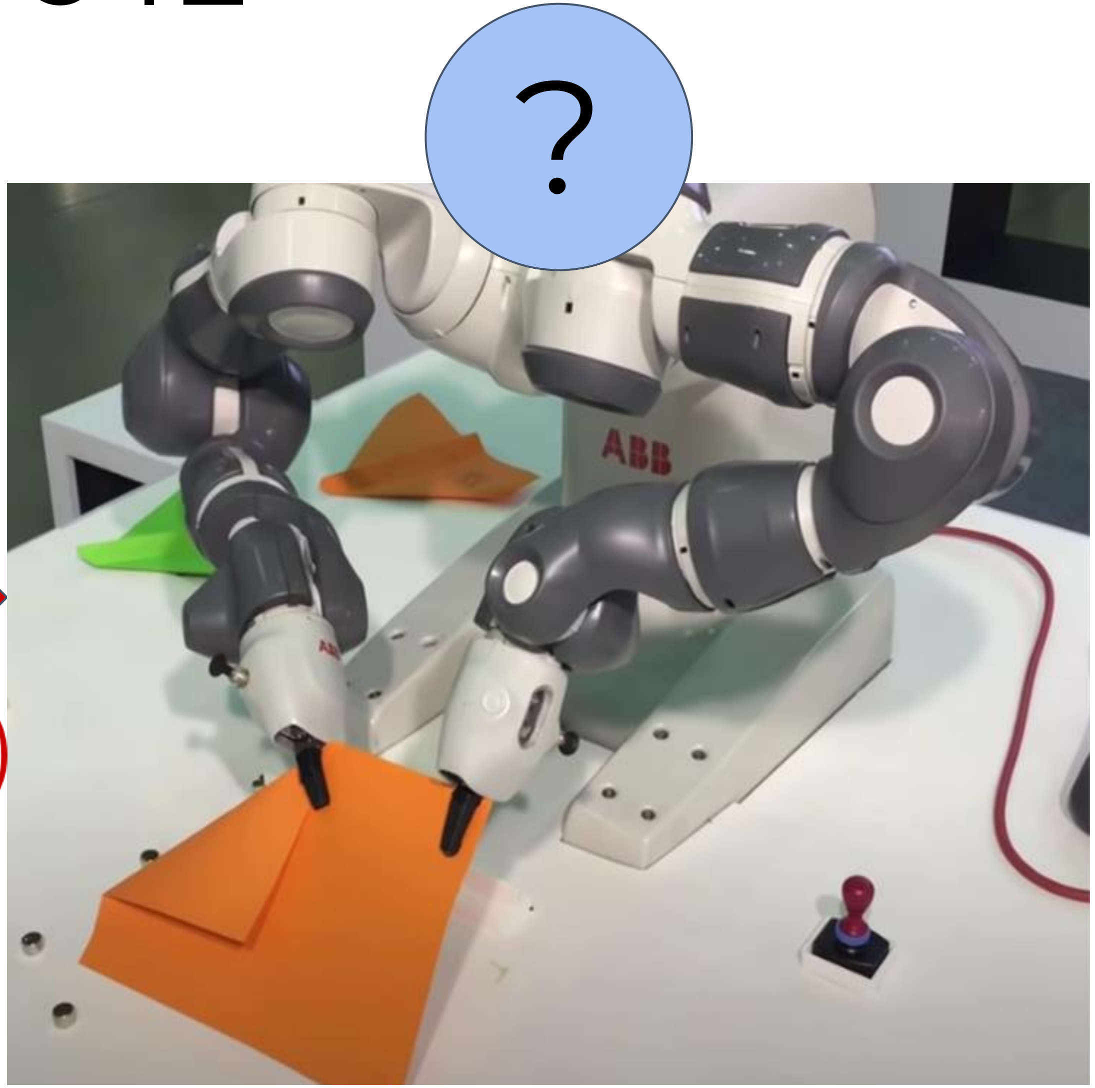


Interactive IL

$\pi_H(s)$

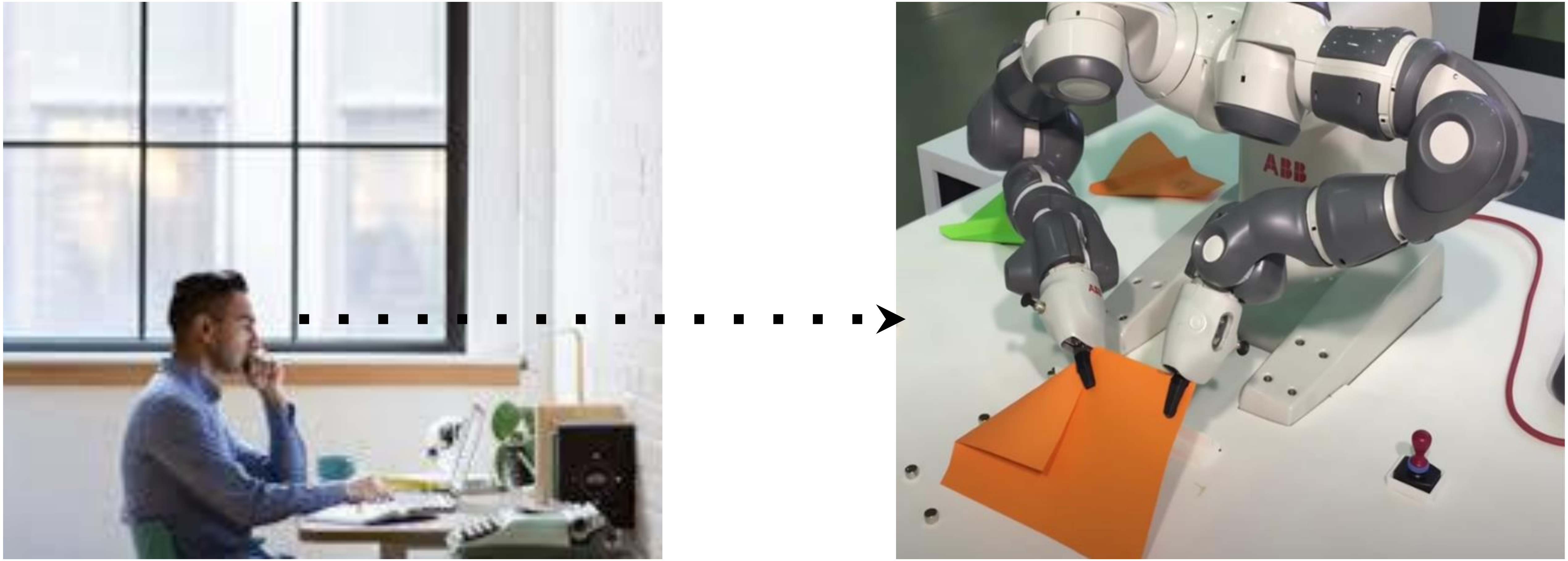
$\pi_{\text{meta}}(s)$

???



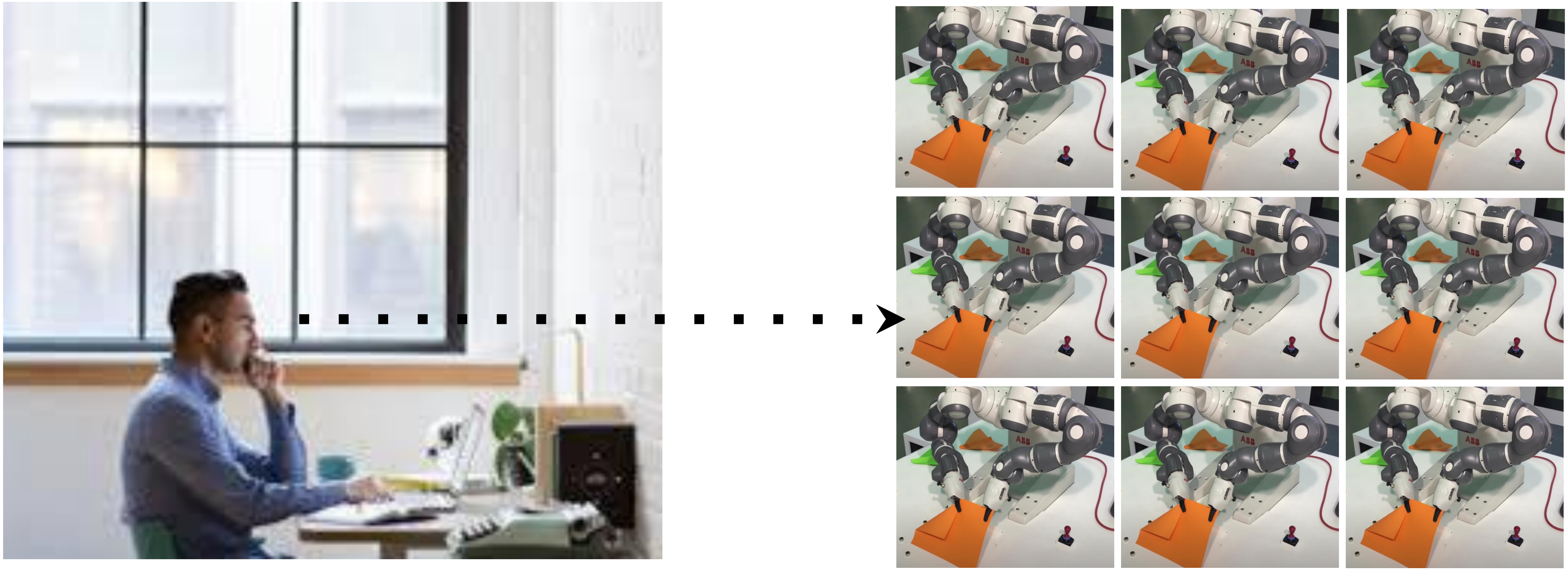
$\pi_R(s)$

Human-Gated Interactive IL



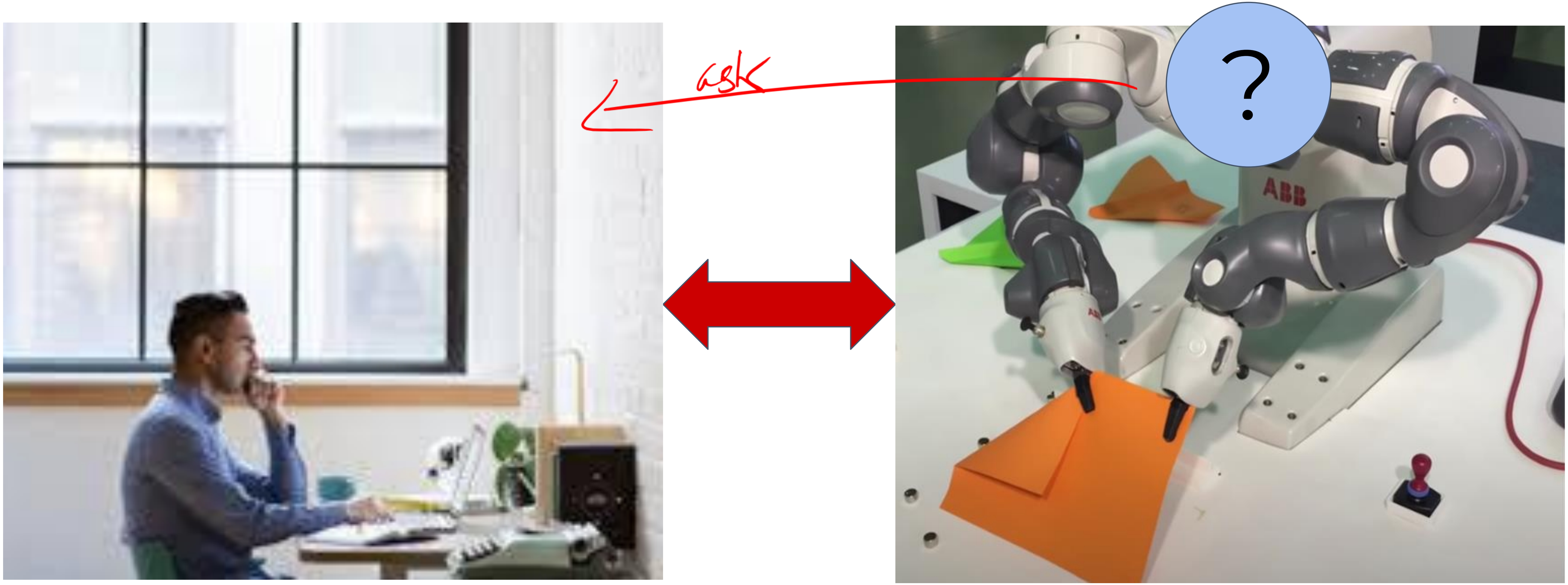
[3] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. HG-DAgger: Interactive Imitation Learning with Human Experts. ICRA 2019.

Human-Gated Interactive IL



[3] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. HG-DAgger: Interactive Imitation Learning with Human Experts. ICRA 2019.

Robot-Gated Interactive IL



[4] J. Zhang, K. Cho. Query-Efficient Imitation Learning for End-to-End Autonomous Driving. AAAI 2017.
[5] K. Menda, K. Driggs-Campbell, M. Kochenderfer. EnsembleDAgger: A Bayesian Approach to Safe Imitation Learning. IROS 2019.

When should a robot ask for help?

Novel (and risky)

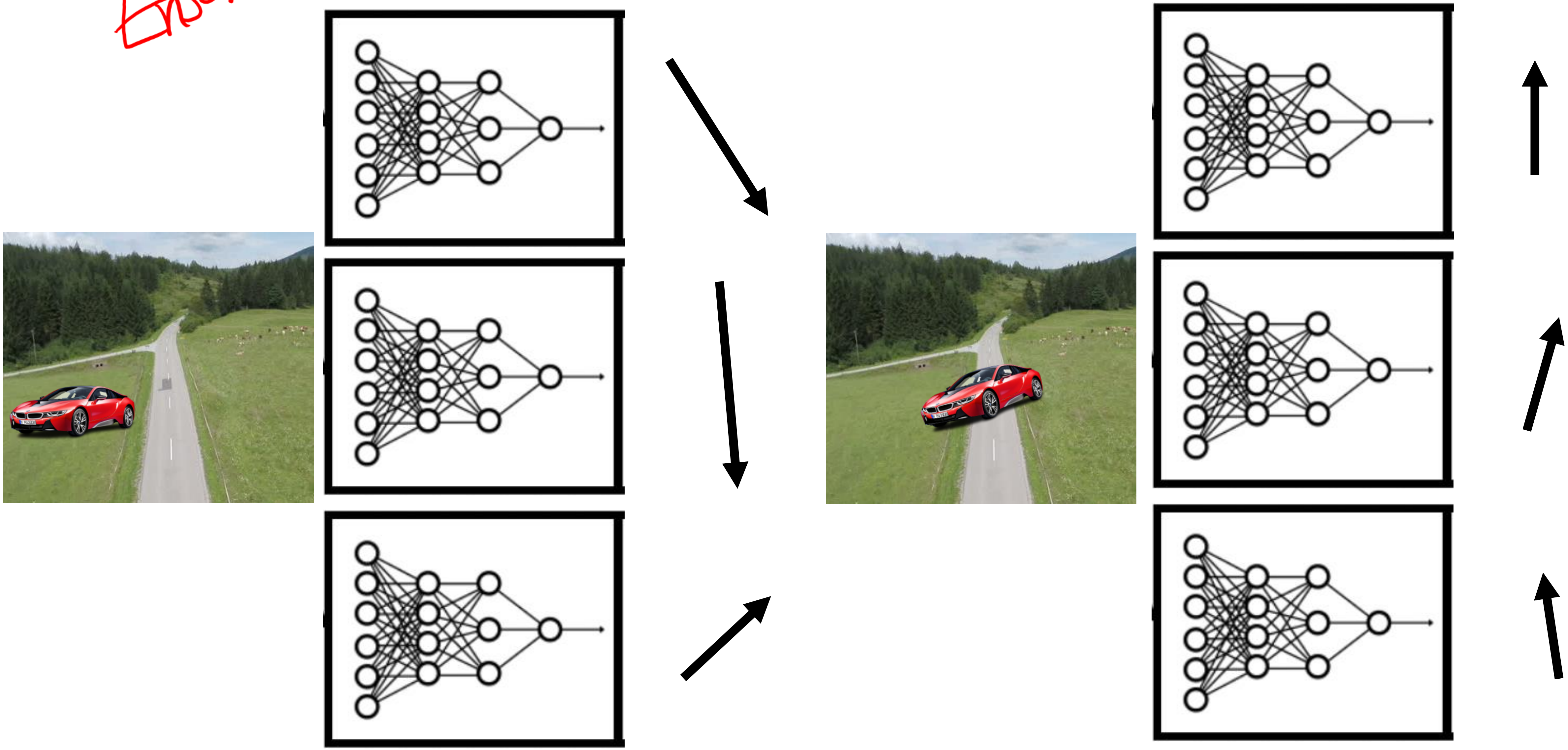
When should a robot ask for help?

Novel (and risky)

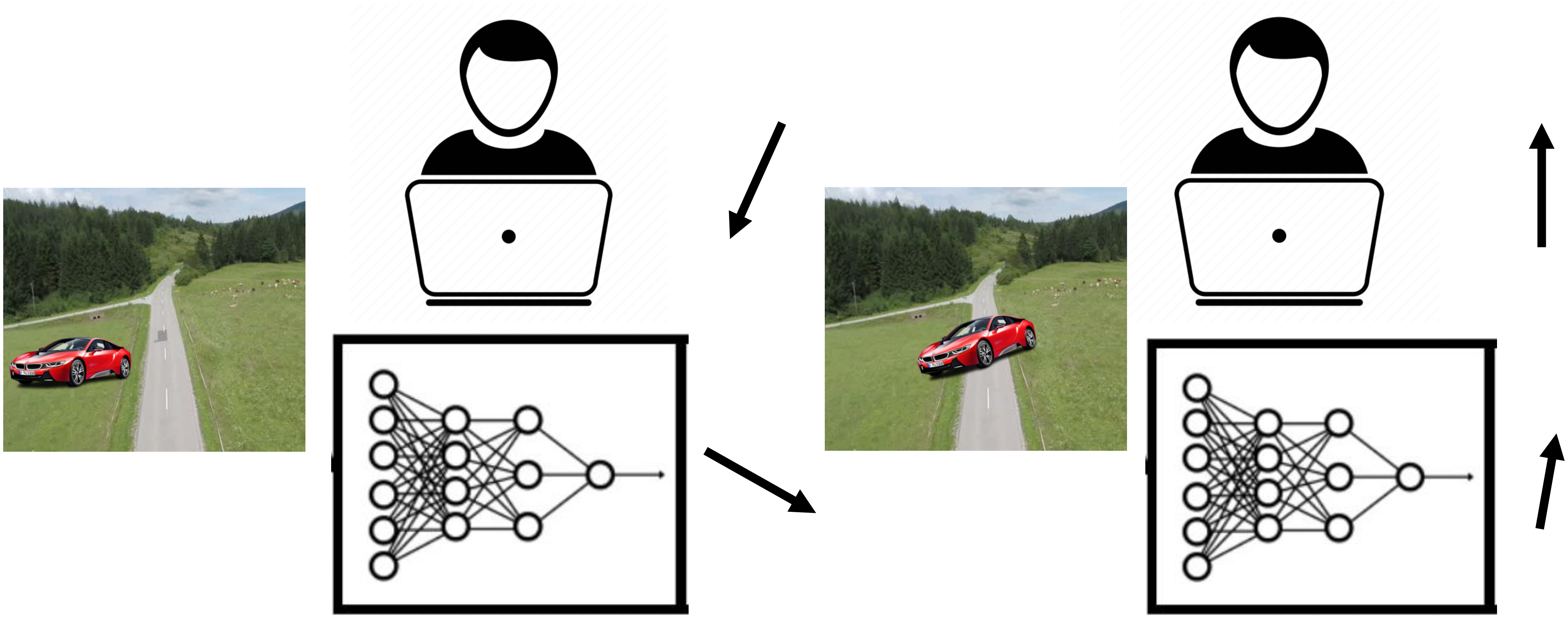
Risky (but not novel)

Novelty Estimation

Ensemble



Novelty Estimation: Supervisor Mode



Risk Estimation

$$Q_{\mathcal{G}}^{\pi_r}(s_t, a_t) = \mathbb{E}_{\pi_r}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} \mathbb{1}_{\mathcal{G}}(s'_t) | s_t, a_t\right]$$

Risk Estimation

$$Q_{\mathcal{G}}^{\pi_r}(s_t, a_t) = \mathbb{E}_{\pi_r}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} \mathbb{1}_{\mathcal{G}}(s'_t) | s_t, a_t\right]$$

$$\text{Risk}^{\pi_r}(s, a) = 1 - \hat{Q}_{\phi,\mathcal{G}}^{\pi_r}(s, a)$$

Risk Estimation

$$Q^{\pi_r}_{\mathcal{G}}(s_t, a_t) = \mathbb{E}_{\pi_r}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} \mathbb{1}_{\mathcal{G}}(s'_t) | s_t, a_t\right]$$

$$\textbf{Risk}^{\pi_r}(s, a) = 1 - \hat{Q}^{\pi_r}_{\phi,\mathcal{G}}(s, a)$$

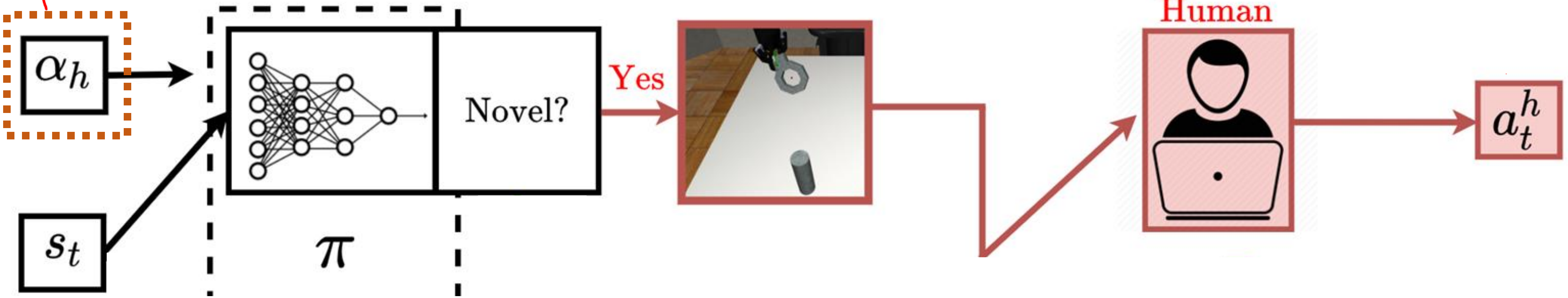
TD-Loss

$$J^{Q}_{\mathcal{G}}(s_t, a_t, s_{t+1}; \phi) =$$
$$\frac{1}{2}\left(\hat{Q}^{\pi_r}_{\phi,\mathcal{G}}(s_t, a_t) - (\mathbb{1}_{\mathcal{G}}(s_t) + (1 - \mathbb{1}_{\mathcal{G}}(s_t))\gamma \hat{Q}^{\pi_r}_{\phi,\mathcal{G}}(s_{t+1}, \pi_r(s_{t+1})))\right)^2$$

ThriftyDAgger

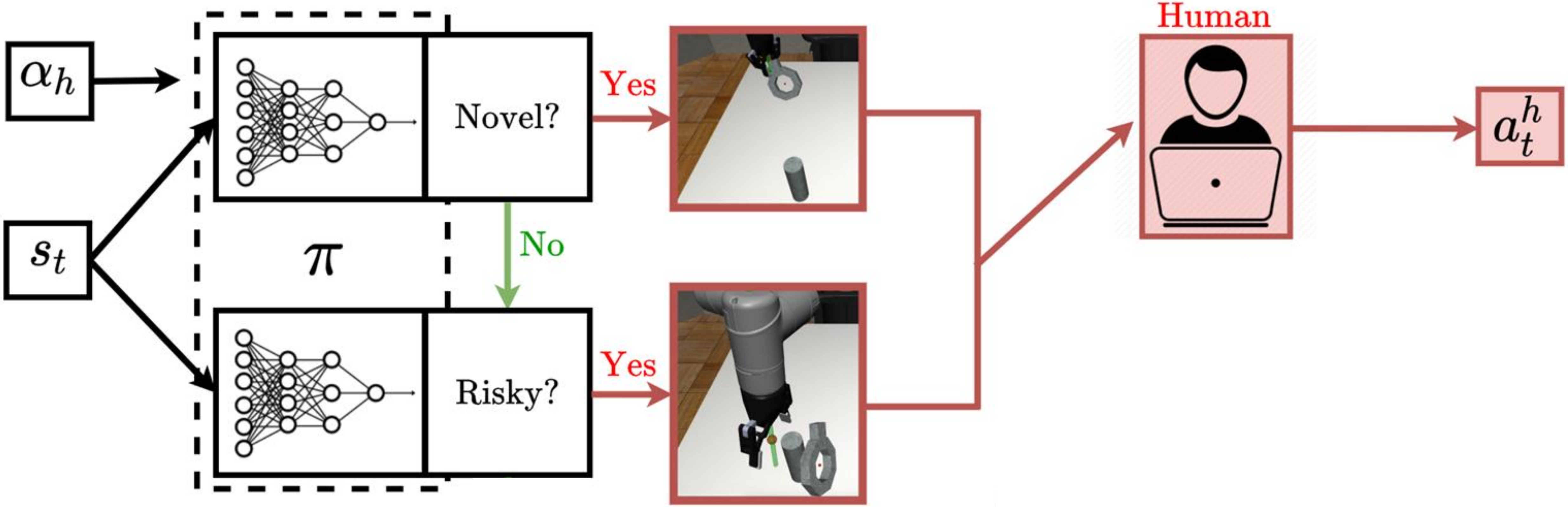
Target percent of time human wants to give interventions.

percent time intervene



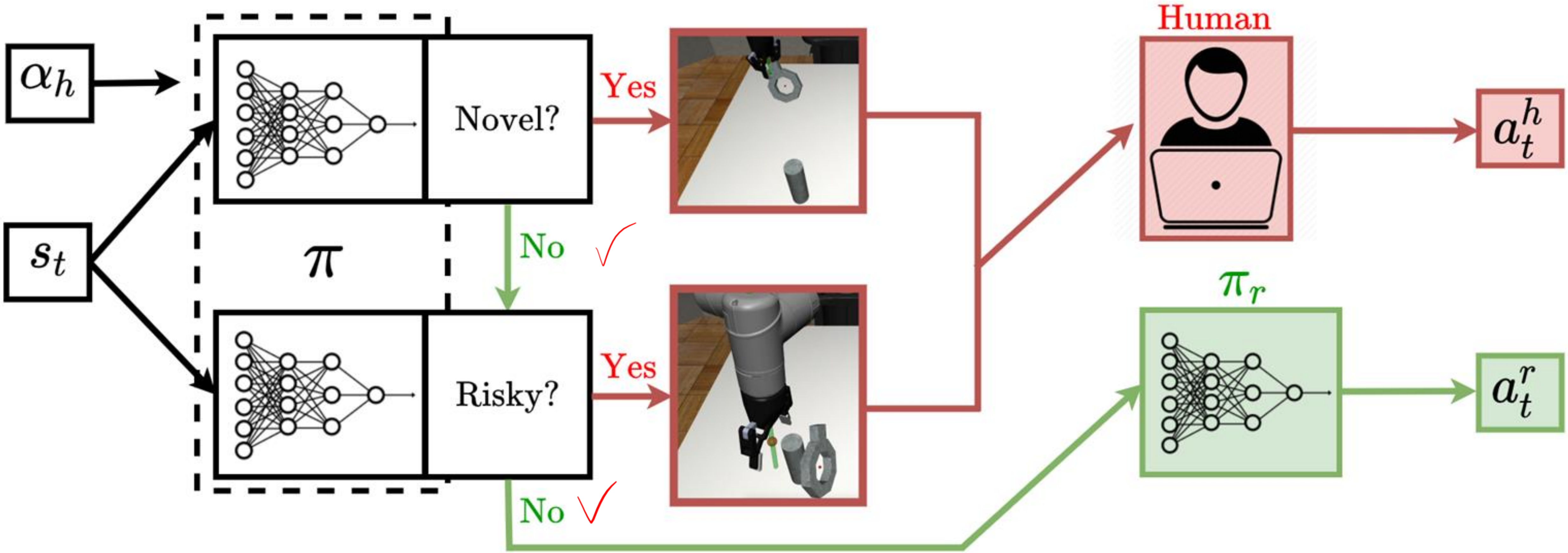
Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

ThriftyDAgger

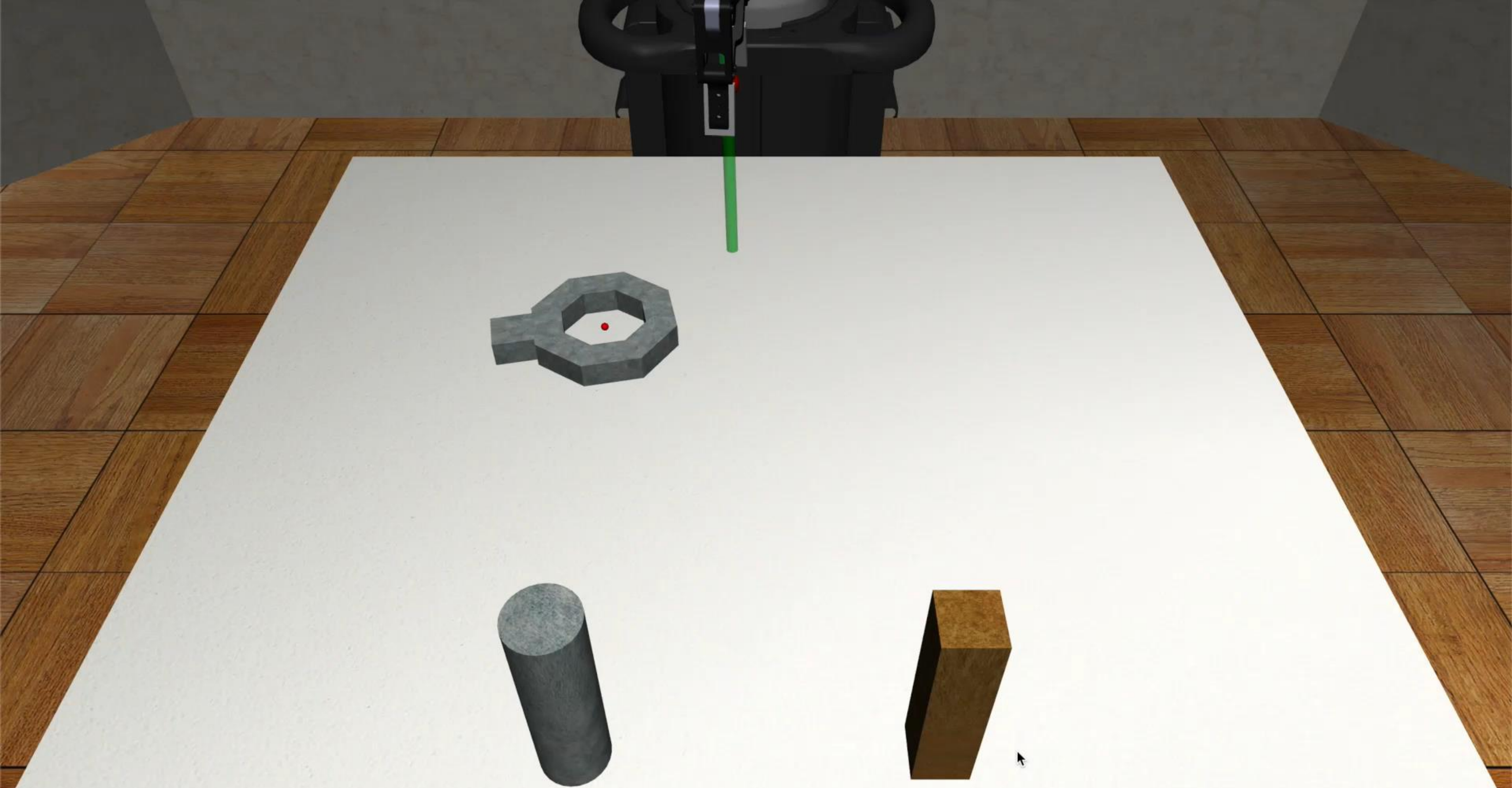


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

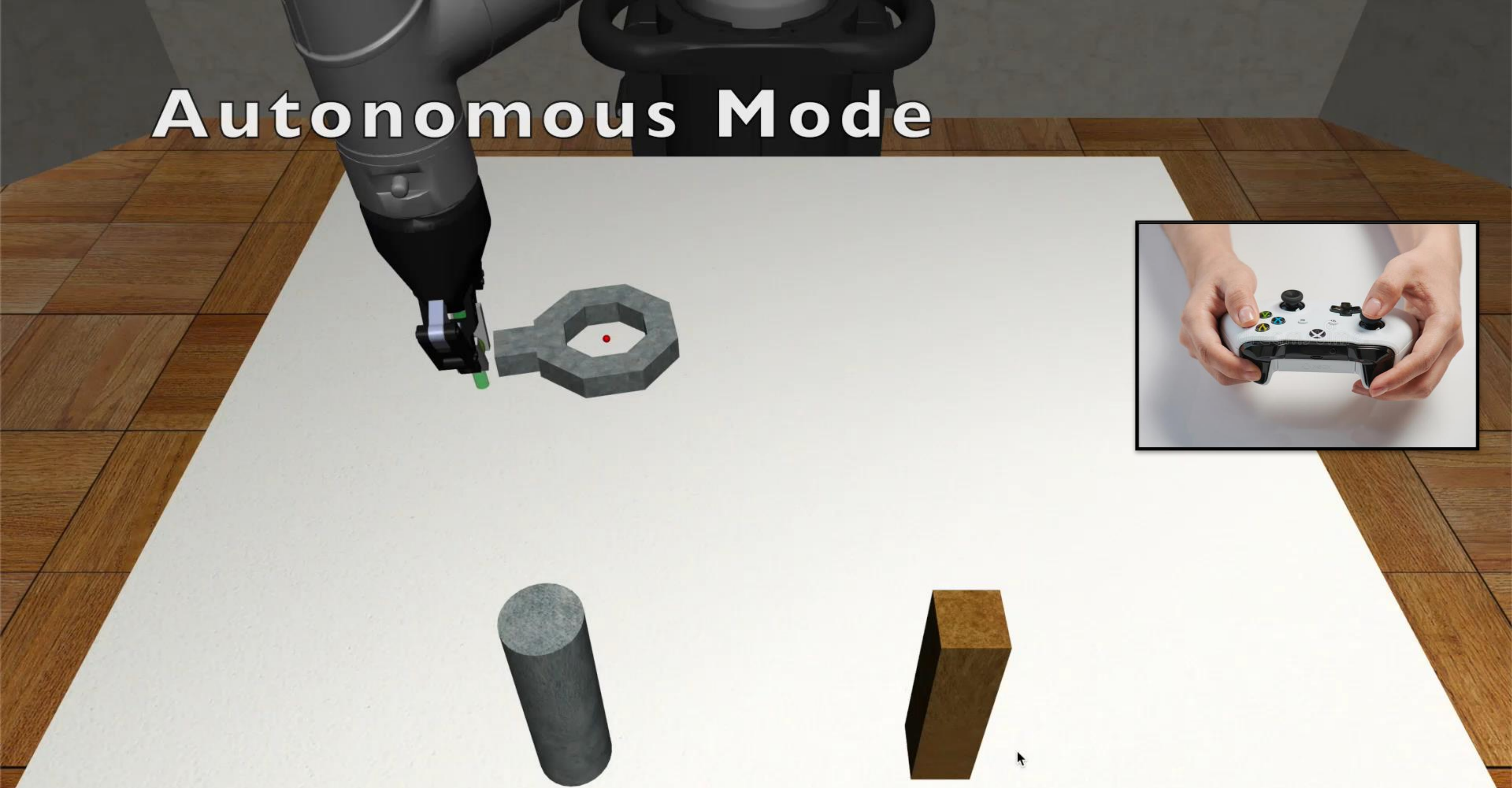
ThriftyDAgger



Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

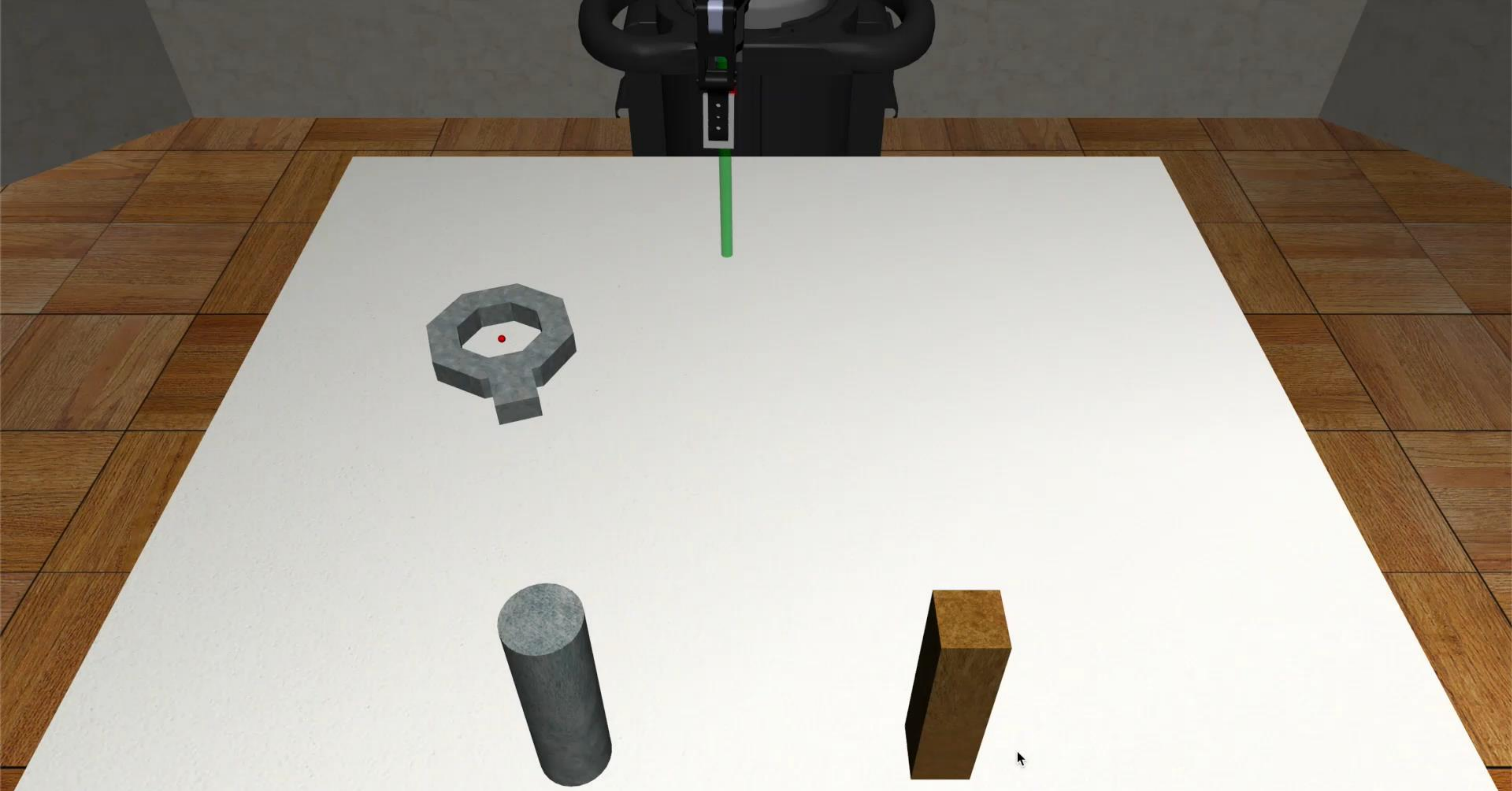


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

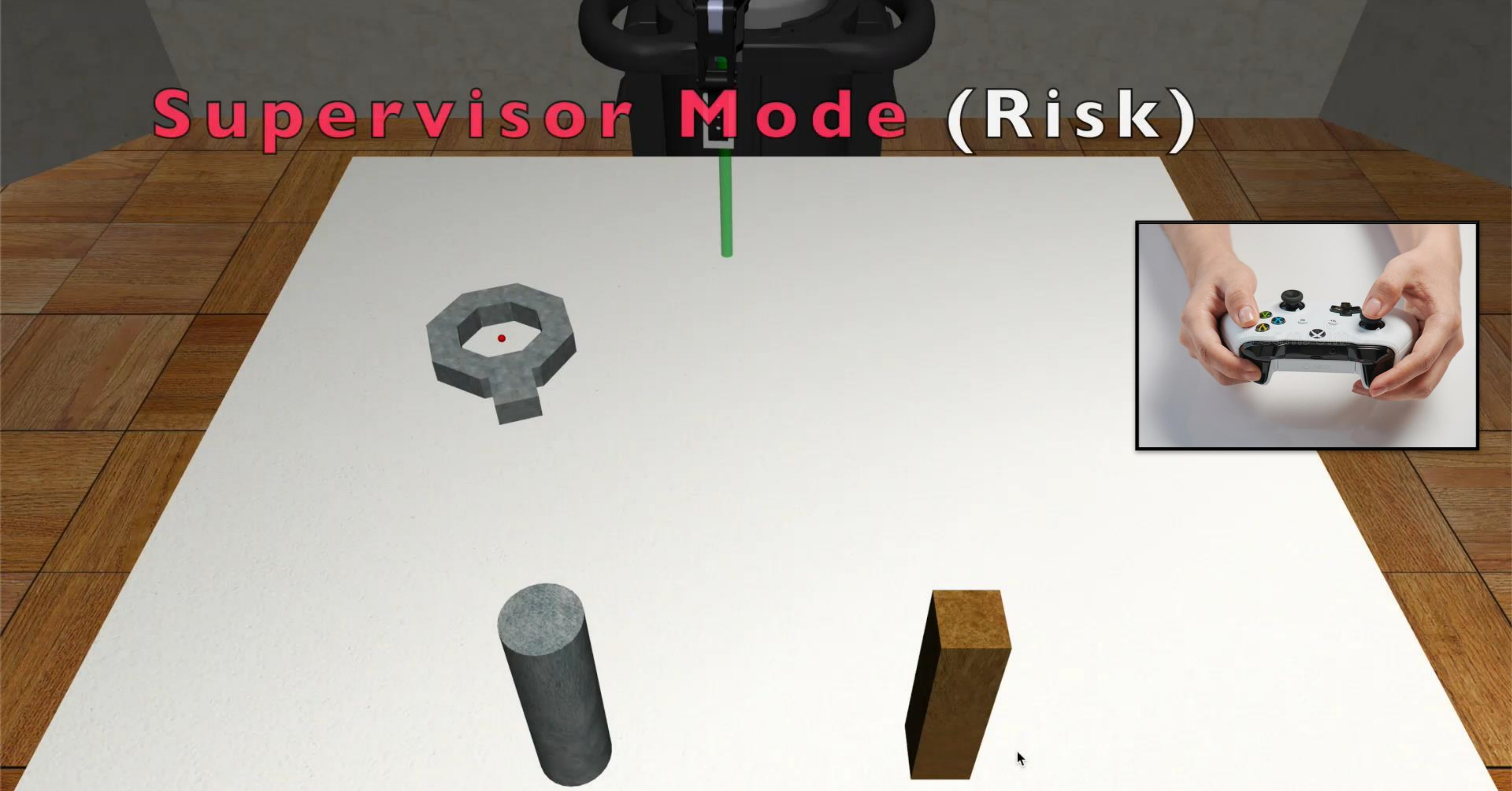


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

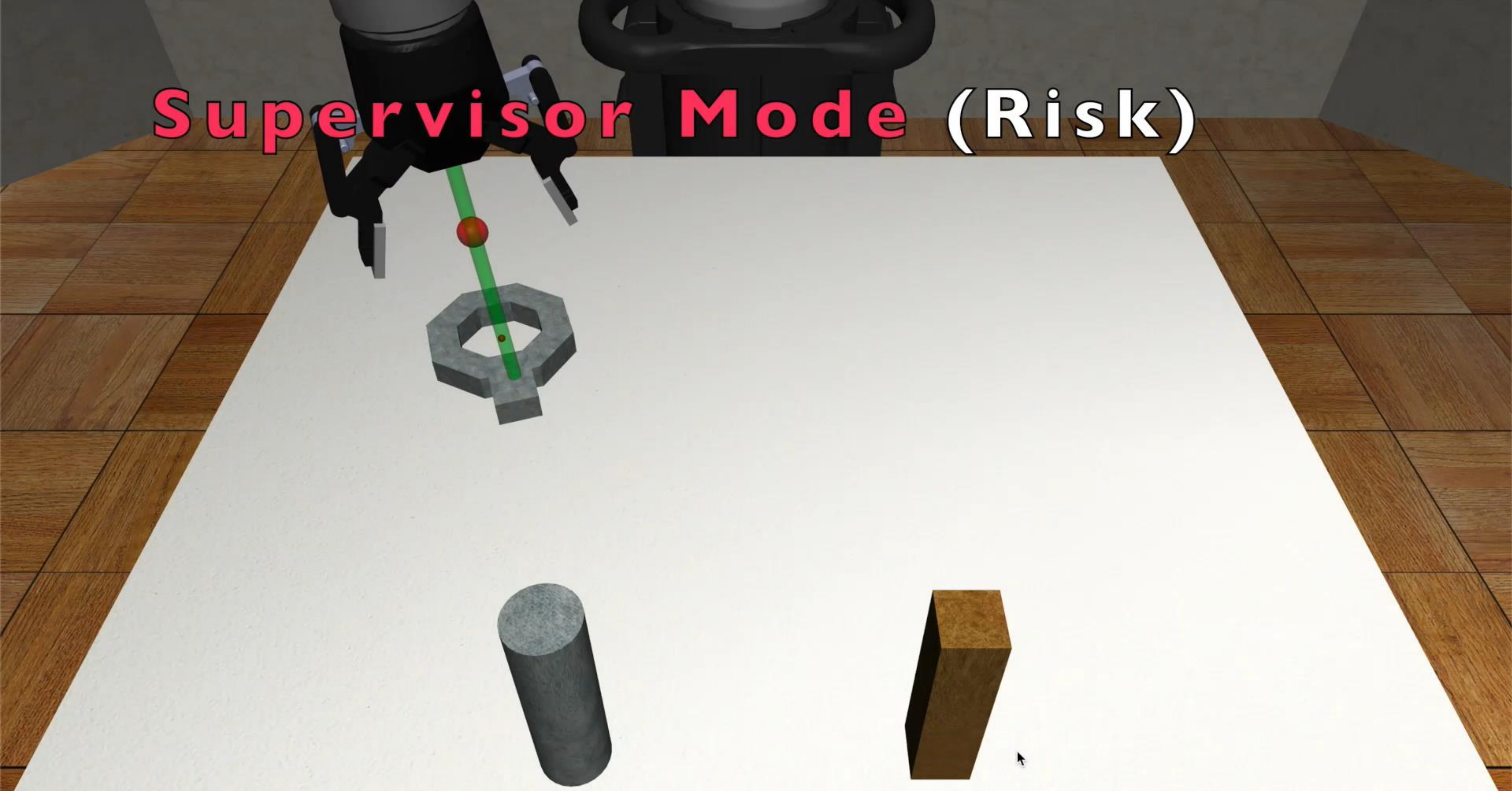
Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.



Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

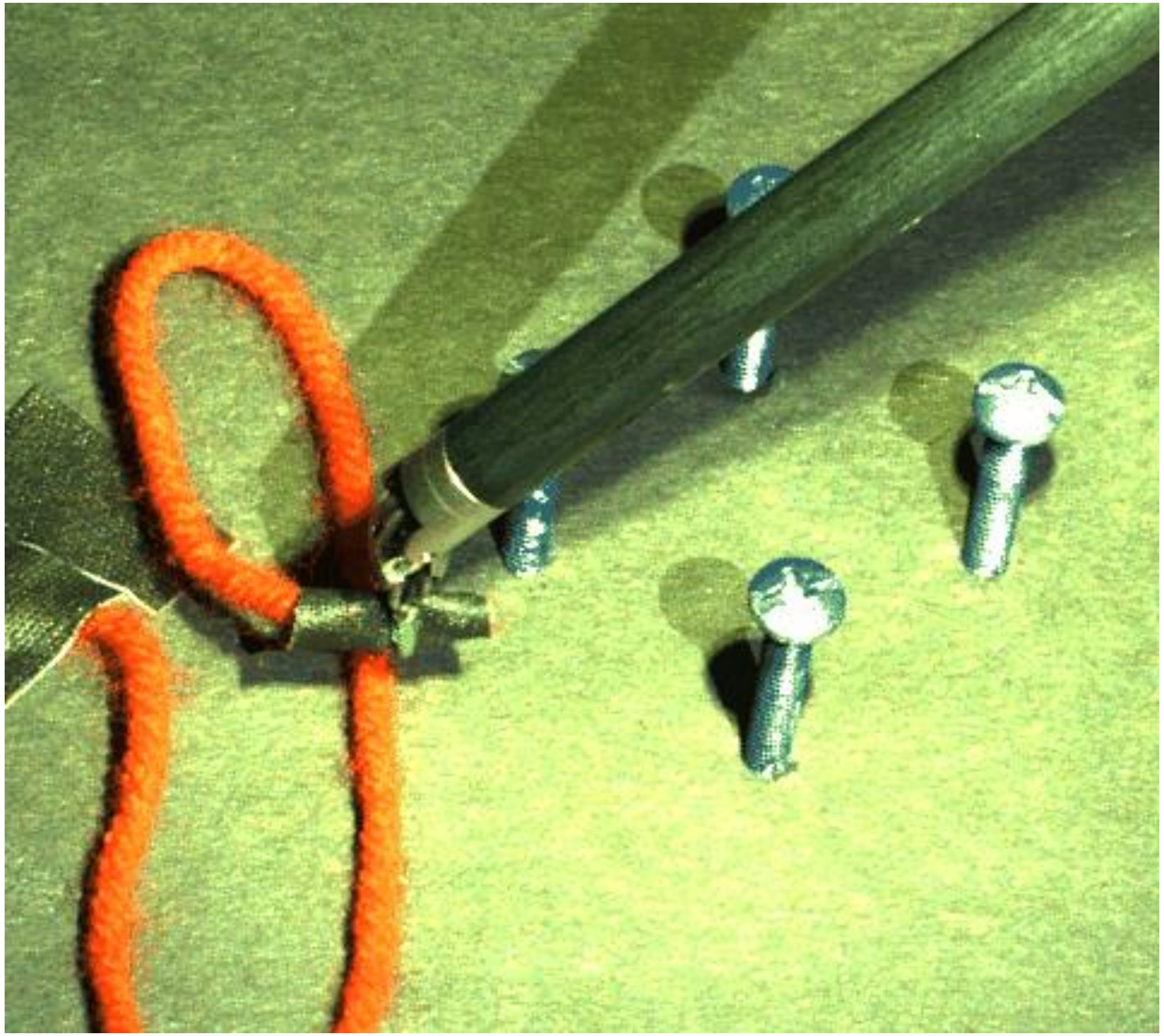


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

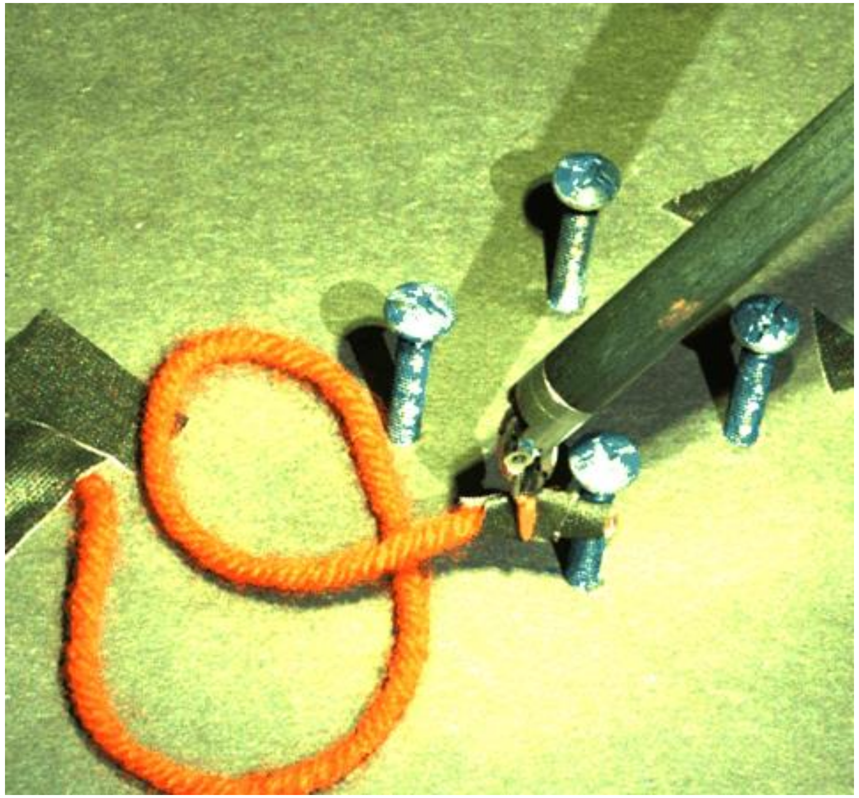


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

Human Demonstration

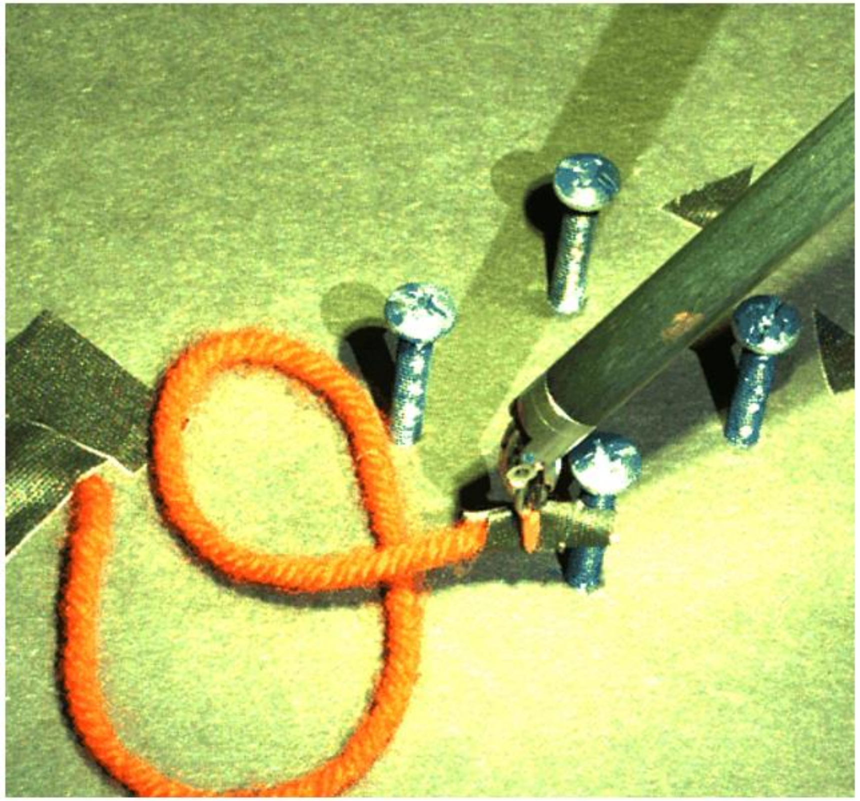


Behavior Cloning

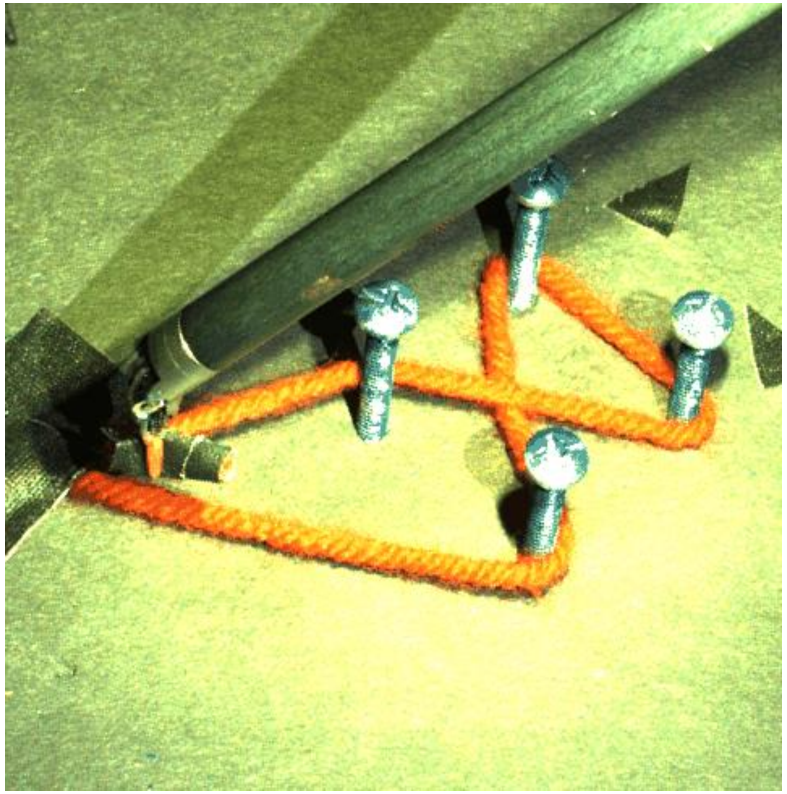


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

Behavior Cloning

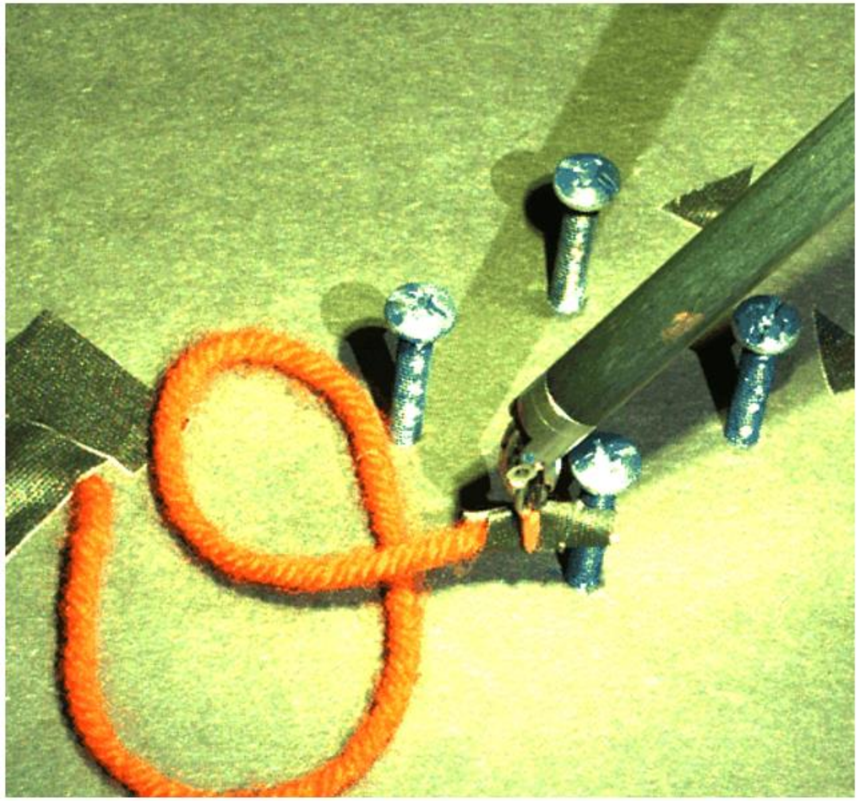


ThriftyDAgger (autonomous)

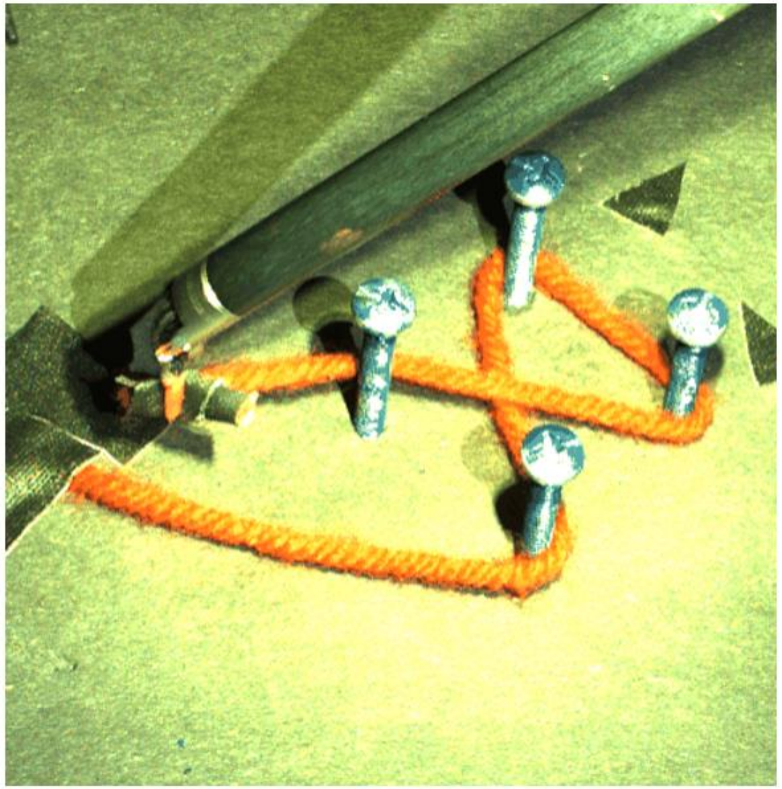


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

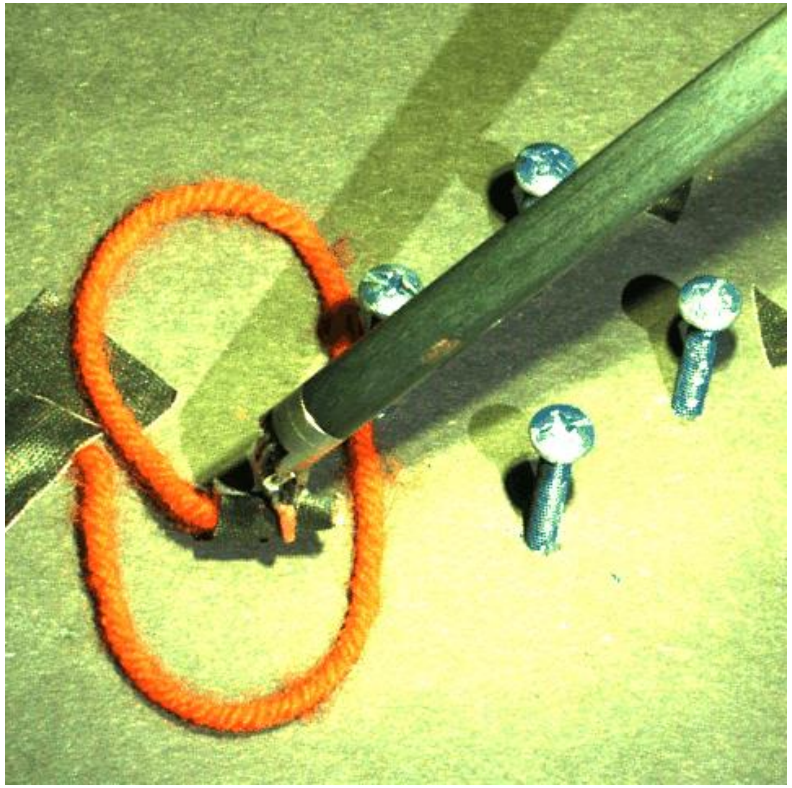
Behavior Cloning



ThriftyDAgger (autonomous)



ThriftyDAgger (+human)

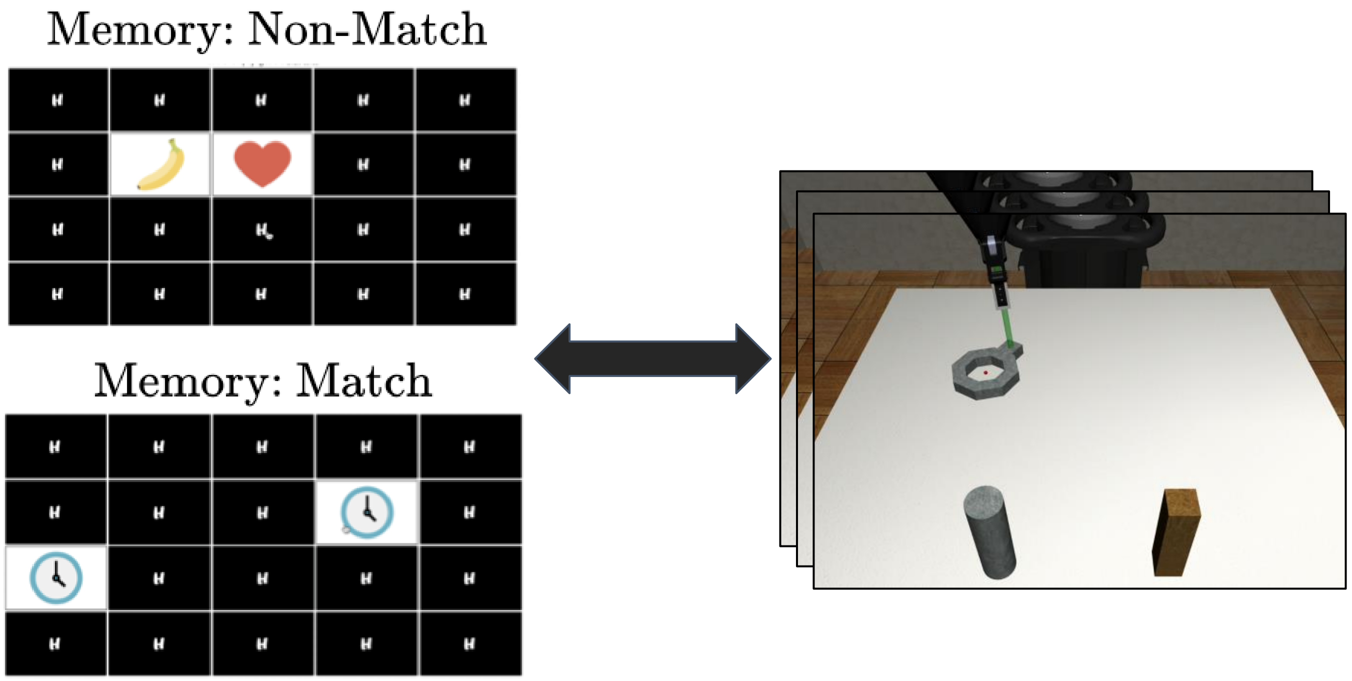


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

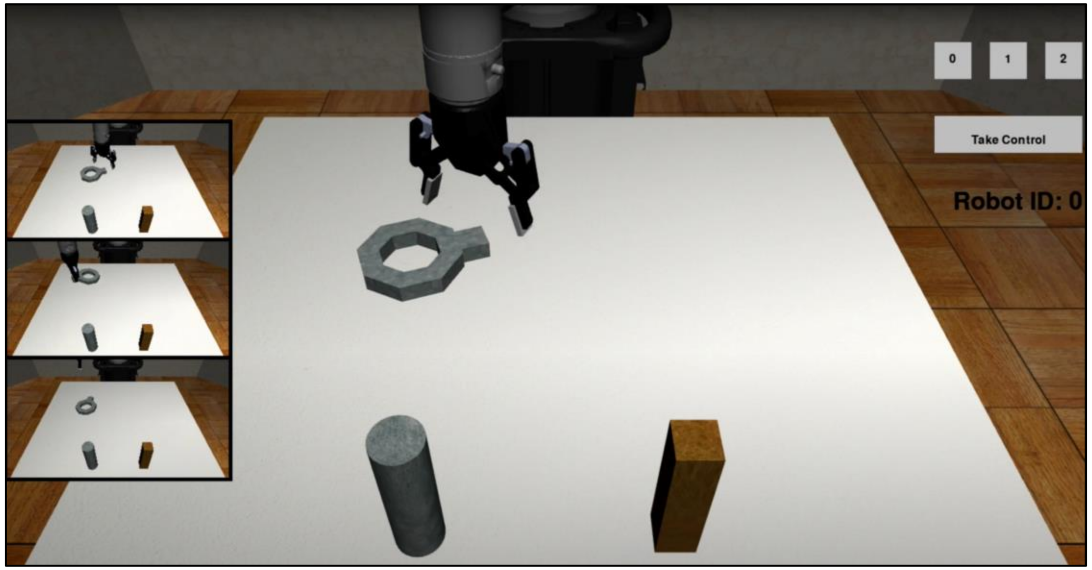
User Study

N=10 subjects each control 3 robots in simulation.

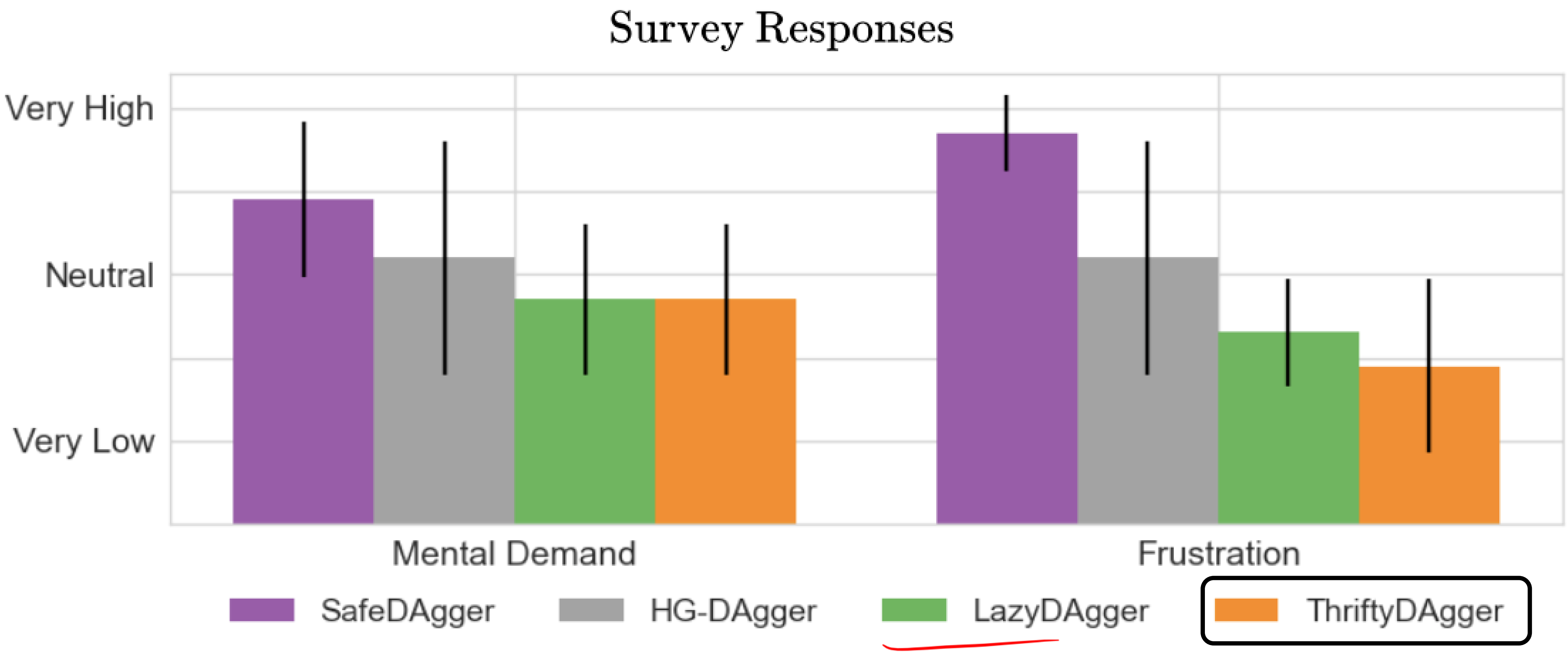
Robot-Gated



Human-Gated



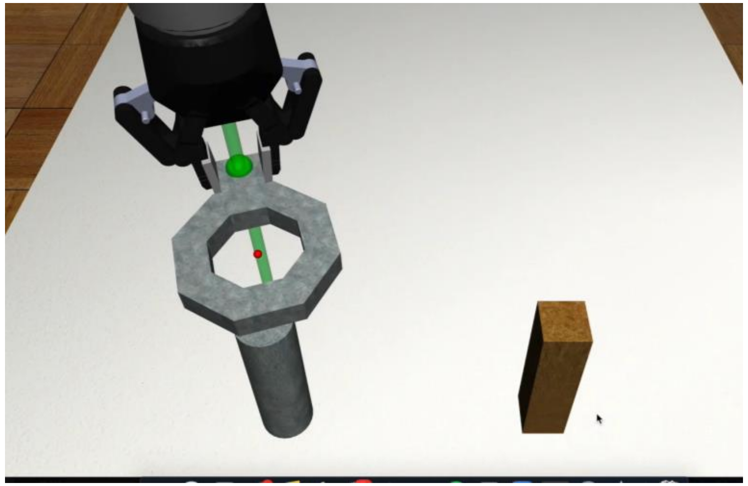
ThriftyDAgger Qualitative Results



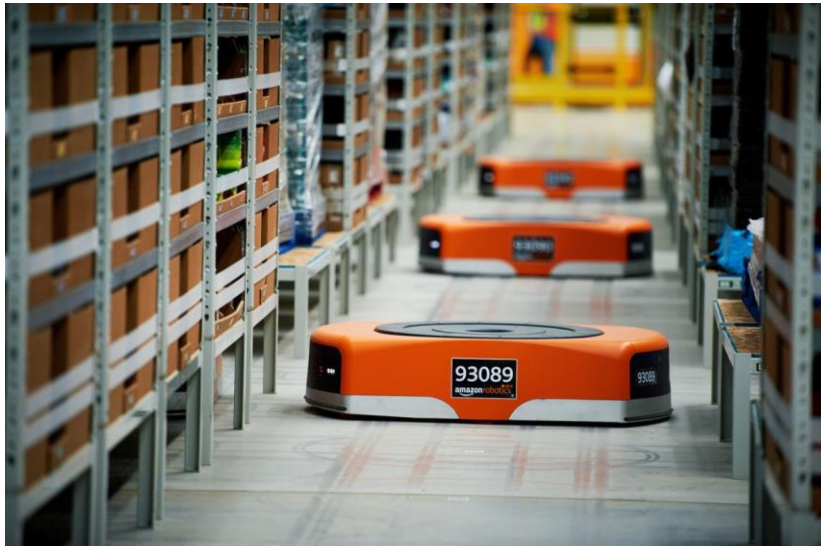
User Study Quantitative Results

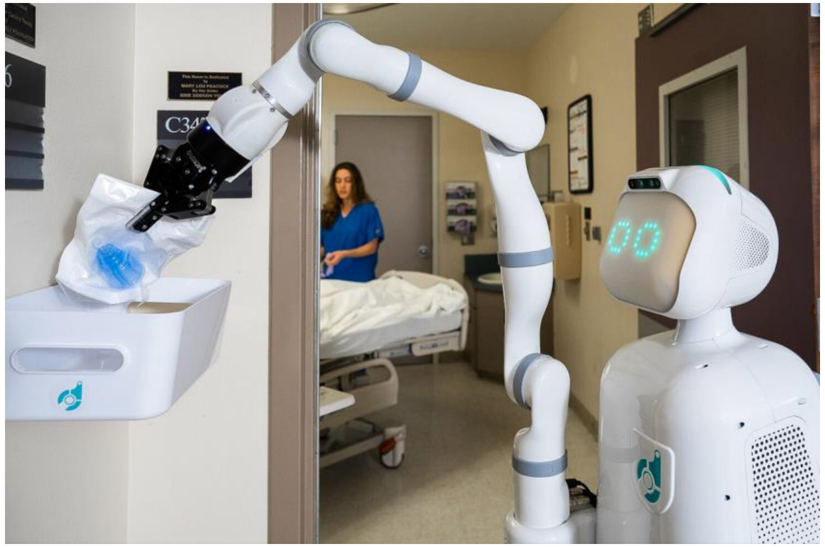
ThriftyDAgger had

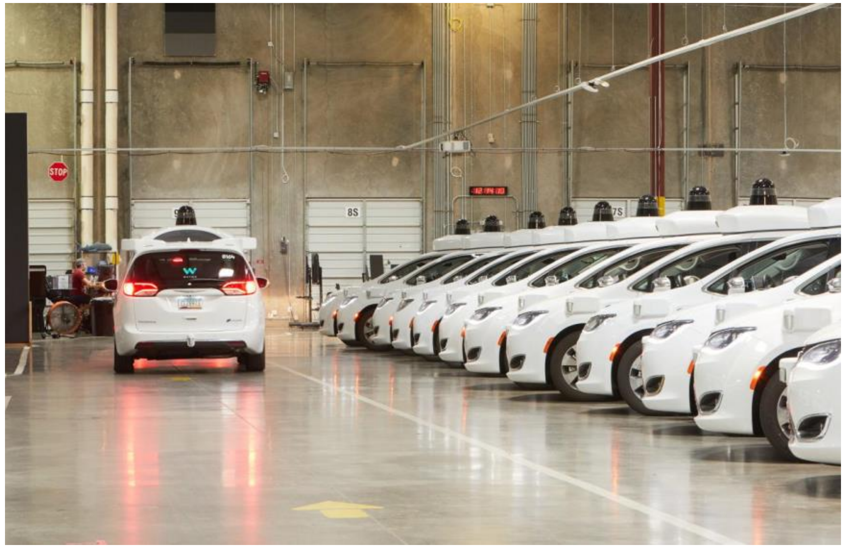
- 21% fewer human interventions
- 57% more concentration pairs found
- 80% more throughput



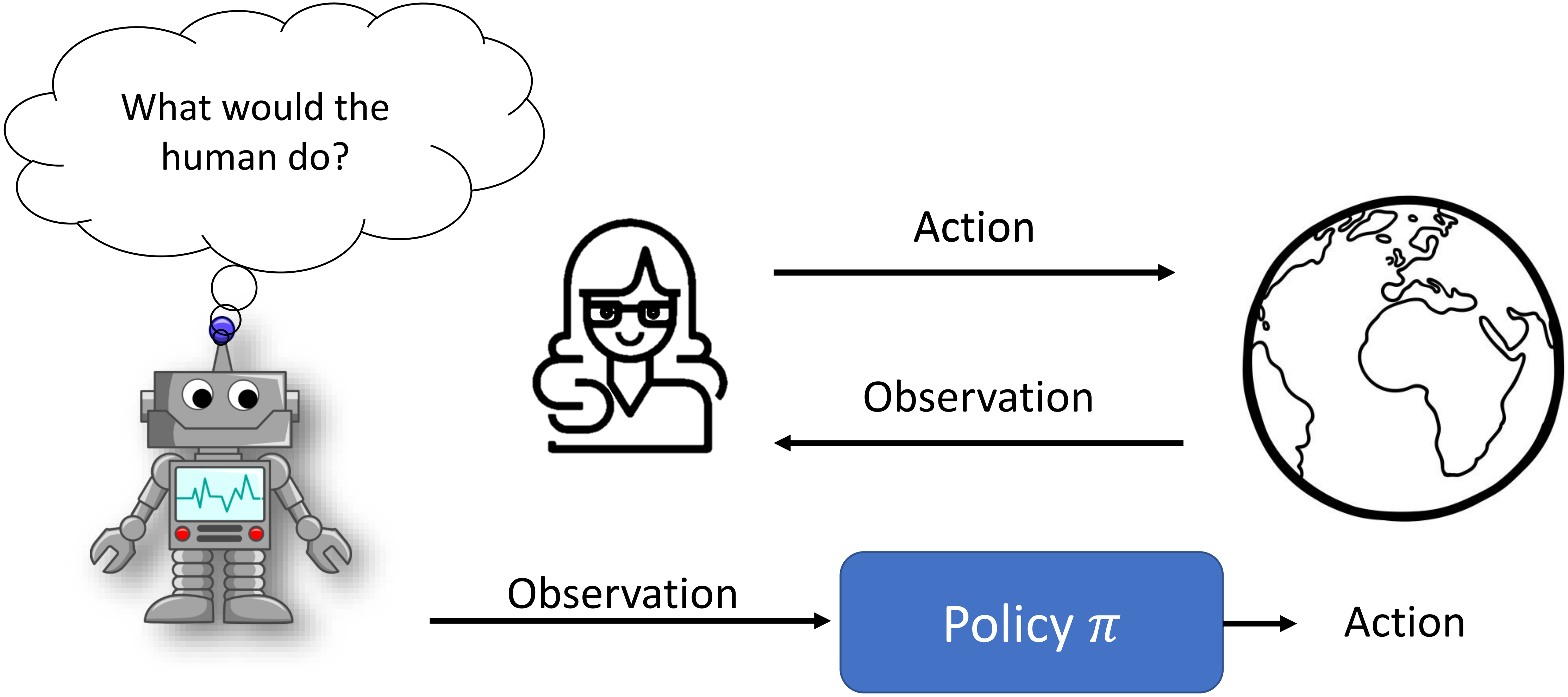
Lots of potential applications of human interaction with robot fleets!



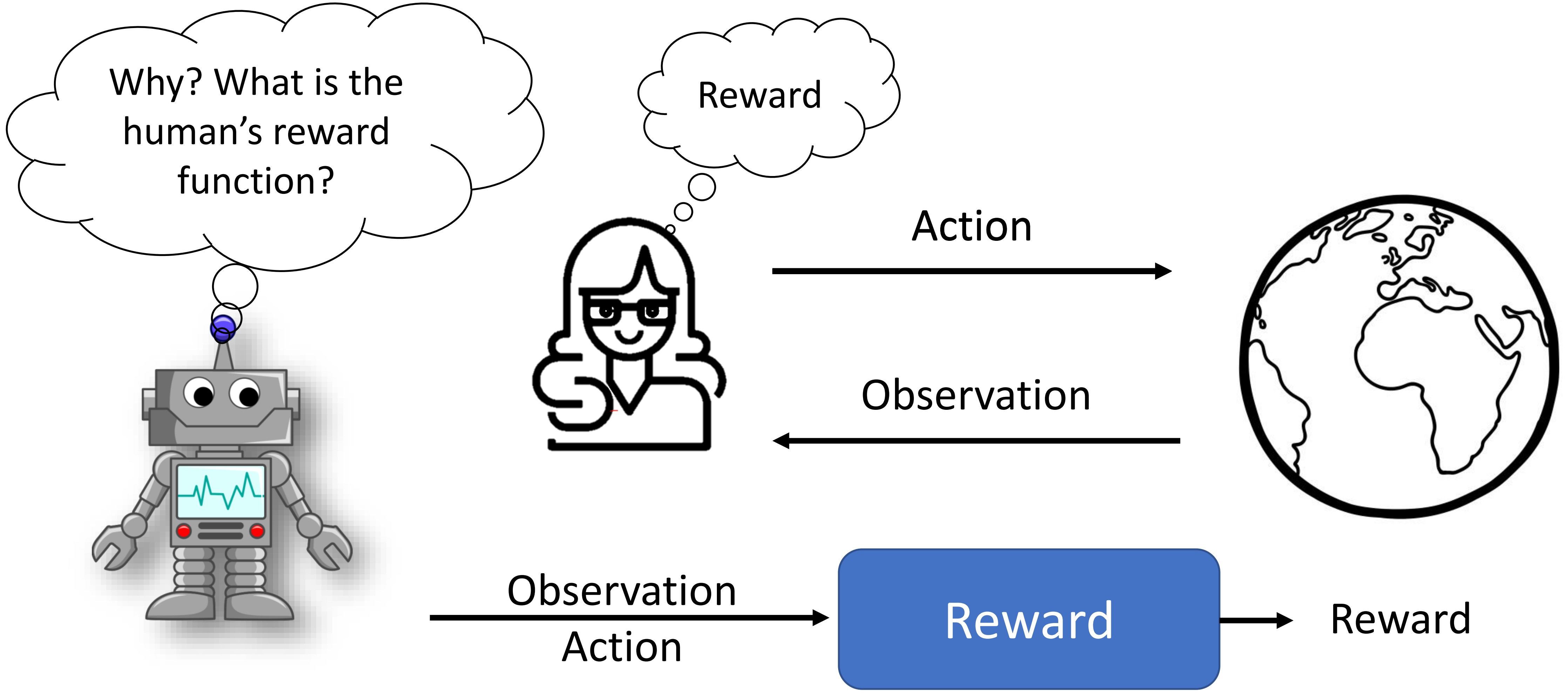




Behavioral Cloning



Reward Learning
(Inverse Reinforcement Learning)



Credit: Simone Giertz

Human Intent Inference

Optimal Control

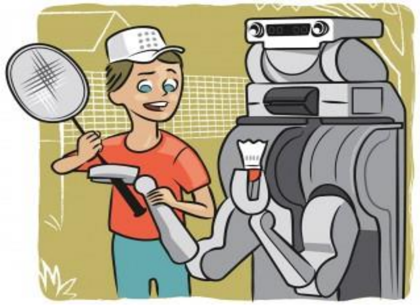
Inverse Reinforcement Learning

- Given
  - MDP without a reward function
  - Demonstrations from an optimal policy π^*

- Recover the reward function $\boldsymbol{R}$ that makes π^* optimal

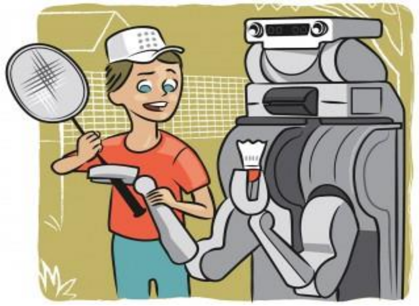
Imitation Learning

Behavioral Cloning

 $\Rightarrow \pi$

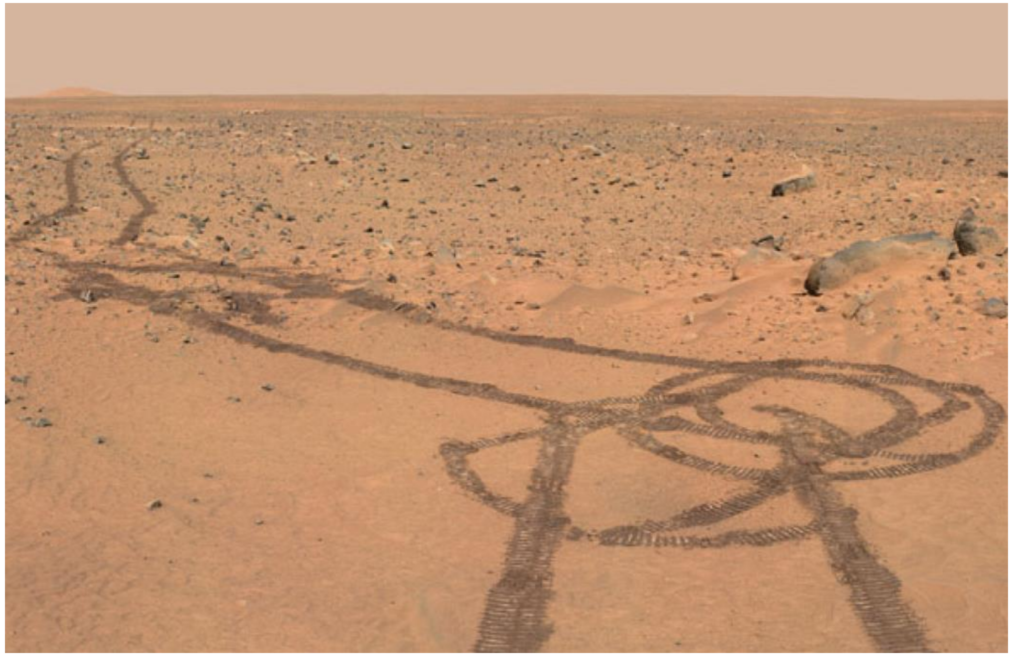
- Answers the “How?” question
- Mimic the demonstrator
- Learn mapping from states to actions
- Computationally efficient
- Compounding errors

Inverse Reinforcement Learning

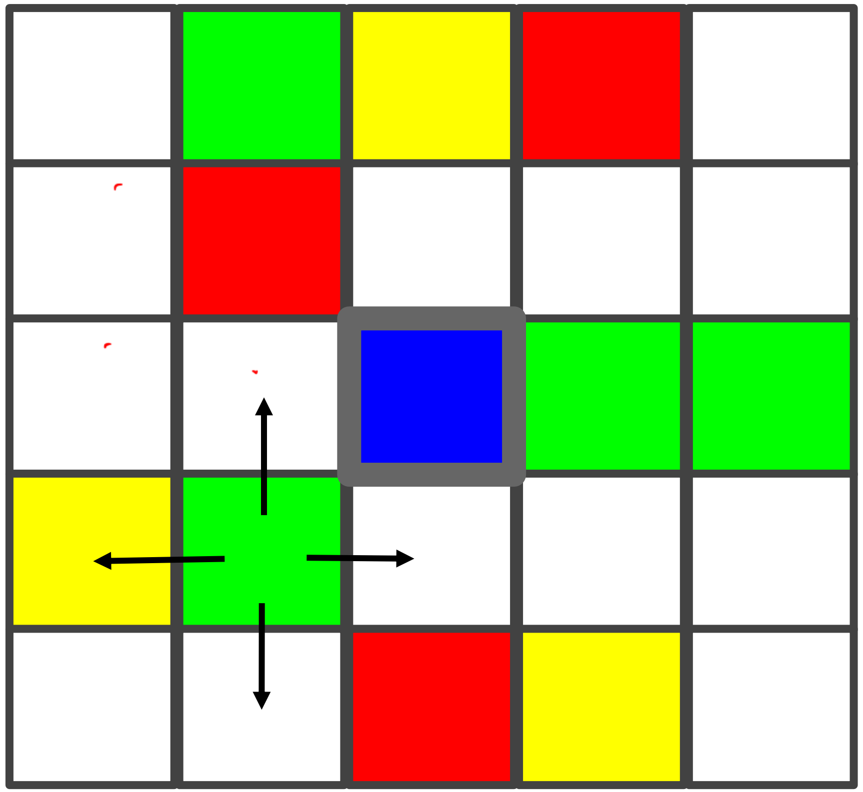
 $\Rightarrow R \Rightarrow \pi$

- Answers the “Why?” question
- Explain the demonstrator’s behavior
- Learn a reward function capturing the demonstrator’s intent
- Can require lots of data and compute
- Better generalization. Can recover from arbitrary states

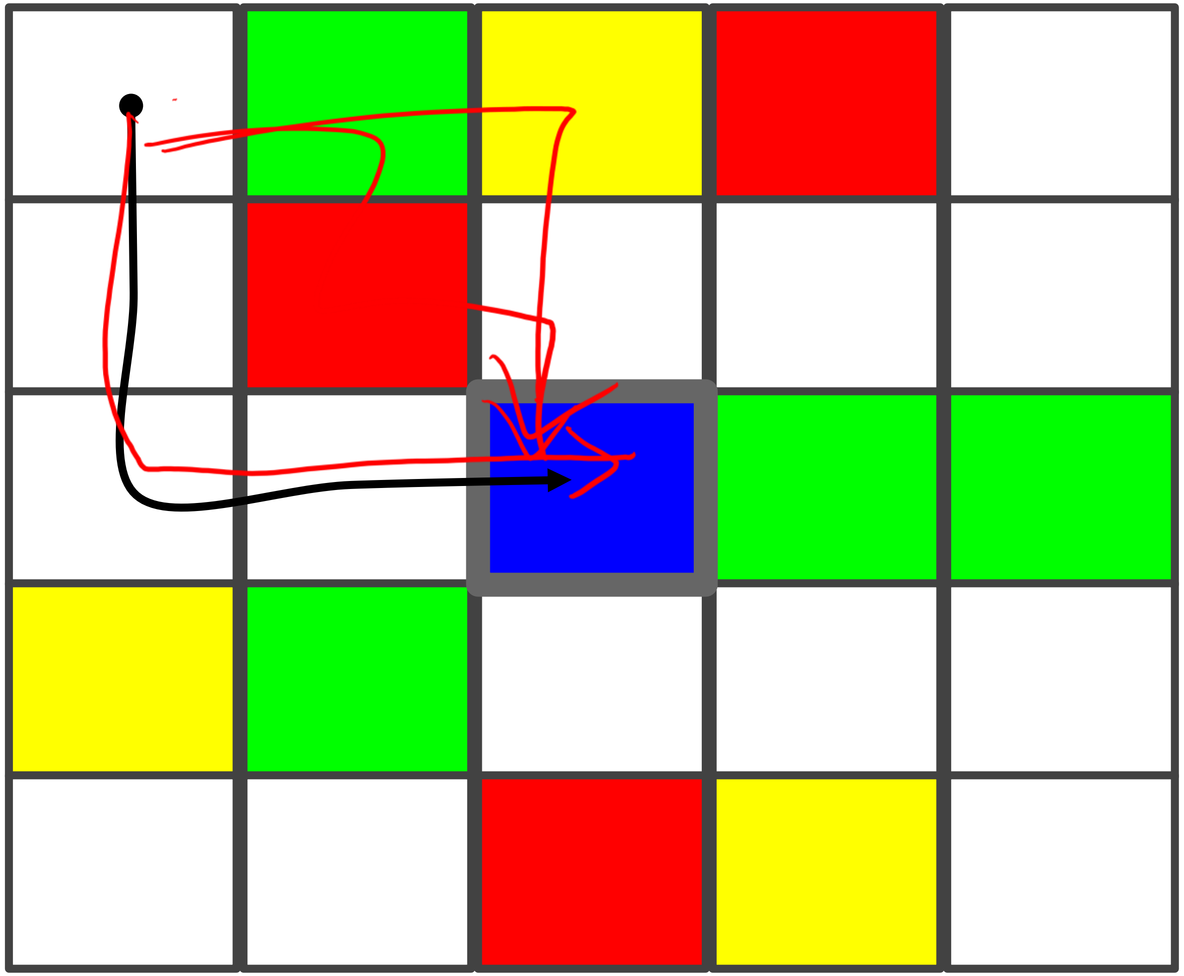
IRL Example: Teaching a robot to navigate through demonstrations

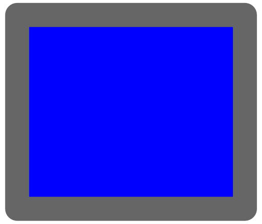


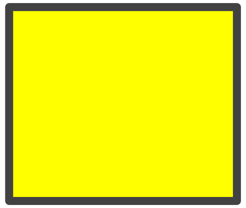
Toy version

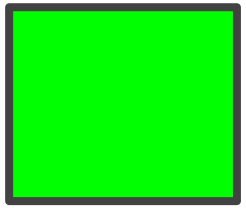


What is the reward?



R()=? >0

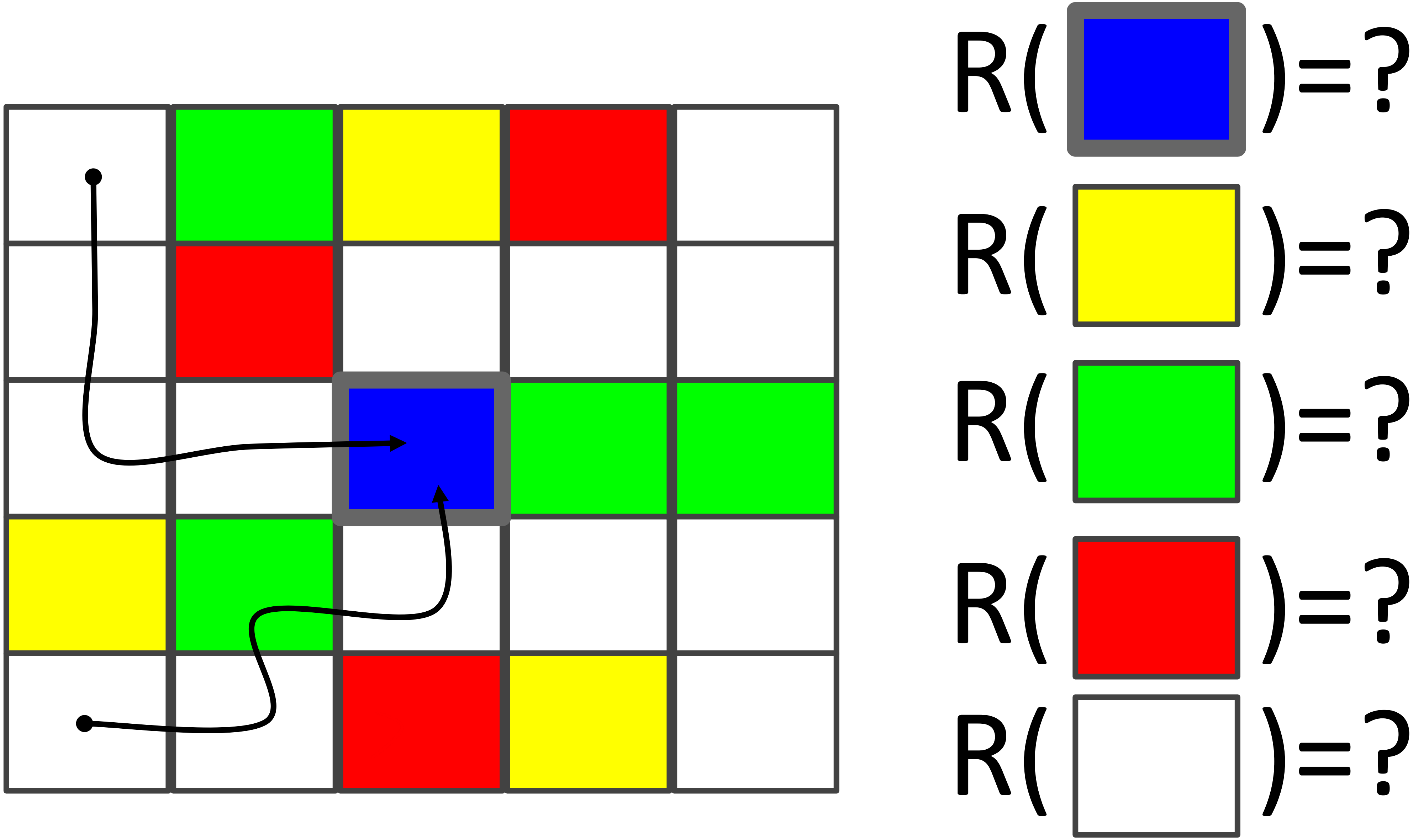
R()=? <0

R()=? <0

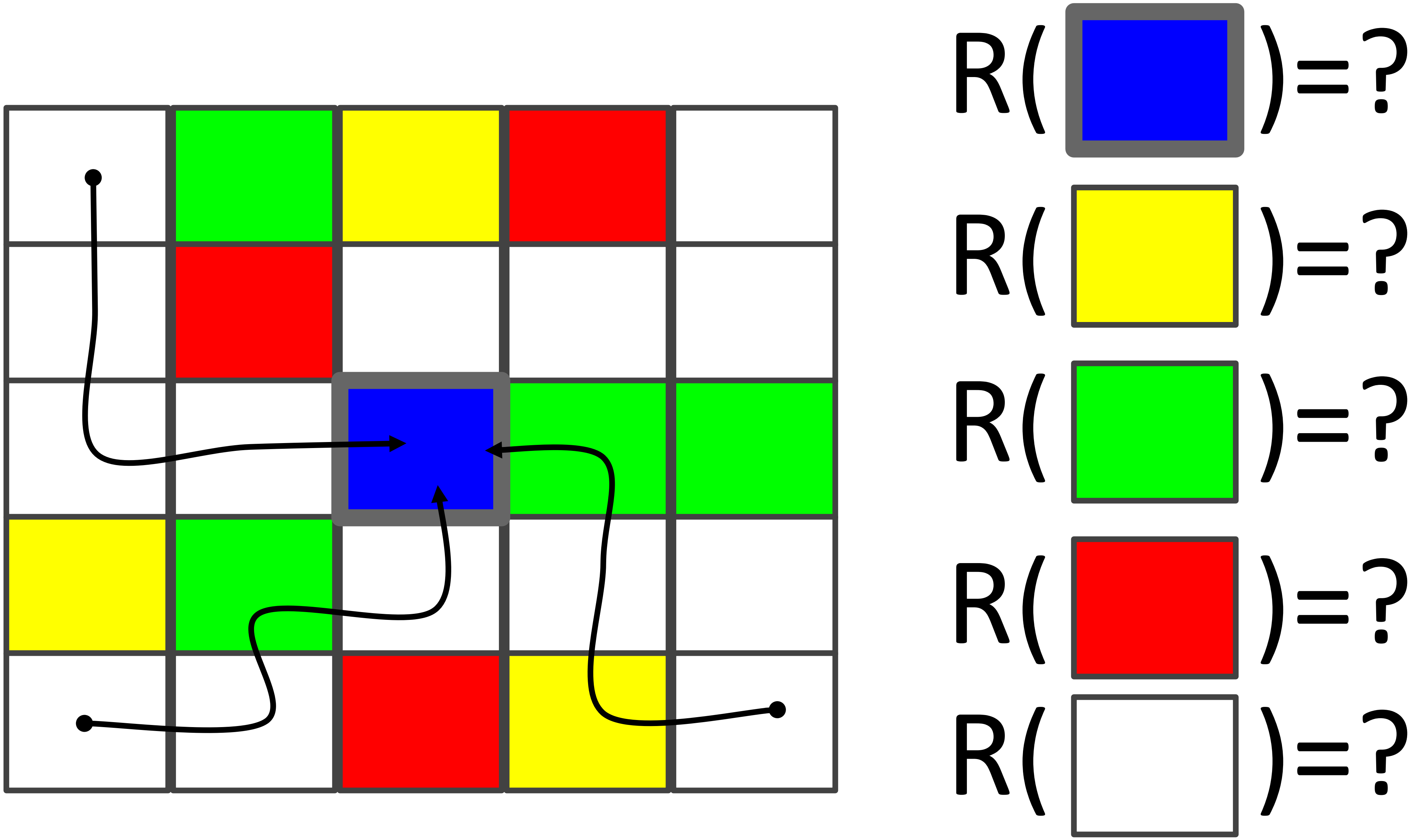
R()=? <0

R()=? 0

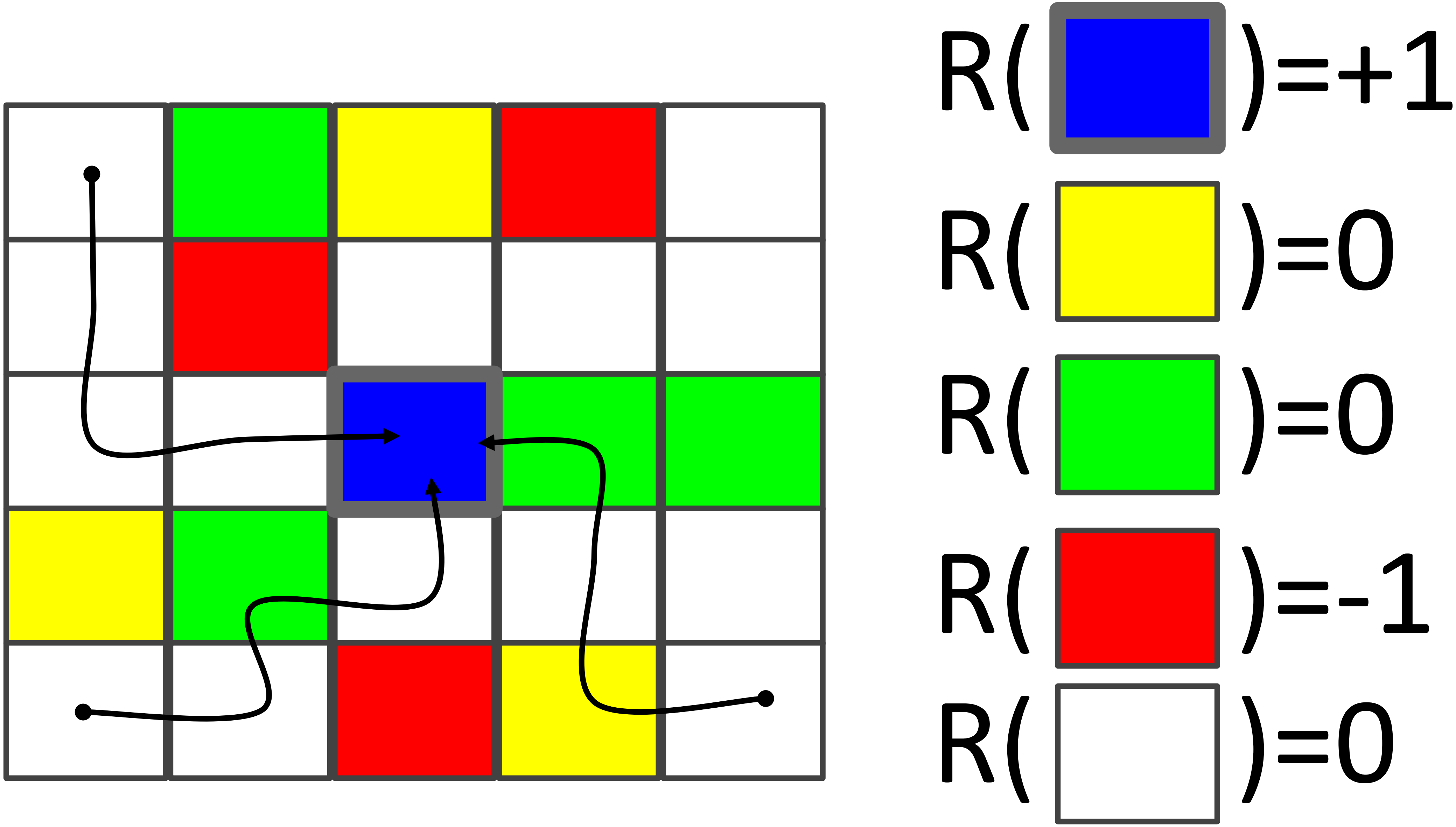
What is the reward?



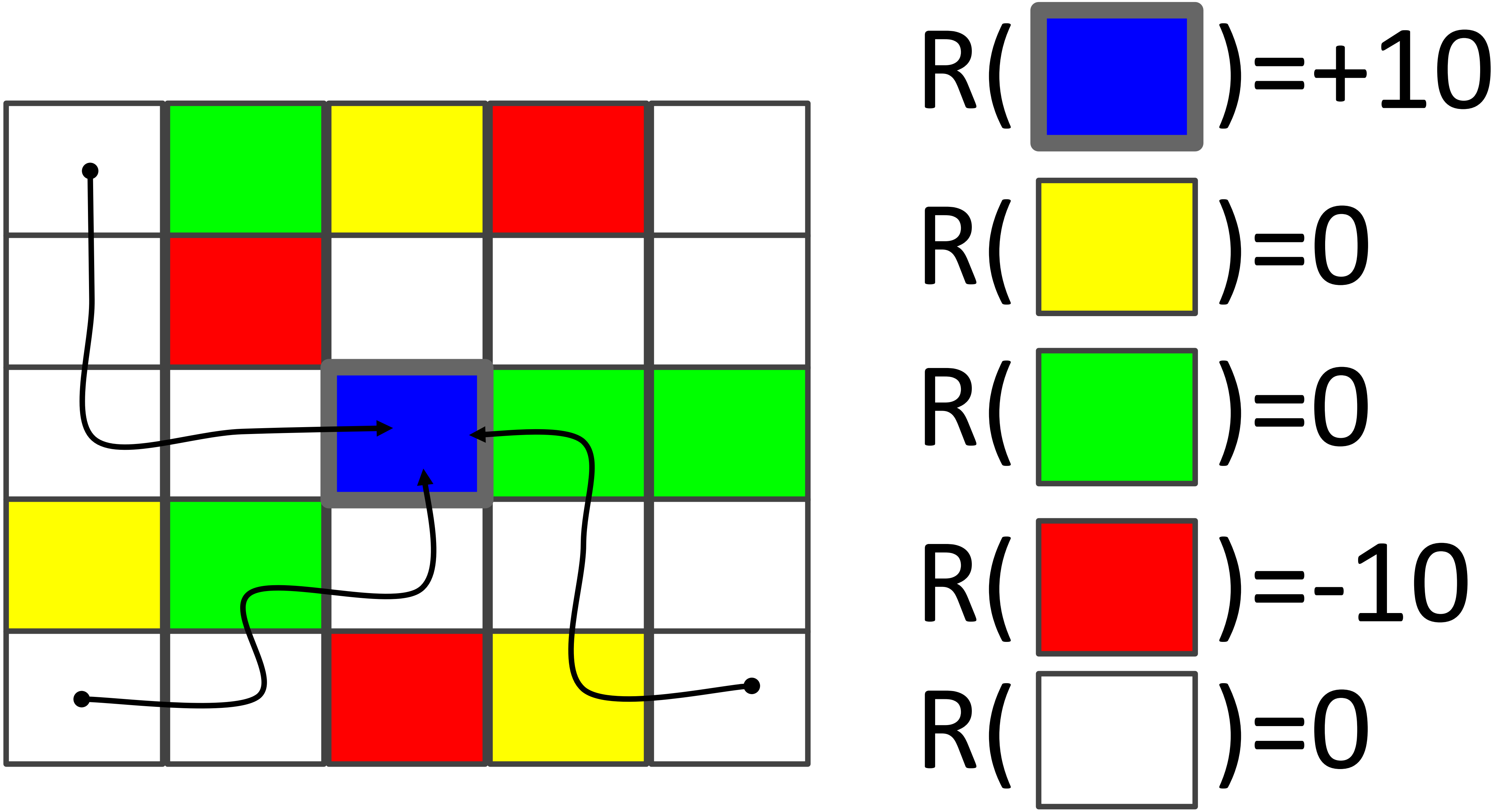
What is the reward?



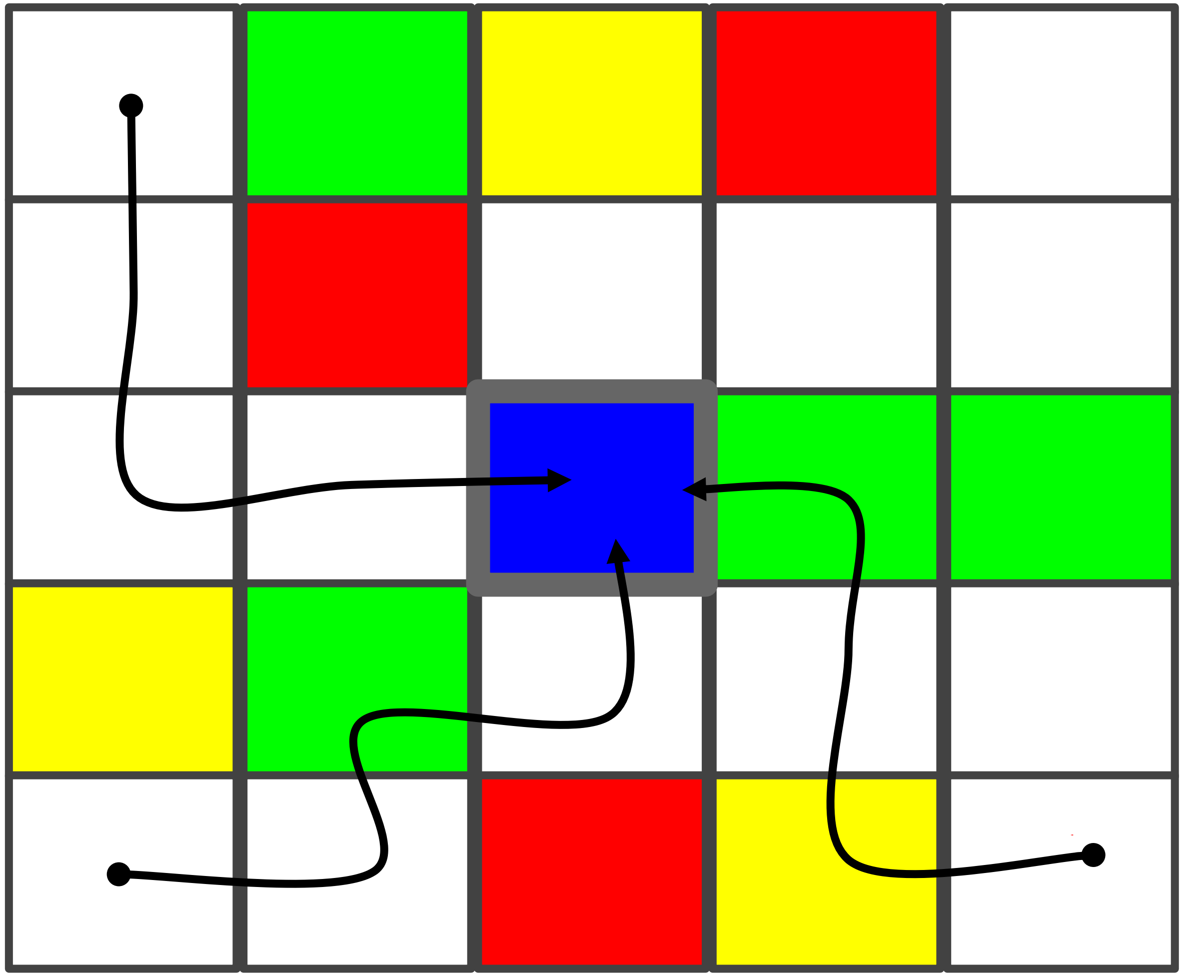
What is the reward?

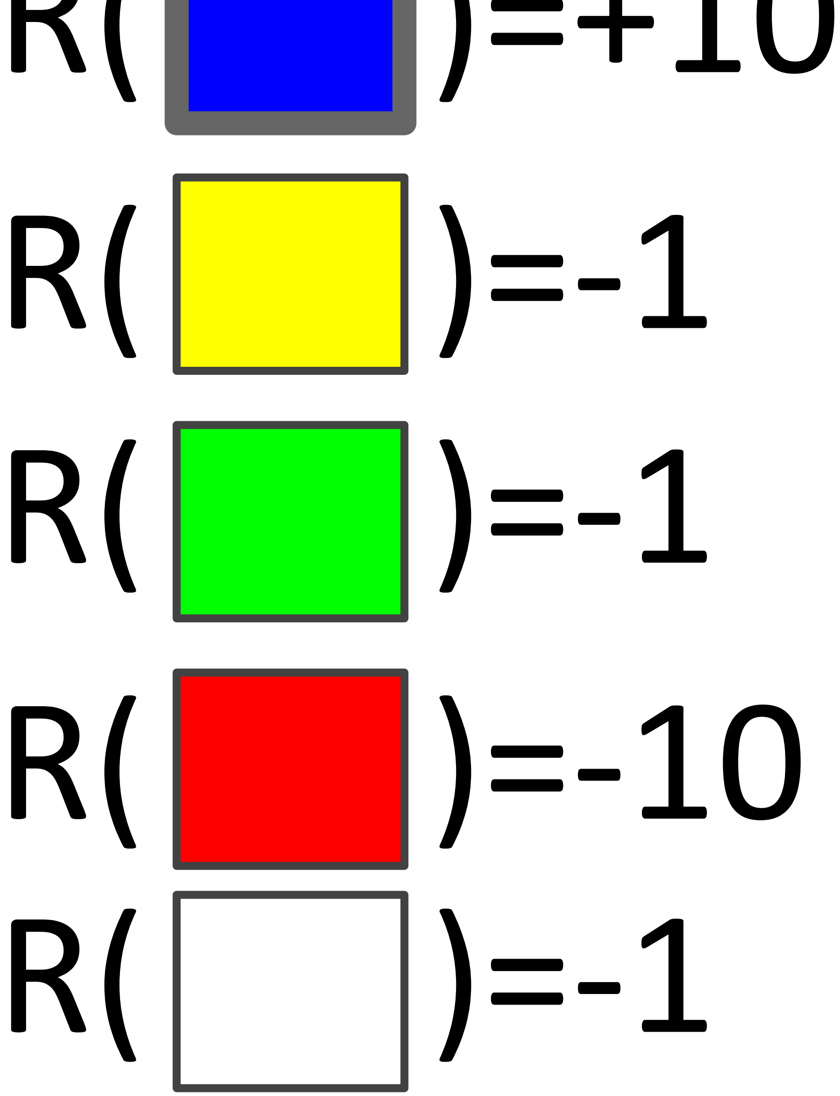


What is the reward?



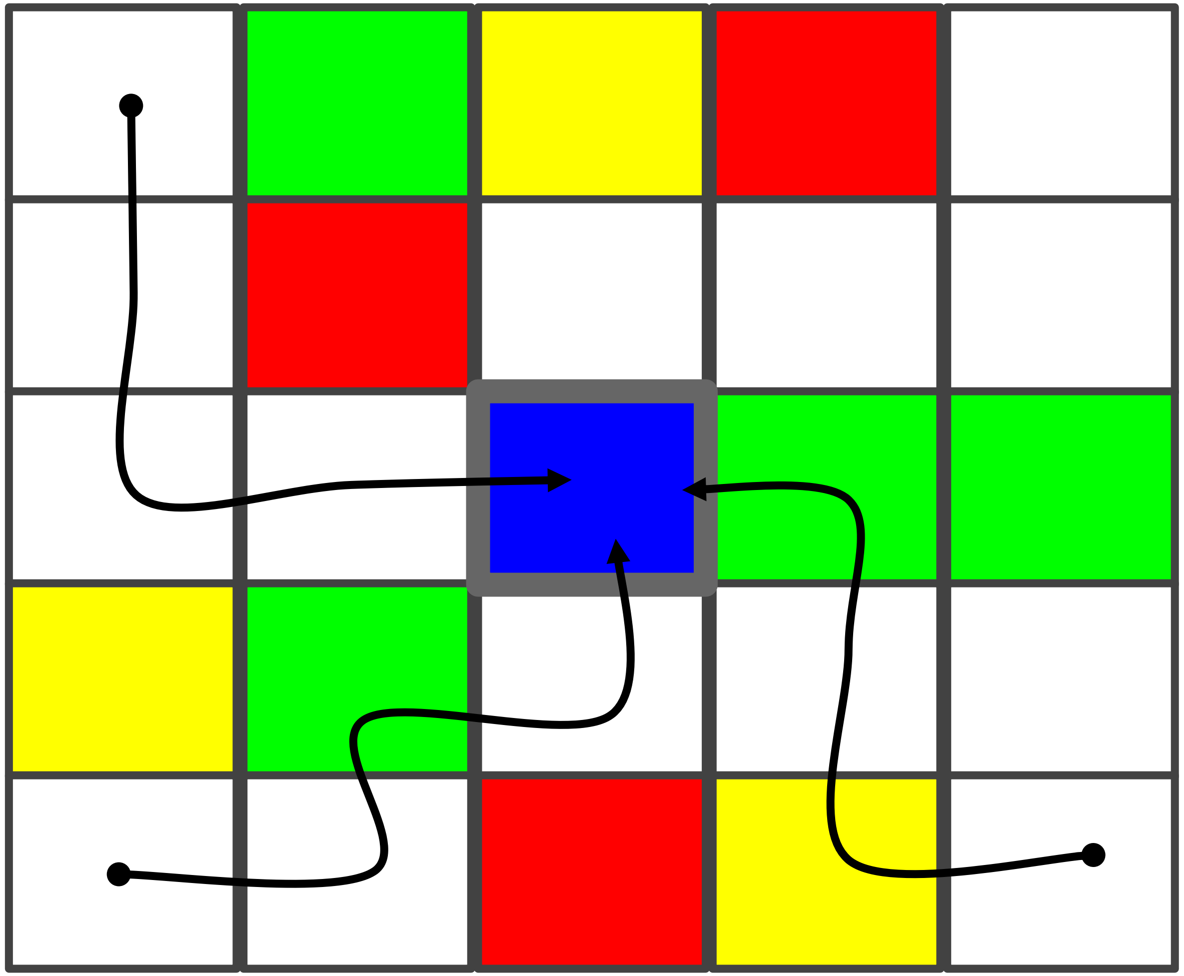
What is the reward?





What is the reward?

$\gamma = 0.9$



R( )=0

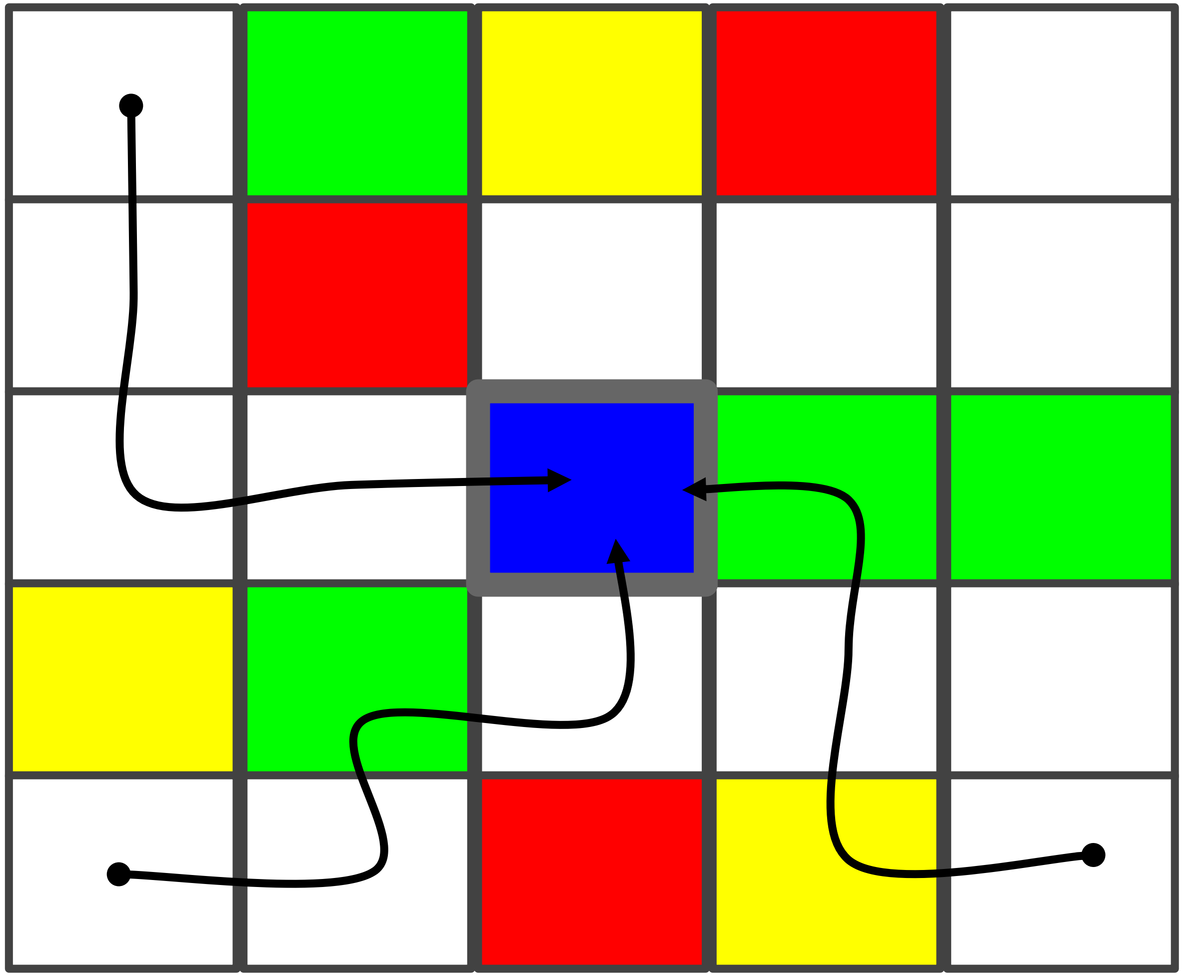
R( )=0

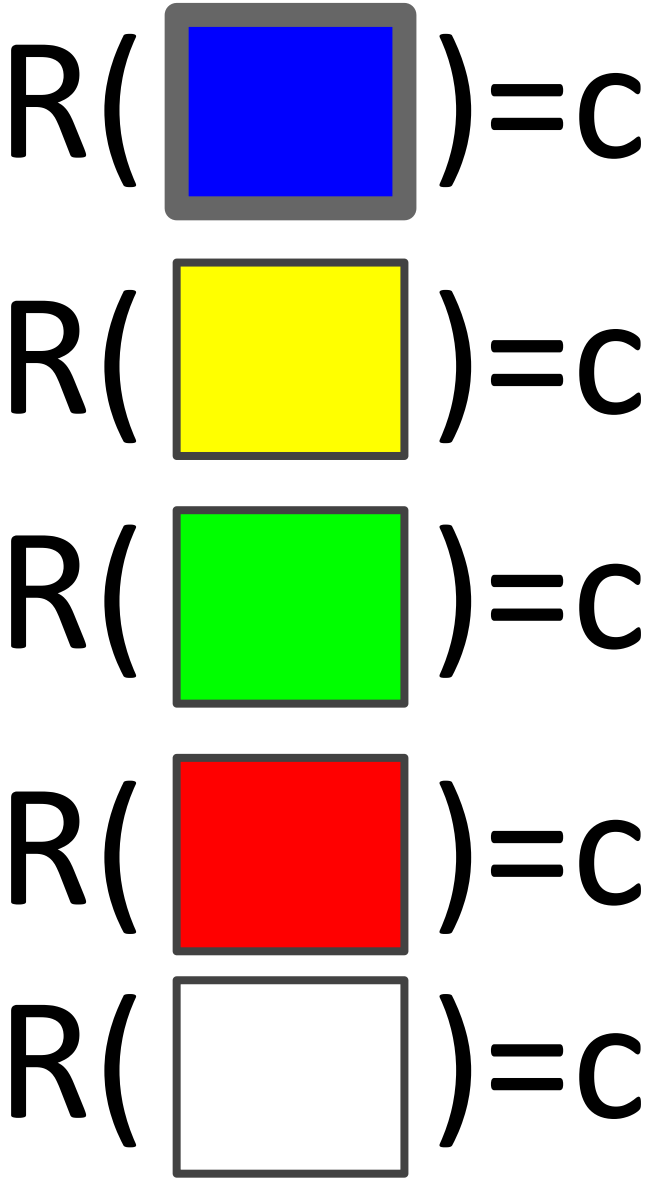
R( )=0

R( )=0

R( )=0

What is the reward?





Inverse Reinforcement Learning Formalism

- Given
  - MDP without a reward function
  - Demonstrations from an optimal policy π^*

- Recover a reward function $\boldsymbol{R}$ that makes π^* optimal

- Ill-Posed Problem
  - Infinite number of reward functions that can make π^* optimal
    - Trivial all zero reward
    - Constant reward
    - $aR + c$ (positive scaling a>0, and affine shifts)

Basic IRL Algorithm

- Start with demonstrations, D
- Guess initial reward function R_0
- $\hat{R} = R_0$
- Loop:
  - Solve for optimal policy $\pi^*_{\hat{R}}$
  - Compare D and $\pi^*_{\hat{R}}$
  - Update $\hat{R}$ to try and make D and $\pi^*_{\hat{R}}$ more similar

Flashback: Approximate Q-Learning

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \ldots + w_n f_n(s,a)$$

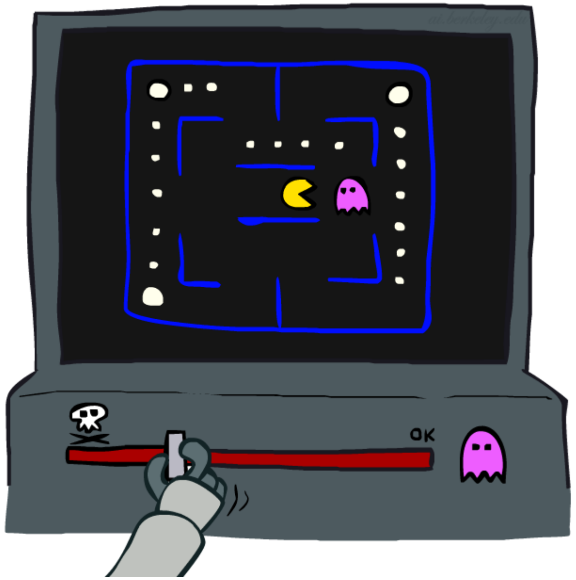
- Q-learning with linear Q-functions:

$$\text{transition } = (s,a,r,s')$$

$$\text{difference} = \left[r + \gamma \max_{a'} Q(s',a')\right] - Q(s,a)$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha\,[\text{difference}] \quad \text{Exact Q's}$$

$$w_i \leftarrow w_i + \alpha\,[\text{difference}]\, f_i(s,a) \quad \text{Approximate Q's}$$



- Intuitive interpretation:
  - Adjust weights of active features
  - E.g., if something unexpectedly bad happens, blame the features that were on: disprefer all states with that state's features

- Formal justification: online least squares

Feature count matching

- Assume the reward function is a linear combination of features:

$$R(s) = \mathbf{w}^T \phi(s) \quad = \sum_{i=1}^{K} w_i \cdot \phi_i(s)$$

- Value function becomes linear combination of (discounted) feature expectations:

$$V_R^\pi \quad = \quad \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \right]$$

Abbeel and Ng, "Apprenticeship learning via inverse reinforcement learning." ICML, 2004.

Feature count matching

- Assume the reward function is a linear combination of features:

$$R(s) = \mathbf{w}^T \phi(s)$$

- Value function becomes linear combination of (discounted) feature expectations:

$$V_R^\pi \quad = \quad \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \mathbf{w}^t \phi(s_t) \right]$$

Abbeel and Ng, "Apprenticeship learning via inverse reinforcement learning." ICML, 2004.

Feature count matching

- Assume the reward function is a linear combination of features:

$$R(s) = \mathbf{w}^T \phi(s)$$

- Value function becomes linear combination of (discounted) feature expectations:

$$V_R^\pi \quad = \quad \mathbf{w}^T \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) \right] = w^T \mu_\pi$$

Abbeel and Ng, "Apprenticeship learning via inverse reinforcement learning." ICML, 2004.

Inverse reinforcement learning: feature matching
(Abbeel and Ng 2004, Syed and Schapire 2007)

$|x^\top y| \leq \|x\|_1 \|y\|_\infty$

- If $||\boldsymbol{w}||_1 \leq 1$, then

$$\begin{aligned} V_R^{\pi^*} - V_R^{\pi_{\text{robot}}} &= \mathbf{w}^T(\mu_{\pi^*} - \mu_{\pi_{\text{robot}}}) \\ &\leq \|\mu_{\pi^*} - \mu_{\pi_{\text{robot}}}\|_\infty \end{aligned}$$

- If feature expectations match, then expected returns are identical.
- Idea: Can we update the reward guess $\hat{R}$ so the feature counts get closer?

Abbeel and Ng, "Apprenticeship learning via inverse reinforcement learning." ICML, 2004.

Problem: Many different policies can lead to same expected feature counts

Maximum Entropy IRL (Ziebart et al. 2008)

$$P(\tau) = \frac{e^{R_w(\tau)}}{Z}$$

$$R(s) = \mathbf{w}^T \phi(s)$$

- Collect M demonstrations $\mathrm{D} = \{\tau_1, \dots, \tau_M\}$
- Initialize reward weights $\boldsymbol{w}$
- **Loop**
  - Solve for (soft) optimal policy $\pi(a|s)$ via Value Iteration

    Soft
  - Solve for expected feature counts of $\pi(a|\mathrm{s})$
  - Compute weight update

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha(\mu_D - \mu_\pi)$$