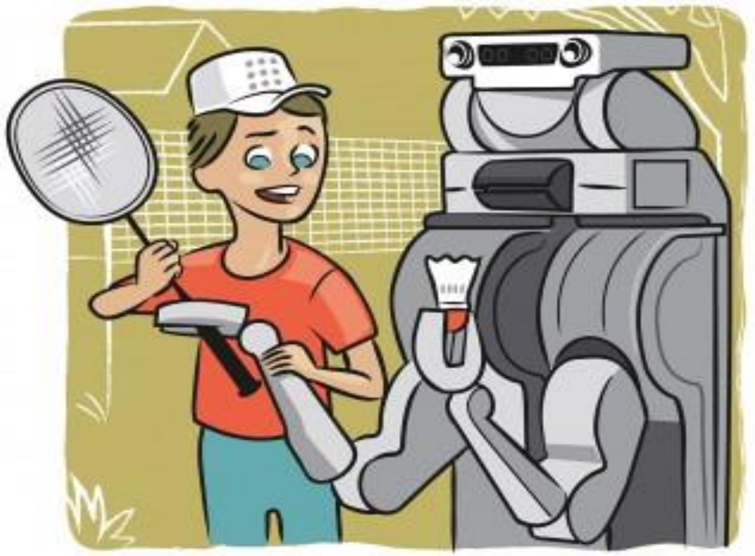
Behavioral Cloning and Interactive Imitation Learning

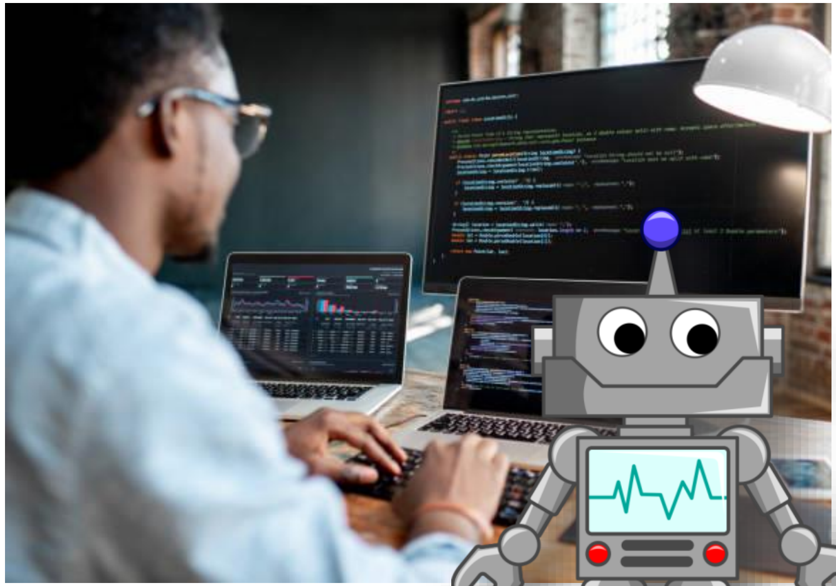


Instructor: Daniel Brown

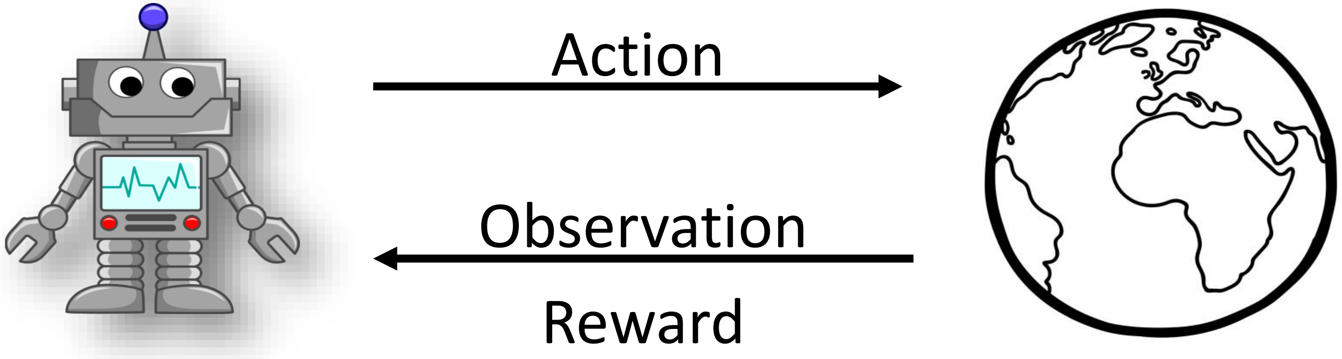
[Some slides adapted from Sergey Levine (CS 285) and Alina Vereshchaka (CSE4/510)]

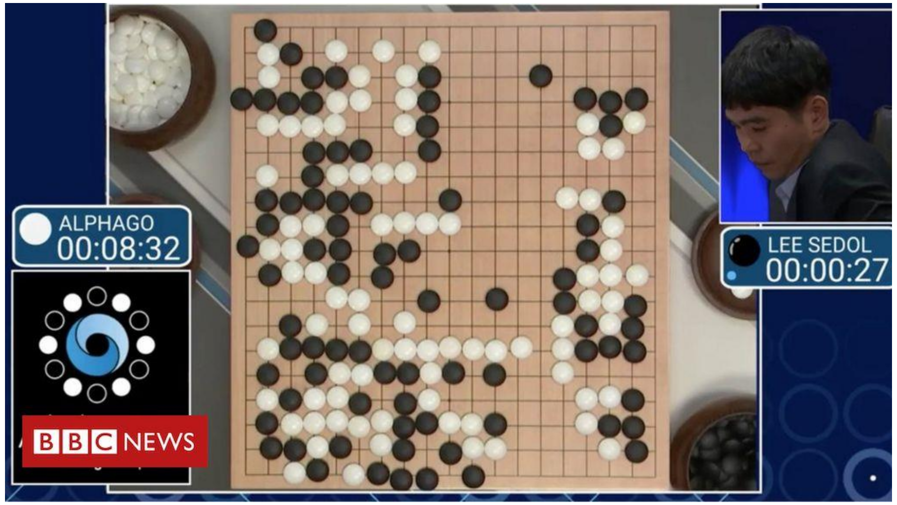
Course feedback is open

- Extra credit if class response rate is 70% or higher
  - Sliding scale if we reach 70%:
    - Extra credit points = response rate percentage / 10

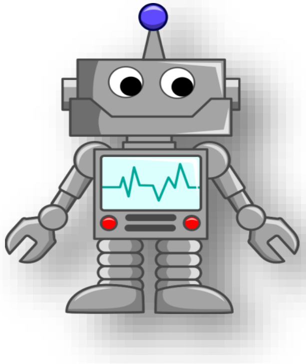


Reinforcement Learning





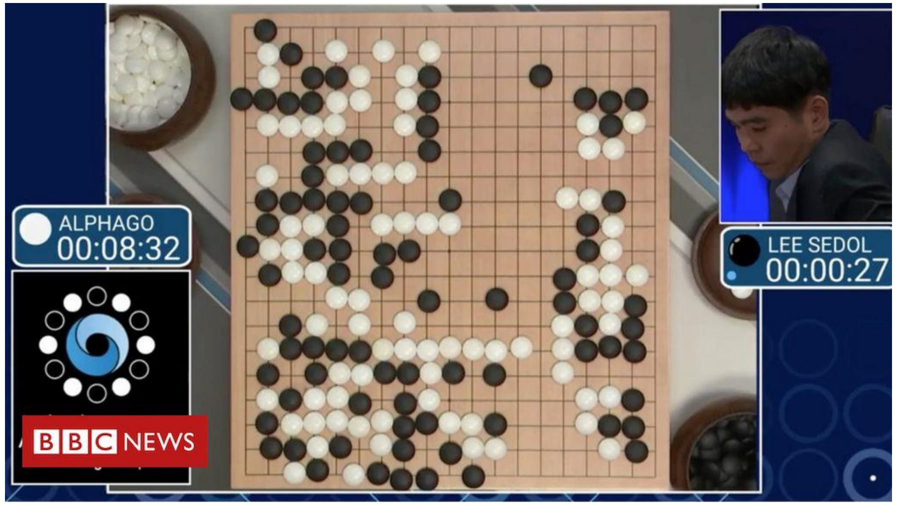
Reinforcement Learning

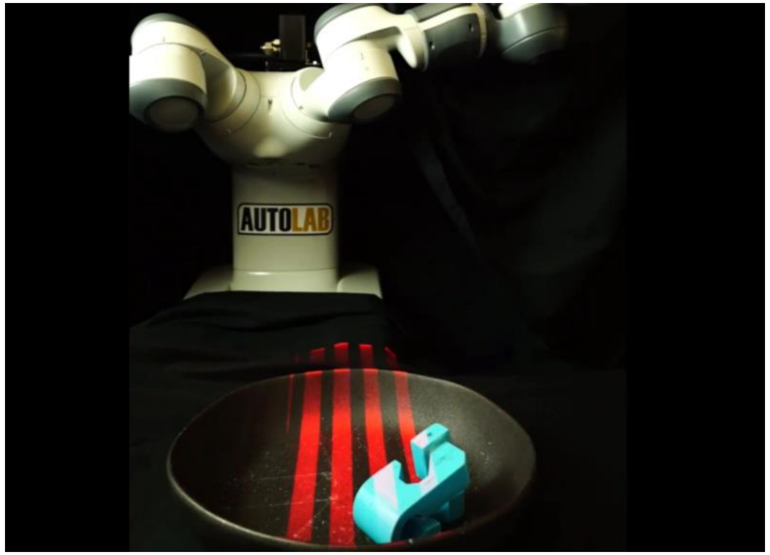


Action →

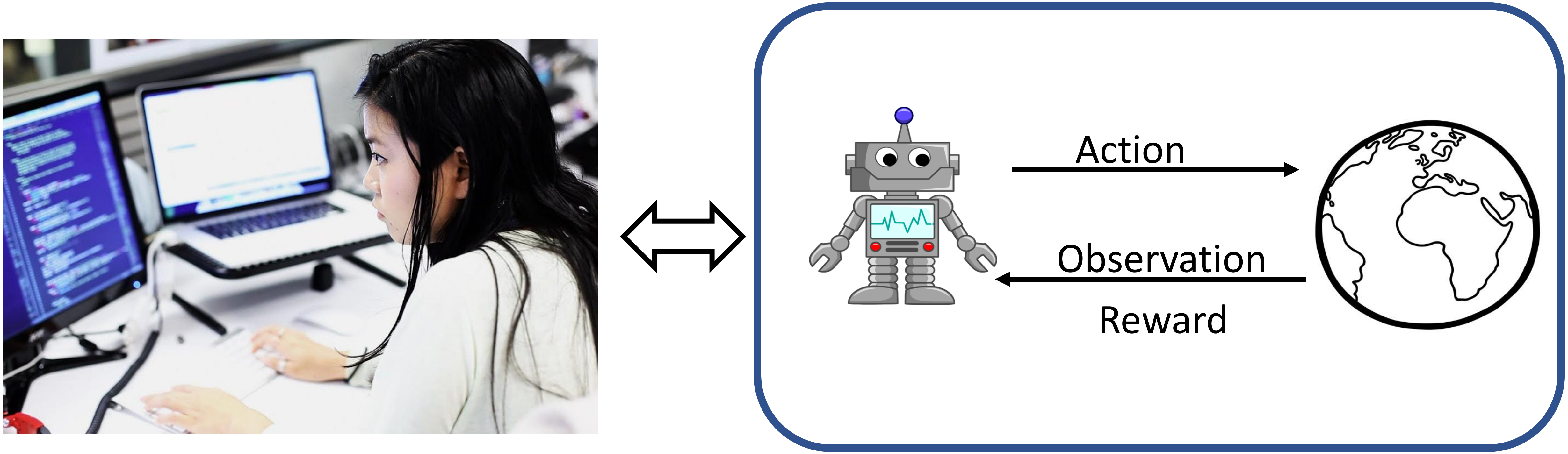
← Observation

Reward

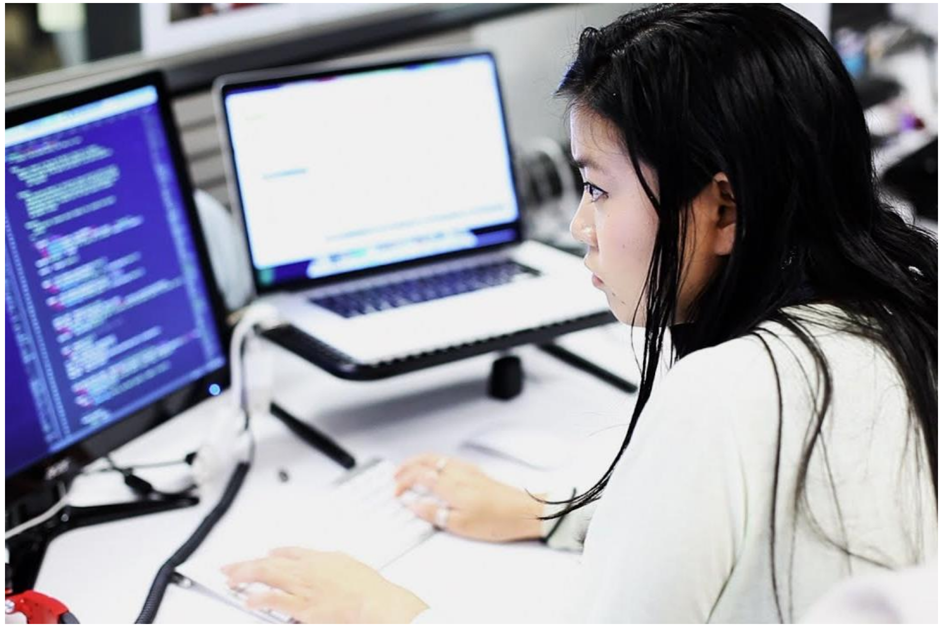


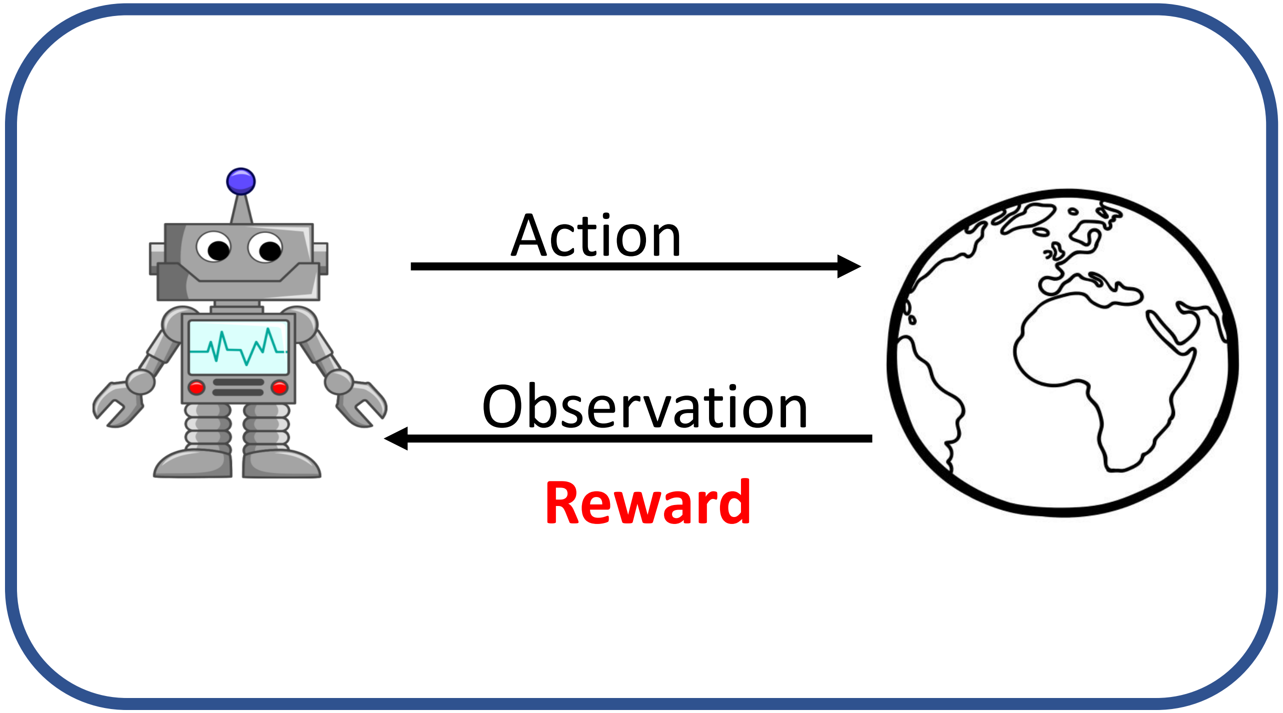


Reward engineering is hard!

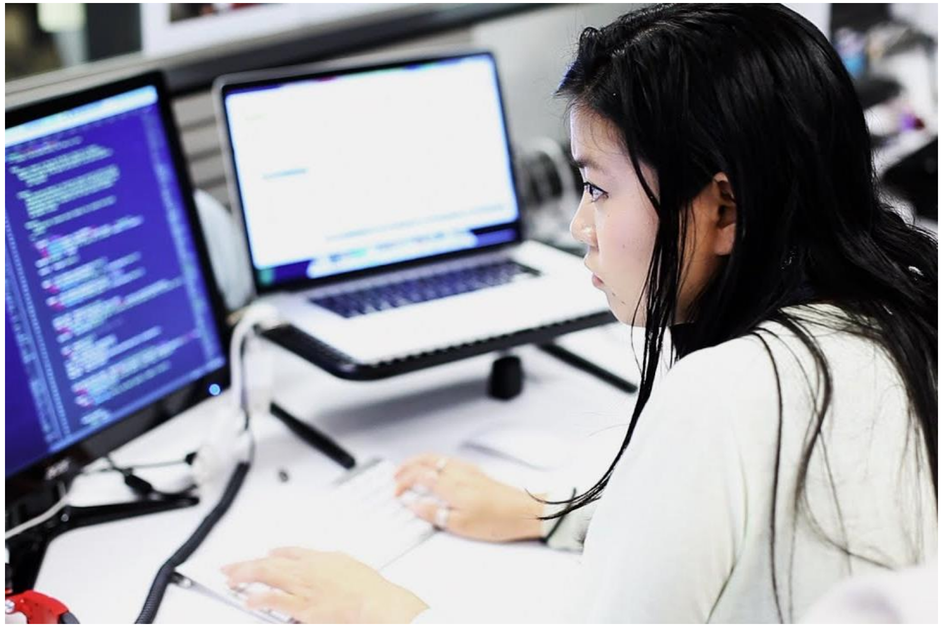


Reward engineering is hard!

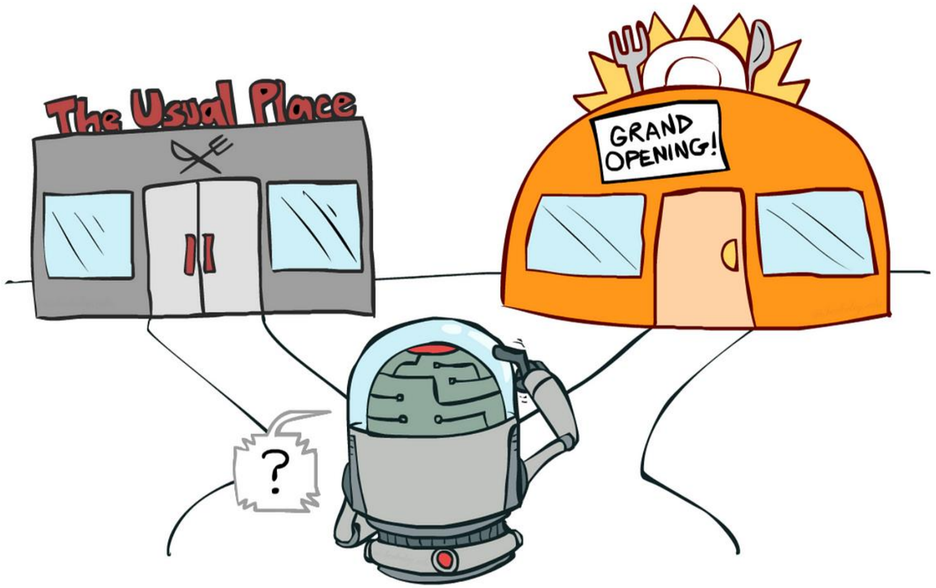




Reward engineering is hard!



Reinforcement learning is hard…even with a reward function!



Imitation Learning:

Learn a policy from examples of good behavior.

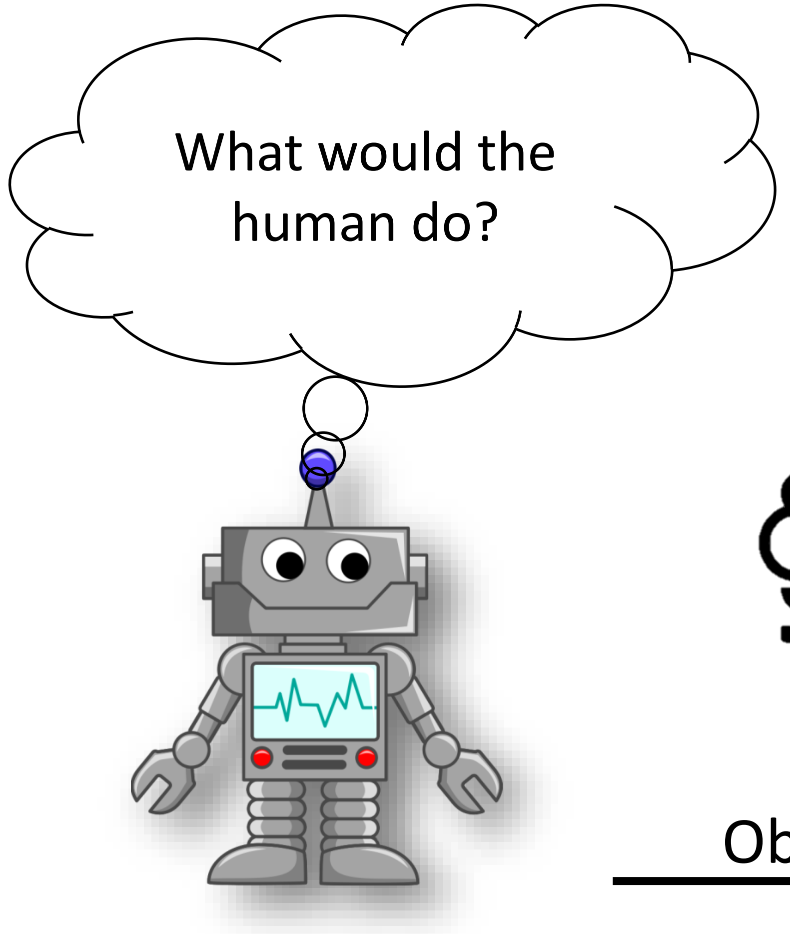
- Often showing is easier than telling.
- Alleviates problem of exploration.

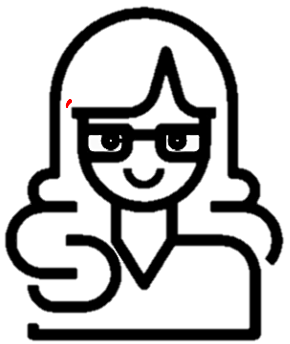
Behavioral Cloning

$D = \{(s_0, a_0)\ (s_1, a_1) \ldots \}$

disc actions   cont. actions

$s_0 \rightarrow \boxed{} \rightarrow a_0$ classifier, regression





Action →

← Observation

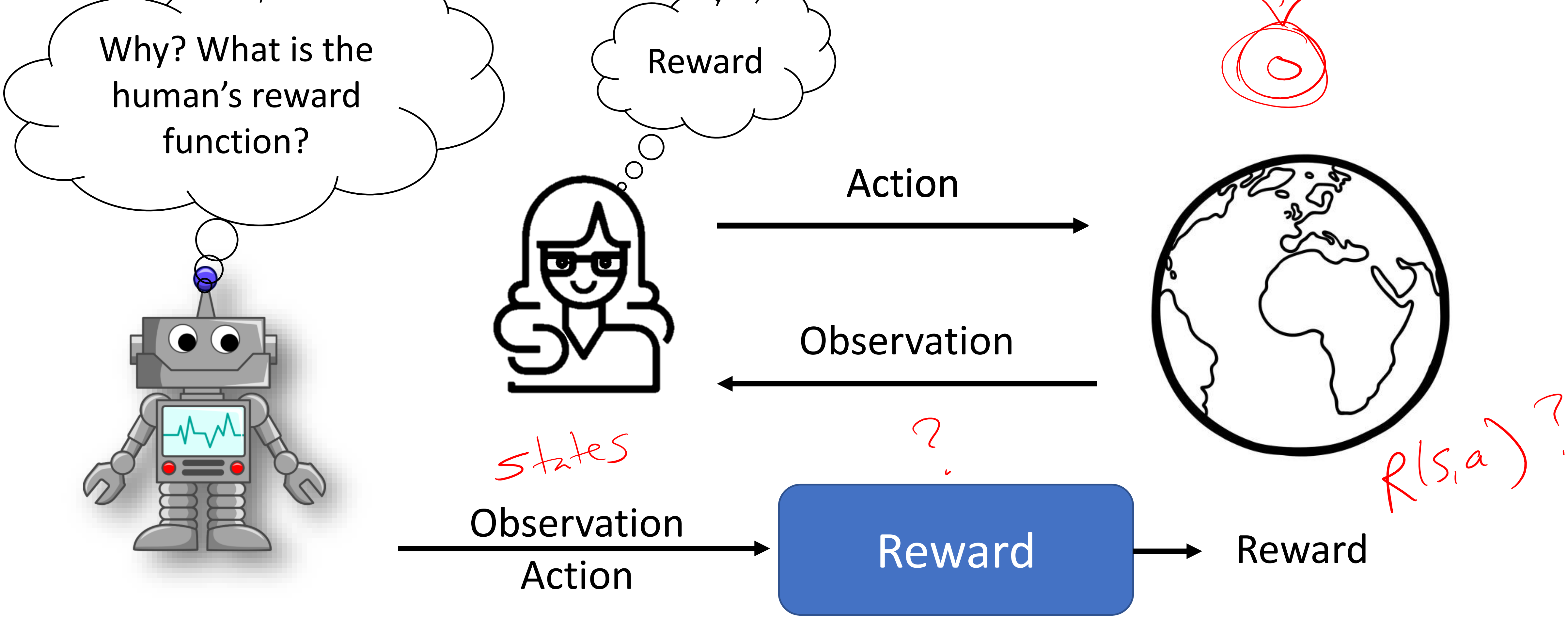
Observation → Policy π → Action

Inverse Reinforcement Learning

s

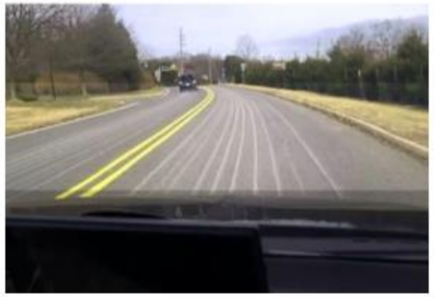
We'll talk about this on Thursday!

o

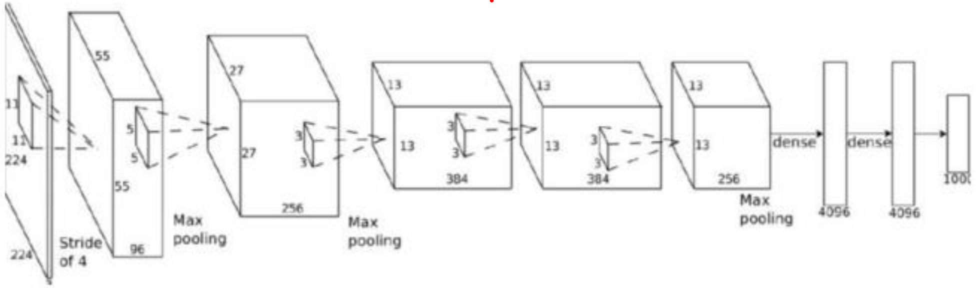


Imitation Learning via Behavioral Cloning

Deep



$\mathbf{o}_t$



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

$\mathbf{a}_t$

Supervised machine learning

input output

x, y $f(x) = y$

$\mathbf{o}_t$

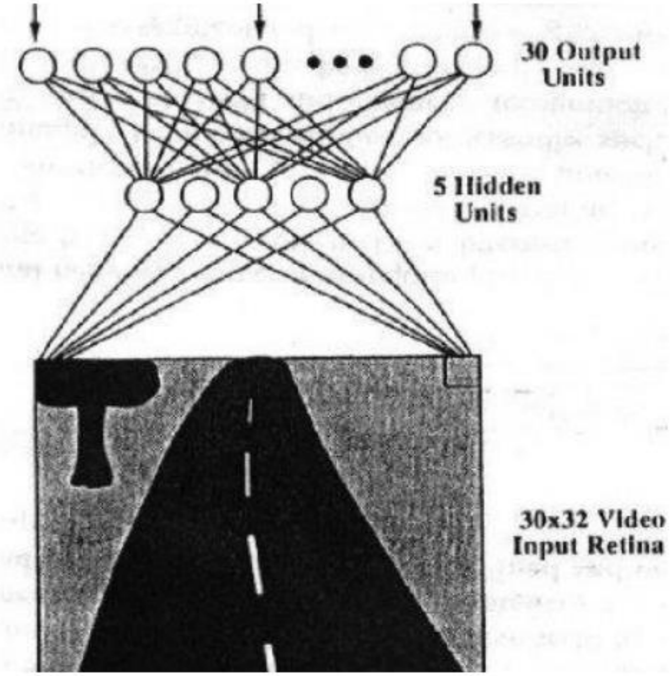
$\mathbf{a}_t$

→ training data → supervised learning

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

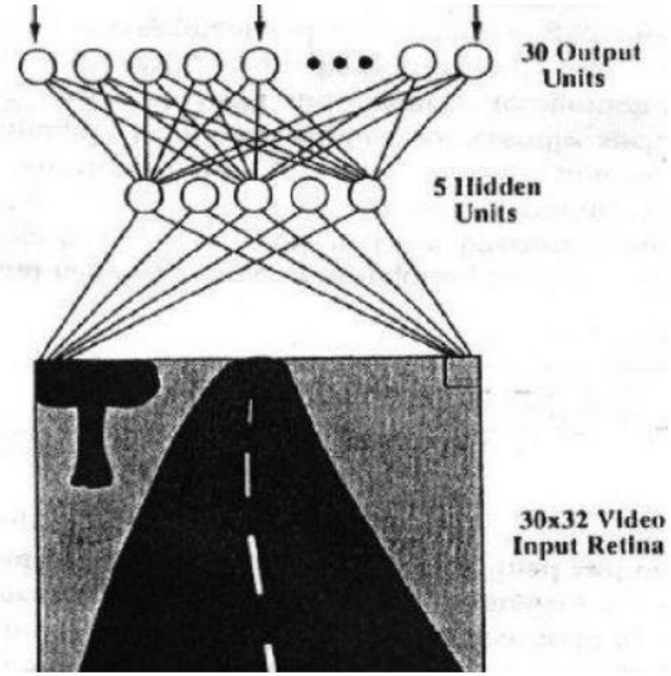
ALVINN: One of the first imitation learning systems

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



ALVINN: One of the first imitation learning systems

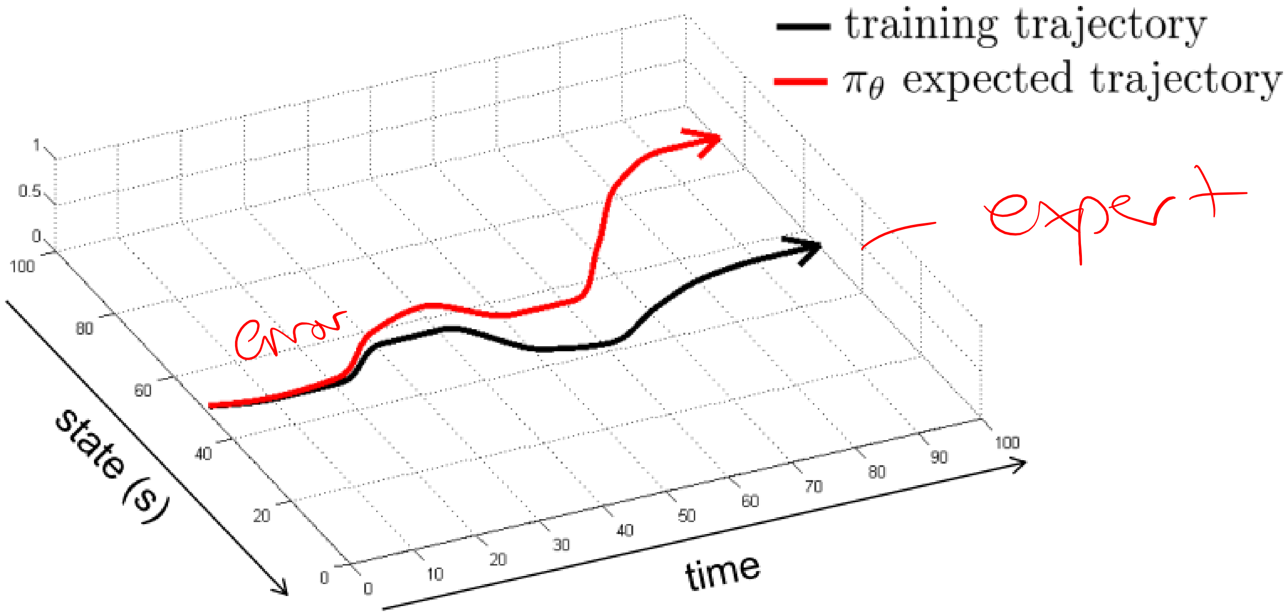
ALVINN: **A**utonomous **L**and **V**ehicle **I**n a **N**eural **N**etwork
1989



What could go wrong?

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

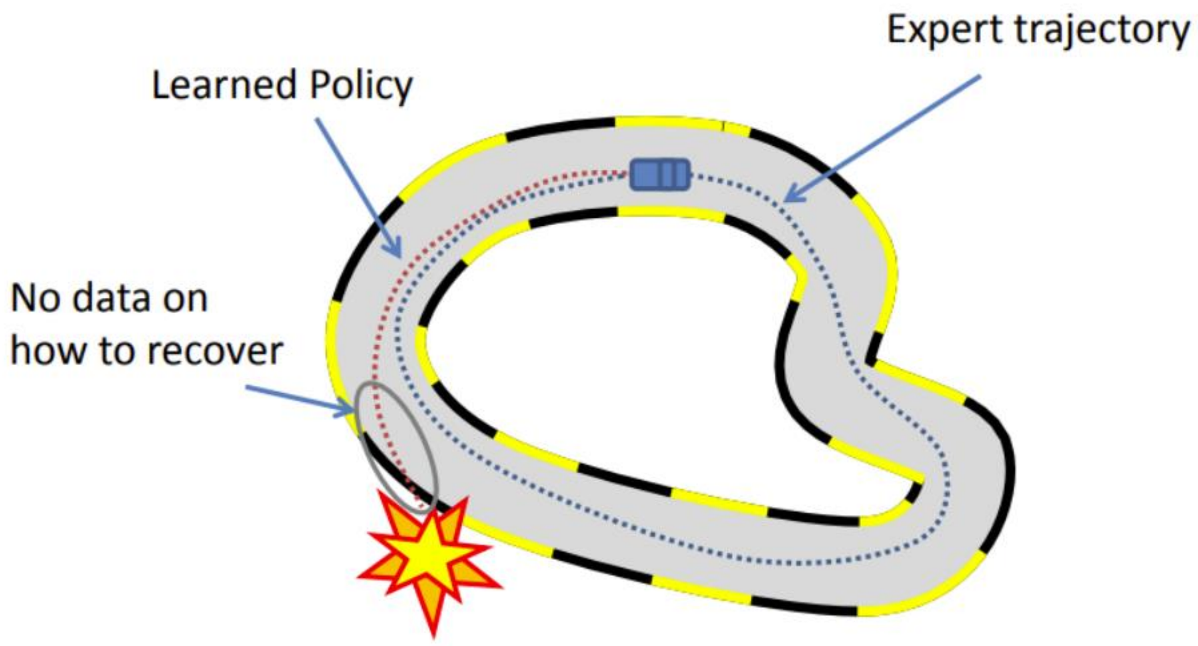
→ training data → supervised learning → $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$



Distribution Shift

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

Demonstrator/Expert policy π^*



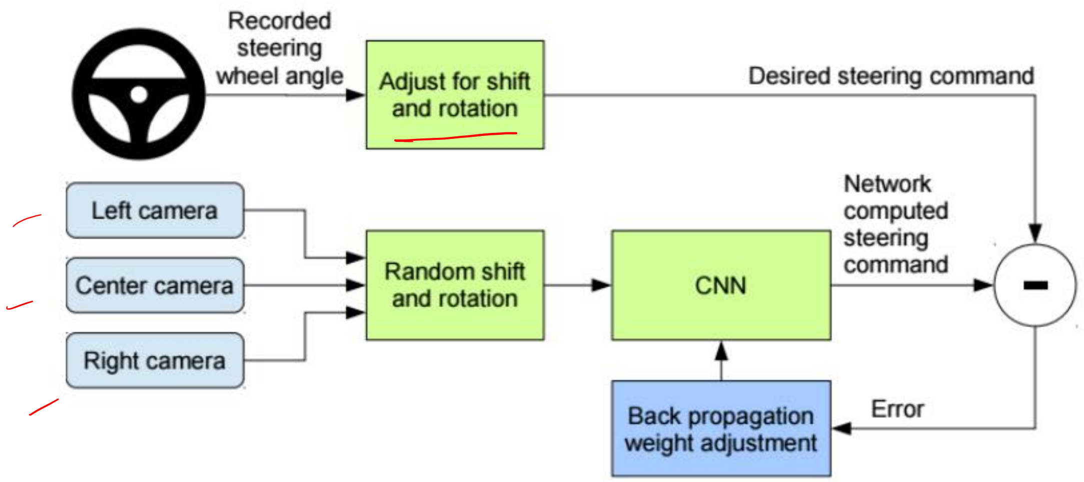
	Supervised Learning	Supervised Learning + Control
Train	$(x, y) \sim D$	$s \sim T(s, a, \pi^*(s))$
Test	$(x, y) \sim D$	$s \sim T(s, a, \pi(s))$

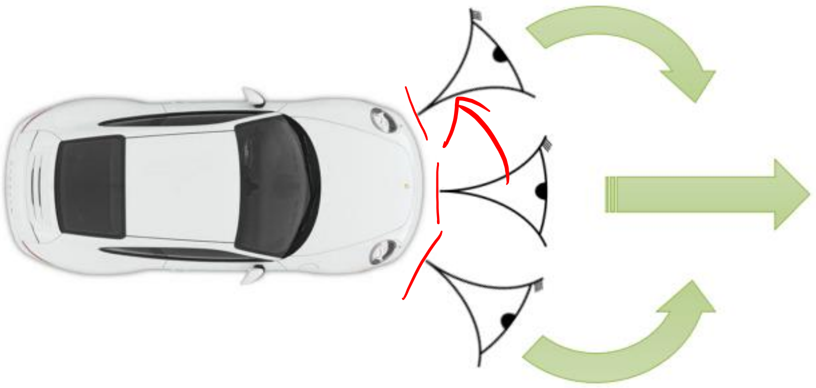
learned policy

But it still can work in practice…

Bojarski et al. ‘16, NVIDIA

How?

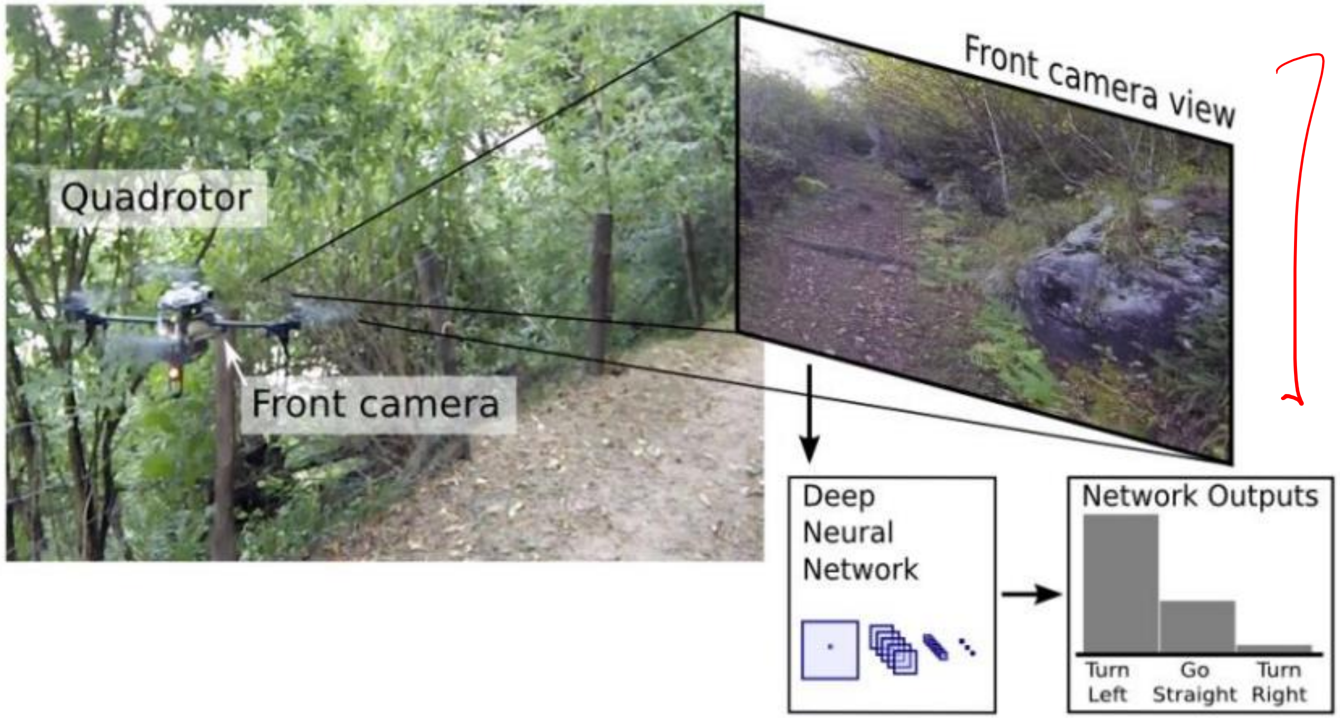




Bojarski et al. ‘16, NVIDIA

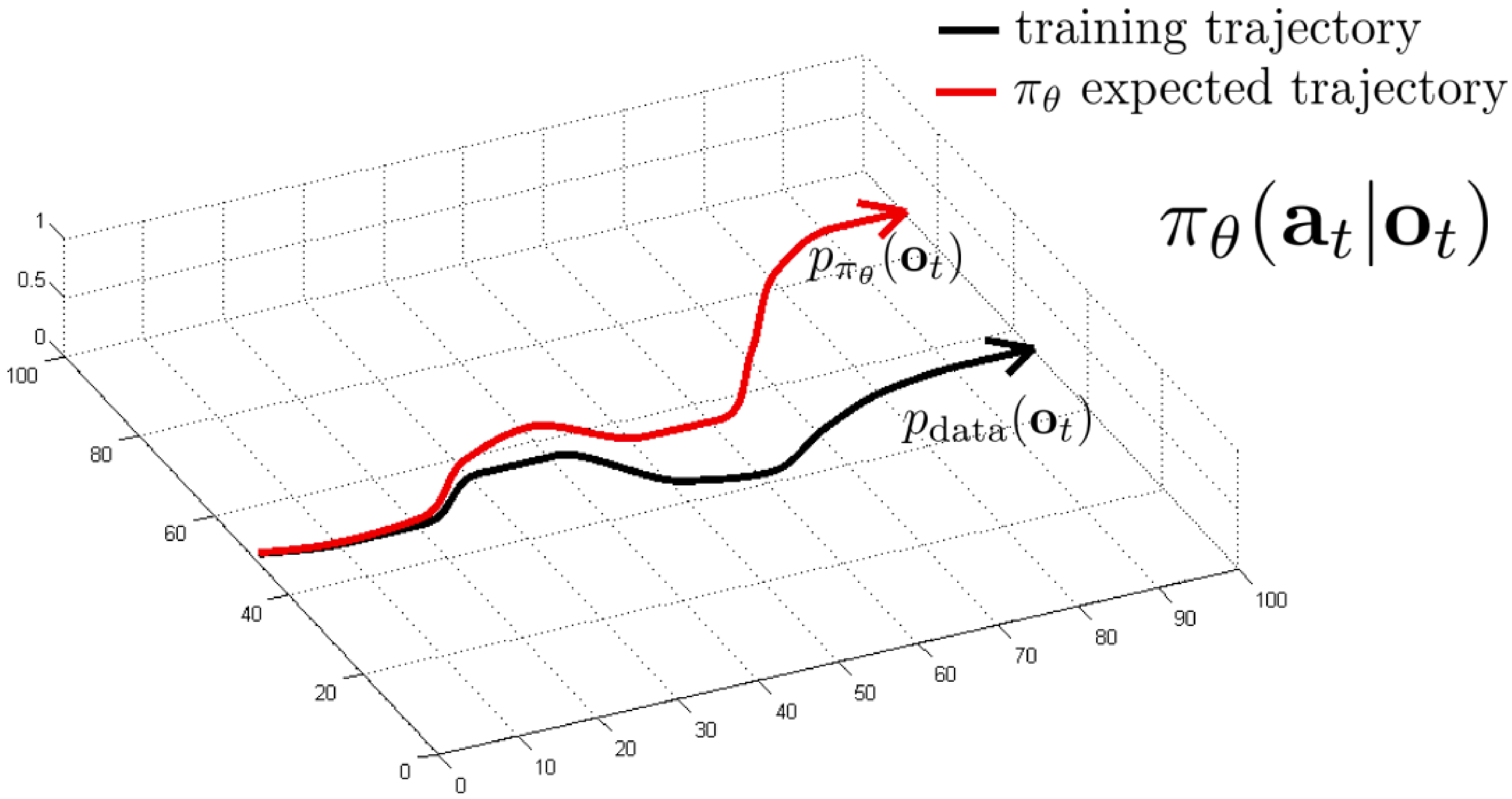
A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots

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]

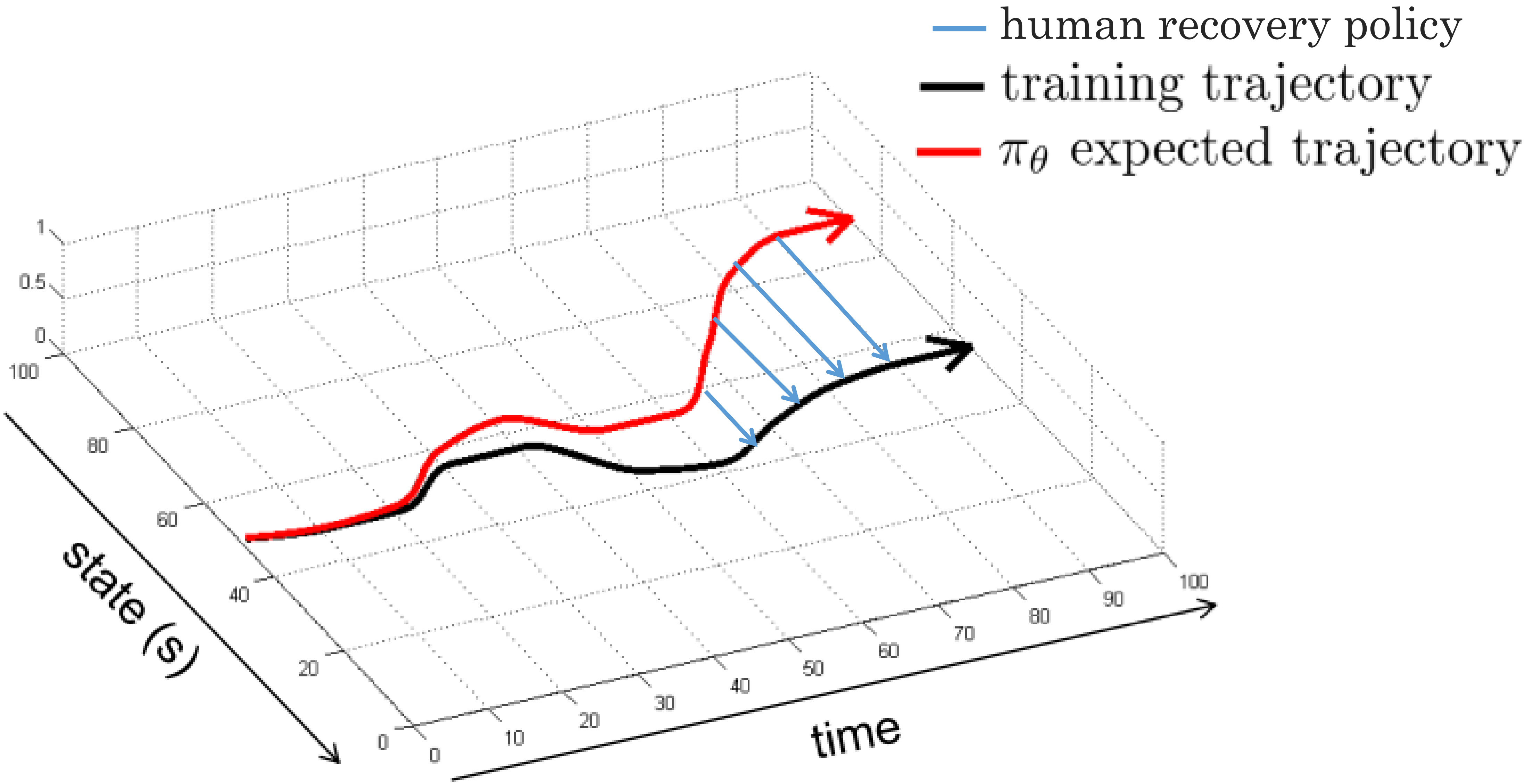



Deep Neural Network
Control Signal

Can we make it work more often?



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

π_θ learned policy

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 $\underline{\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)}$ $\quad O_{t=0} \cdots O_{t=T}$

but need labels $\underline{\mathbf{a}_t}$! $\quad a_0 \cdots a_T$

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

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 $p_{\text{data}}(\mathbf{o}_t)$

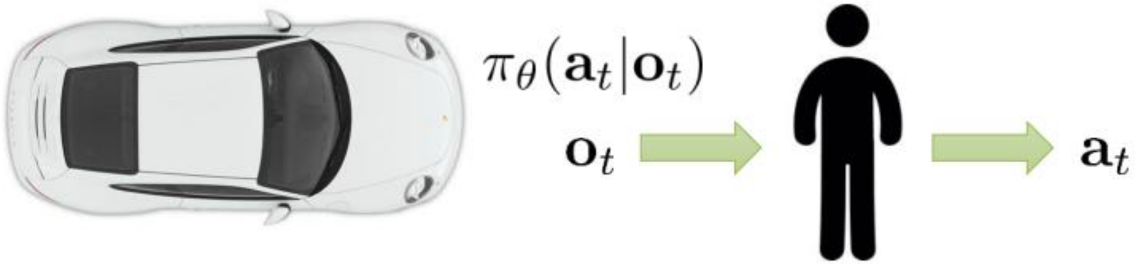
how? just run $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

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~~ to label $\mathcal{D}_\pi$ with actions $\mathbf{a}_t$
4. Aggregate: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_\pi$

supervisor

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?

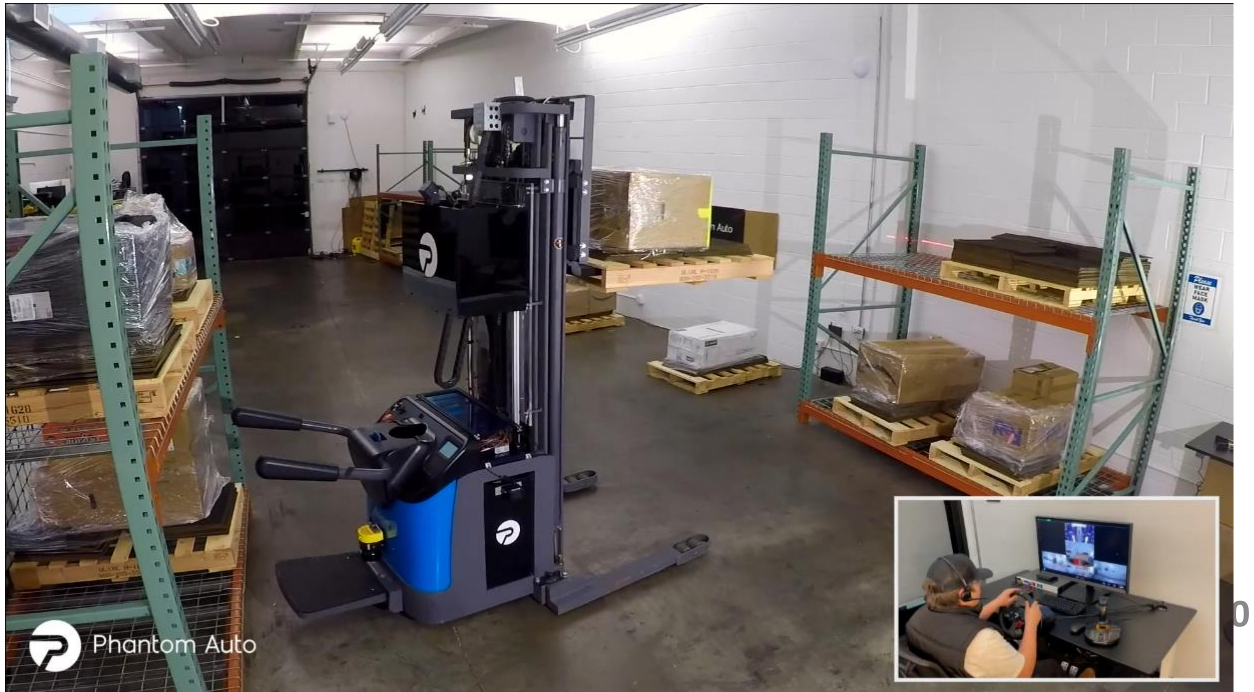
PLUS ONE ROBOTICS

WAYMO

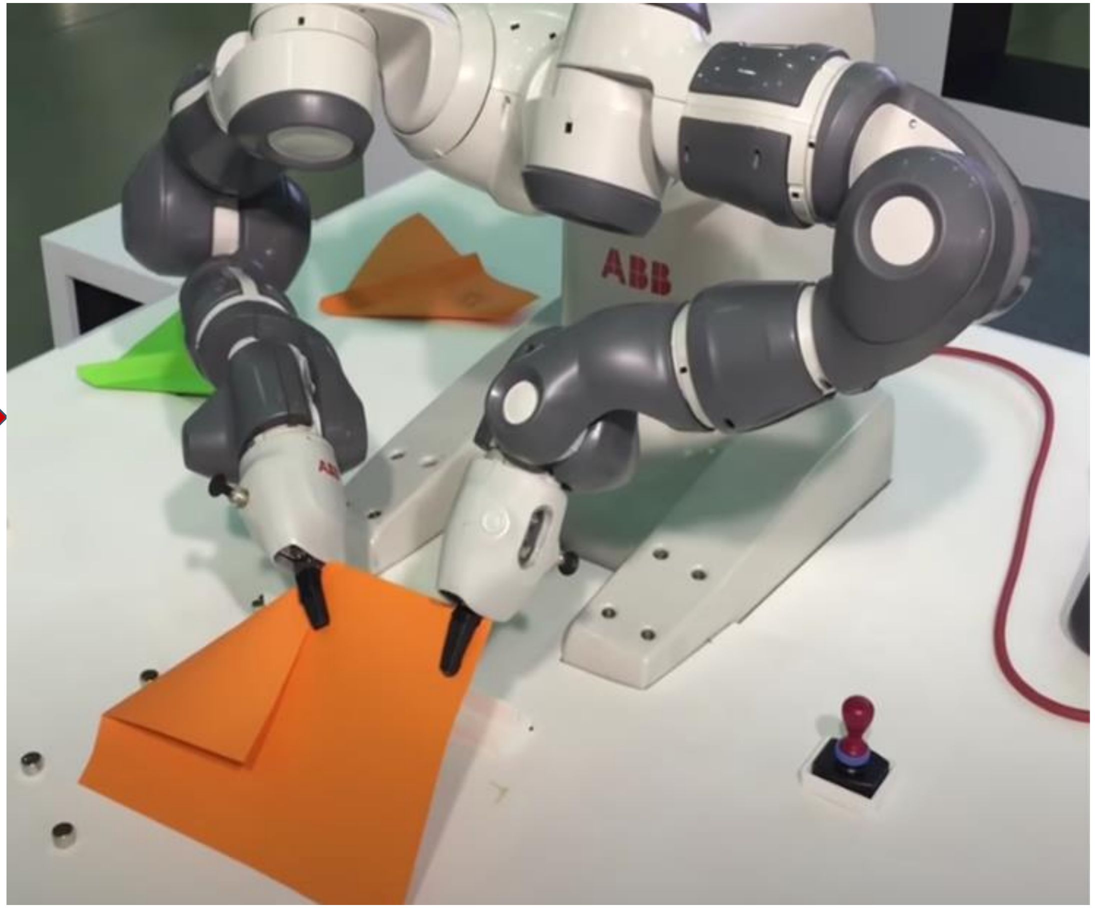
ZOOX

nimble

Phantom Auto



Interactive IL



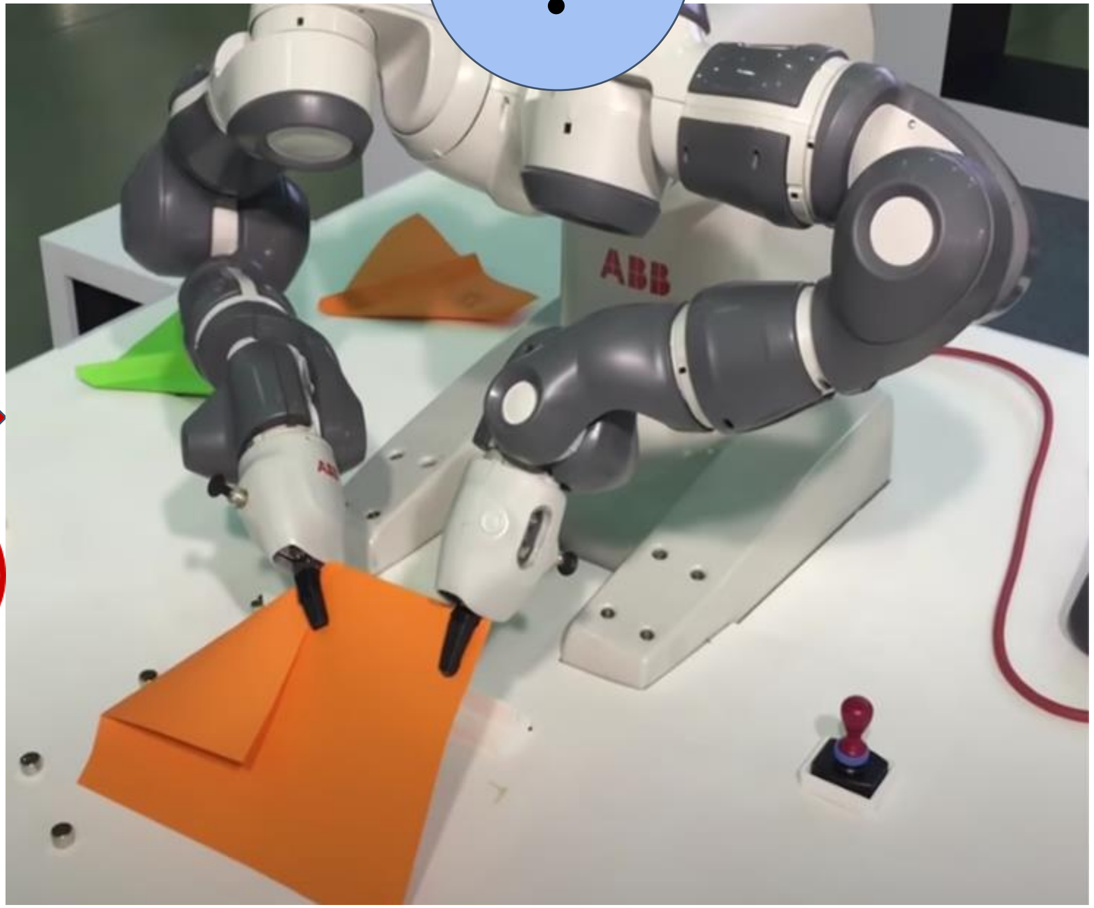
Interactive IL

$\pi_H(s)$

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

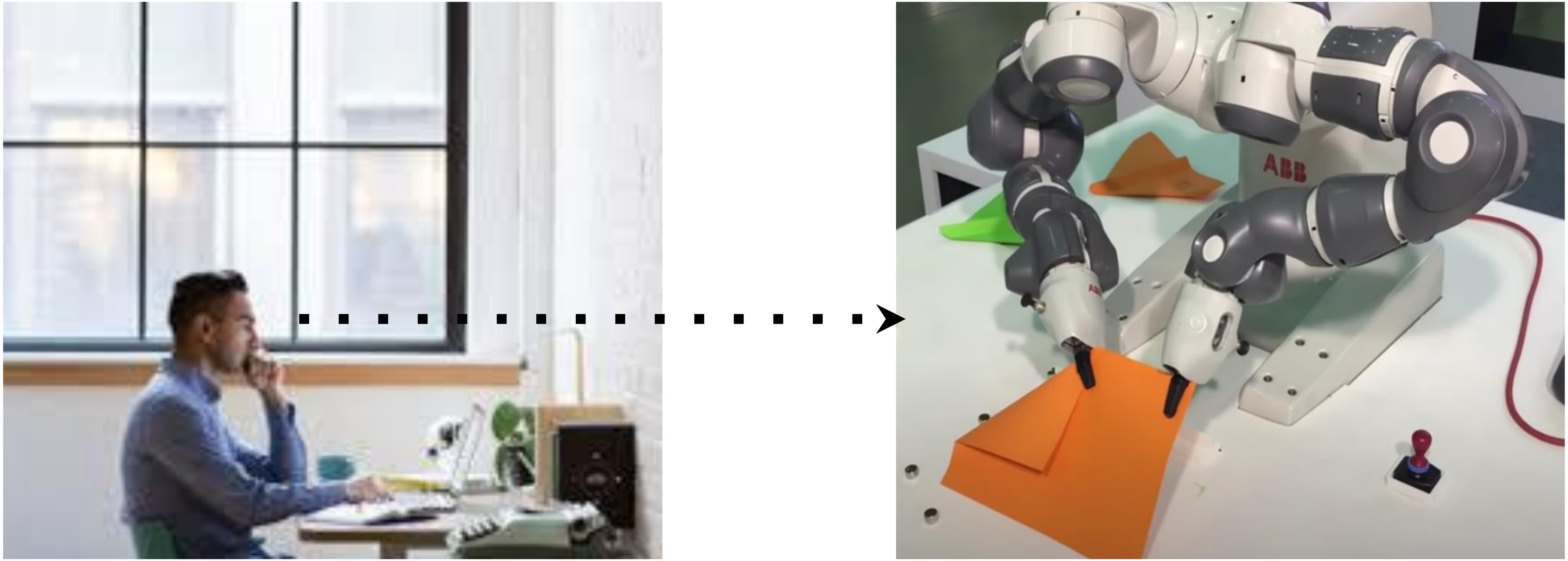
???

?



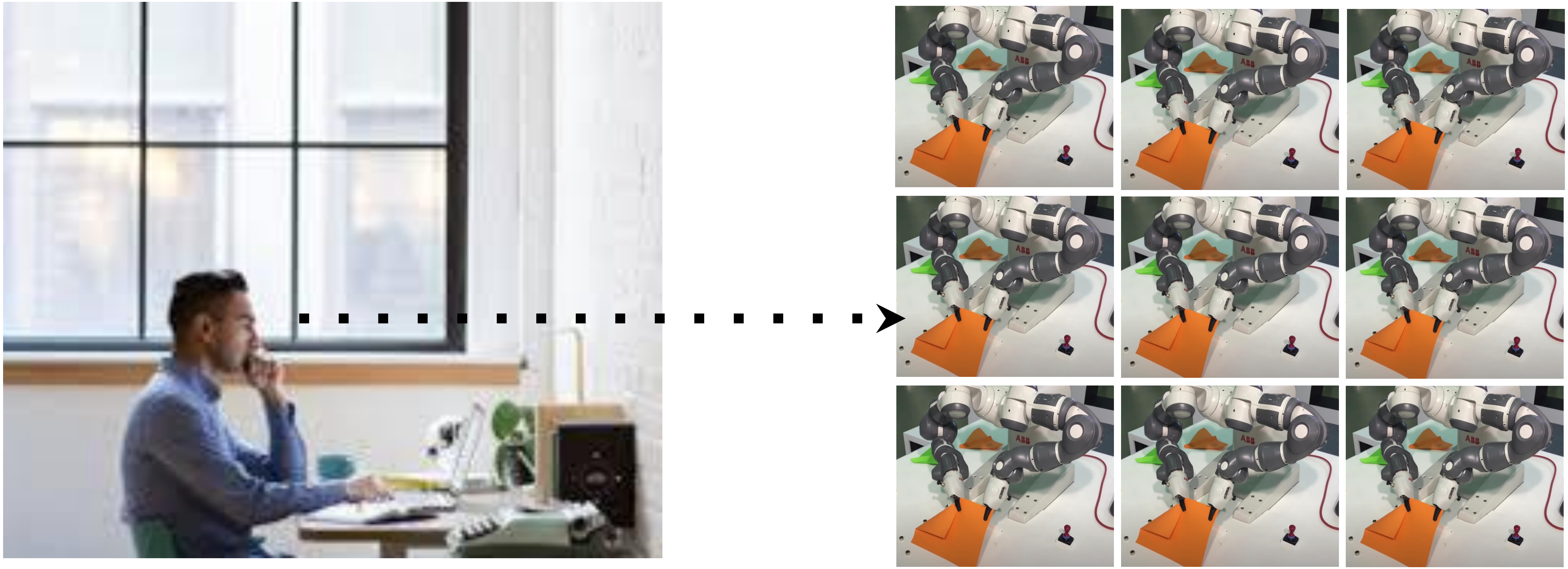
$\pi_R(s)$

Human-Gated Interactive IL



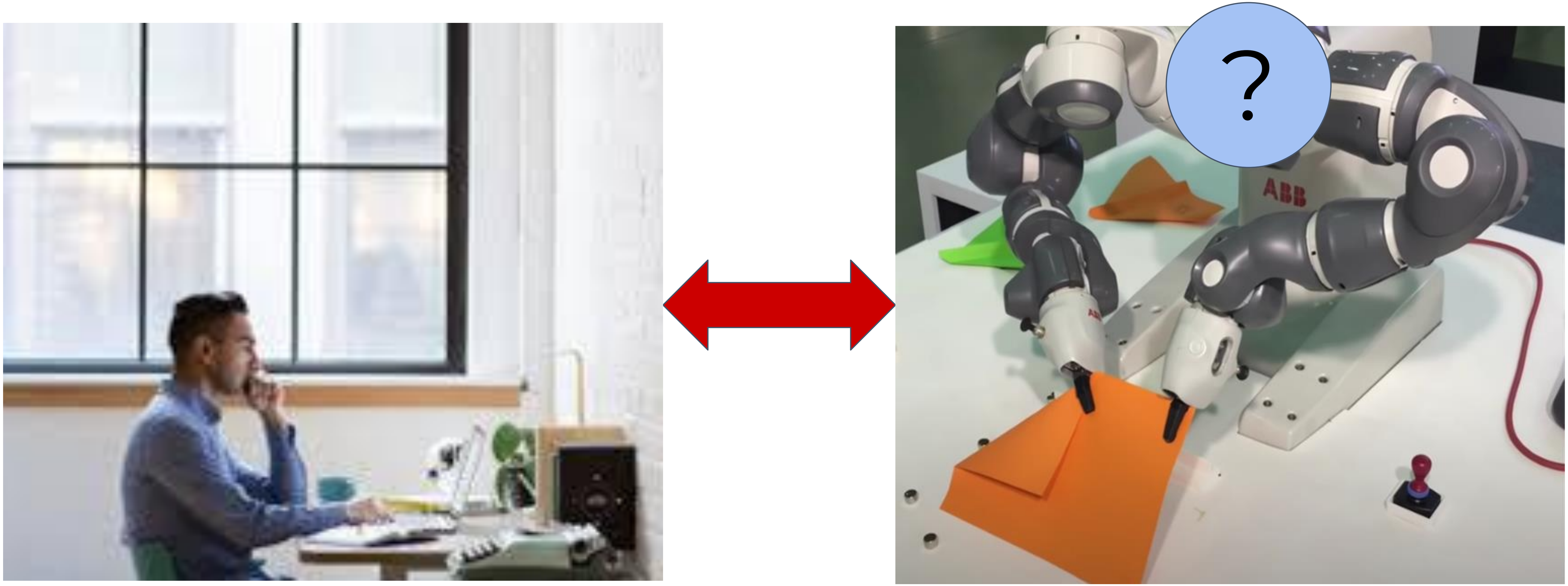
[3] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. HG-DAgger: Interactive Imitation Learning with Human Experts. ICRA 2019.

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.

Minimizing Supervisor Burden

- C = Number of context switches
- L = Latency of context switching
- I = Expected number of supervisor actions per intervention

$$B(\pi) \triangleq C(\pi) \cdot (L + I(\pi))$$

Ideally, we want

Loss of policy, policy error rate

$$\boxed{\begin{gathered}\pi = \arg\min_{\pi' \in \Pi} L(\pi_r') \\ s.t.\ B(\pi') \leq \Gamma_b\end{gathered}}$$

Minimizing Supervisor Burden

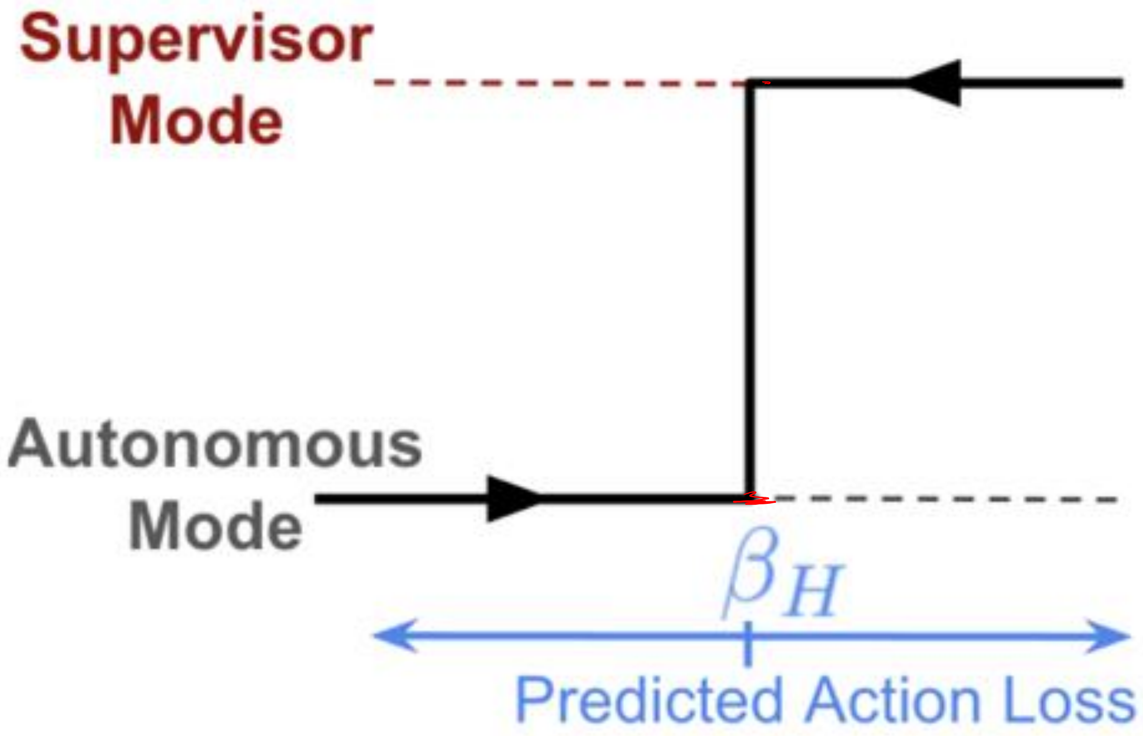
- C = number of context switches
- L = Latency of context switching
- I = expected number of supervisor actions per intervention

$$B(\pi) \triangleq \boldsymbol{C}(\pi) \cdot (L + I(\pi))$$

In practice, we approximate this by focusing on limiting the number of interventions (number of context switches)

$$\boxed{\begin{gathered}\pi = \arg\min_{\pi' \in \Pi} L(\pi'_r) \\ s.t.\ B(\pi') \leq \Gamma_b\end{gathered}}$$

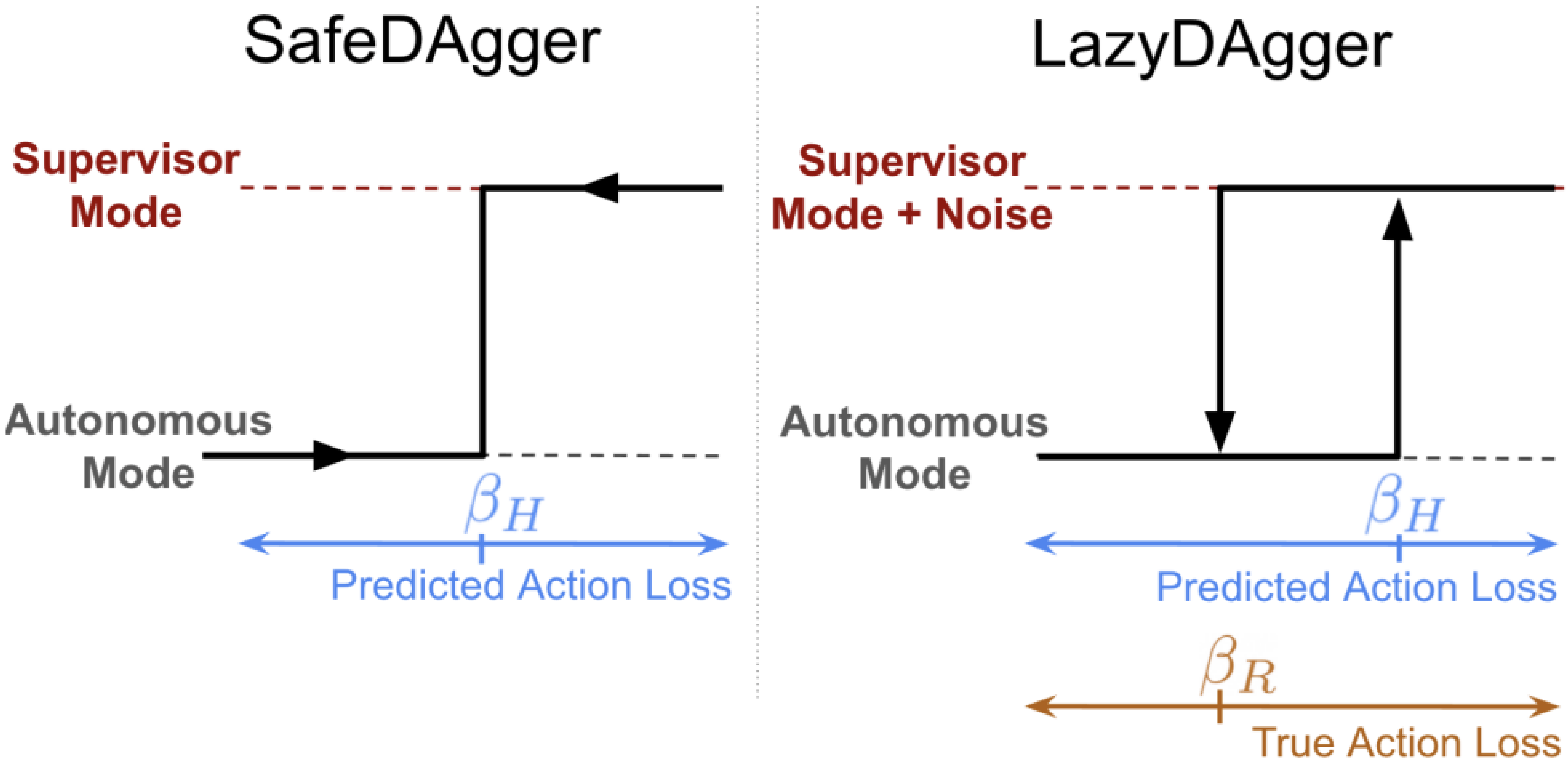
SafeDAgger



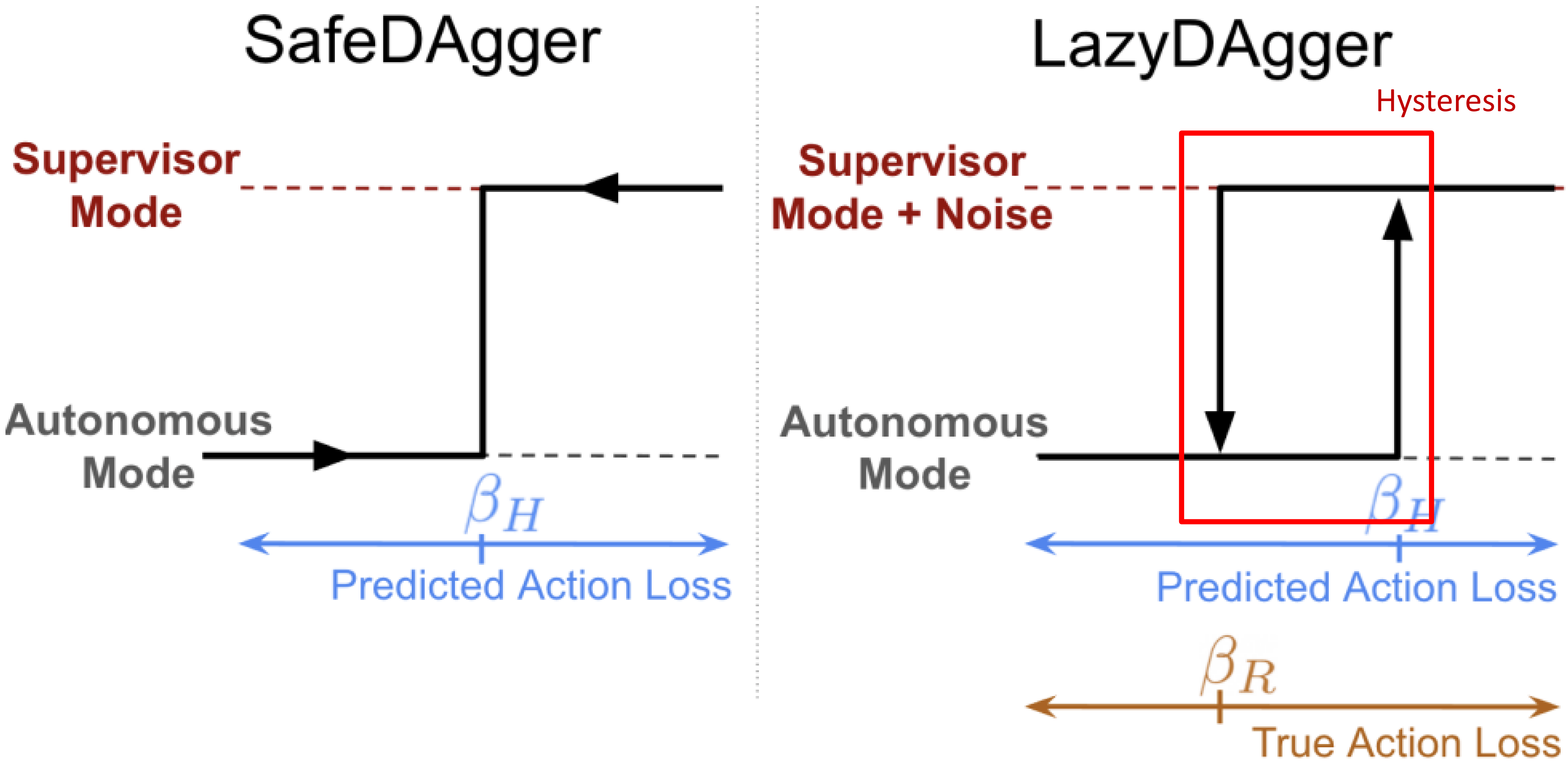
Predicted action loss = predicted difference between human and robot action.

Trained using held-out set of data from human.

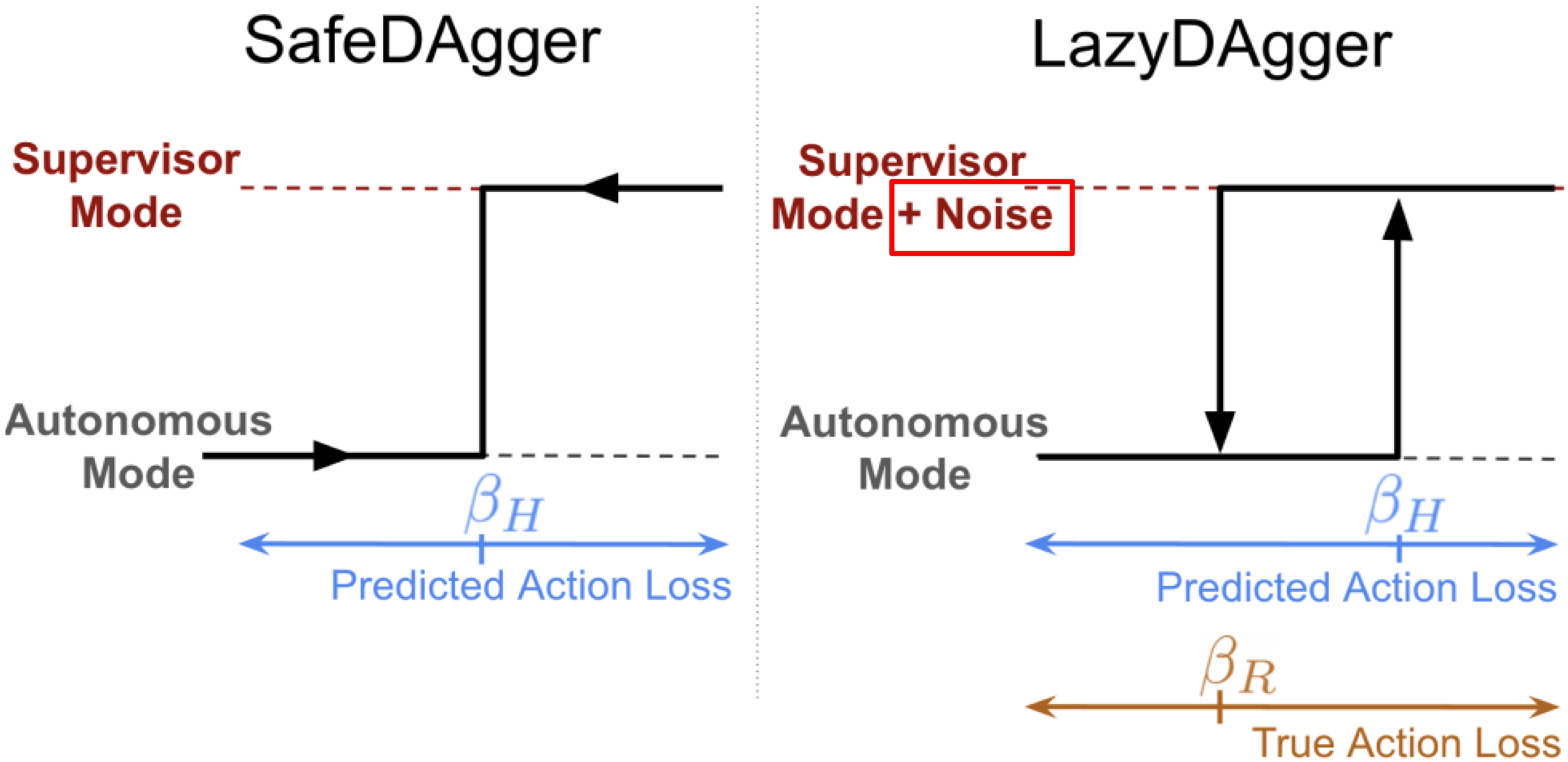
J. Zhang, K. Cho. Query-Efficient Imitation Learning for End-to-End Autonomous Driving. AAAI 2017.



Hoque et al. 2021.

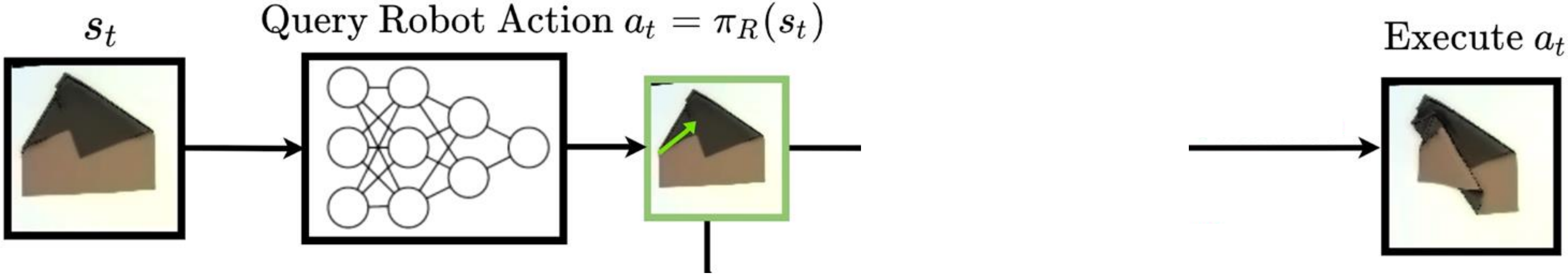


Hoque et al. 2021.



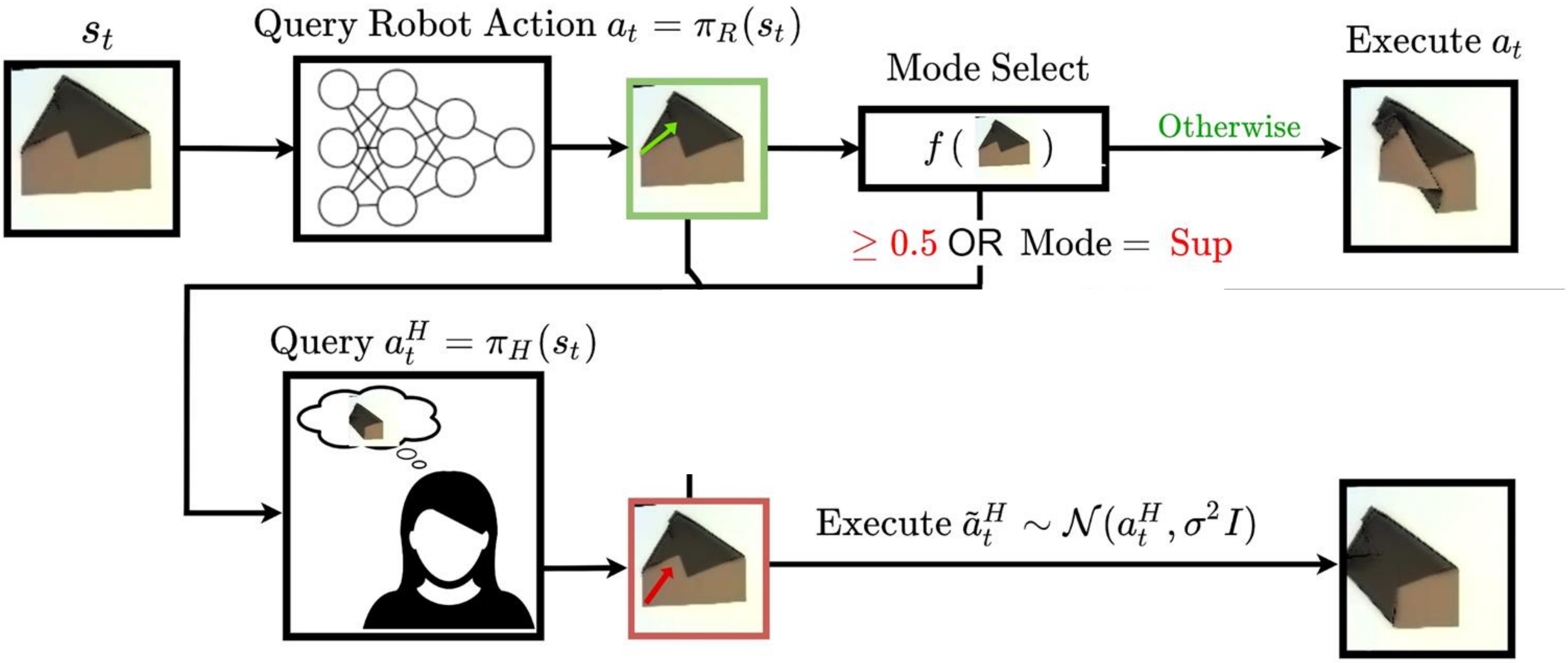
[5] M. Laskey, J. Lee, R. Fox, A. Dragan, K. Goldberg. DART: Noise Injection for Robust Imitation Learning. CoRL 2017.

LazyDAgger



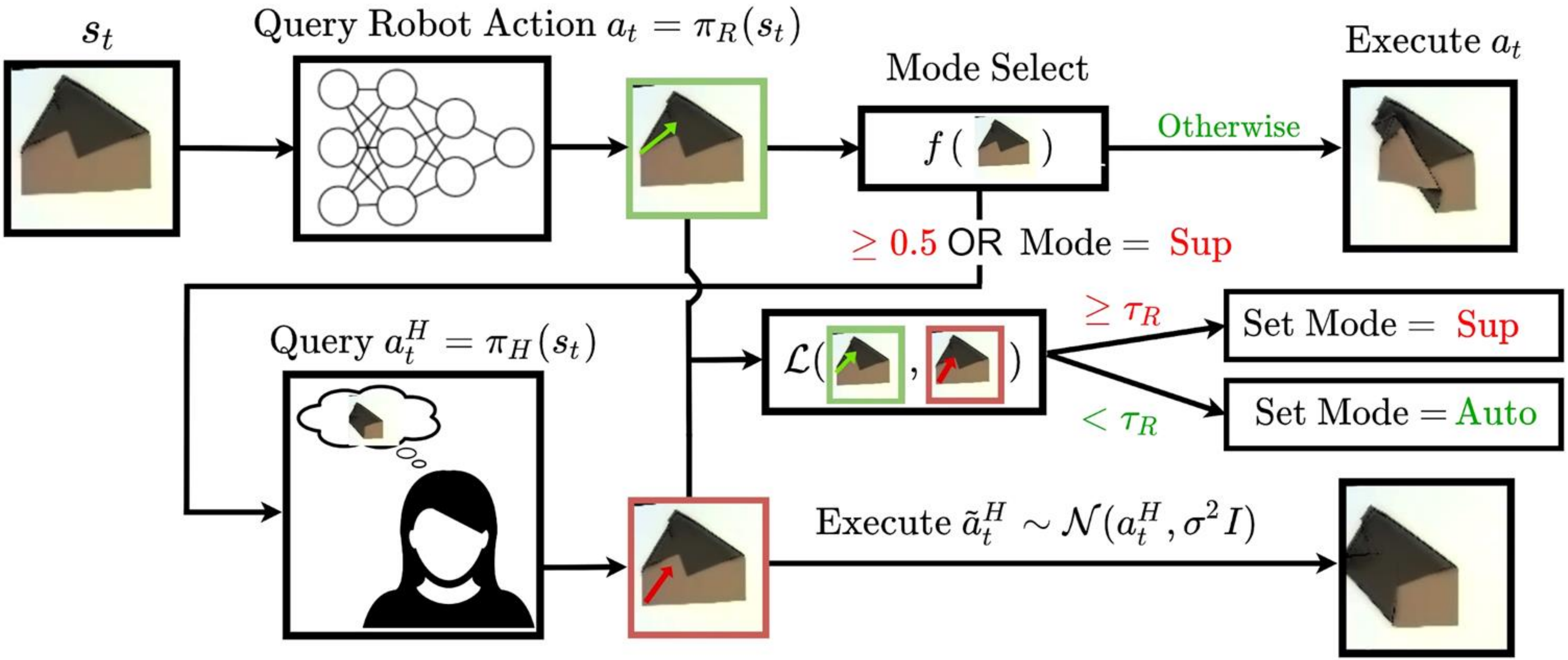
Hoque et al. 2021.

LazyDAgger



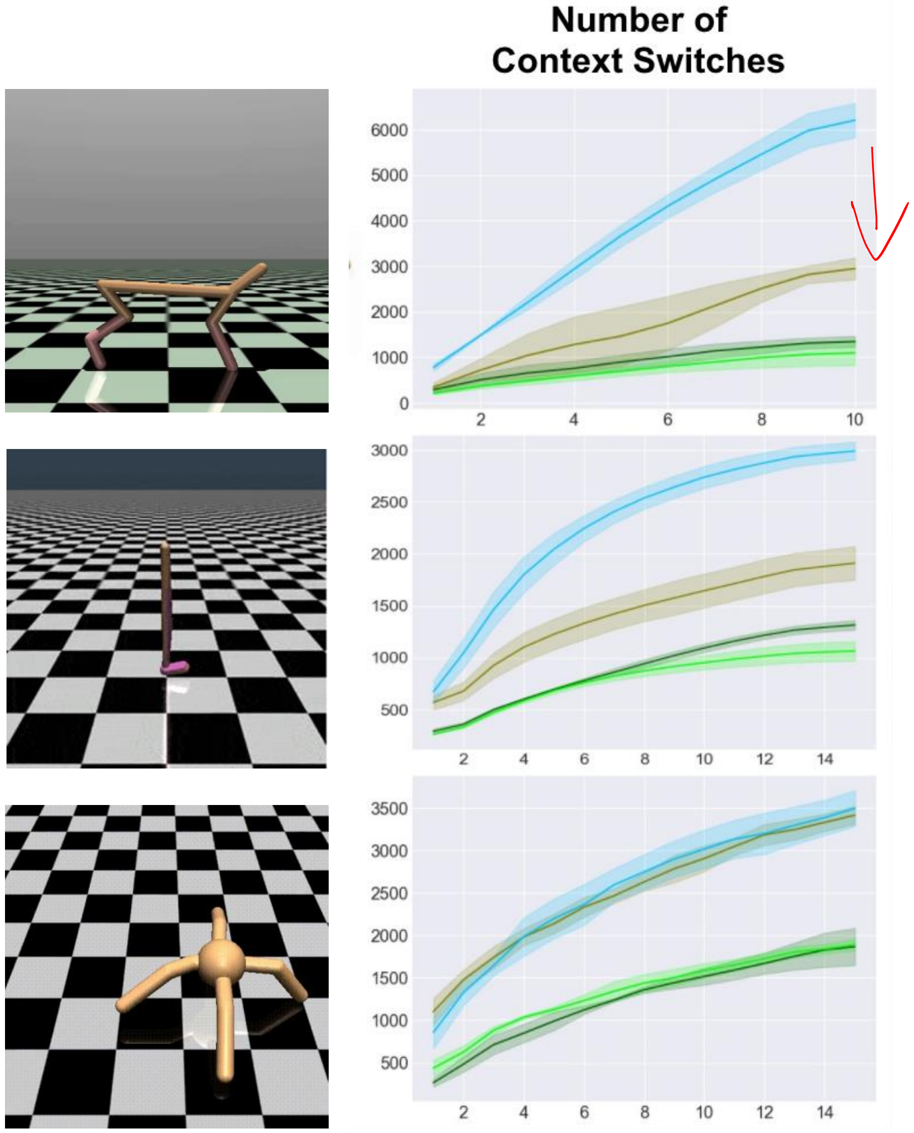
s_t
Query Robot Action $a_t = \pi_R(s_t)$
Mode Select
$f($ $)$
Otherwise
Execute a_t
≥ 0.5 OR Mode $=$ Sup
Query $a_t^H = \pi_H(s_t)$
Execute $\tilde{a}_t^H \sim \mathcal{N}(a_t^H, \sigma^2 I)$

LazyDAgger



s_t
Query Robot Action $a_t = \pi_R(s_t)$
Mode Select
$f(\ )$
Otherwise
Execute a_t
≥ 0.5 OR Mode = Sup
$\geq \tau_R$
Set Mode = Sup
$\mathcal{L}(\ ,\ )$
$< \tau_R$
Set Mode = Auto
Query $a_t^H = \pi_H(s_t)$
Execute $\tilde{a}_t^H \sim \mathcal{N}(a_t^H, \sigma^2 I)$

Simulation Experiments



Context Switching Reduction

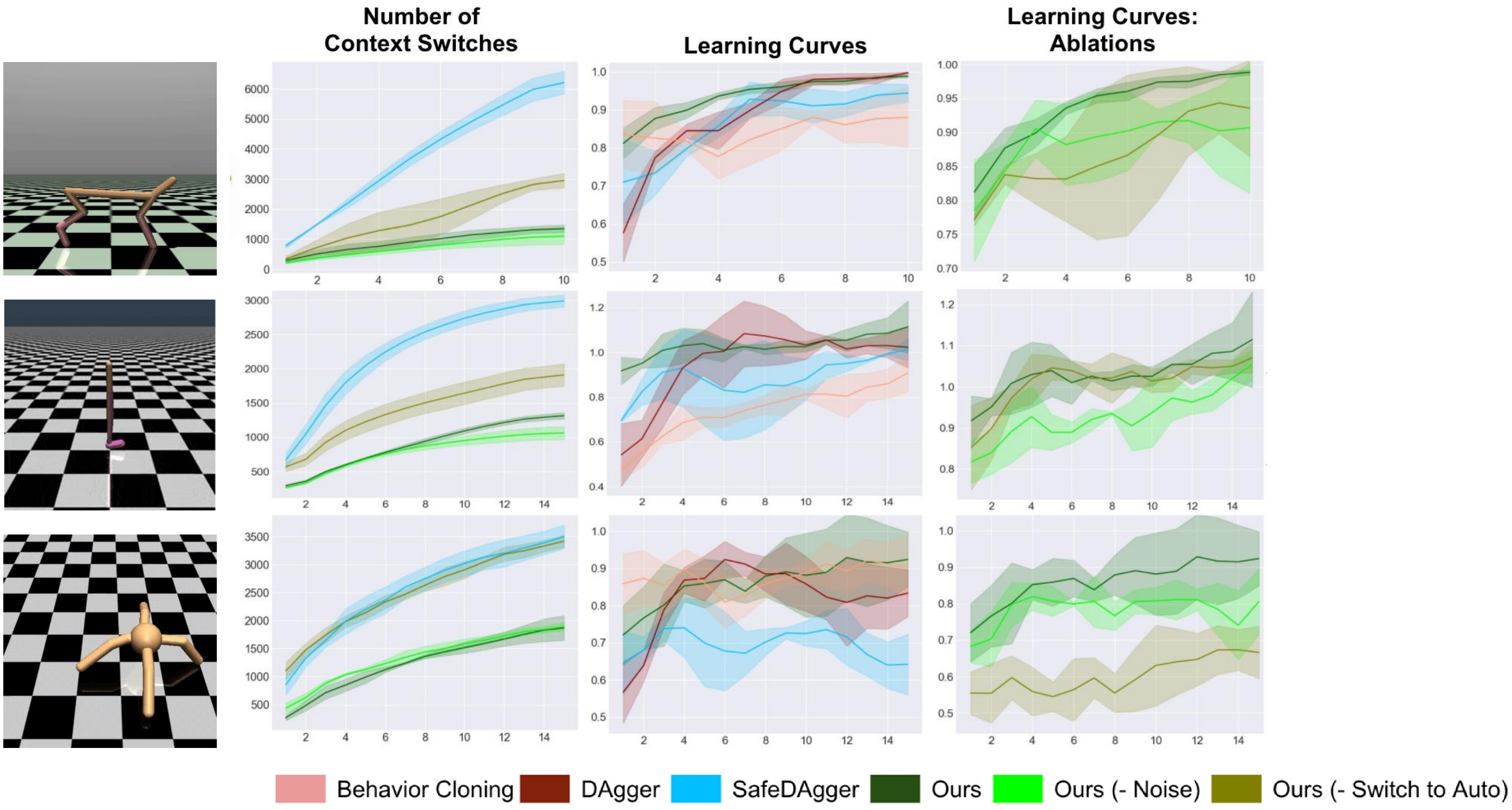
79%

56%

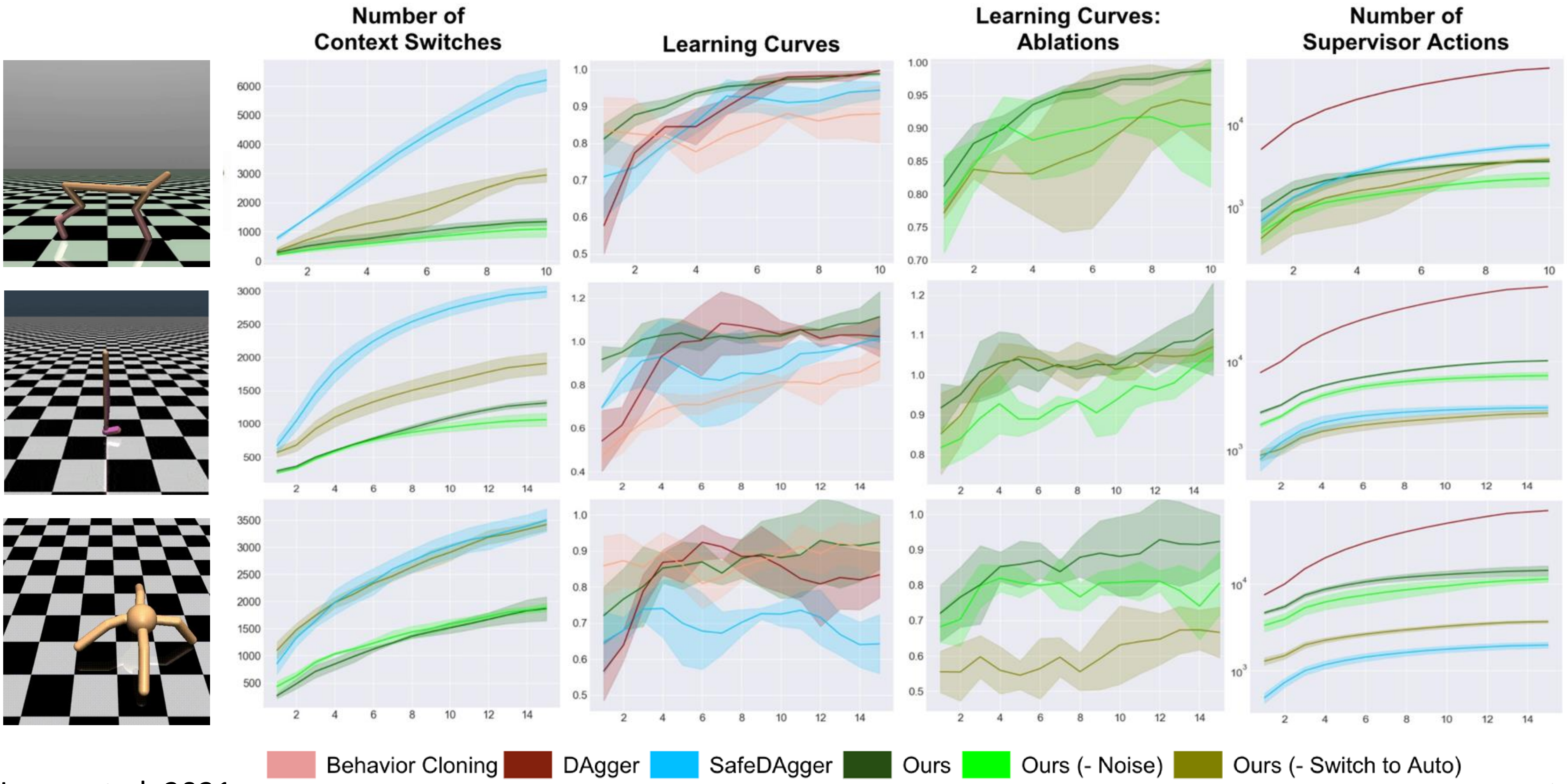
46%

Behavior Cloning DAgger SafeDAgger Ours Ours (- Noise) Ours (- Switch to Auto)

Simulation Experiments



Simulation Experiments



Hoque et al. 2021.

$$L = \frac{\text{(Time to perform one context switch)}}{\text{(Time to perform one action)}}$$

$$\boldsymbol{L} = \frac{\textbf{(Time to perform one context switch)}}{\textbf{(Time to perform one action)}}$$

$$B(\pi) \triangleq C(\pi) \cdot (L + I(\pi))$$

$$L = \frac{\text{(Time to perform one context switch)}}{\text{(Time to perform one action)}}$$

$$B(\pi) \triangleq C(\pi) \cdot (L + I(\pi))$$

C = 20 switches
I = 10 actions
B = 20L + 100

$$L = \frac{\textbf{(Time to perform one context switch)}}{\textbf{(Time to perform one action)}}$$

$$B(\pi) \triangleq C(\pi) \cdot (L + I(\pi))$$

C = 20 switches
I = 2 actions
B = 20L + 40

C = 4 switches
D = 20 actions
B = 4*L* + 80

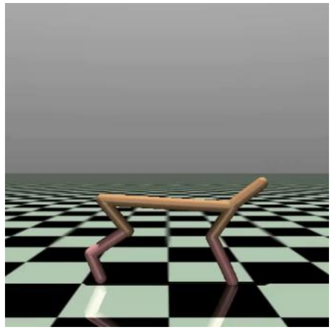
$$\boldsymbol{L} = \frac{\textbf{(Time to perform one context switch)}}{\textbf{(Time to perform one action)}}$$

$$B(\pi) \triangleq C(\pi) \cdot (L + I(\pi))$$

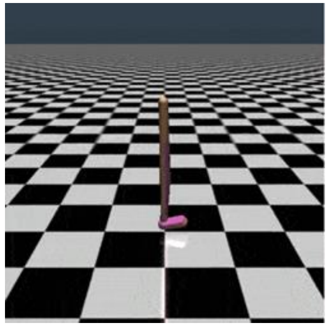
Define "cut-off latency" $L^* \geq 0,$ as the minimum value such that

B(SafeDAgger) > B(LazyDAgger) for all $L \geq L^*$

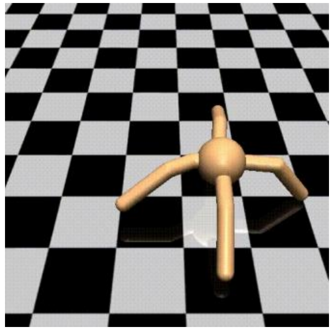
Simulation Experiments



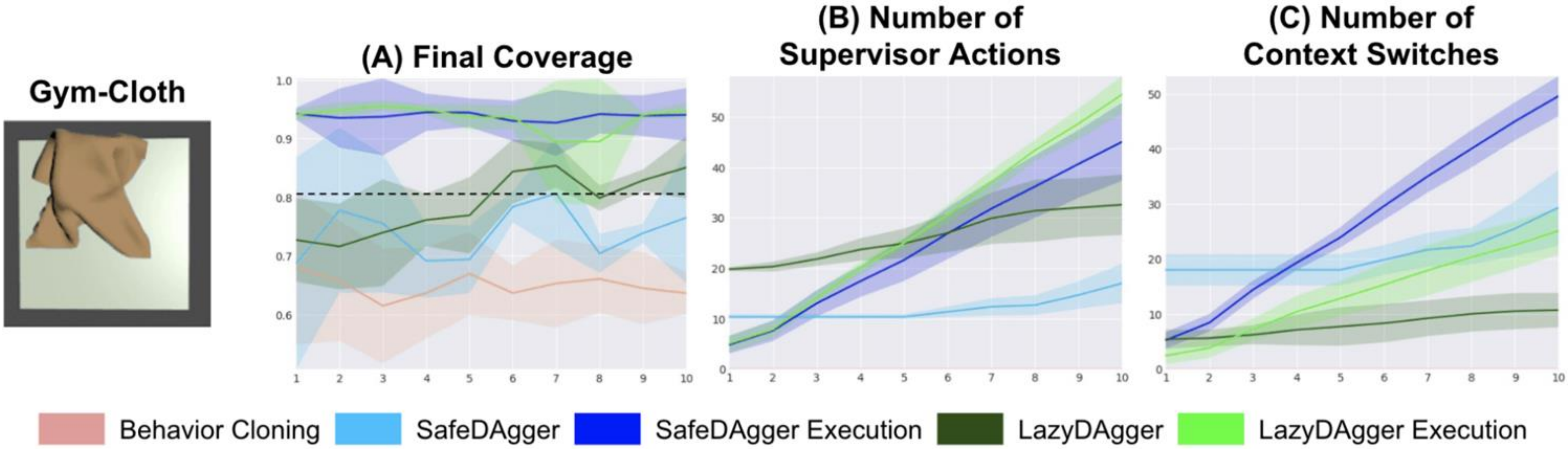
$L^* = 0.0$

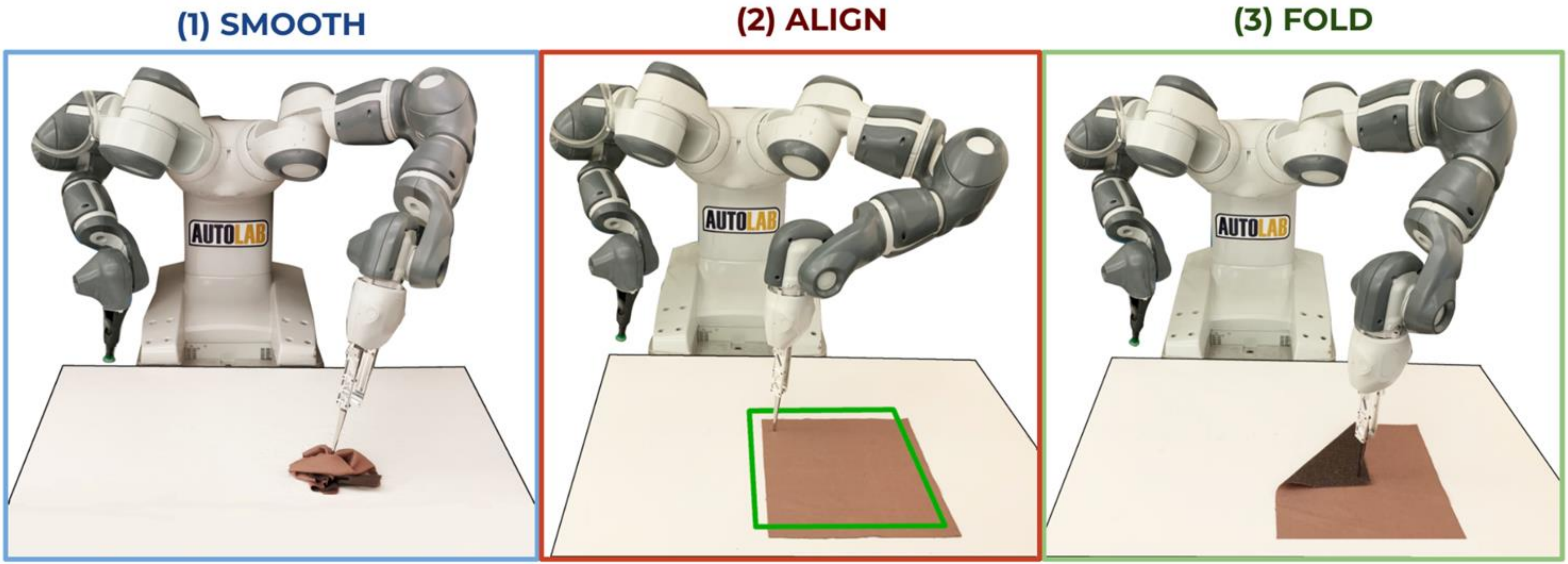


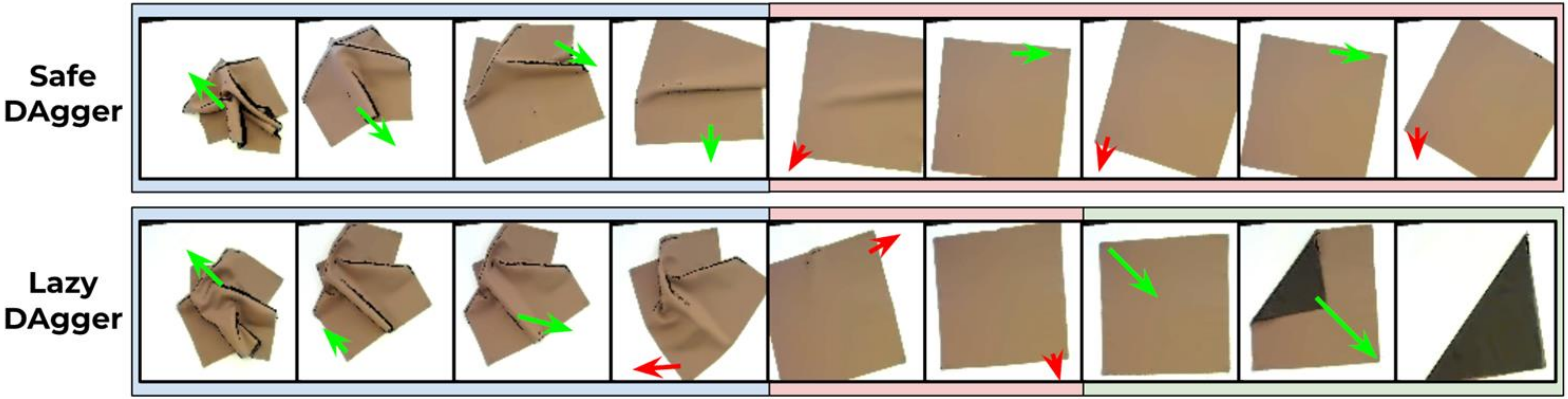
$L^* = 4.3$

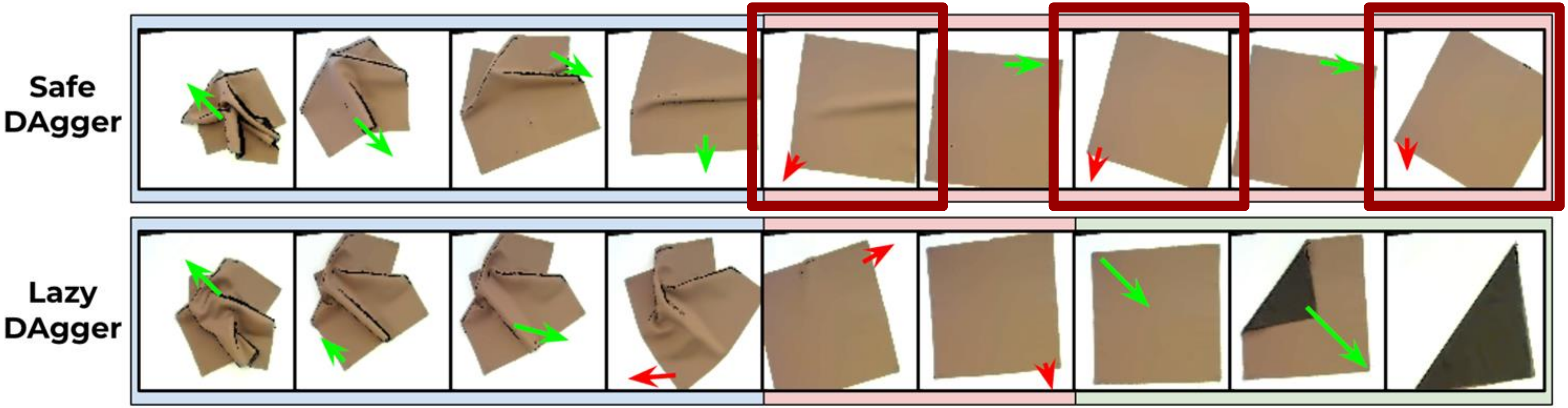


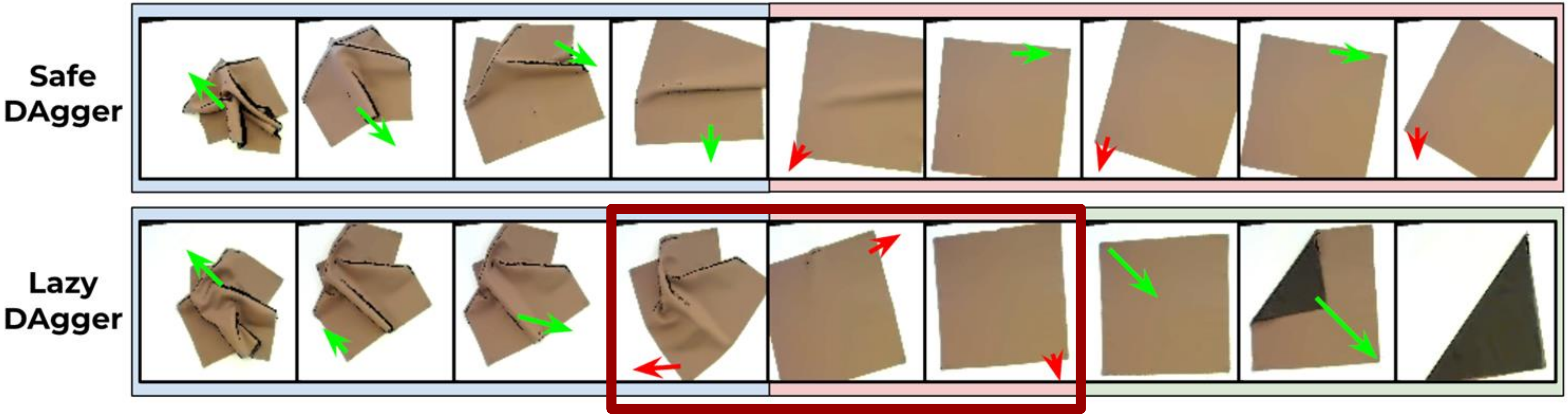
$L^* = 7.6$


Gym-Cloth
(A) Final Coverage
(B) Number of Supervisor Actions
(C) Number of Context Switches
Behavior Cloning
SafeDAgger
SafeDAgger Execution
LazyDAgger
LazyDAgger Execution


(1) SMOOTH
(2) ALIGN
(3) FOLD
AUTOLAB
AUTOLAB
AUTOLAB

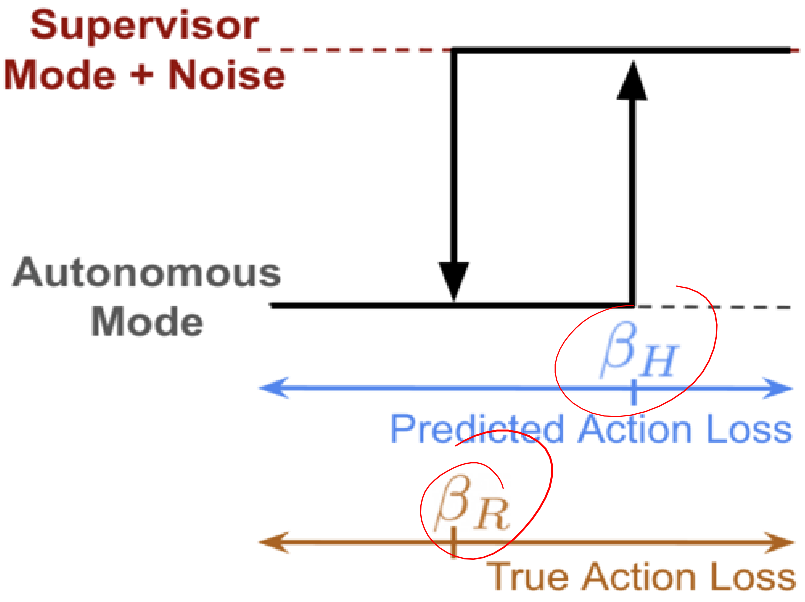

Safe DAgger
Lazy DAgger

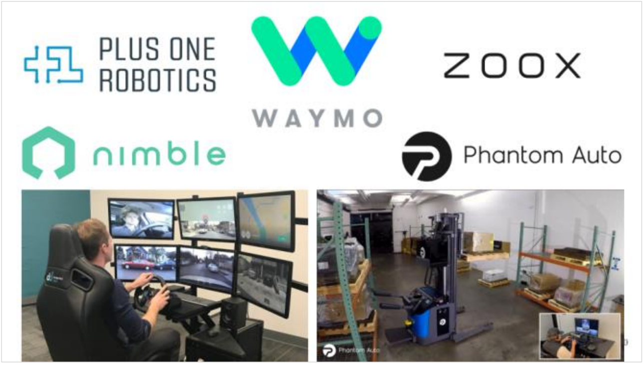

Safe DAgger
Lazy DAgger


Safe DAgger
Lazy DAgger

Limitations

- Parameter tuning
- Hard to know how many interventions will be requested.
- One human managing one robot.





When should a robot ask for help?

Novel (and risky)

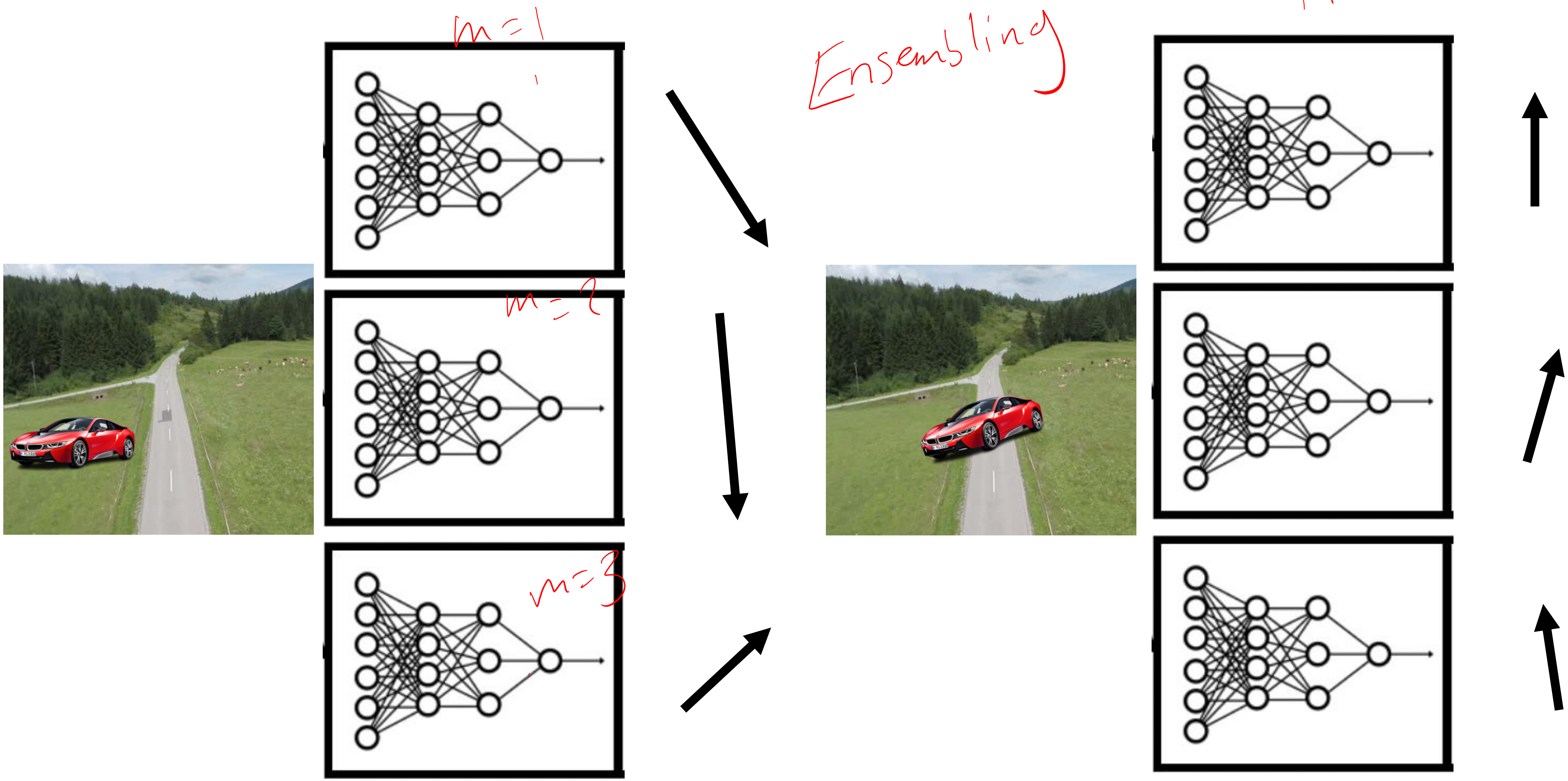
When should a robot ask for help?

Novel (and risky)

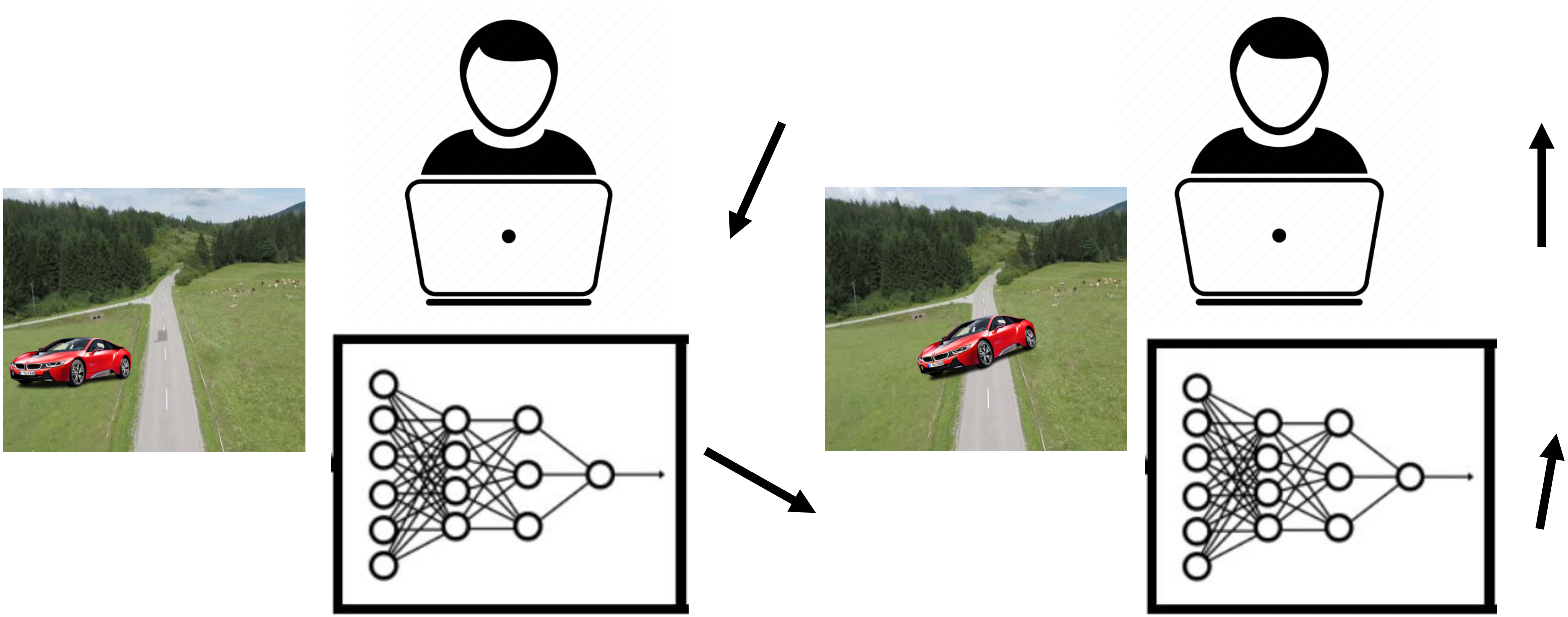
Risky (but not novel)

Novelty Estimation

in distribution



Novelty Estimation: Supervisor Mode



Risk Estimation

Assume Goal identifier

R = 1 if at goal, 0 otherwise

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

Expected prob of reaching goal / task success

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

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

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

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

Putting it all together…

AUTONOMOUS MODE

$$\text{Novelty}(s_t) > \delta_h$$

OR

$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h$$

$\longrightarrow$ Switch to SUPERVISOR MODE

Putting it all together…

AUTONOMOUS MODE

$$\text{Novelty}(s_t) > \delta_h$$

OR

$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h$$

→ Switch to SUPERVISOR MODE

not novel

SUPERVISOR MODE

$$||\pi_r(s_t) - \pi_h(s_t)||_2^2 < \delta_r$$

AND

$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) < \beta_r$$

not risky

→ Switch to AUTONOMOUS MODE

Putting it all together…

AUTONOMOUS MODE

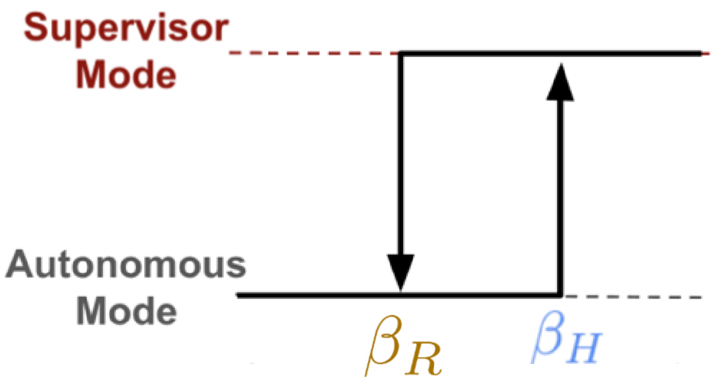
$$\text{Novelty}(s_t) > \delta_h$$
OR
$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h$$

⟶ Switch to SUPERVISOR MODE

SUPERVISOR MODE

$$||\pi_r(s_t) - \pi_h(s_t)||_2^2 < \delta_r$$
AND
$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) < \beta_r$$

⟶ Switch to AUTONOMOUS MODE



Wait, didn't we just double the number of hyperparameters?

Putting it all together…

AUTONOMOUS MODE

$$\text{Novelty}(s_t) > \boxed{\delta_h}$$

OR

$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \boxed{\beta_h}$$

Set to 1-α quantiles of empirical data

$\longrightarrow$ Switch to SUPERVISOR MODE

SUPERVISOR MODE

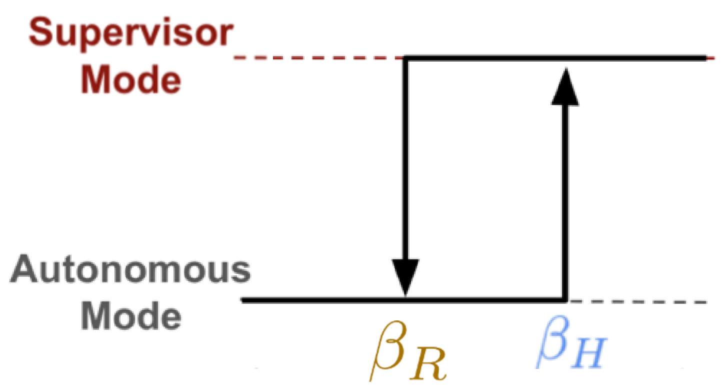
$$||\pi_r(s_t) - \pi_h(s_t)||_2^2 < \delta_r$$

AND

$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) < \beta_r$$

$\longrightarrow$ Switch to AUTONOMOUS MODE

$$\alpha = \frac{\text{desired \# interventions}}{\text{\# robot actions}}$$



Putting it all together...

AUTONOMOUS MODE

$$\text{Novelty}(s_t) > \delta_h$$

OR

$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h$$

Set to 1-α quantiles of empirical data

→ Switch to SUPERVISOR MODE

SUPERVISOR MODE

$$||\pi_r(s_t) - \pi_h(s_t)||_2^2 < \delta_r$$

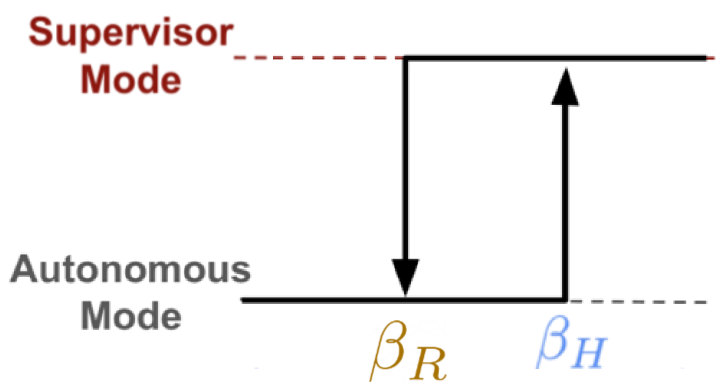
AND

$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) < \beta_r$$

→ Switch to AUTONOMOUS MODE

Set to medians of empirical data

$$\alpha = \frac{\text{desired \# interventions}}{\text{\# robot actions}}$$



Putting it all together...

AUTONOMOUS MODE

$$\text{Novelty}(s_t) > \delta_h$$
OR
$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) > \beta_h$$

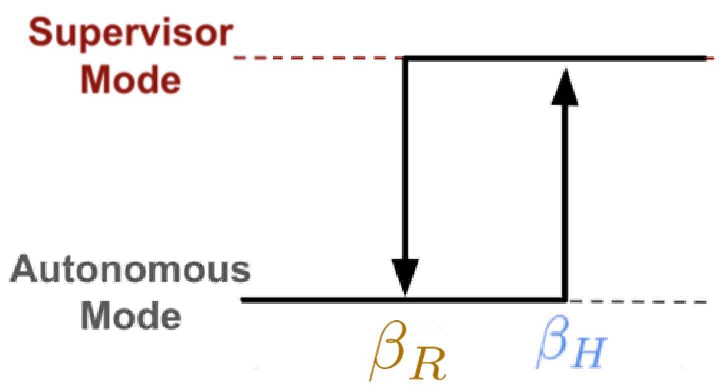
→ Switch to SUPERVISOR MODE

SUPERVISOR MODE

$$||\pi_r(s_t) - \pi_h(s_t)||_2^2 < \delta_r$$
AND
$$\text{Risk}^{\pi_r}(s_t, \pi_r(s_t)) < \beta_r$$

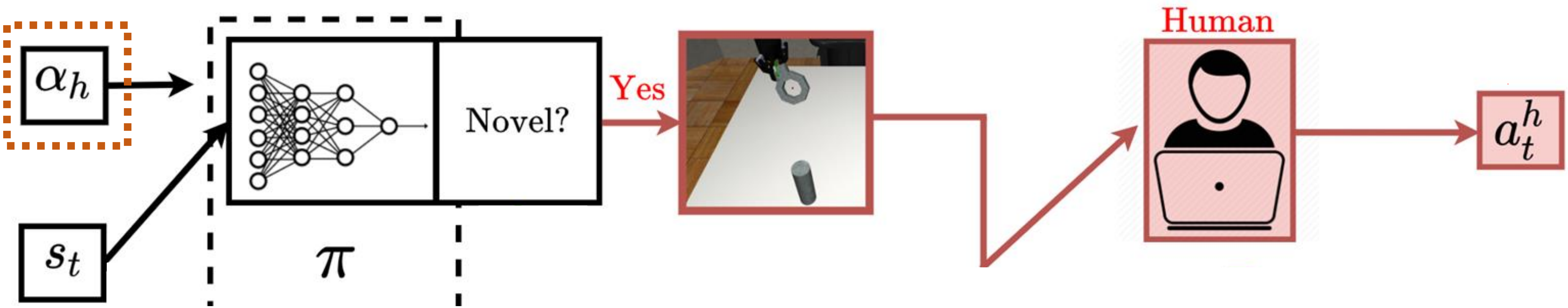
→ Switch to AUTONOMOUS MODE

$$\alpha = \frac{\text{desired \# interventions}}{\text{\# robot actions}}$$



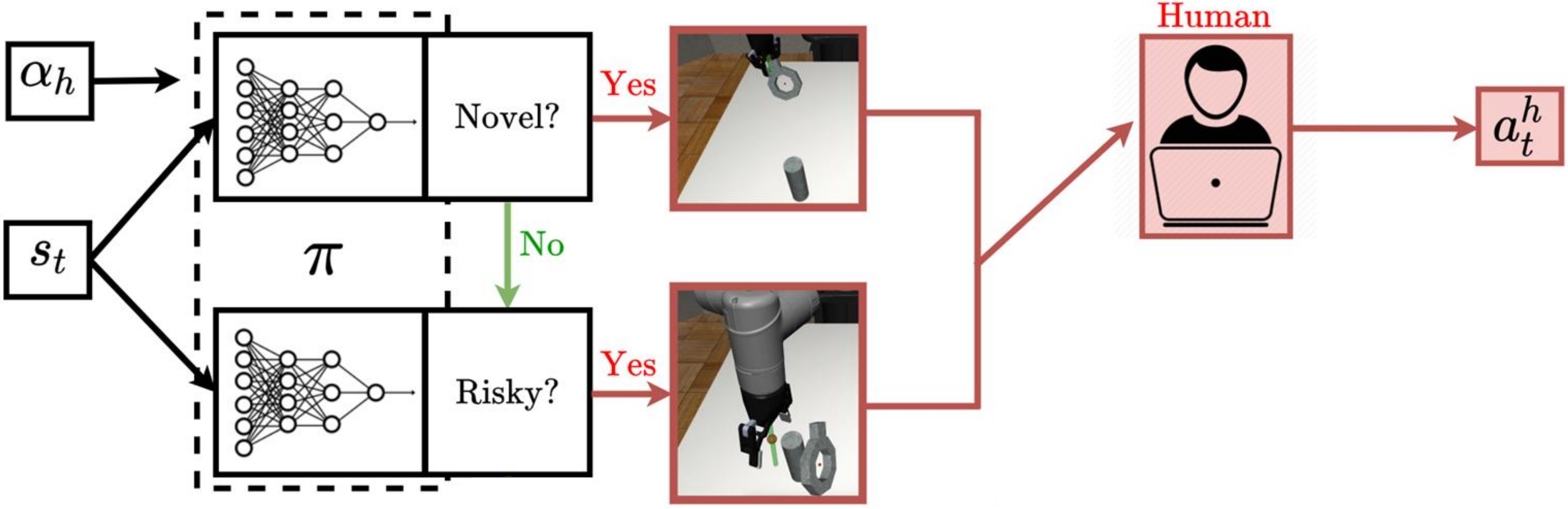
ThriftyDAgger

Target percent of time human wants to give interventions.



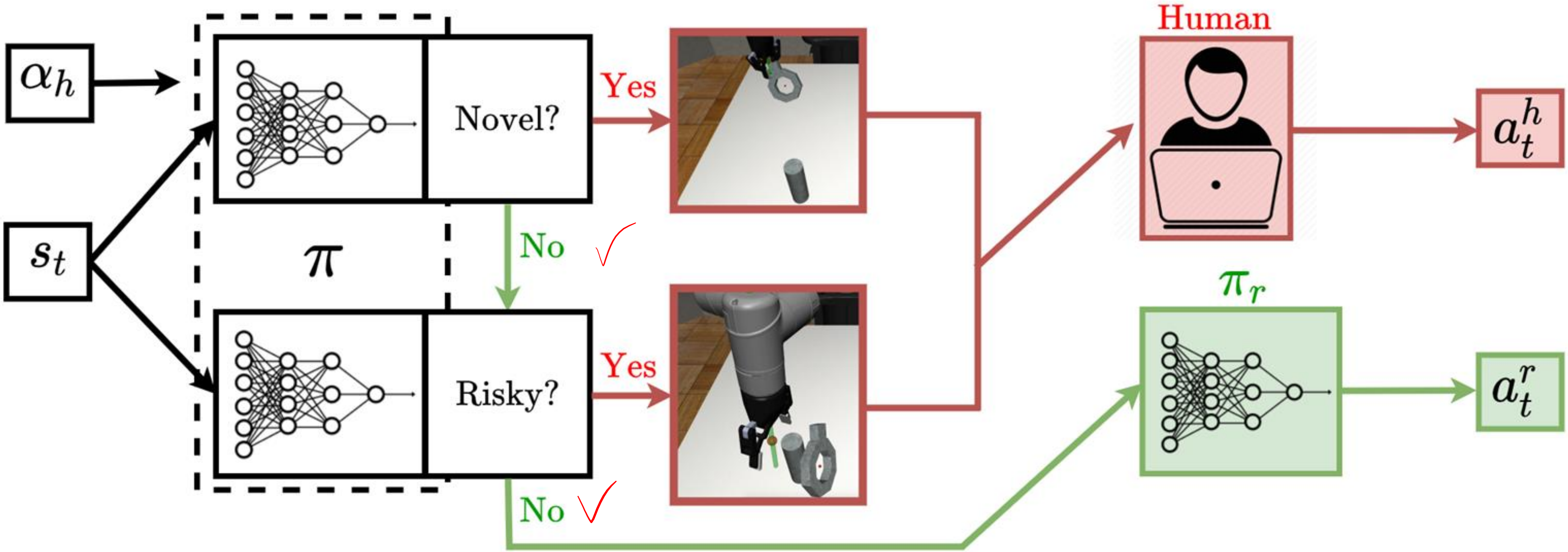
Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

ThriftyDAgger

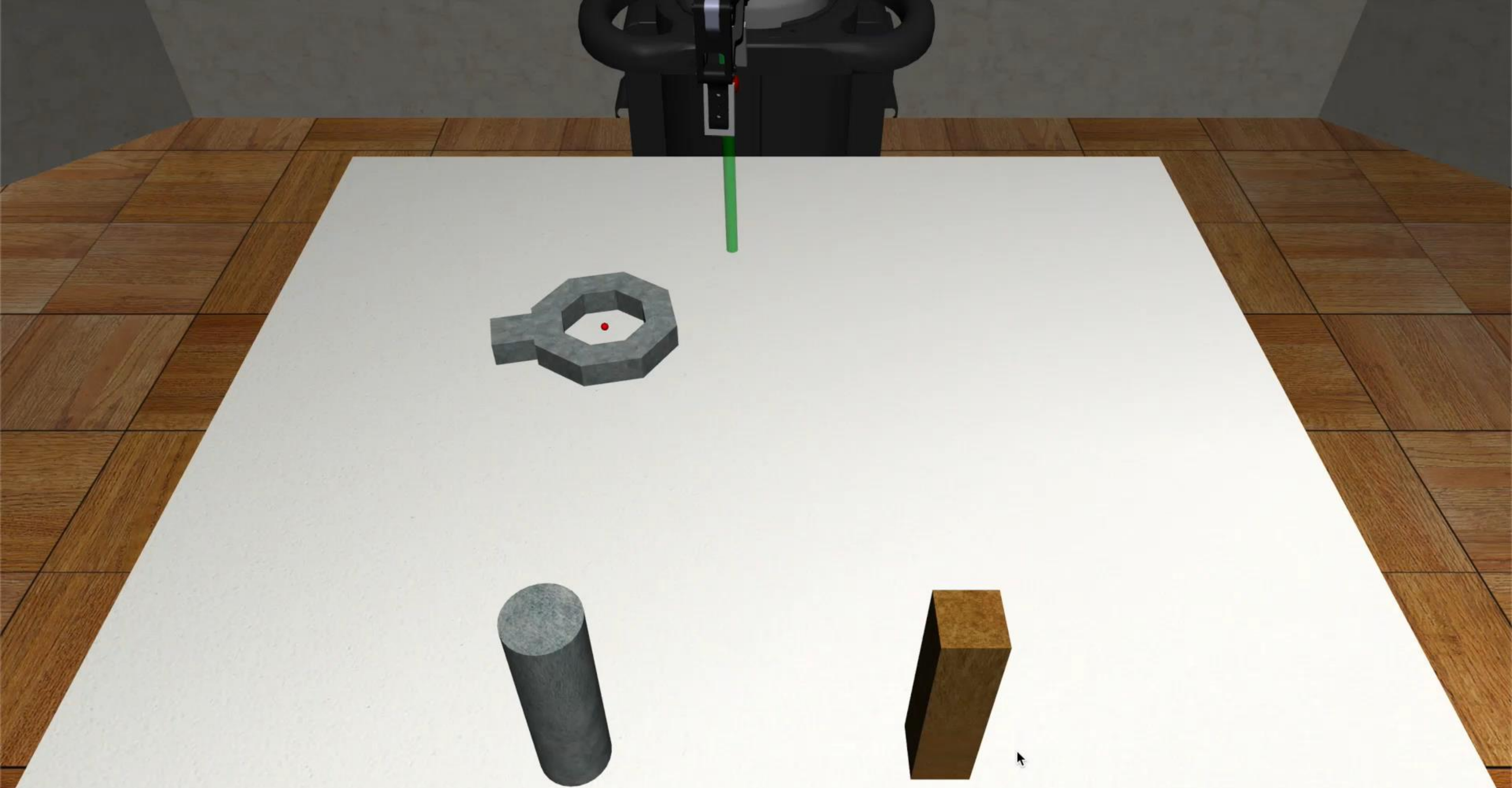


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

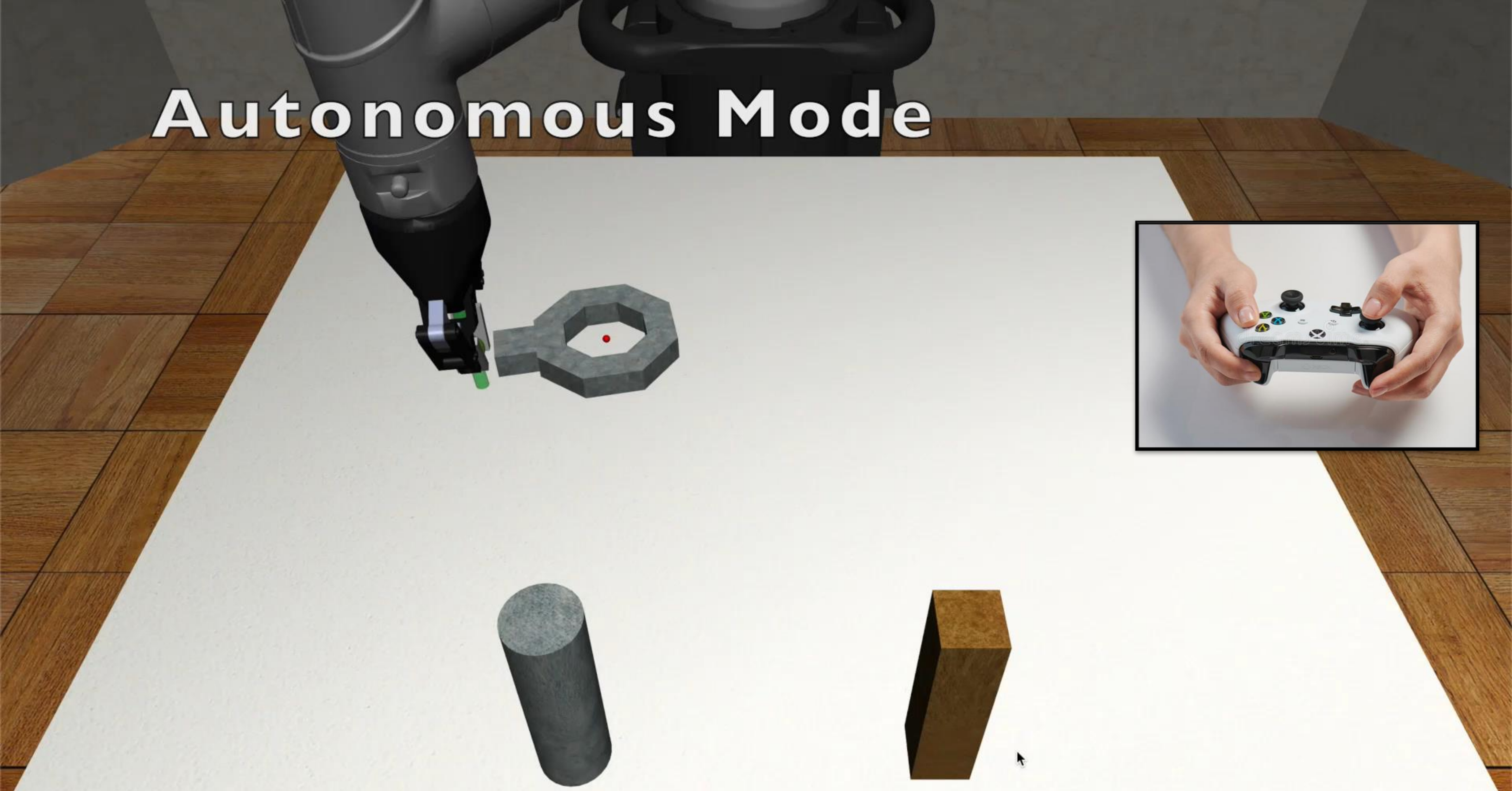
ThriftyDAgger



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

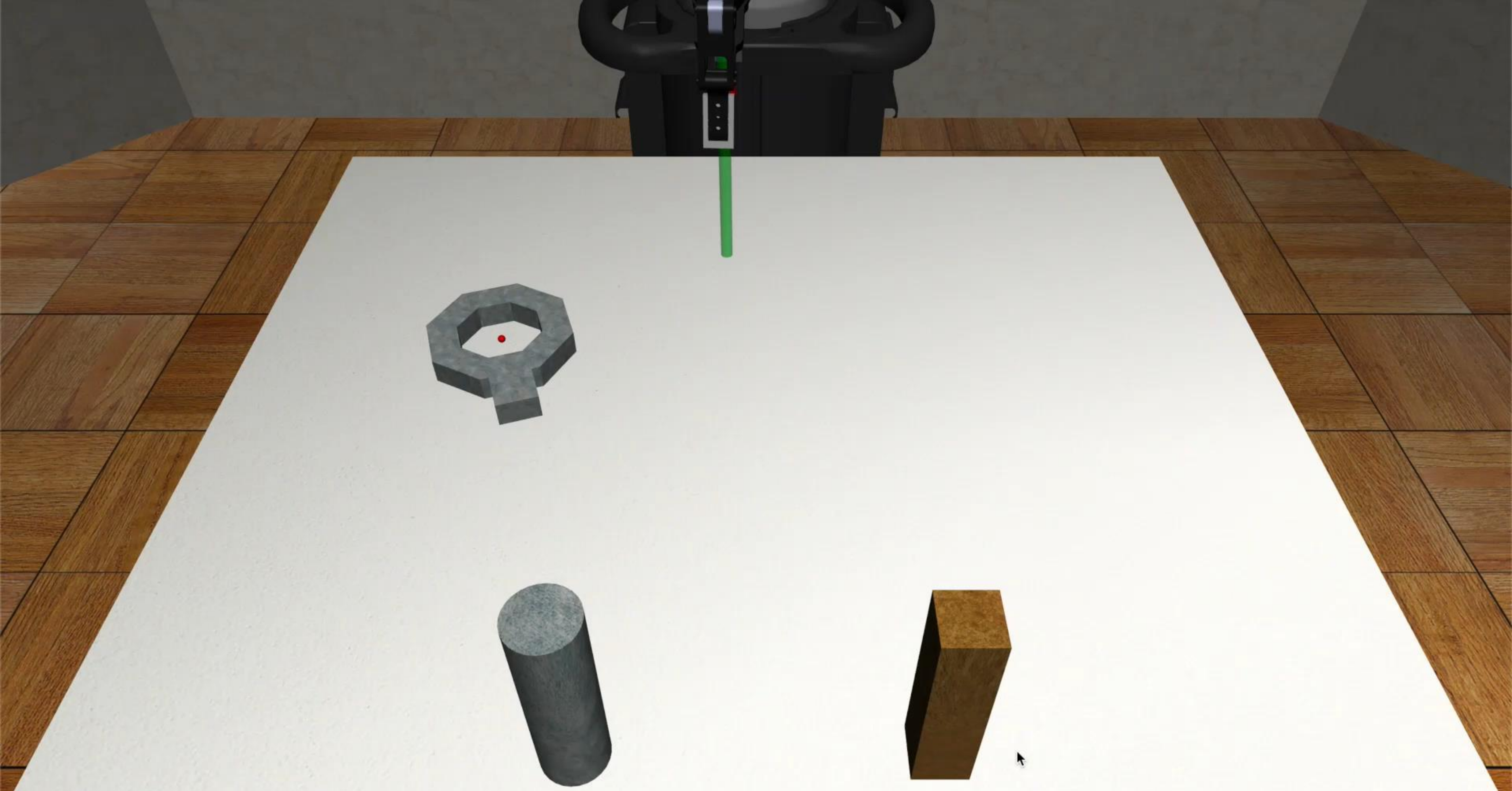


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

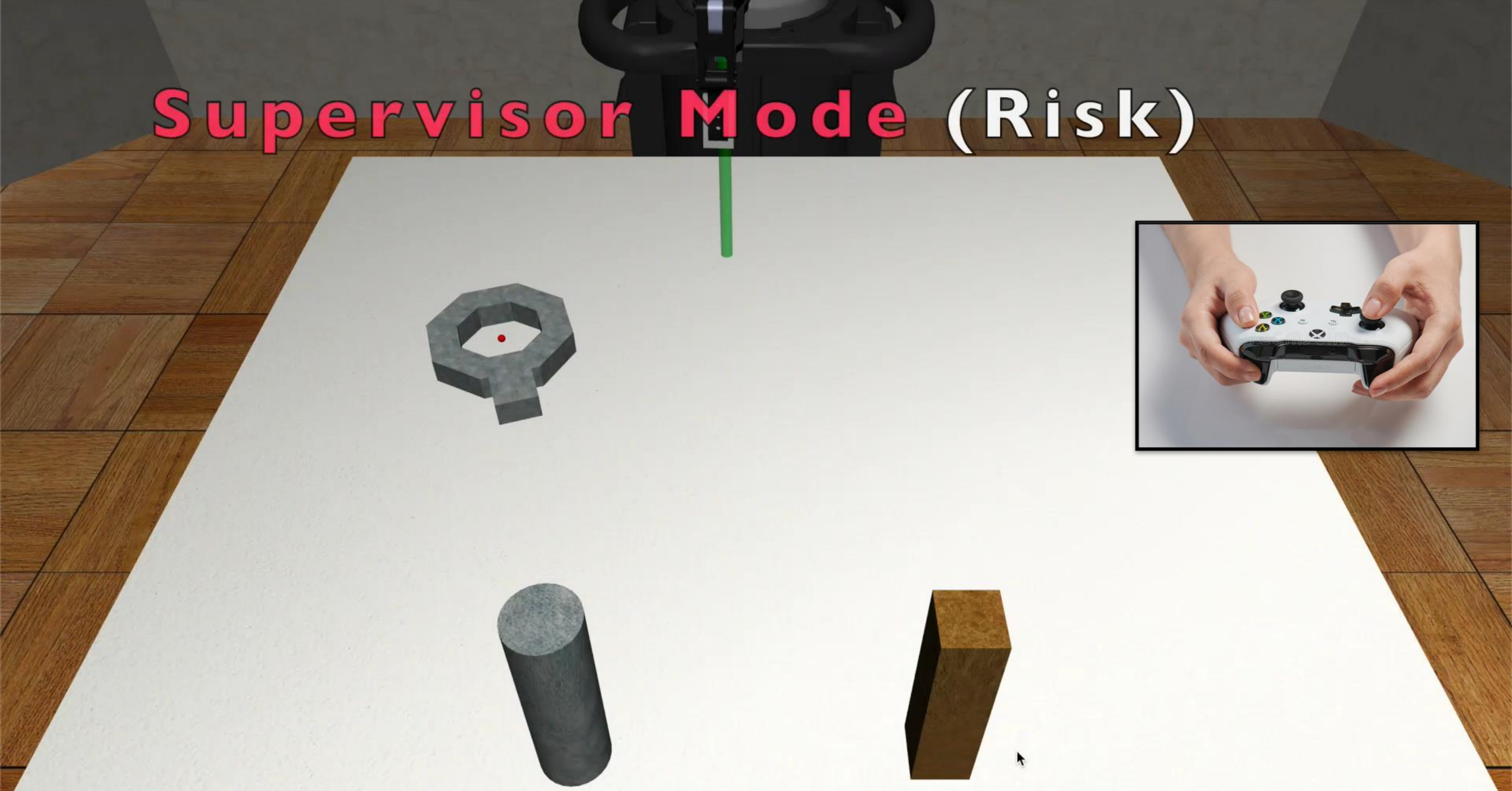


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

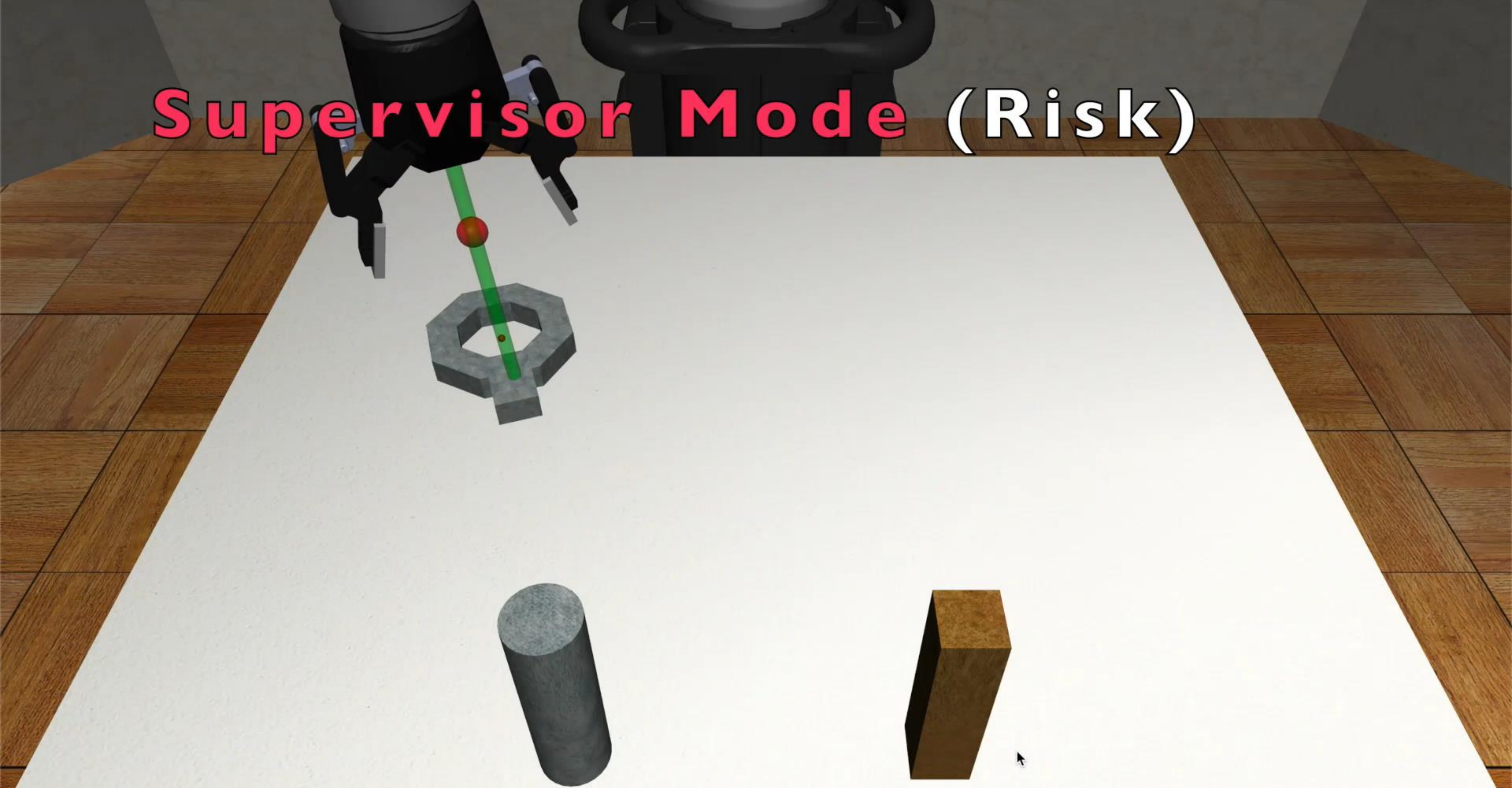
Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.



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

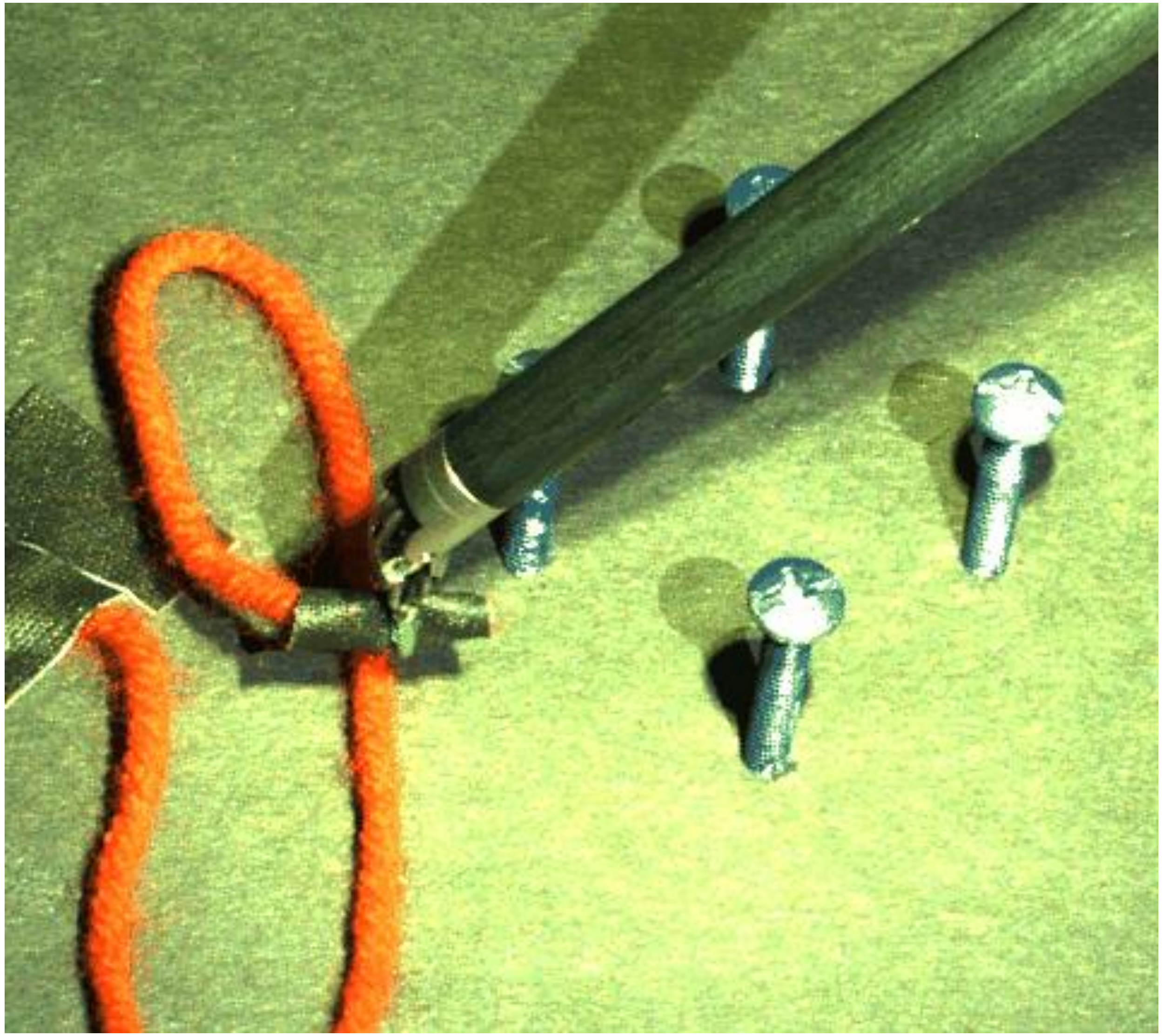


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

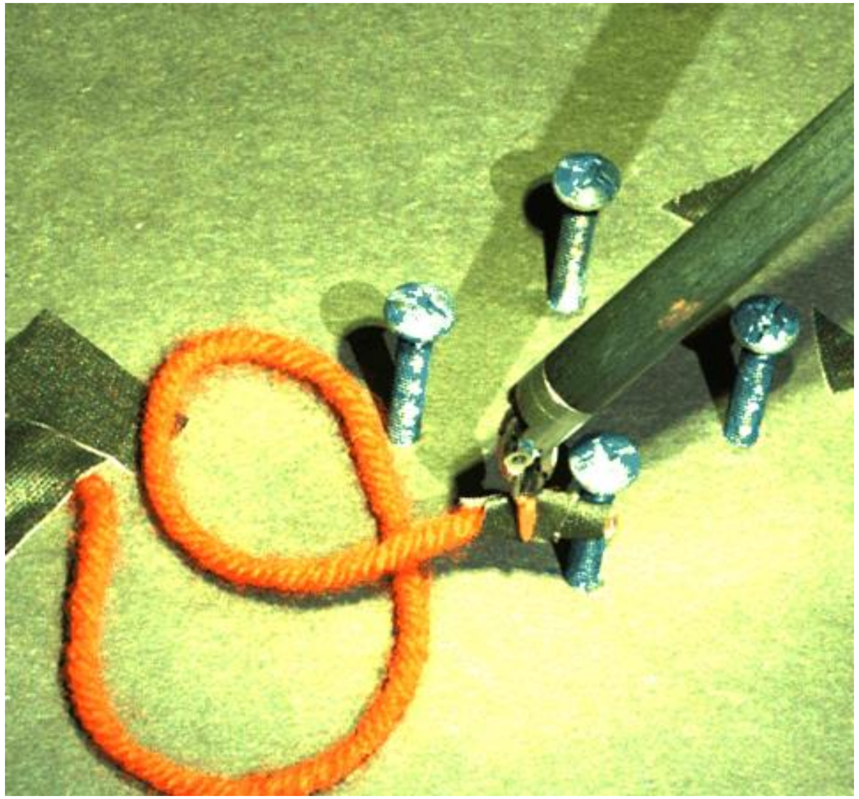


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

Human Demonstration

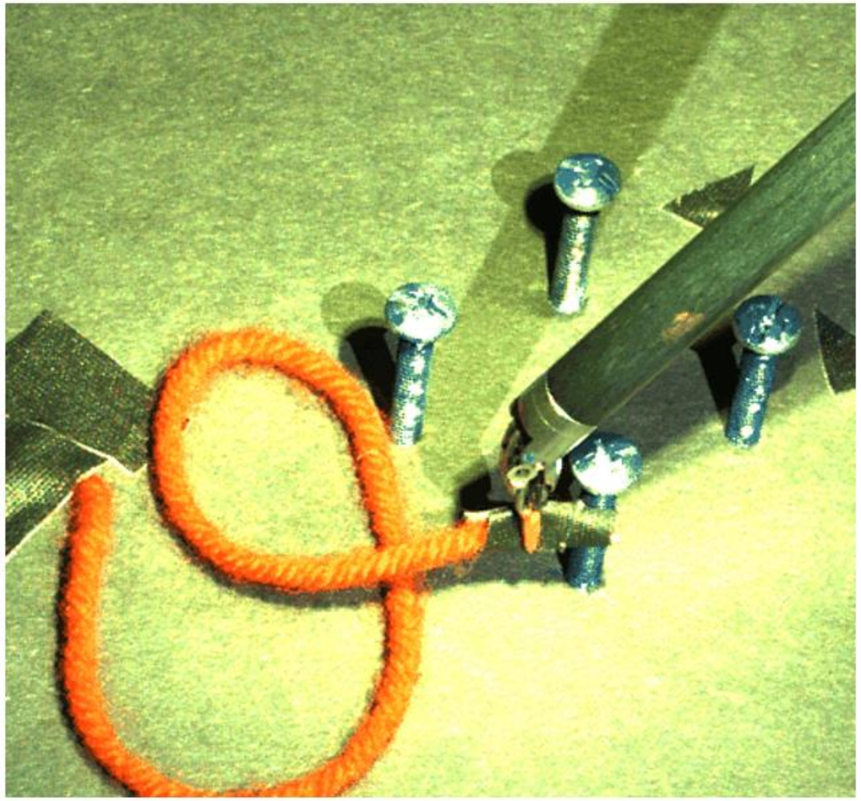


Behavior Cloning

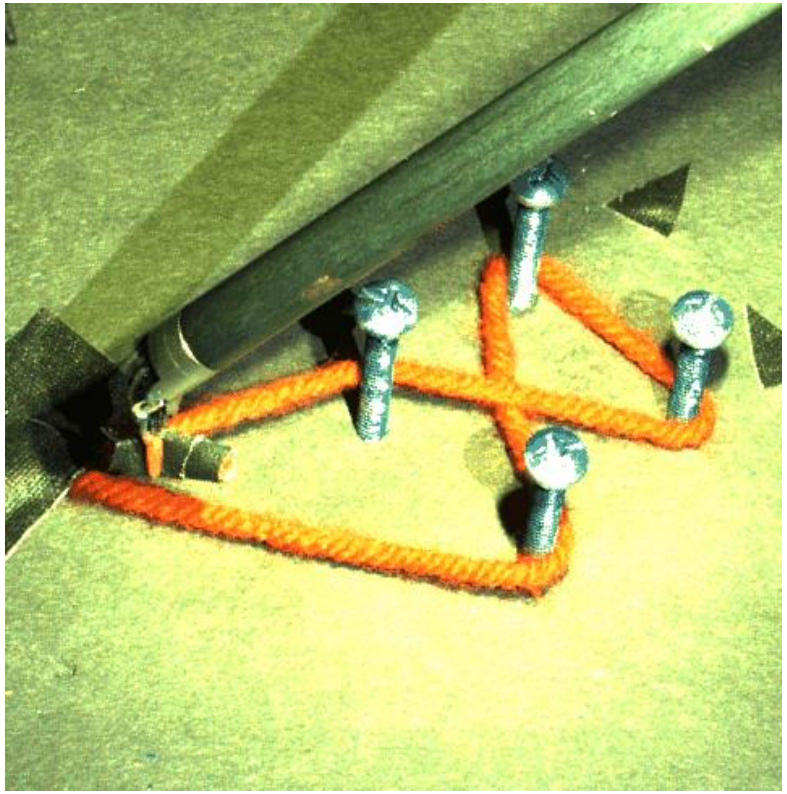


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

Behavior Cloning

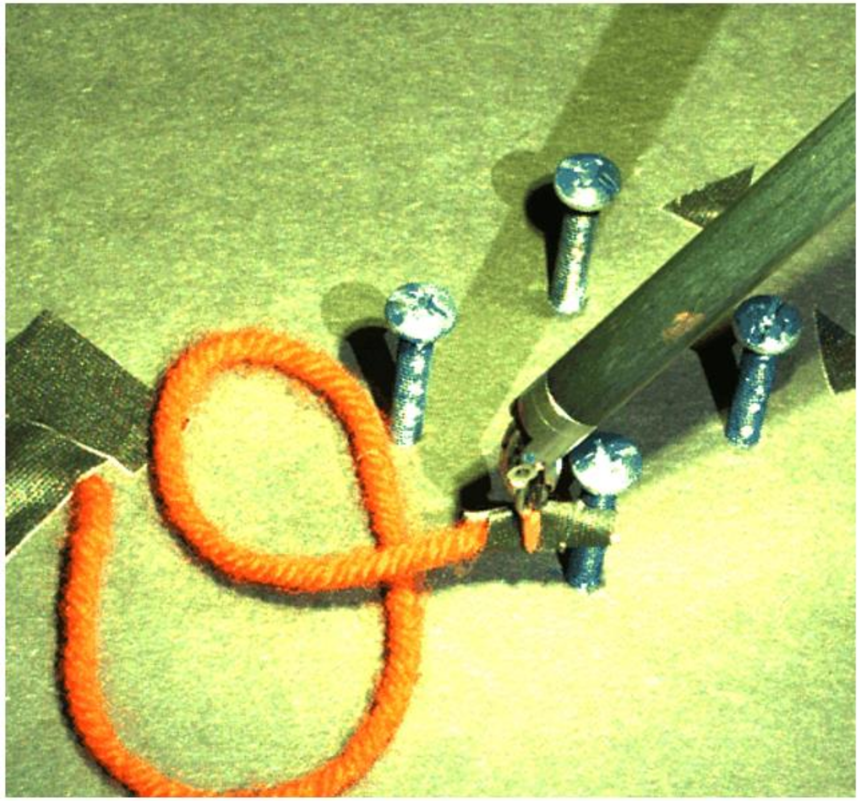


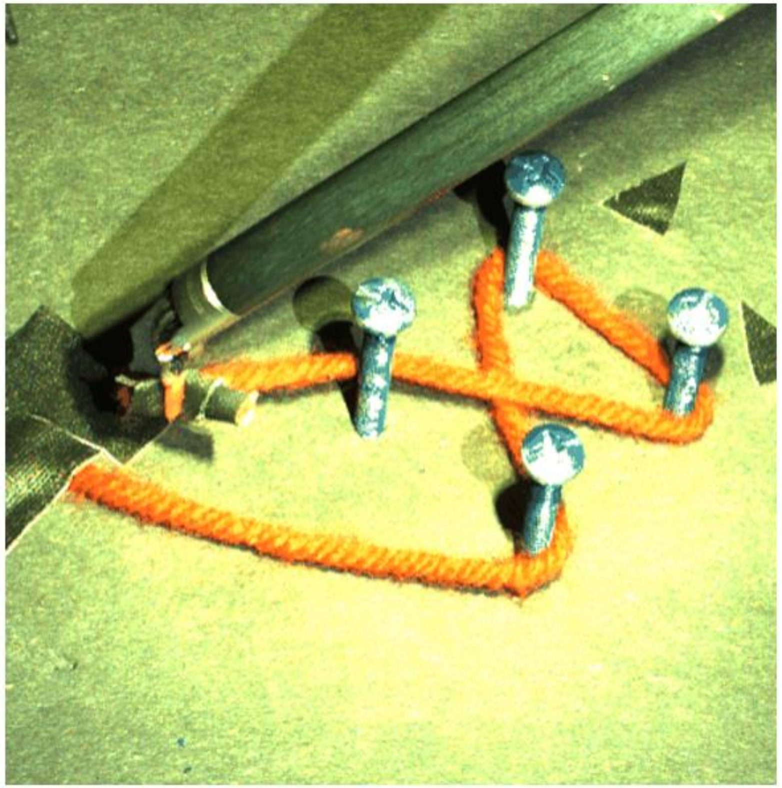
ThriftyDAgger (autonomous)

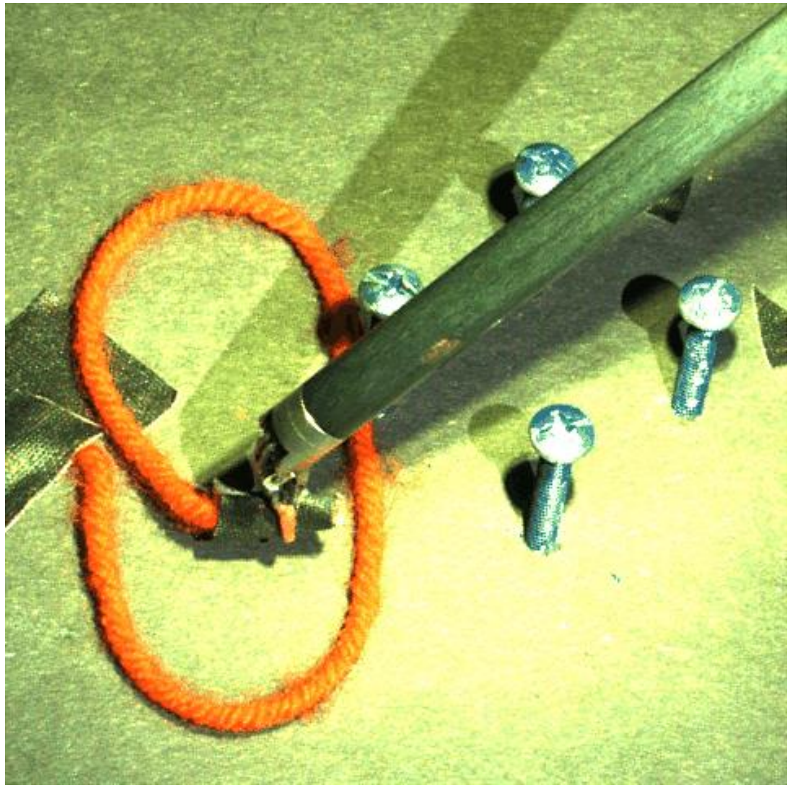


Hoque et al. "ThriftyDAgger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning." CoRL 2021.

Behavior Cloning | **ThriftyDAgger (autonomous)** | **ThriftyDAgger (+human)**







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

User Study

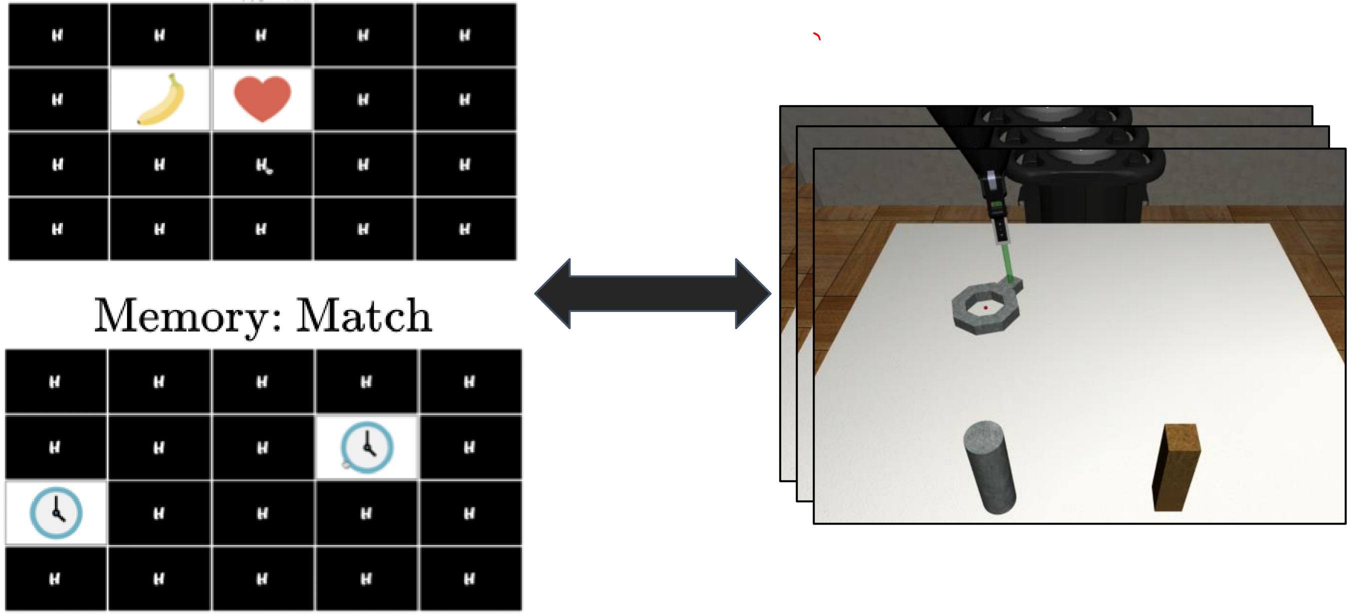
N=10 subjects each control 3 robots in simulation.

humans

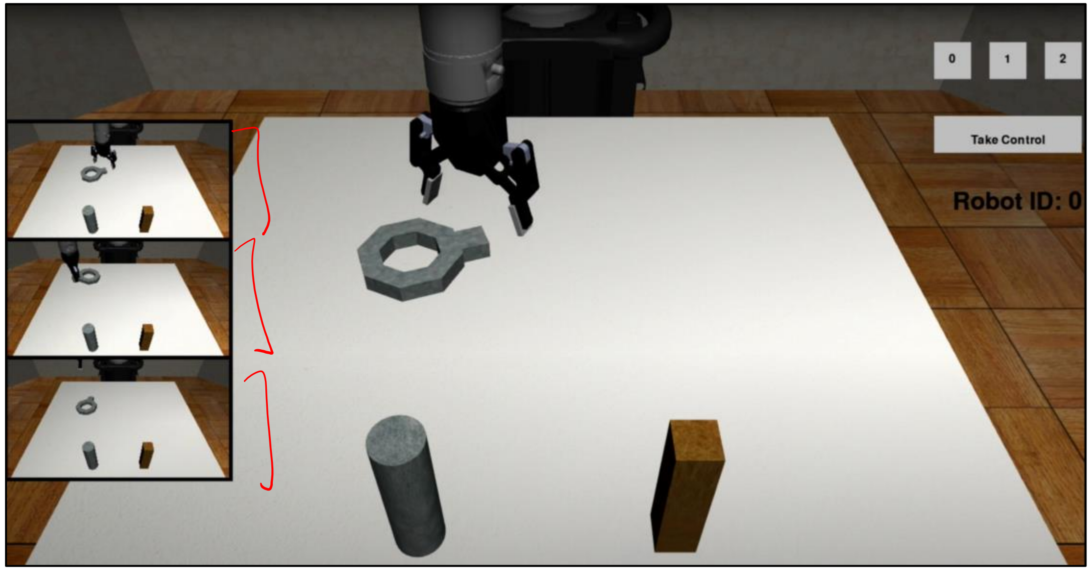
Robot-Gated

Memory: Non-Match

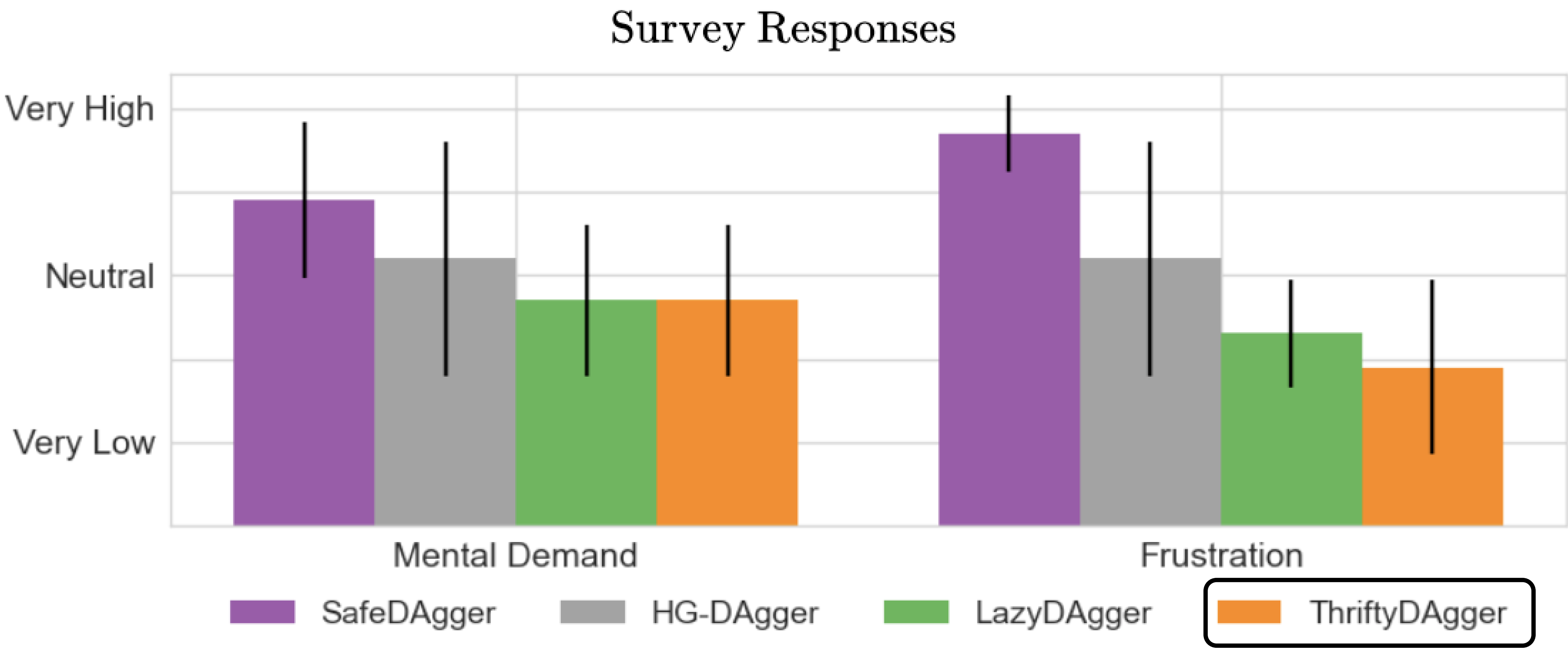
Memory: Match



Human-Gated



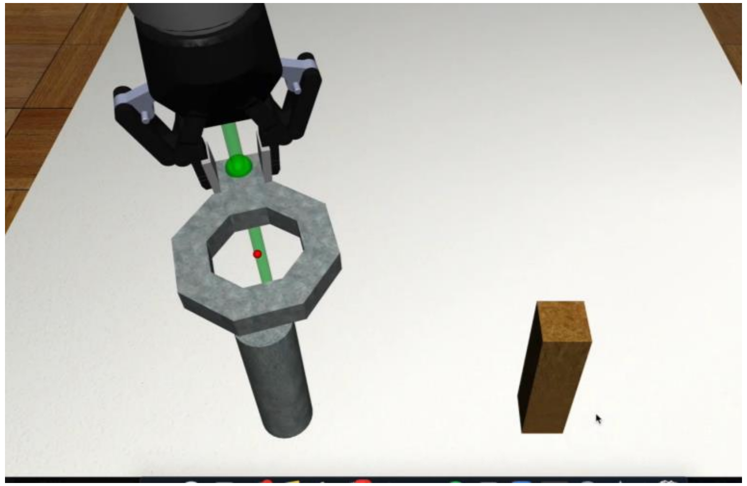
ThriftyDAgger Qualitative Results



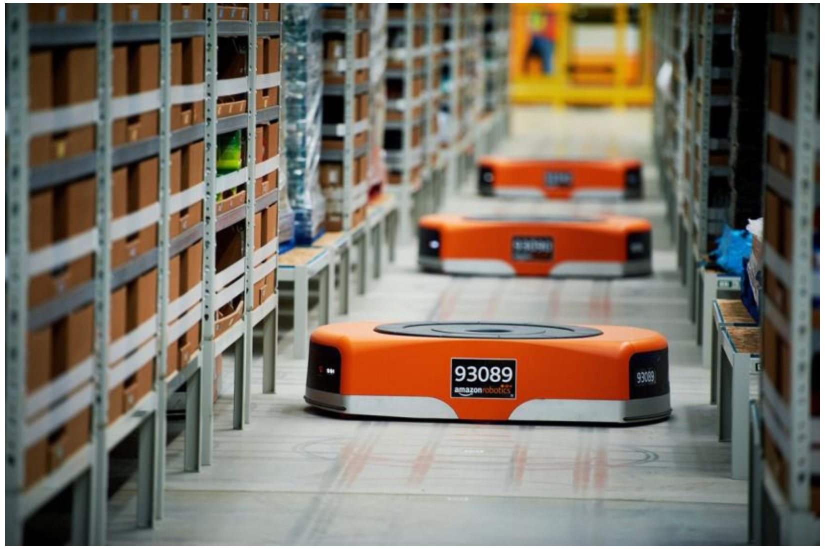
User Study Quantitative Results

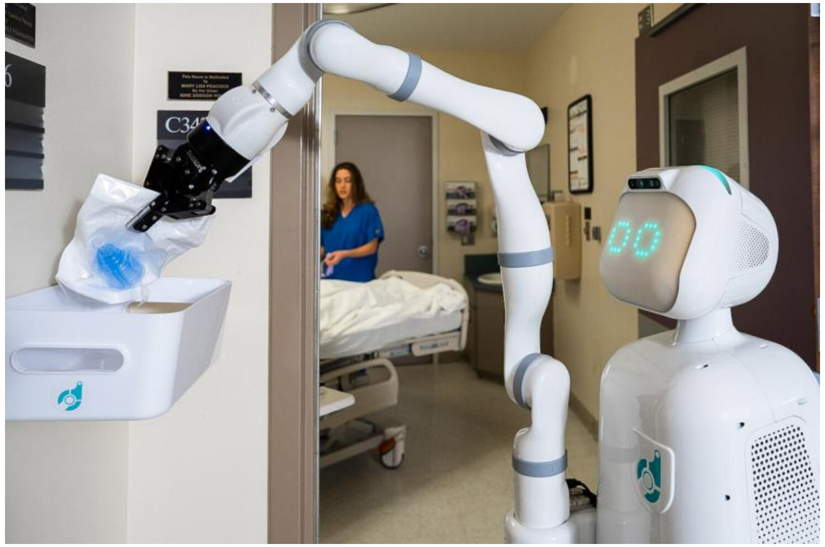
ThriftyDAgger had

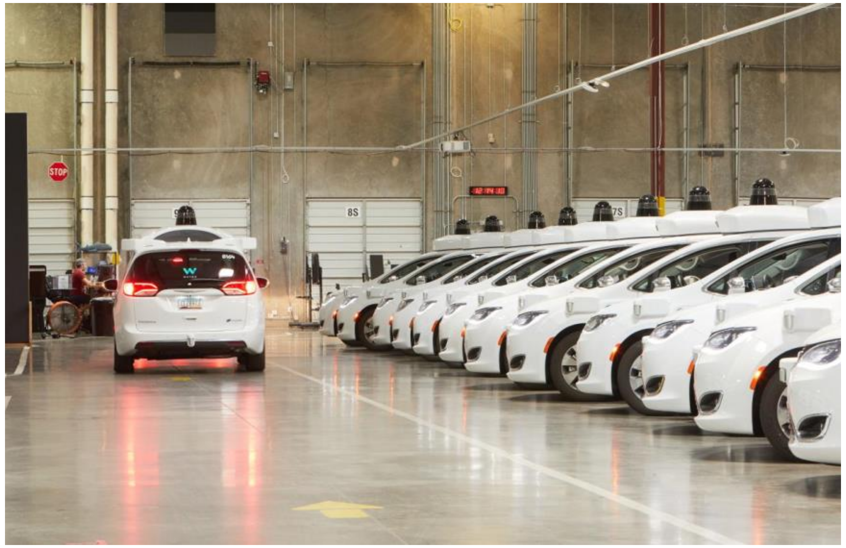
- 21% fewer human interventions
- 57% more concentration pairs found
- 80% more throughput



Scalable and safe robot fleets are possible when robots ask for help in ways that minimize human supervisor burden.







Next time: Inverse Reinforcement Learning!