Inverse Reinforcement Learning

Chelsea Finn

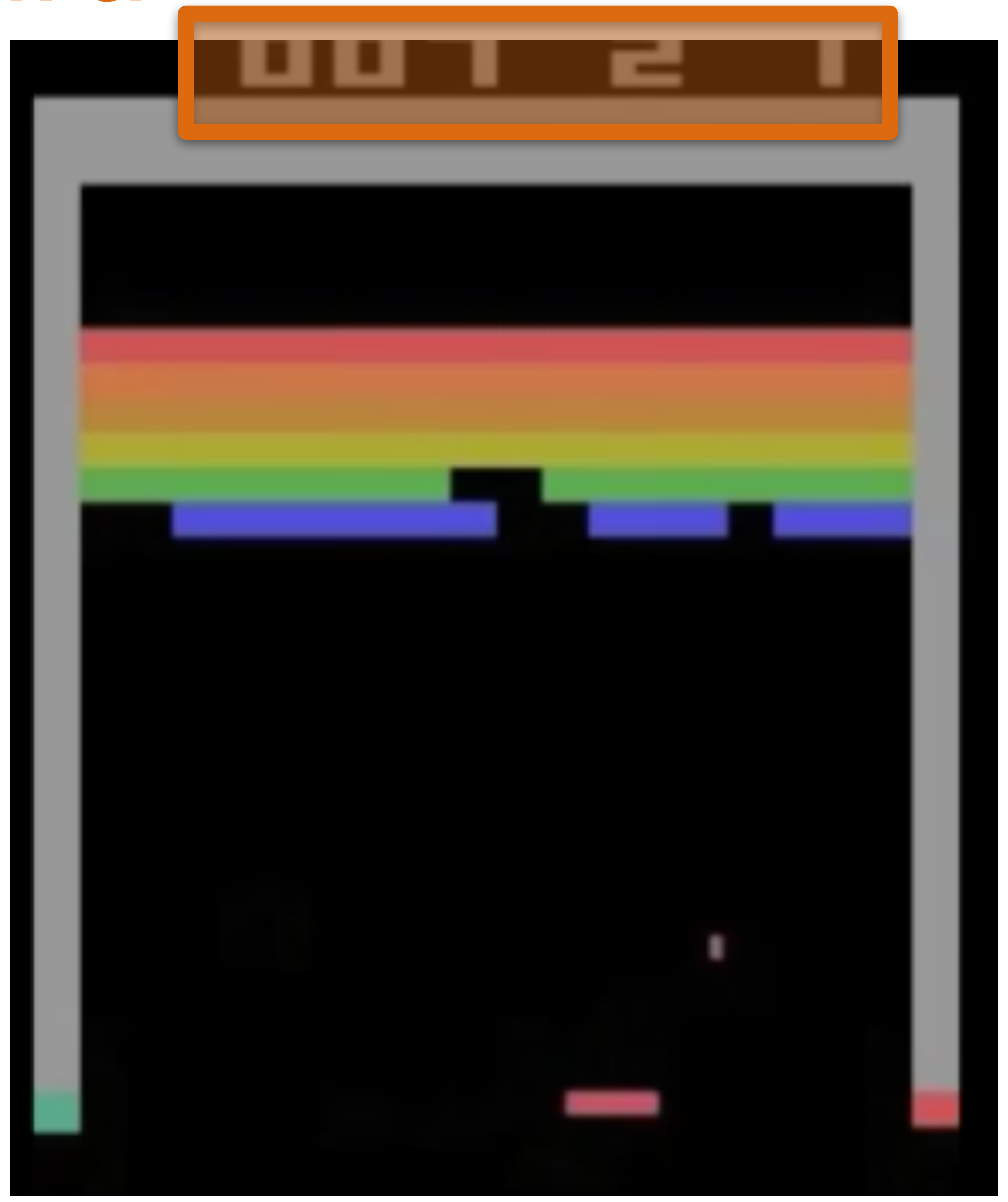
Deep RL Bootcamp

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Where does the reward come from?

Computer Games

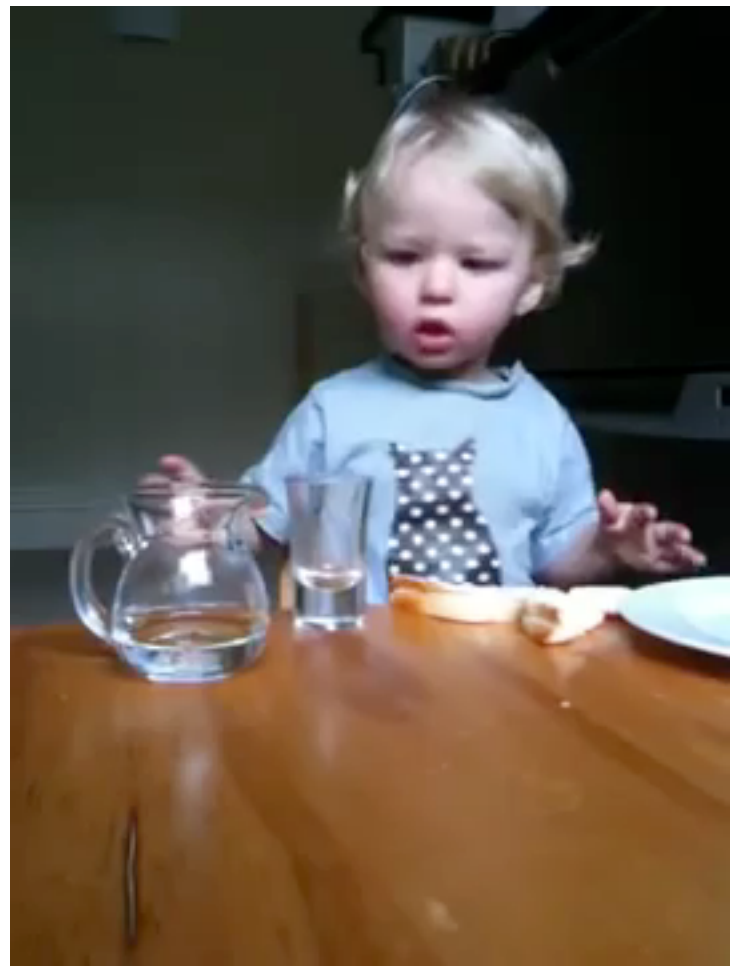
reward



Mnih et al. ‘15

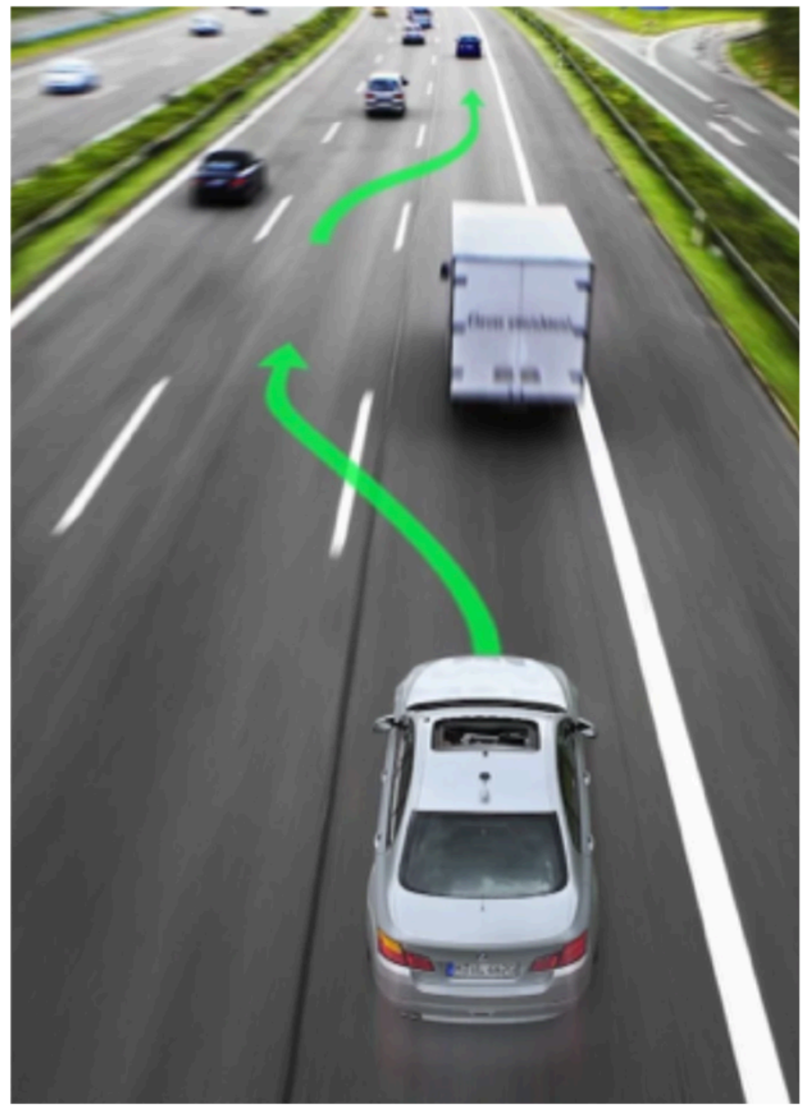
Real World Scenarios

robotics



dialog

autonomous driving



what is the reward?
often use a proxy

frequently easier to provide expert data

Approach: infer reward function from roll-outs of expert policy

Why infer the reward?

Behavioral Cloning/Direct Imitation: Mimic actions of expert
- but no reasoning about outcomes or dynamics
- the expert might have different degrees of freedom

Can we reason about what the expert is trying to achieve?

Inverse Optimal Control / Inverse Reinforcement Learning:

infer reward function from demonstrations

(IOC/IRL) (Kalman '64, Ng & Russell '00)

given:
- state & action space
- Roll-outs from π^*
- dynamics model (sometimes)

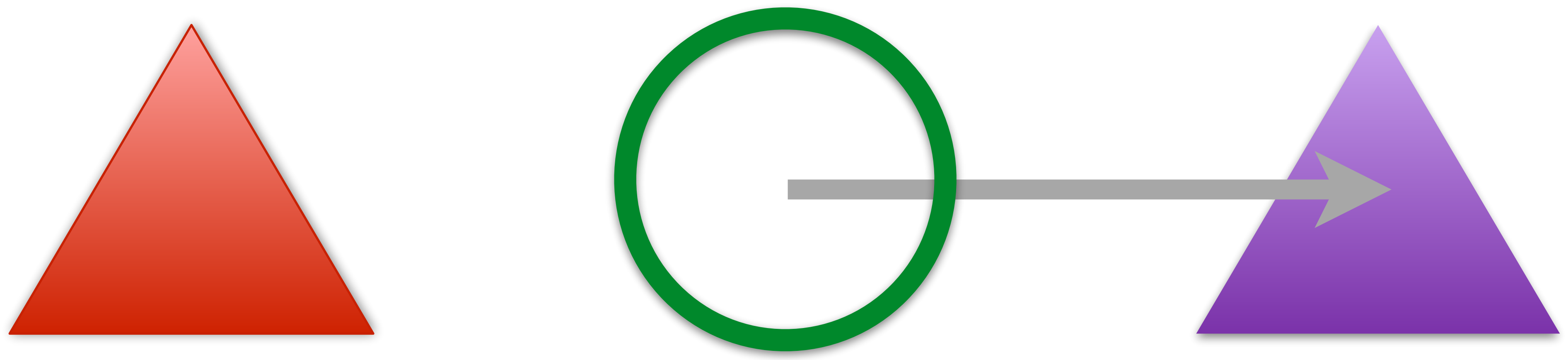
goal:
- recover reward function
- then use reward to get policy

Challenges

underdefined problem

difficult to evaluate a learned reward

demonstrations may not be precisely optimal



Maximum Entropy Inverse RL

(Ziebart et al. '08)

handle ambiguity using probabilistic model of behavior

Notation:

$$\tau = \{s_1, a_1, ..., s_t, a_t, ..., s_T\}$$

trajectory

$$R_\psi(\tau) = \sum_t r_\psi(s_t, a_t)$$

learned reward

$$\mathcal{D} : \{\tau_i\} \sim \pi^*$$

expert demonstrations

MaxEnt formulation:

$$p(\tau) = \frac{1}{Z} \exp(R_\psi(\tau))$$

$$Z = \int \exp(R_\psi(\tau)) d\tau$$

$$\max_\psi \sum_{\tau \in \mathcal{D}} \log p_{r_\psi}(\tau)$$

(energy-based model for behavior)

Maximum Entropy IRL Optimization

$$\max_{\psi} \mathcal{L}(\psi) = \sum_{\tau \in \mathcal{D}} \log p_{r_\psi}(\tau)$$

$$= \sum_{\tau \in \mathcal{D}} \log \frac{1}{Z} \exp(R_\psi(\tau))$$

$$= \sum_{\tau \in \mathcal{D}} R_\psi(\tau) - M \log Z$$

$$= \sum_{\tau \in \mathcal{D}} R_\psi(\tau) - M \log \sum_{\tau} \exp(R_\psi(\tau))$$

$$\nabla_\psi \mathcal{L}(\psi) = \sum_{\tau \in \mathcal{D}} \frac{dR_\psi(\tau)}{d\psi} - M \frac{1}{\sum_\tau \exp(R_\psi(\tau))} \sum_{\tau} \exp(R_\psi(\tau)) \frac{dR_\psi(\tau)}{d\psi}$$

Maximum Entropy IRL Optimization

$$\nabla_\psi \mathcal{L}(\psi) = \sum_{\tau \in \mathcal{D}} \frac{dR_\psi(\tau)}{d\psi} - M \frac{1}{\sum_\tau \exp(R_\psi(\tau))} \underbrace{\sum_\tau \exp(R_\psi(\tau)) \frac{dR_\psi(\tau)}{d\psi}}$$

$$\sum_\tau p(\tau \mid \psi) \frac{dR_\psi(\tau)}{d\psi}$$

$$\sum_{\mathbf{s}} p(\mathbf{s} \mid \psi) \frac{dr_\psi(\mathbf{s})}{d\psi}$$

blackboard

Maximum Entropy Inverse RL

(Ziebart et al. '08)

Algorithm:

0. Initialize ψ, gather demonstrations $\mathcal{D}$
1. Solve for optimal policy $\pi(\mathbf{a}|\mathbf{s})$ w.r.t. reward r_ψ
2. Solve for state visitation frequencies $p(\mathbf{s}|\psi)$
3. Compute gradient $\nabla_\psi \mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{\tau_d \in \mathcal{D}} \frac{dr_\psi}{d\psi}(\tau_d) - \sum_s p(s|\psi) \frac{dr_\psi}{d\psi}(s)$
4. Update ψ with one gradient step using $\nabla_\psi \mathcal{L}$

How can we:

(1) handle unknown dynamics? (2) avoid solving the MDP in the inner loop

$$\max_{\psi} \sum_{\tau \in \mathcal{D}} \log p_{r_\psi}(\tau)$$

$$p(\tau) = \frac{1}{Z} \exp(R_\psi(\tau))$$

$$Z = \int \exp(R_\psi(\tau)) d\tau$$

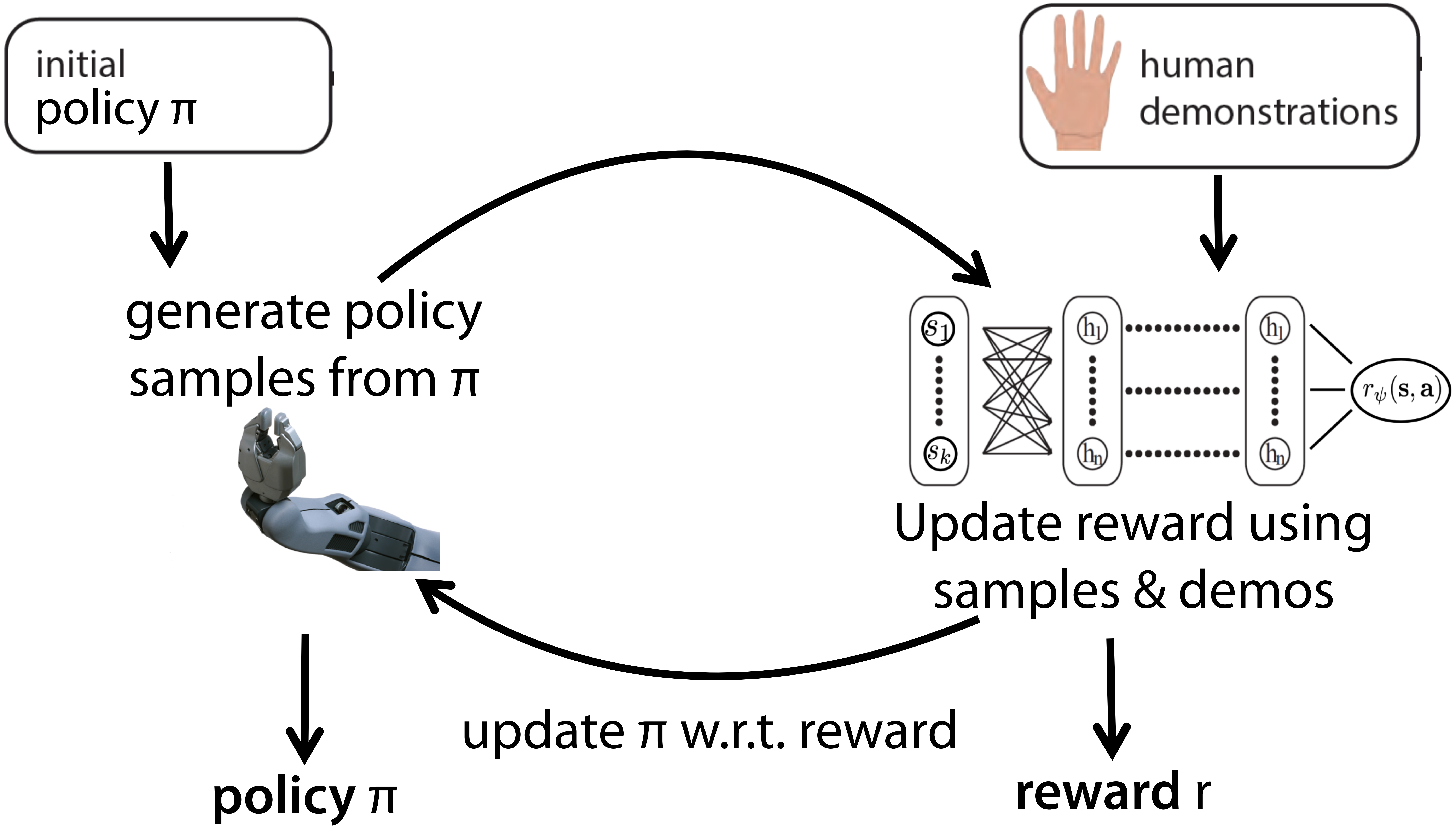
sample to estimate Z

sample *adaptively* to estimate Z

[by constructing a policy]

guided cost learning algorithm

(Finn et al. ICML '16)



guided cost learning algorithm

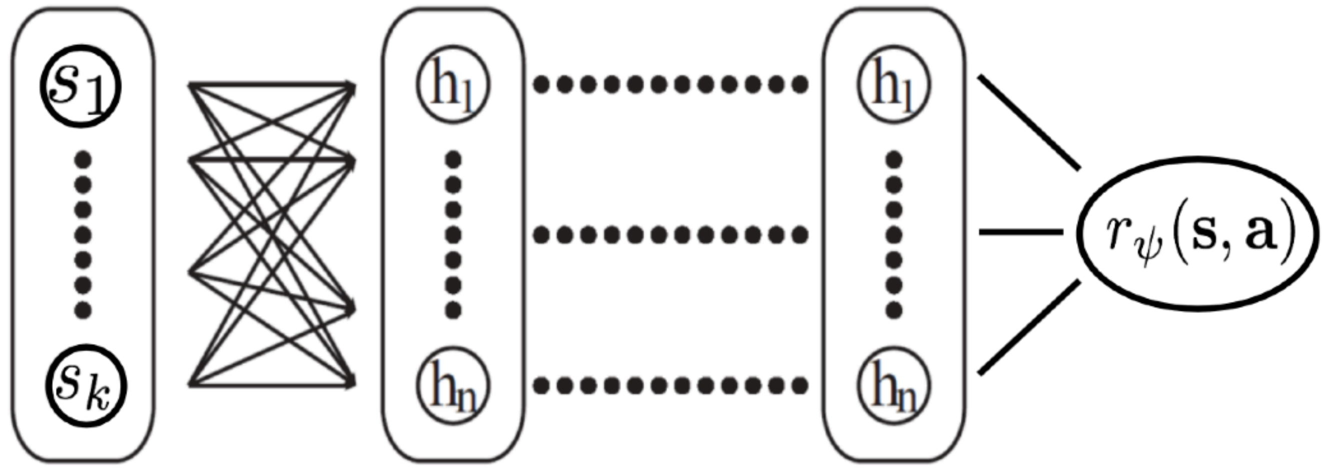
(Finn et al. ICML '16)

initial
policy π

human
demonstrations

generate policy
samples from π

generator



Update reward using
samples & demos

discriminator

update π w.r.t. reward
take 1 policy opt. step

policy π

reward r

Update reward in inner loop of policy optimization

Ho & Ermon, NIPS '16

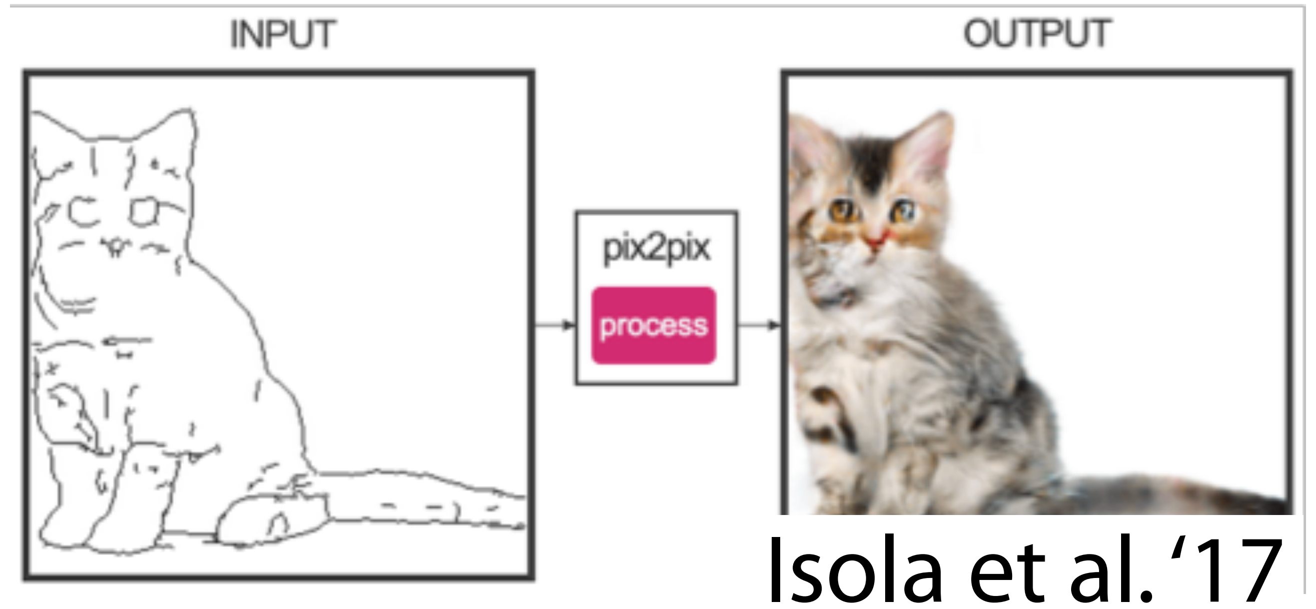
Aside: Generative Adversarial Networks

(Goodfellow et al. '14)

Similar to inverse RL, **GANs** learn an objective for generative modeling.


Zhu et al. '17


Arjovsky et al. '17



Isola et al. '17

	trajectory τ	⟷	**sample x**	
Inverse RL	**policy** $\pi \sim q(\tau)$	⟷	**generator** G	GANs
	reward r	⟶	**discriminator** D	

(Finn*, Christiano*, et al. '16)

Connection to Generative Adversarial Networks

(Goodfellow et al. '14)

Inverse RL			GANs
trajectory τ	⟷	**sample x**	
policy π~q(τ)	⟷	**generator** G	
reward r	⟶	**discriminator** D	

data distribution p

Reward/discriminator optimization:

GCL:

$$D^*(\tau) = \frac{p(\tau)}{p(\tau) + q(\tau)}$$

$$D_\psi(\tau) = \frac{\frac{1}{Z}\exp(R_\psi)}{\frac{1}{Z}\exp(R_\psi) + q(\tau)}$$

GAIL:

$D_\psi(\tau) =$ some classifier

Both:

$$\mathcal{L}_{\text{discriminator}}(\psi) = \mathbb{E}_{\tau\sim p}[-\log D_\psi(\tau)] + \mathbb{E}_{\tau\sim q}[-\log(1 - D_\psi(\tau))]$$

(Finn*, Christiano*, et al. '16)

Connection to Generative Adversarial Networks

(Goodfellow et al. '14)

Inverse RL			GANs
trajectory τ	⟷	**sample x**	
policy π~q(τ)	⟷	**generator** G	
reward r	⟶	**discriminator** D	

data distribution p

Policy/generator optimization:

$$\mathcal{L}_{\text{generator}}(\theta) = \mathbb{E}_{\tau\sim q}[\log(1 - D_\psi(\tau)) - \log D_\psi(\tau)]$$
$$= \mathbb{E}_{\tau\sim q}[\log q(\tau) + \log Z - R_\psi(\tau)]$$

entropy-regularized RL

Unknown dynamics: train generator/policy with RL

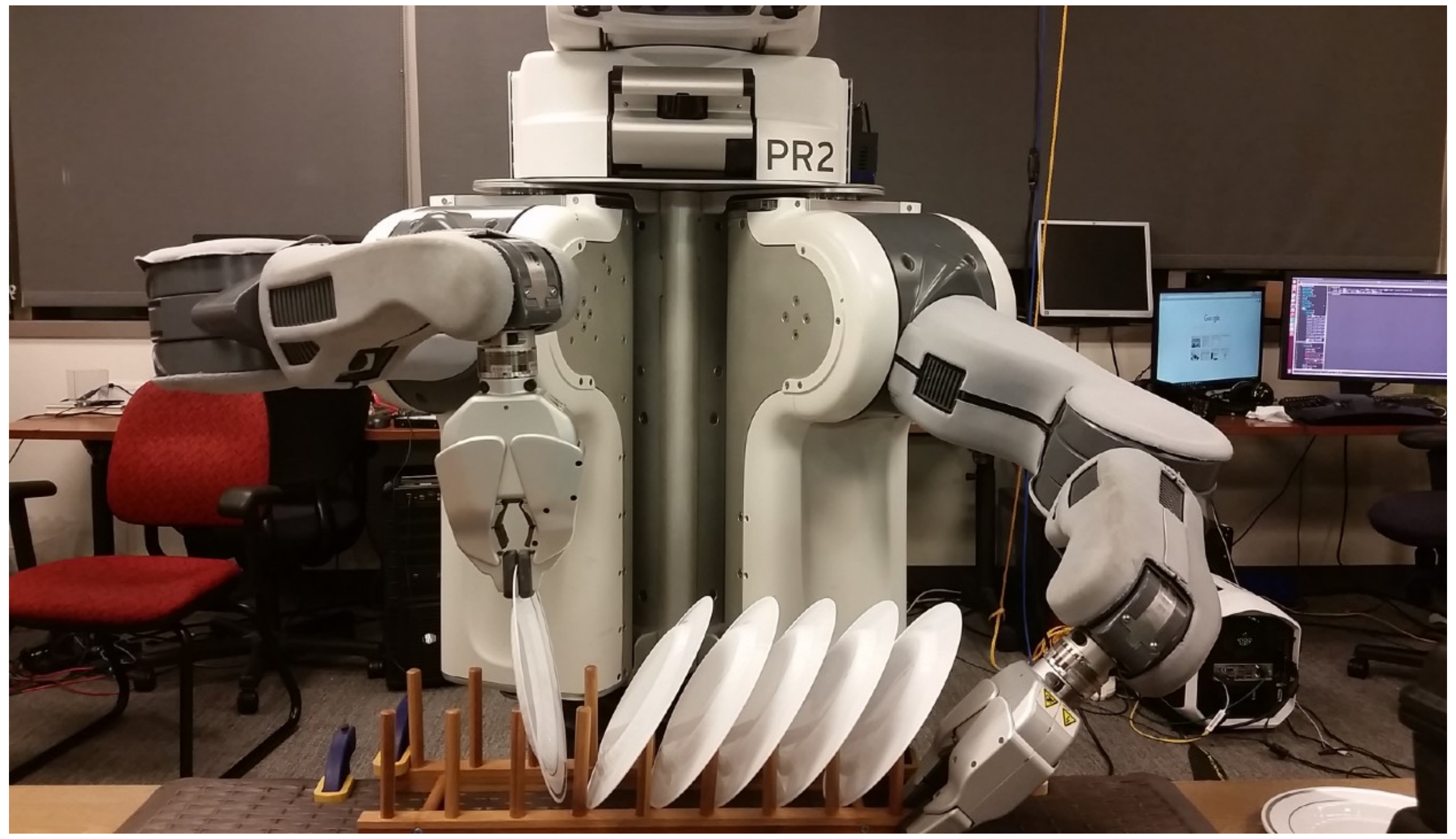
Baram et al. ICML '17: use learned dynamics model to backdrop through discriminator

(Finn*, Christiano*, et al. '16)

Guided Cost Learning Experiments

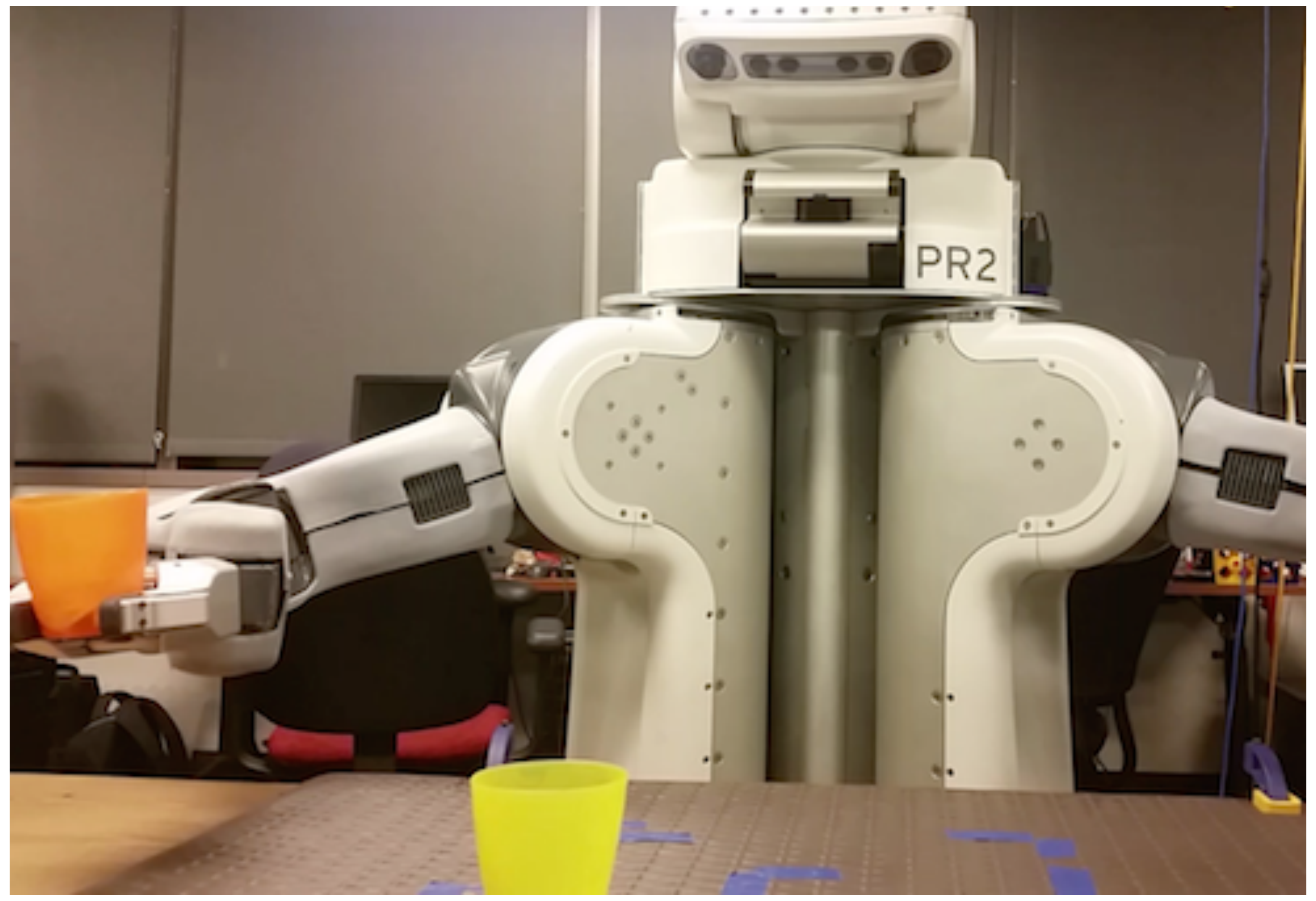
Real-world Tasks

dish placement



state includes goal plate pose

pouring almonds



state includes unsupervised visual features [Finn et al. '16]

Comparison

Relative Entropy IRL
(Boularias et al. '11)

Comparisons

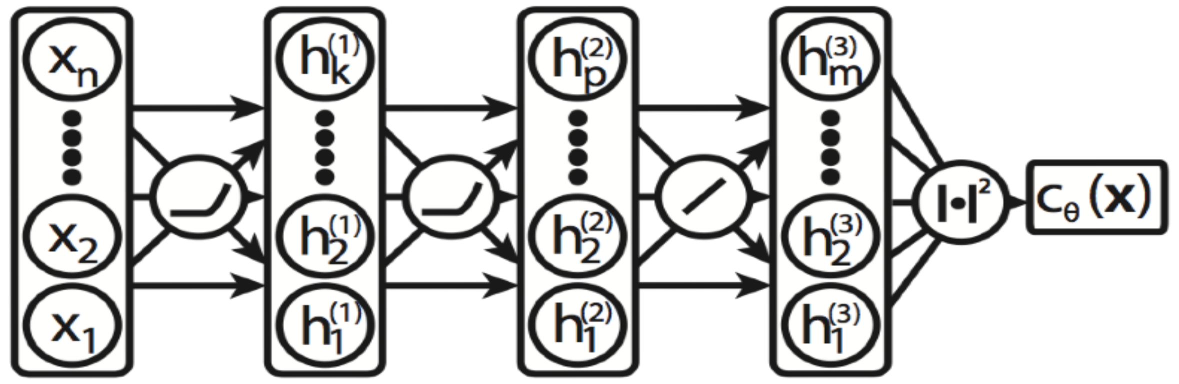
Path Integral IRL
(Kalakrishnan et al. '13)

Relative Entropy IRL
(Boularias et al. '11)

initial distribution q_0

generate policy samples from q

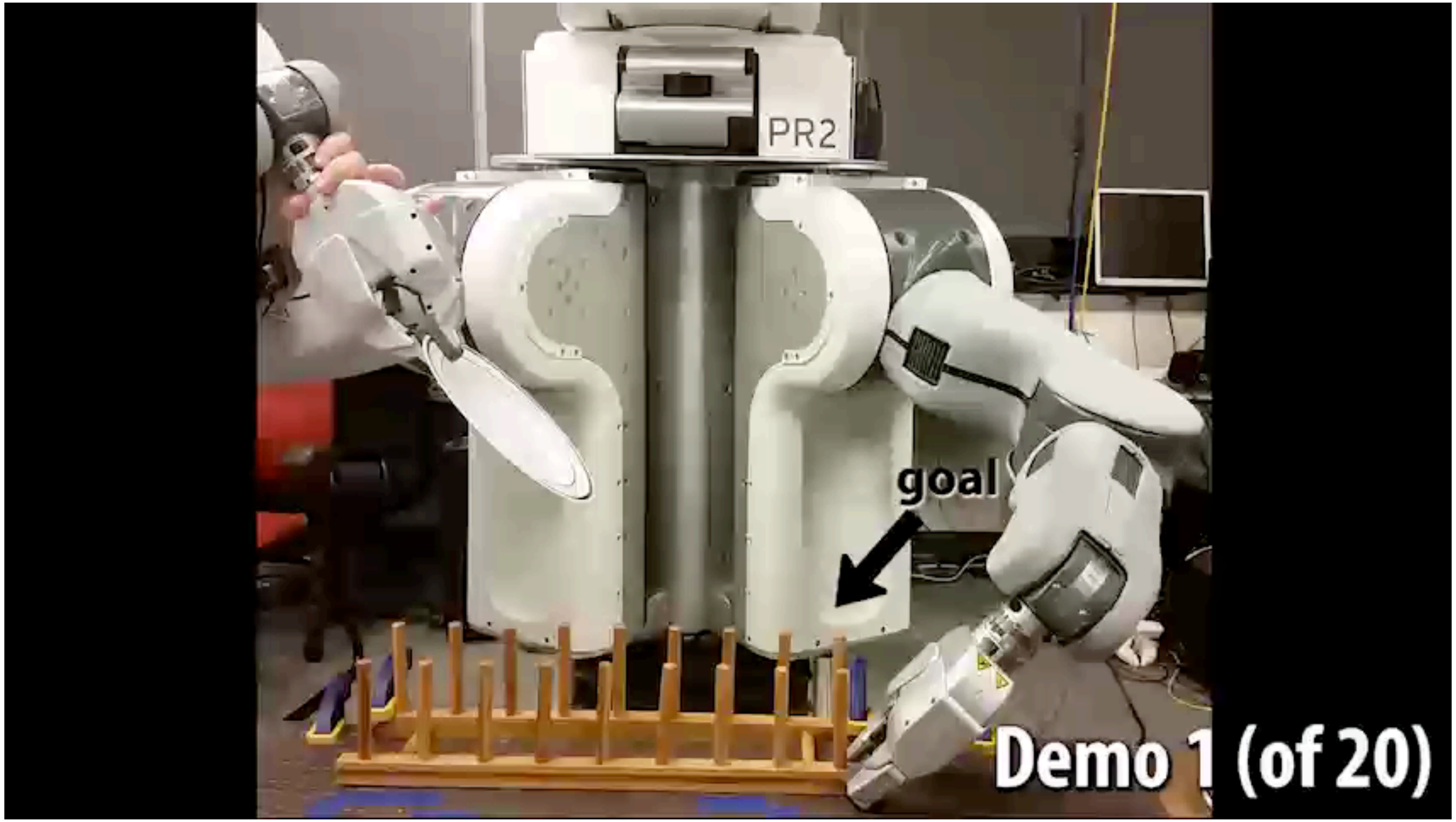
human demonstrations



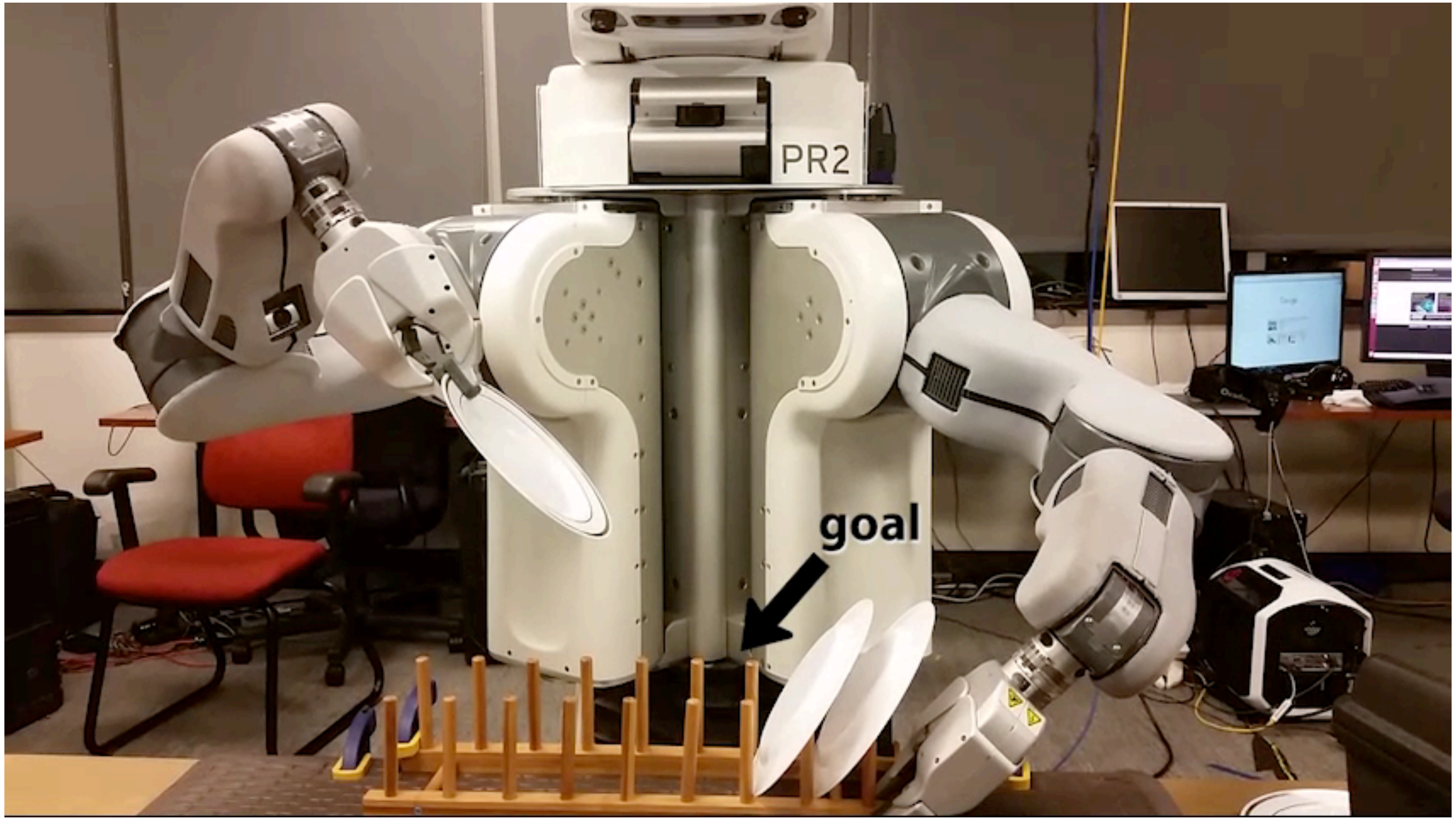
Update reward using samples & demos

reward r

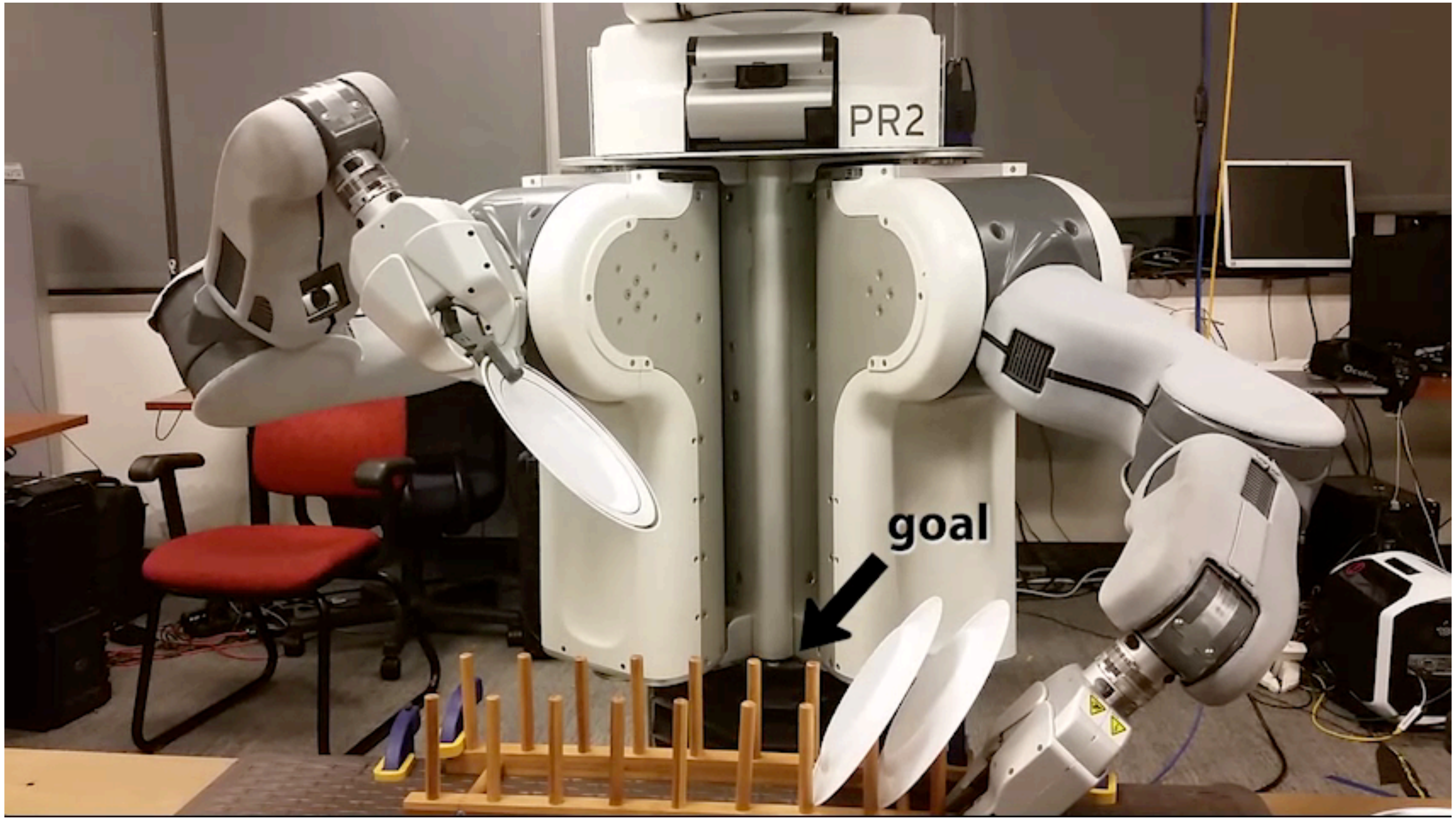
Dish placement, demos



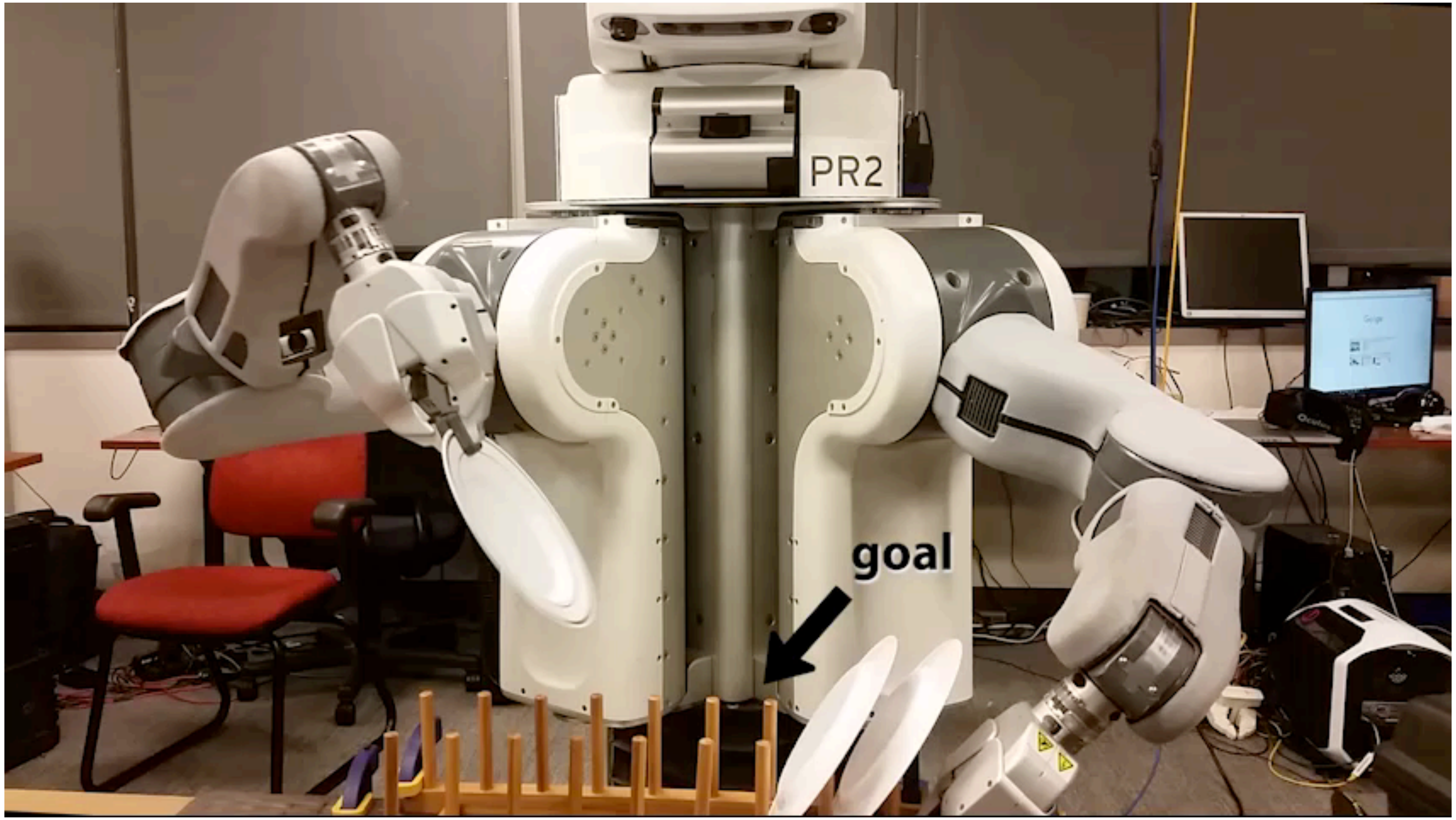
Dish placement, standard reward



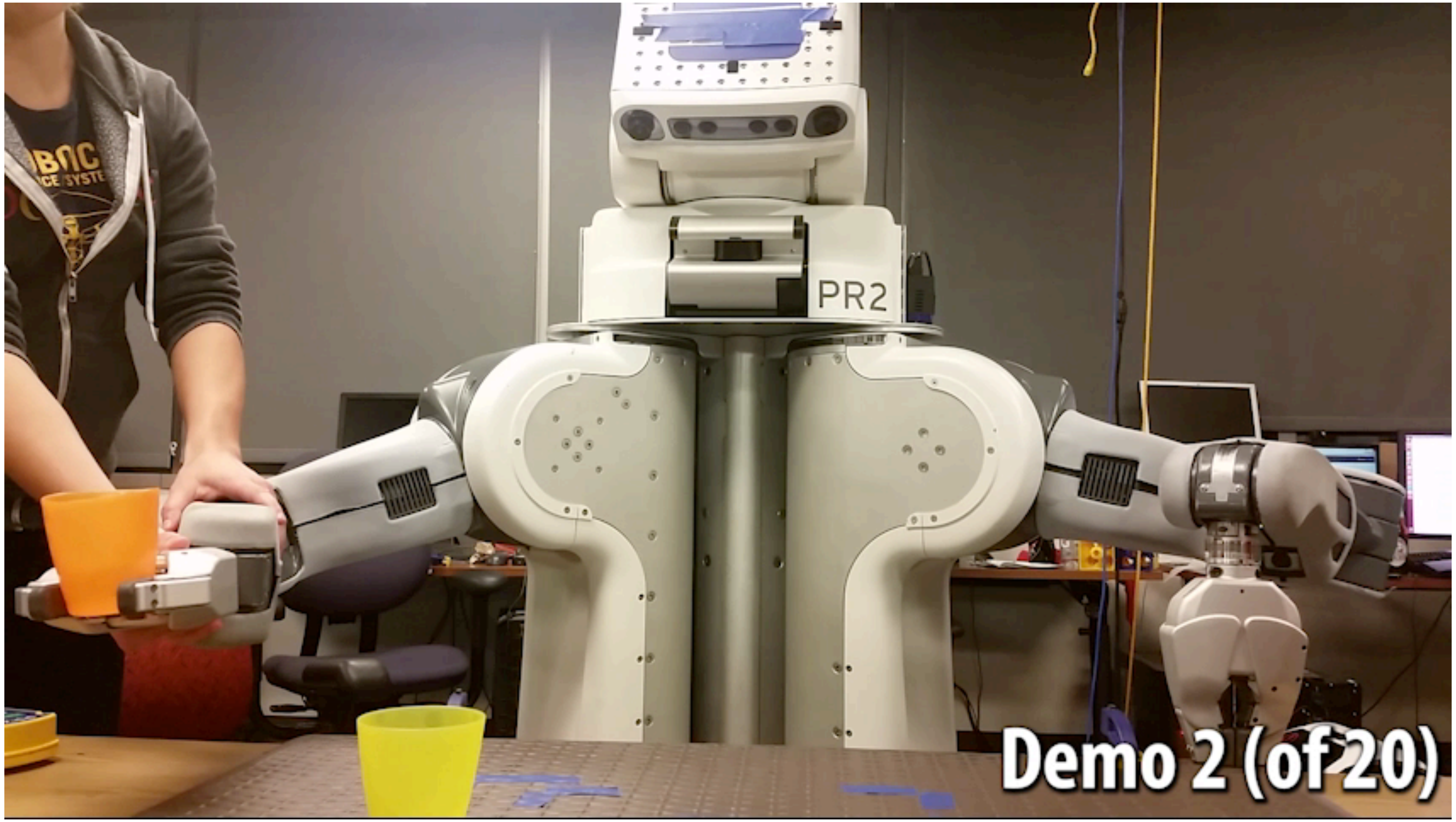
Dish placement, RelEnt IRL



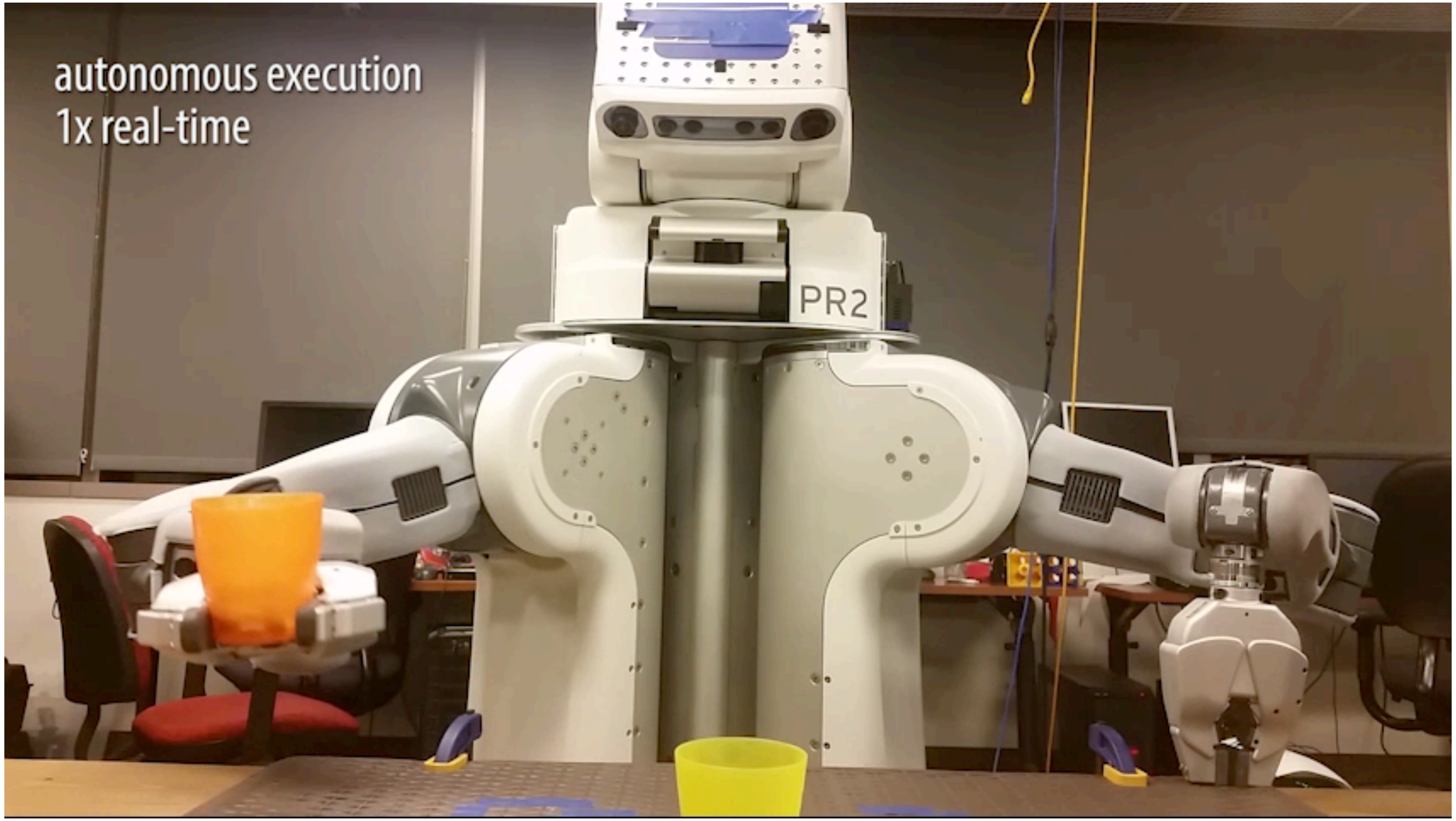
Dish placement, GCL policy



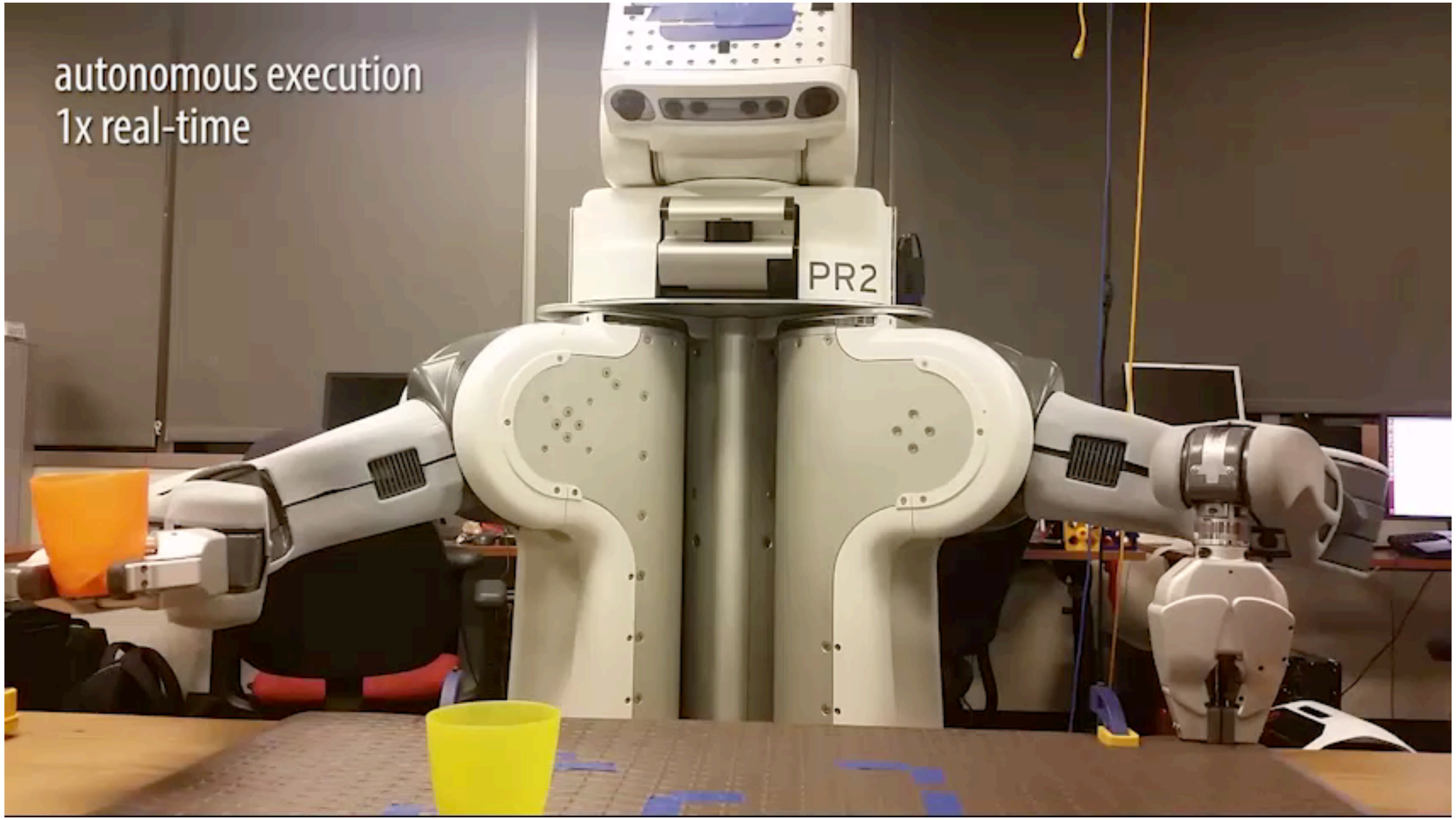
Pouring, demos



Pouring, RelEnt IRL



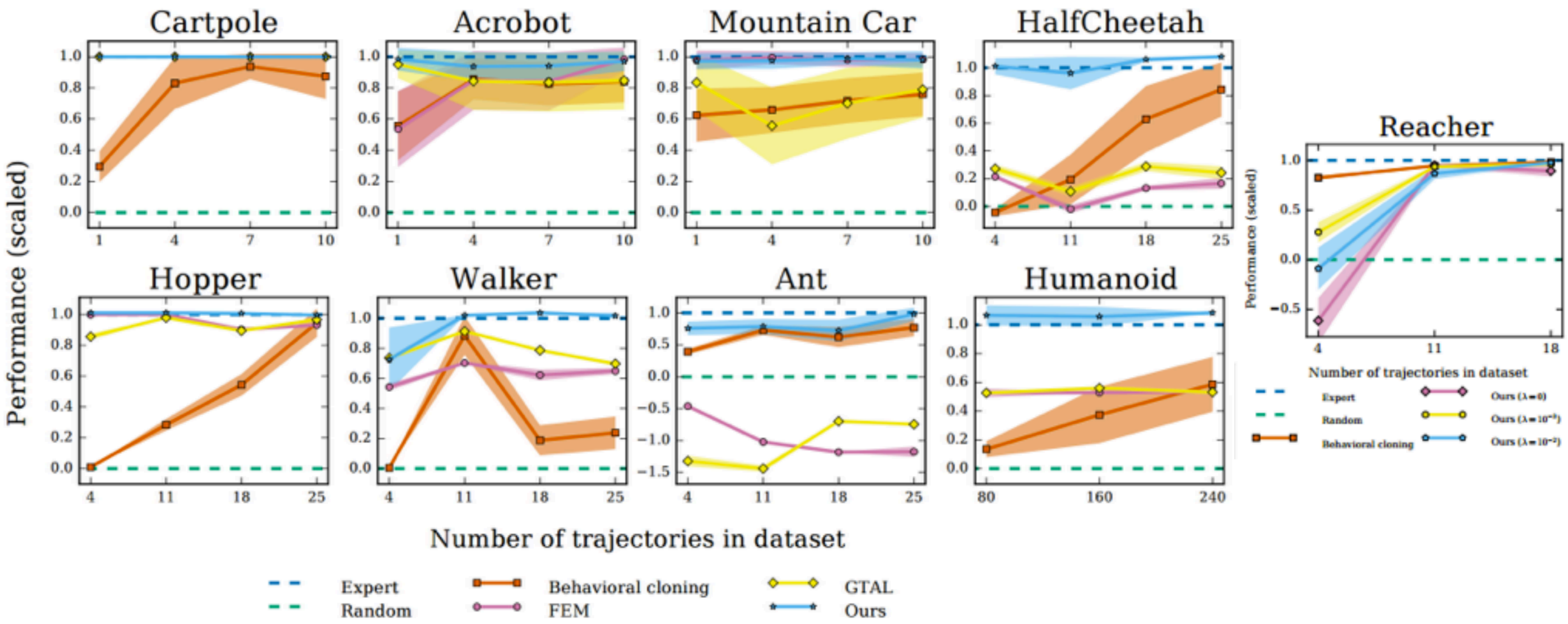
Pouring, GCL policy



Generative Adversarial Imitation Learning Experiments

(Ho & Ermon NIPS '16)

- demonstrations from TRPO-optimized policy
- use TRPO as a policy optimizer

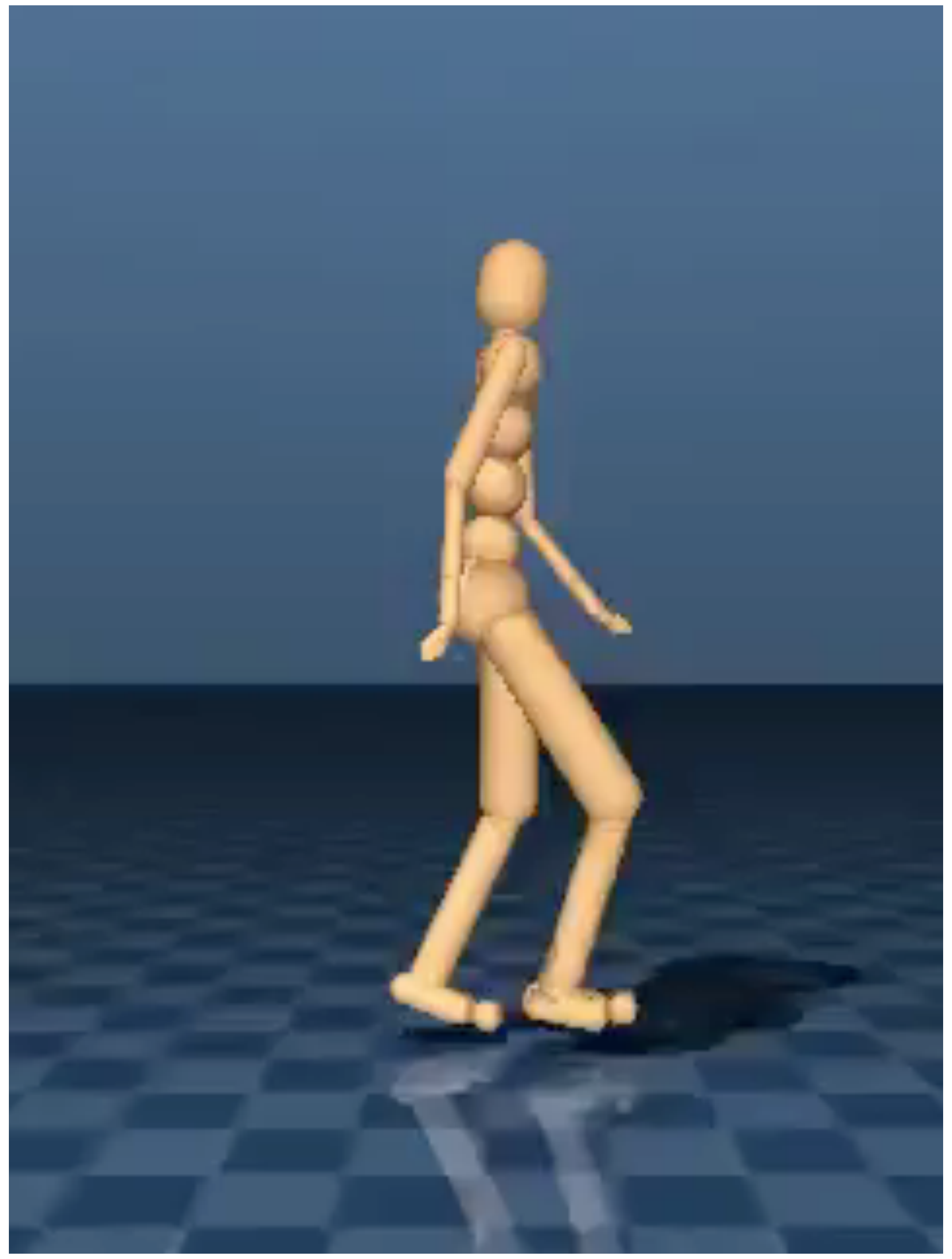


Conclusion: IRL requires fewer demonstrations than behavioral cloning

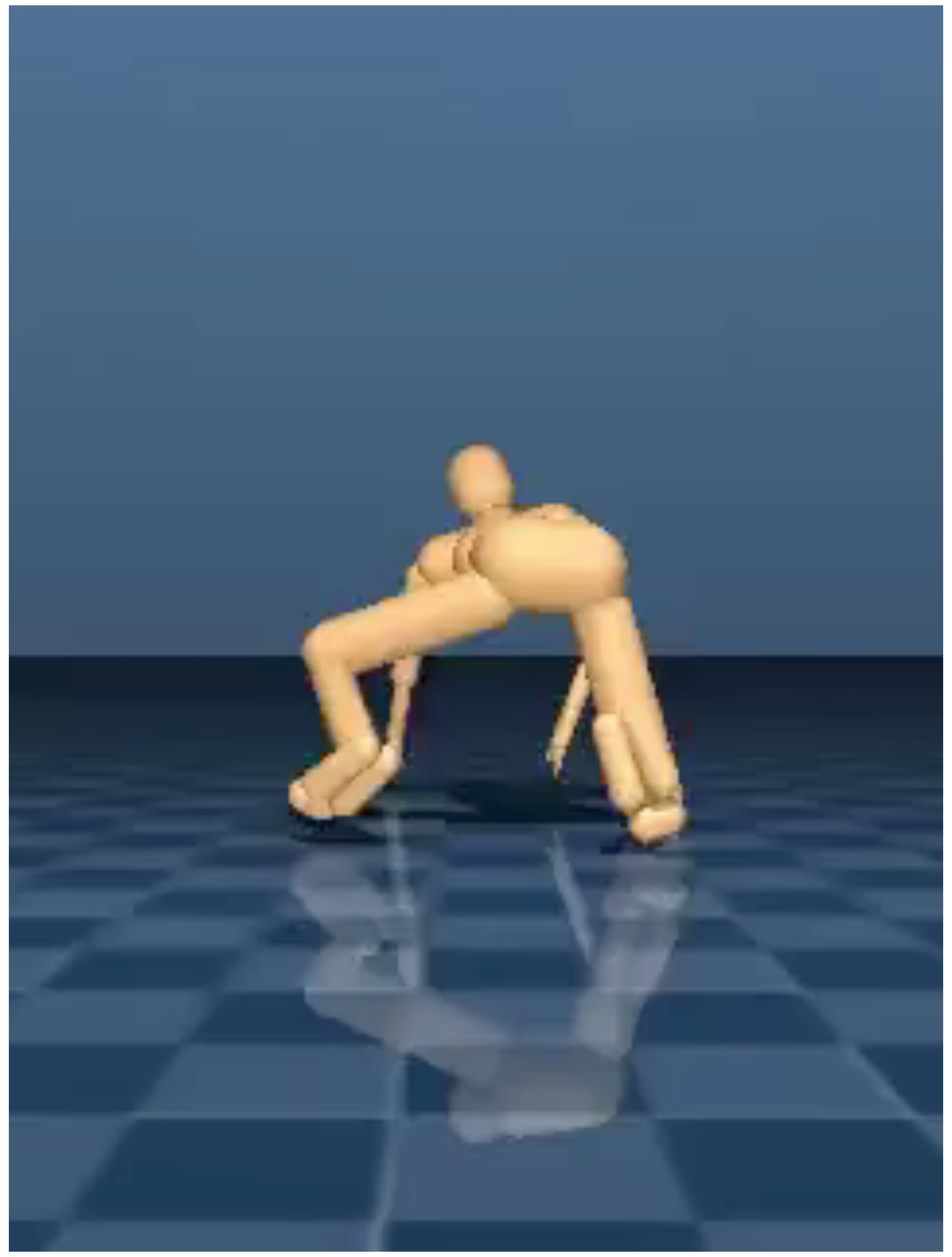
Generative Adversarial Imitation Learning Experiments

(Ho & Ermon NIPS '16)

learned behaviors from human motion capture
Merel et al. '17



walking



falling & getting up

GCL & GAIL: Pros & Cons

Strengths

- can handle unknown dynamics
- scales to neural net costs
- efficient enough for real robots (with an efficiency policy optimizer)

Limitations

- adversarial optimization is hard
- can't scale to raw pixel observations of demos
- demonstrations typically collected with kinesthetic teaching or teleoperation (first person)

Inverse Reinforcement Learning Review

Acquiring a reward function is important (and challenging!)

Goal of Inverse RL: infer reward function underlying expert demonstrations

Evaluating the partition function:

- initial approaches solve the MDP in the inner loop and/or assume known dynamics
- with unknown dynamics, estimate Z using samples

Connection to generative adversarial networks:

- sampling-based MaxEnt IRL is a GAN with a special form of discriminator and uses RL to optimize the generator

Suggested Reading on Inverse RL

<u>Classic Papers</u>:

Abbeel & Ng ICML ’04. *Apprenticeship Learning via Inverse Reinforcement Learning.* Good introduction to inverse reinforcement learning

Ziebart et al. AAAI ’08. *Maximum Entropy Inverse Reinforcement Learning.* Introduction to probabilistic method for inverse reinforcement learning

<u>Modern Papers</u>:

Wulfmeier et al. arXiv ’16. *Deep Maximum Entropy Inverse Reinforcement Learning.* MaxEnt inverse RL using deep reward functions

Finn et al. ICML ’16. *Guided Cost Learning.* Sampling based method for MaxEnt IRL that handles unknown dynamics and deep reward functions

Ho & Ermon NIPS ’16. *Generative Adversarial Imitation Learning.* Inverse RL method using generative adversarial networks

Further Reading on Inverse RL

MaxEnt-based IRL: Ziebart et al. AAAI '08, Wulfmeier et al. arXiv '16, Finn et al. ICML '16
Adversarial IRL: Ho & Ermon NIPS '16, Finn*, Christiano* et al. arXiv '16, Baram et al. ICML '17
Handling multimodality: Li et al. arXiv '17, Hausman et al. arXiv '17, Wang, Merel et al. arXiv '17
Handling domain shift: Stadie et al. ICLR '17

Questions?

IOC is under-defined

need regularization:

- encourage slowly changing cost

$$g_{\text{lcr}}(\tau) = \sum_{x_t \in \tau} [(c_\theta(x_{t+1}) - c_\theta(x_t)) - (c_\theta(x_t) - c_\theta(x_{t-1}))]^2$$

- cost of demos decreases strictly monotonically in time

$$g_{\text{mono}}(\tau) = \sum_{x_t \in \tau} [\max(0, c_\theta(x_t) - c_\theta(x_{t-1}) - 1)]^2$$

Regularization ablation