CS 690S: AI Alignment

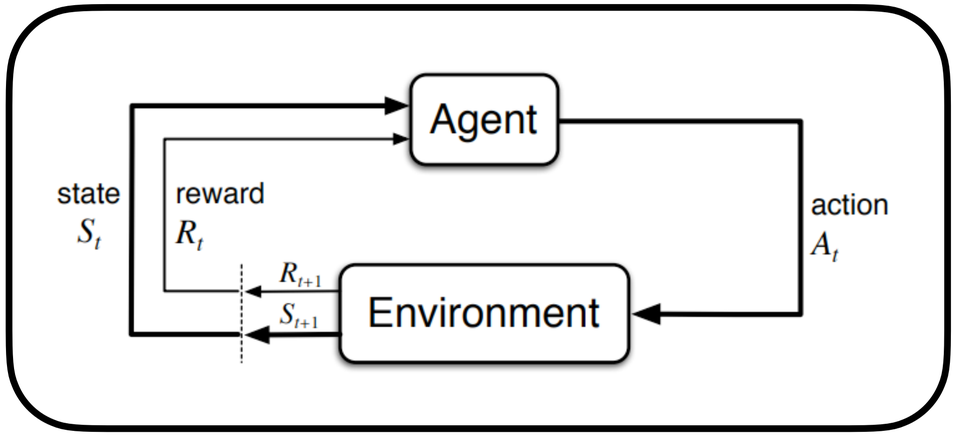
Prof. Scott Niekum

Behavioral Cloning

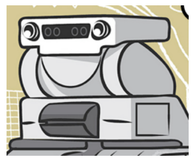
Reinforcement Learning

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

π^* $\quad \pi : S \times A \to [0, 1]$

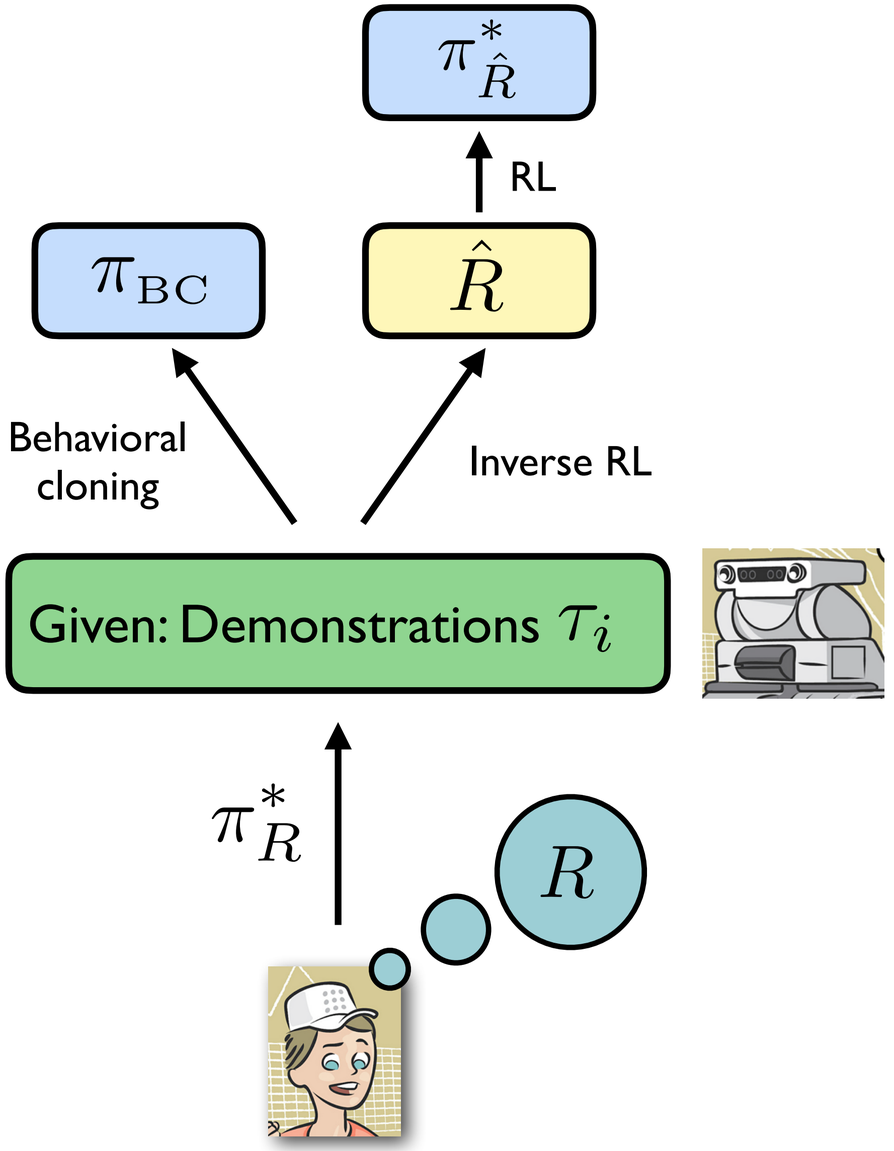


RL

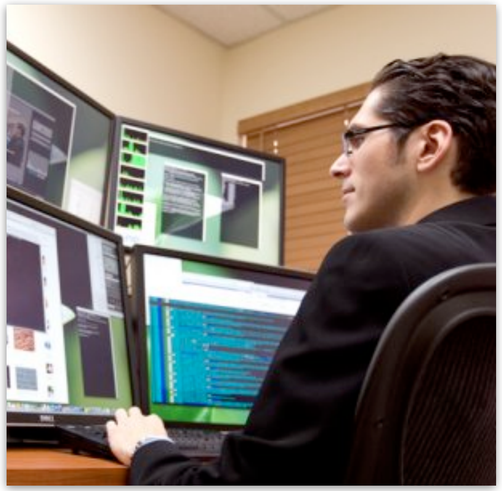


Given: R $\quad R : S \to \mathbb{R}$

Imitation Learning

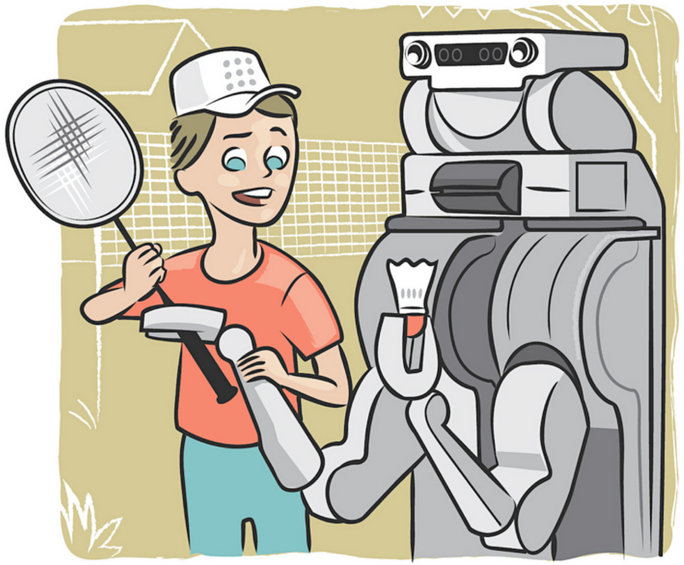


Why learn from demonstrations?



General purpose robot → Specific task → Expert engineer

Why learn from demonstrations?



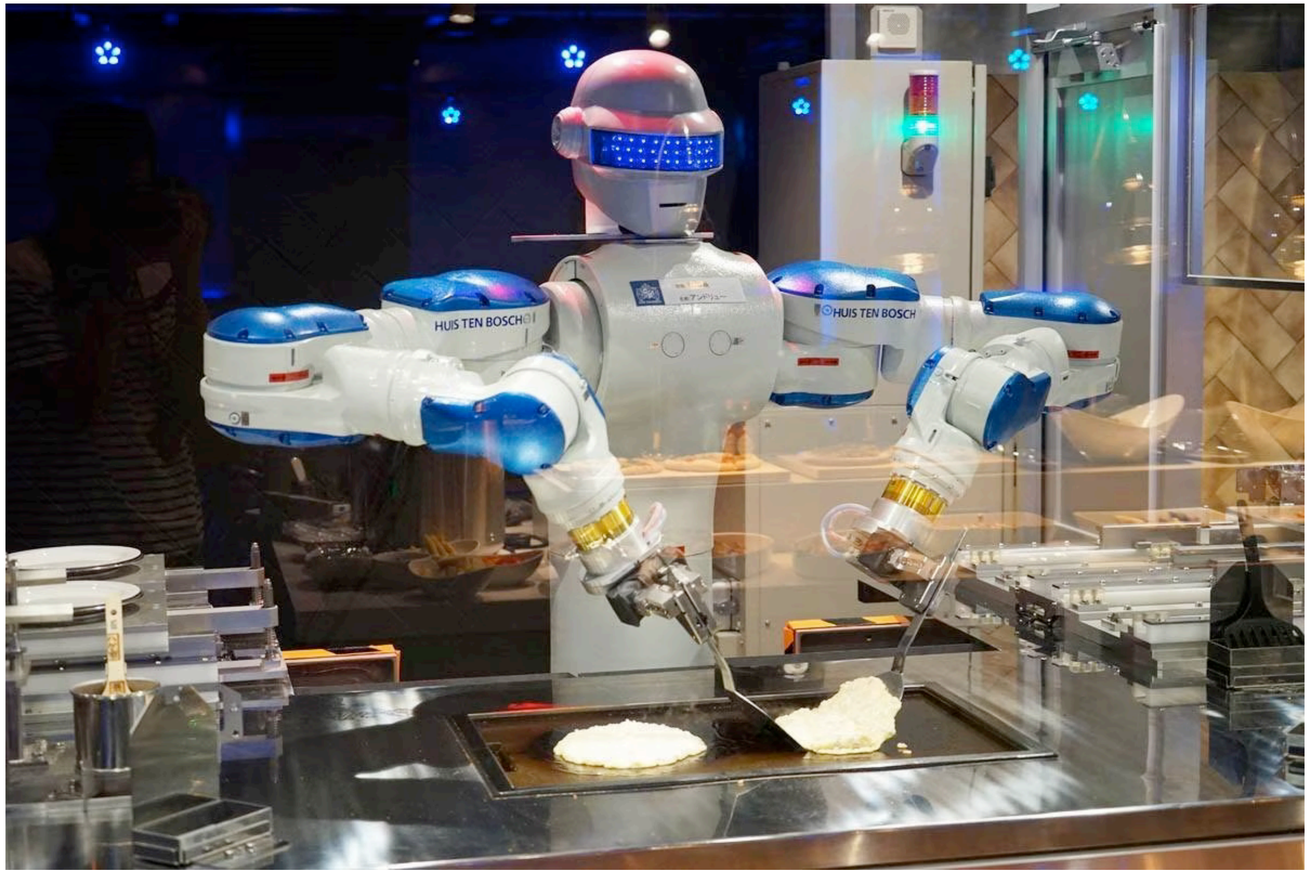
- Natural and expressive
- No expert knowledge required
- Valuable human intuition
- Program new tasks as-needed

General purpose robot → Specific task → End user

Why learn from demonstrations?

What reward function describes a good cartwheel?

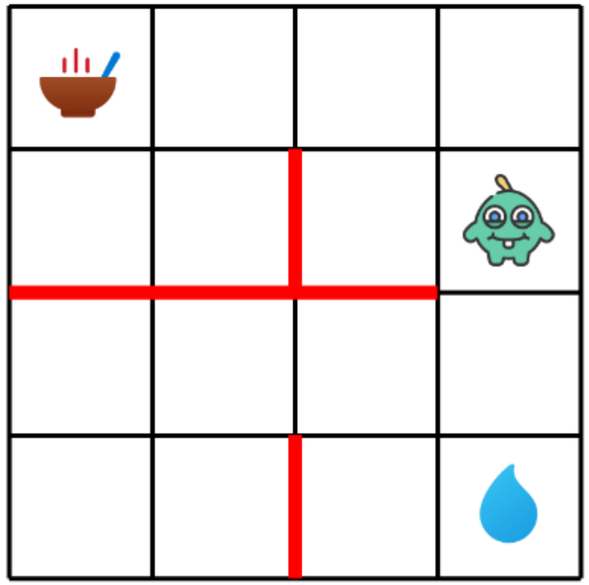
Why learn from demonstrations?



Why learn from demonstrations?

The Perils of Trial-and-Error Reward Design:
Misdesign through Overfitting and Invalid Task Specifications

Serena Booth[1,2,3], W. Bradley Knox[1,2,5], Julie Shah[3],
Scott Niekum[2,4], Peter Stone[2,6], Alessandro Allievi[1,2]



Why learn from demonstrations?

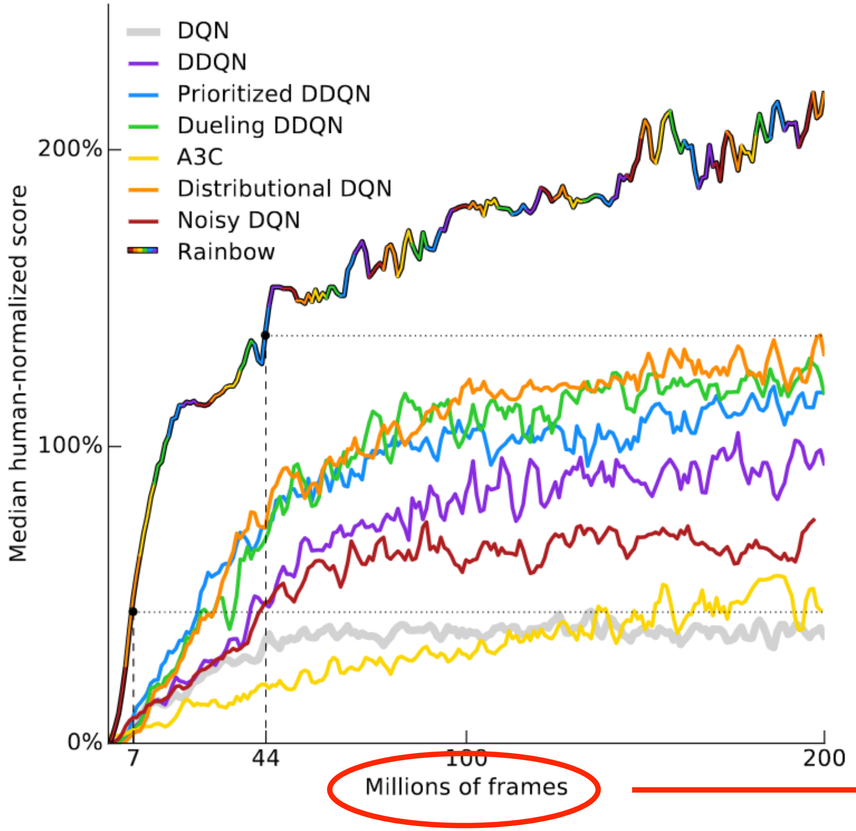
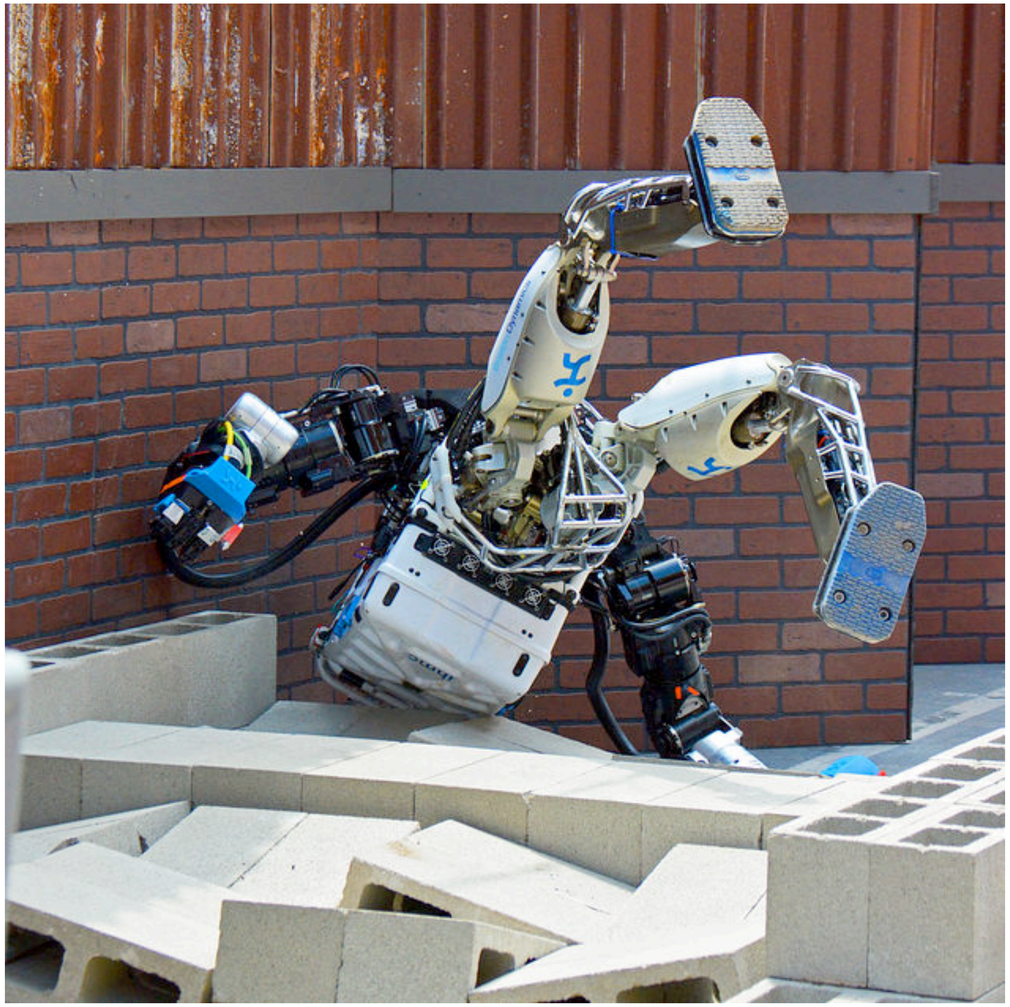


Figure 1: **Median human-normalized performance** across 57 Atari games. We compare our integrated agent (rainbow-

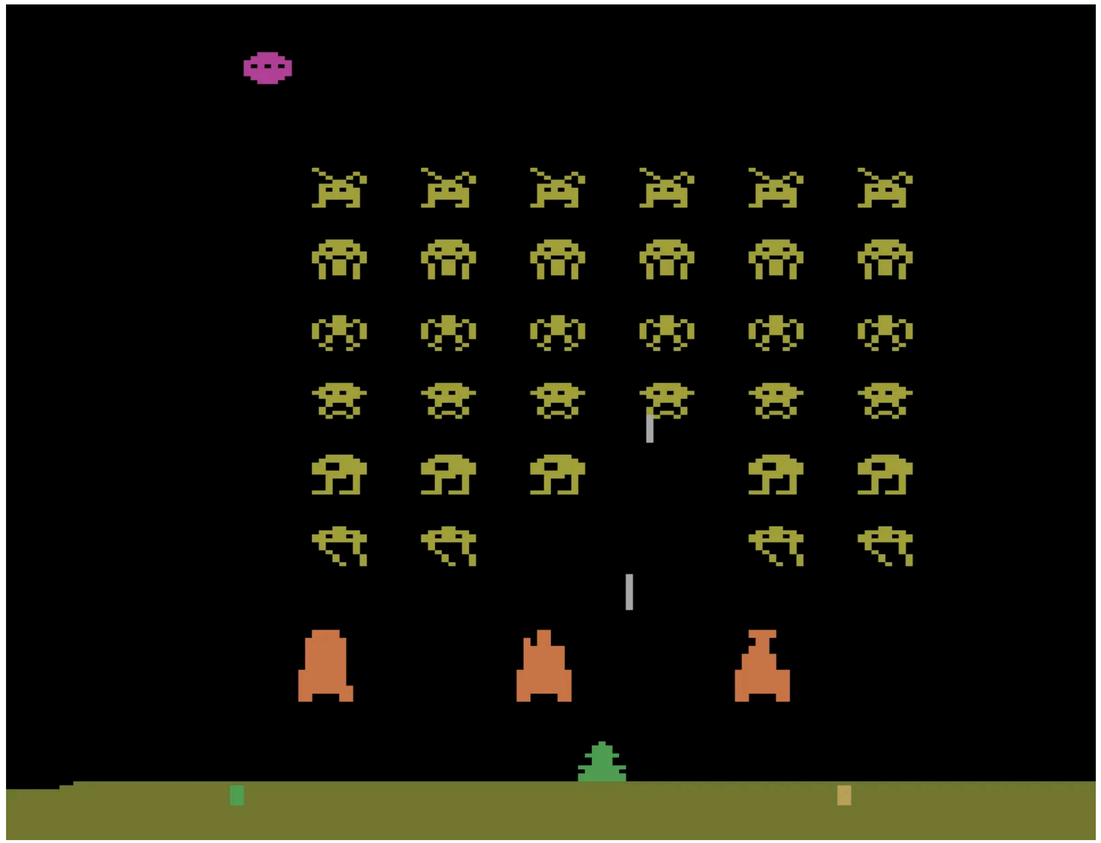
Millions!

Hessel, Matteo, et al. "Rainbow: Combining improvements in deep reinforcement learning." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. No. 1. 2018.

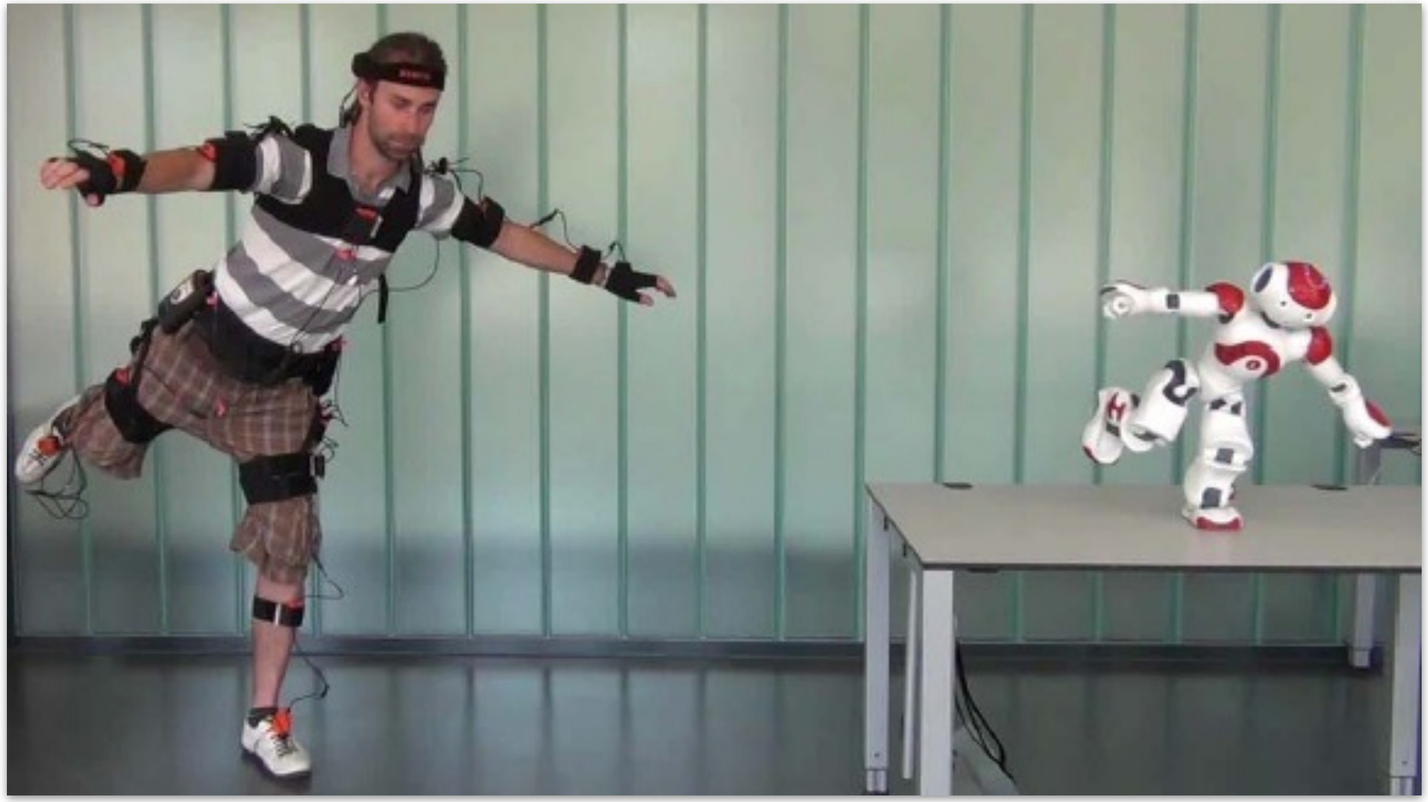
Why learn from demonstrations?



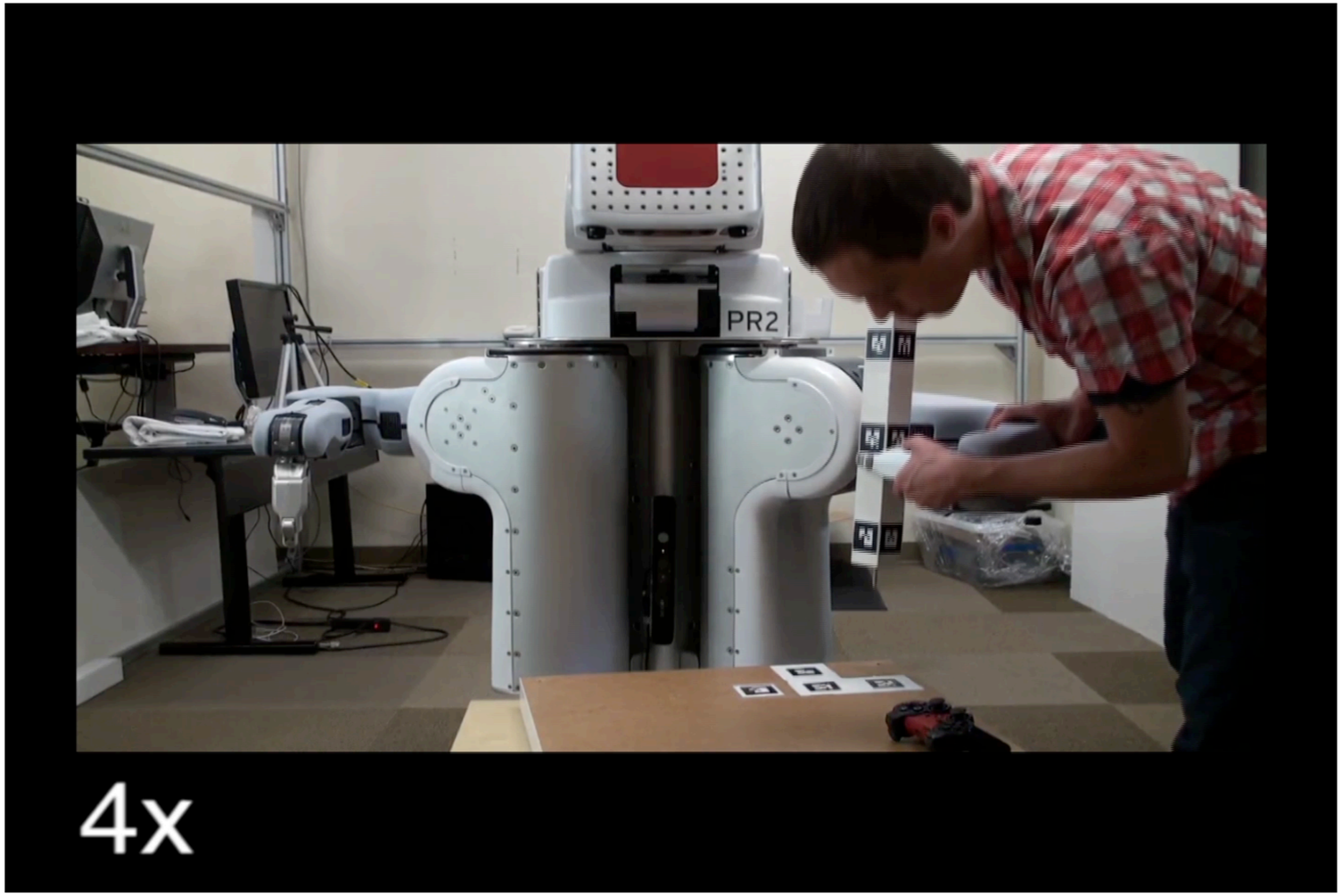
How to provide demonstrations?



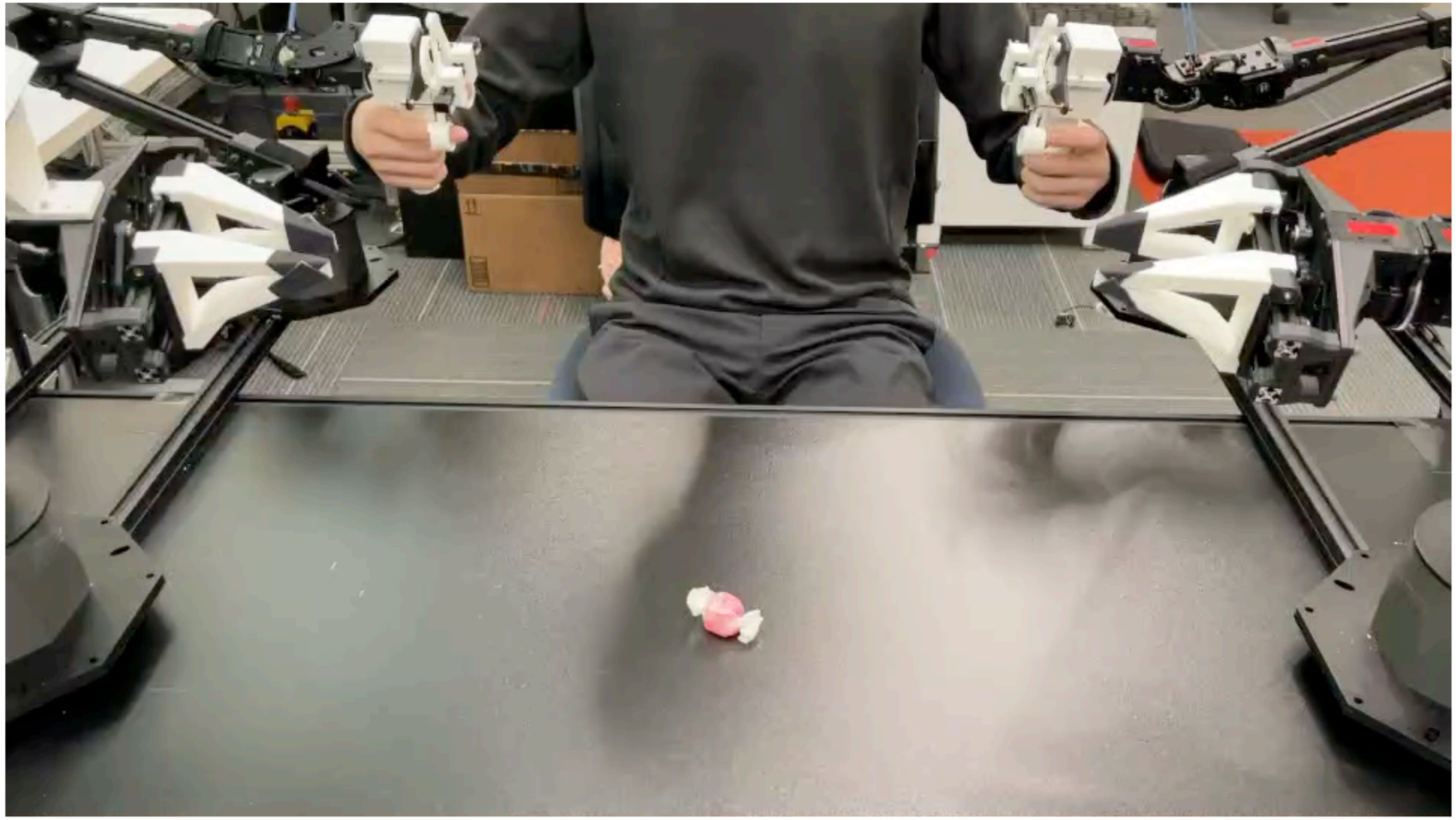
How to provide demonstrations?



How to provide demonstrations?



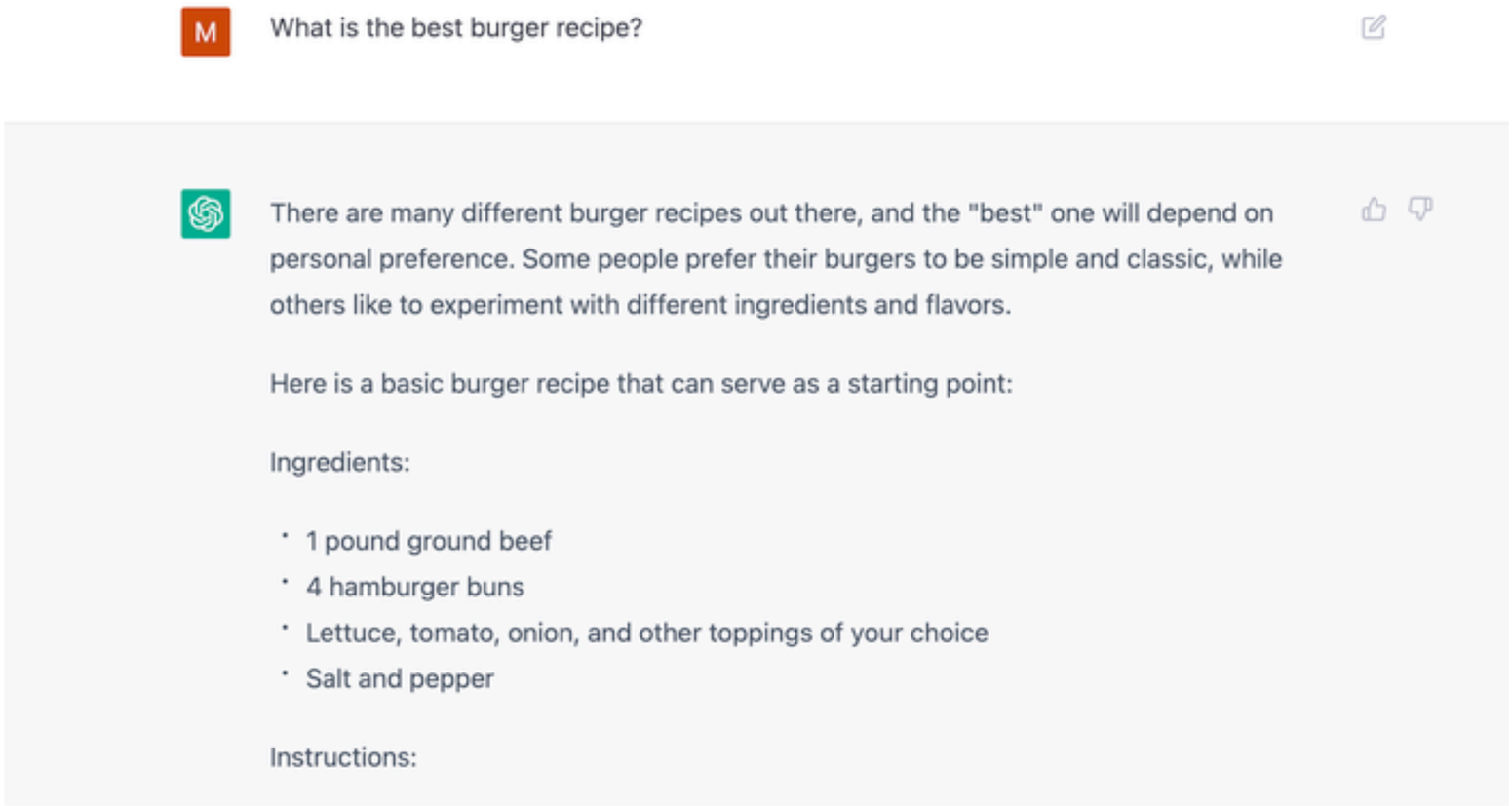
How to provide demonstrations?



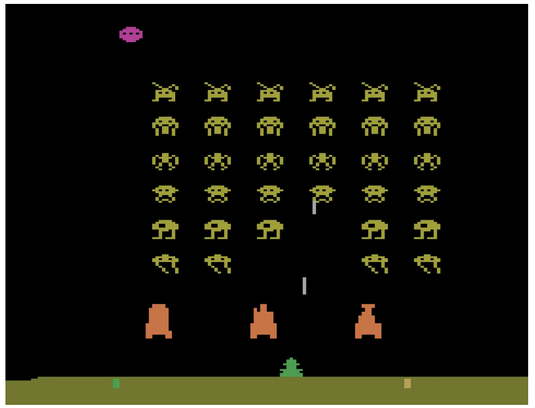
Fu, Zipeng, Tony Z. Zhao, and Chelsea Finn. "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation." *arXiv preprint arXiv:2401.02117* (2024).

How to provide demonstrations?

How to provide demonstrations?



Behavioral cloning



$$D = \{s_t, a_t, s_{t+1}\}_N$$

Learn: $\pi : S \rightarrow A$

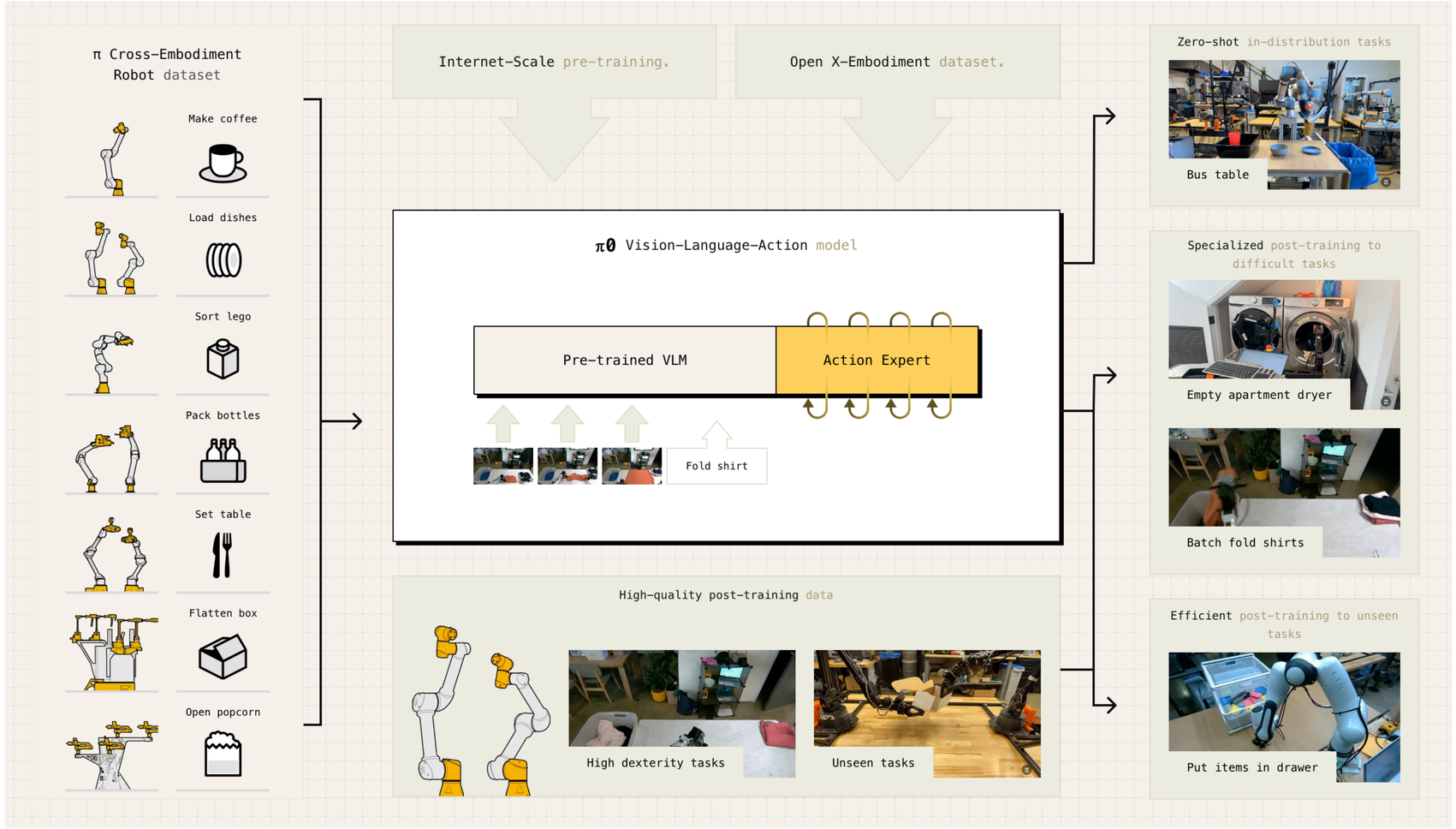
or more generally: $\pi(s, a) := p(a|s)$

Straightforward supervised learning problem

Scaling up BC: $\pi_{0.6}$

https://www.pi.website/blog/pistar06

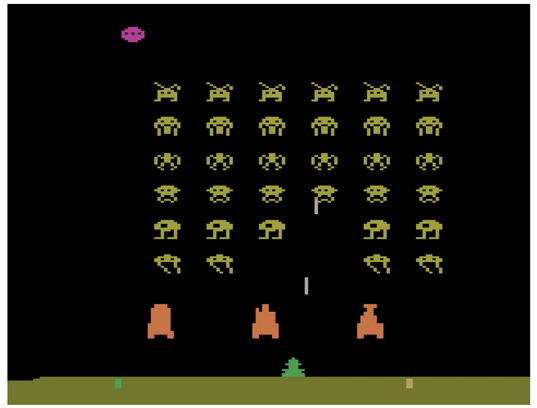
Scaling up BC: π0 (predecessor to 0.6)



https://www.pi.website/blog/pi0

How to provide demonstrations?

Behavioral cloning



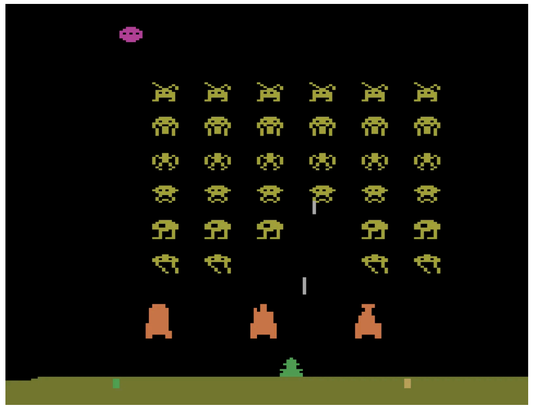
$$D = \{s_t, a_t, s_{t+1}\}_N$$

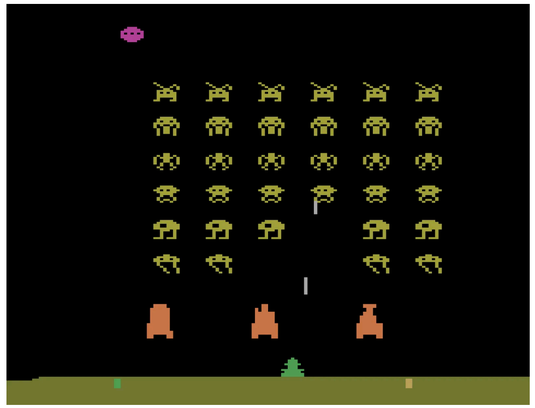
Learn: $\pi : S \rightarrow A$

or more generally: $\pi(s, a) := p(a|s)$

Straightforward supervised learning problem

Behavioral cloning from observation





$$D = \{s_t, \cancel{a_t}, s_{t+1}\}_N$$

Learn: $\pi : S \rightarrow A$

or more generally: $\pi(s, a) := p(a|s)$

How to infer action that caused transition from s_t to s_{t+1} ?

Inverse dynamics

Dynamics: $p(s_{t+1}|s_t, a_t)$

Inverse dynamics: $p(a_t|s_t, s_{t+1})$

Learn inverse dynamics from offline data:

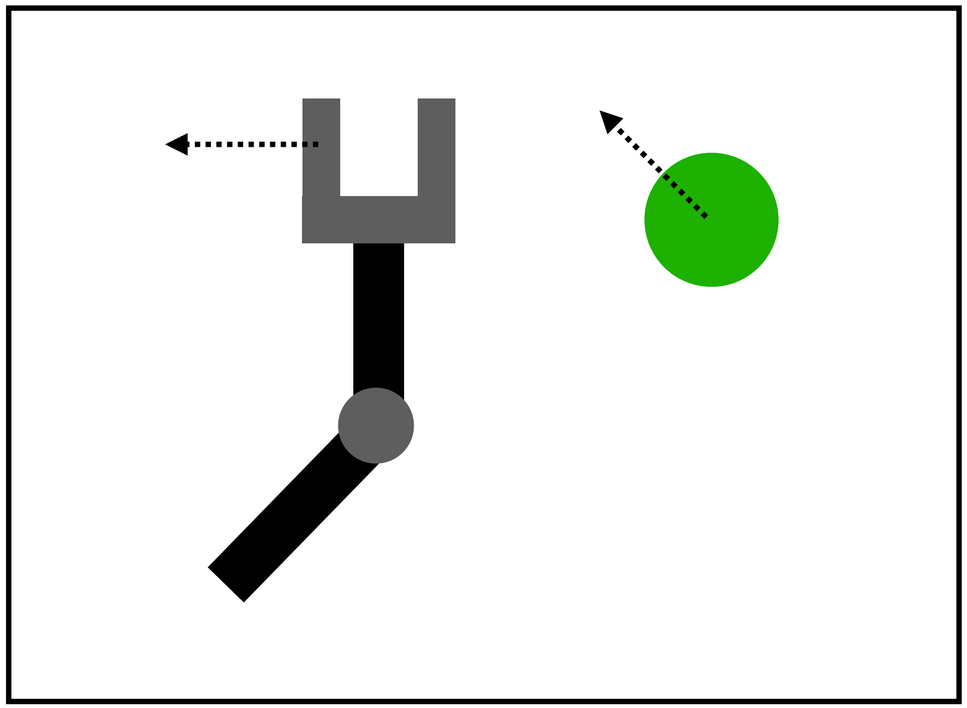
$$\theta^* = \arg\max_\theta \prod_{i=0}^{|\mathcal{I}^{pre}|} p_\theta(a_i \,|s_i^a, s_{i+1}^a)$$

…and guess the missing demonstration actions: $D = \{s_t, \cancel{a_t}, s_{t+1}\}_N$

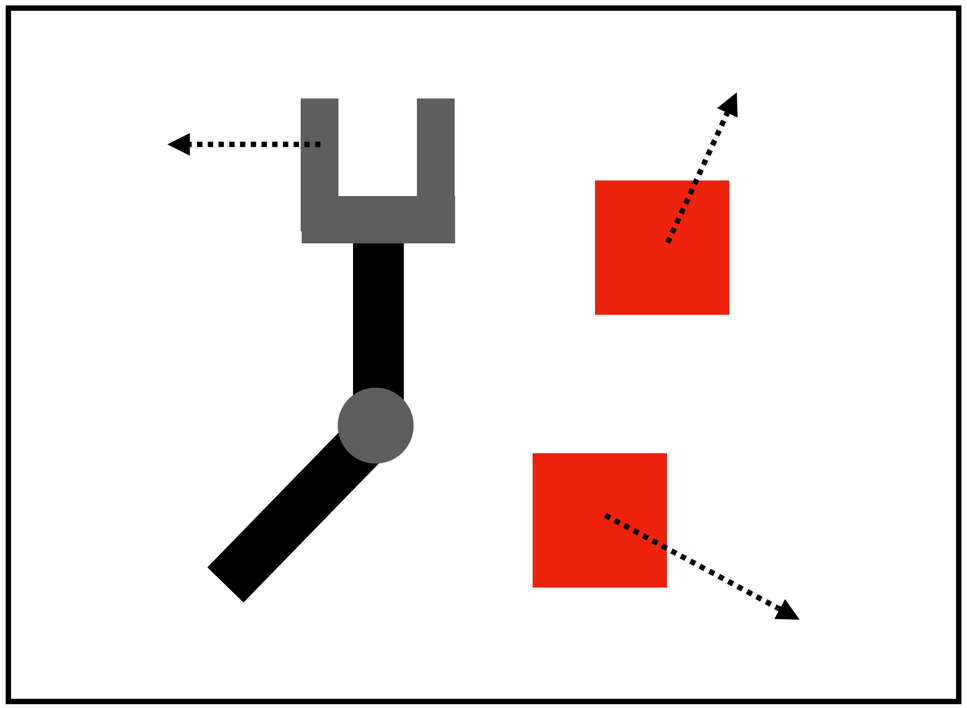
Now we're back to standard BC problem!

Agent-specific vs. task-specific state

$$S = S^a \times S^t$$

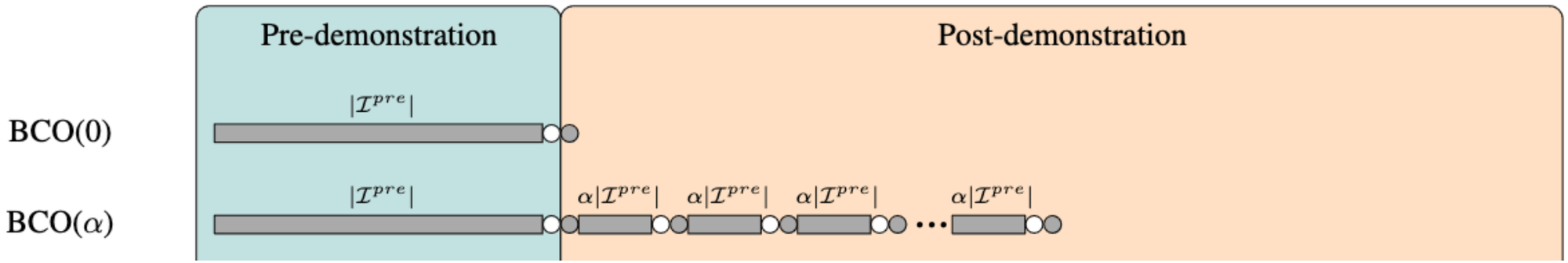


Inverse dynamics learning task



BCO task

BCO(alpha)


Pre-demonstration
Post-demonstration
BCO(0)
BCO(α)
$|\mathcal{I}^{pre}|$
$|\mathcal{I}^{pre}|$
$\alpha|\mathcal{I}^{pre}|$
$\alpha|\mathcal{I}^{pre}|$
$\alpha|\mathcal{I}^{pre}|$
$\alpha|\mathcal{I}^{pre}|$

Analyzing a paper

- Warning: the next paper (A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning) is challenging
- You don’t need to understand everything!
- Analyze at simplest conceptual level before getting stuck in details (and ignore details that you don’t need)
- Be systematic!

Analyzing a paper: Claims (1/3)

- Claims:
  - An inverse model can be learned from pre-demonstration data
  - A *task-agnostic* inverse dynamics model can be learned
  - BCO can accurately imitate with observation-only data
  - BCO allows for imitation without post-demonstration interaction
  - However, post-demonstration data helps, if you’re willing to collect it
  - BCO is better with less data than competing approaches

Analyzing a paper: Baselines

- GAIL and FEM
  - We'll look at methods like these later in the course
  - At a high-level, they aim to match state-action occupancies / feature expectations
  - To do so, need to have post-demonstration data to learn policies that match the state/features well

Analyzing a paper: Claims (2/3)

- Claims supported?
  - An inverse model can be learned from pre-demonstration data
    - No direct experiments testing accuracy of inverse dynamics model, only done in context of how it effects downstream policy learning
    - No experiments examining effects of different amounts of pre-demonstration data
    - If BCO works well overall, then the inverse model must be decent
  - A *task-agnostic* inverse dynamics model can be learned
    - Reacher domain has partitioned agent/task state space
    - Only assess accuracy of overall algorithm, not inverse model
    - Partitioned by hand, and they don't show consequences of not partitioning

Analyzing a paper: Claims (3/3)

- Claims supported?
  - BCO can accurately imitate with observation-only data / without post-demonstration interaction / with less data than competing approaches
    - Performance much better than random, often close to expert, and often close to action-aware BC.
    - Competitive with baselines that have access to actions (kind of like having infinite pre-demonstration data)
    - Performance steadily improves with additional pre-demonstration data
    - 20 trials + small standard error bars provides confidence of correctness of results
    - They show that other methods (e.g. GAIL) need many post-demonstration interactions to do as well as BCO(0). But these are very simple domains, and GAIL has better guarantees than BCO outside of the support of the demonstrations.
  - Post-demonstration data helps, if you're willing to collect it
    - Performance increases as alpha increases and approaches action-aware BC

Analyzing a paper: Reproducibility

- Details provided of each domain, neural network architectures, number of interactions, etc.
- While architecture is specified for dynamics model, they don’t say what it is for policy
- Not clear if/how hyperparameters for GAIL and FEM were tuned

Analyzing a paper: Questions

- All experiments were with synthetic demonstrations from TRPO. How close to optimal were they? Does BCO's performance gracefully degrade with noisy demonstrations, e.g. from humans?

- Can agent/task state space partitioning be learned? How much does performance suffer if you don't partition?

- Domains were quite simple, but the motivation in the introduction was learning from Youtube videos. What would it take to scale? Can partitioning be learned (implicitly, perhaps)?

Analyzing a paper: Citations

BCO cites:

Niekum, S., Osentoski, S., Konidaris, G., Chitta, S., Marthi, B., & Barto, A. G. (2015). Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research*, *34*(2), 131-157.

- Also imitates from observations
- Doesn't need actions due to robot arm controller with known functional form and simplistic generalization based only on changing start/goal locations
- Automatically segments and reuses sub-skills for complex, multi-step tasks

BCO cited by:

Torabi, F., Warnell, G., & Stone, P. (2018). Generative adversarial imitation from observation. *In ICML Workshop on Imitation, Intent, and Interaction*. *arXiv preprint arXiv:1709.04905, 2019*

- Same authors!
- Aims to address compounding error that BCO can experience due to being purely supervised
- Essentially GAIL, but from observation-only data
- Has above advantages compared to BCO, but also requires post-demonstration data

Analyzing a paper: Future work ideas

- **Study approaches and effects of automatic agent/task state partitioning:**
  - In a simple domain, study performance degradation as task-specific variables are leaked into agent's state space for inverse model.
  - Learn a partitioning automatically
  - How does domain complexity influence the above? E.g. real-world robotic manipulation from video vs. reacher domain?