Principles of Robot Autonomy II

Imitation Learning

Stanford University

iliad

intelligent and interactive autonomous systems

Today’s itinerary

- Intro to Imitation Learning
- Behavioral Cloning
- Imitation Learning with Interactive Experts
- Inverse RL (MMP, Max Ent IRL)
- Learning from other sources of data (preferences, physical feedback)

Today’s itinerary

- Intro to Imitation Learning
- Behavioral Cloning
- Imitation Learning with Interactive Experts
- Inverse RL (MMP, Max Ent IRL)
- Learning from other sources of data (preferences, physical feedback)

Why Imitation Learning?

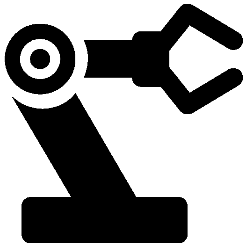
For the Sake of Robot Learning:

- It is difficult to learn from sparse rewards (unless data is cheap and you don't care about seeing lots of failures).
- Hand-designing rewards is hard.

Just design the right reward function

$R_H(s)$

$a_R^* = \text{argmax}_{a_R} R_H(s)$




START
START
SCORE
0
LAPS
--/3
TIME
0:01
TURBO
MORE GAMES

Why Imitation Learning?

For the Sake of Robot Learning:

- It is difficult to learn from sparse rewards (unless data is cheap and you don't care about seeing lots of failures).
- Hand-designing rewards is hard.

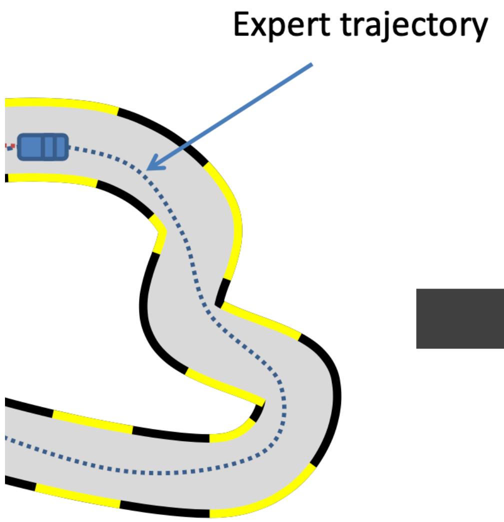
For the Sake of Learning Human Models:

- Learning human's intents, preferences, and underlying reward functions.

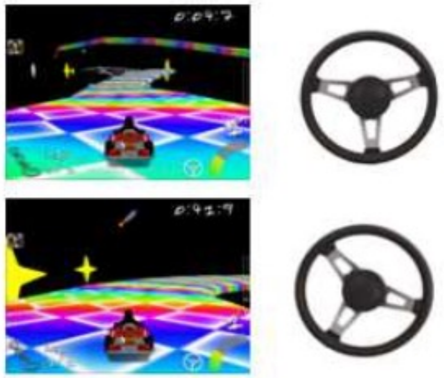
Imitation Learning in a Nutshell

- **Given:** Demonstrations or Demonstrator
- **Goal:** Train a policy to mimic demonstrations

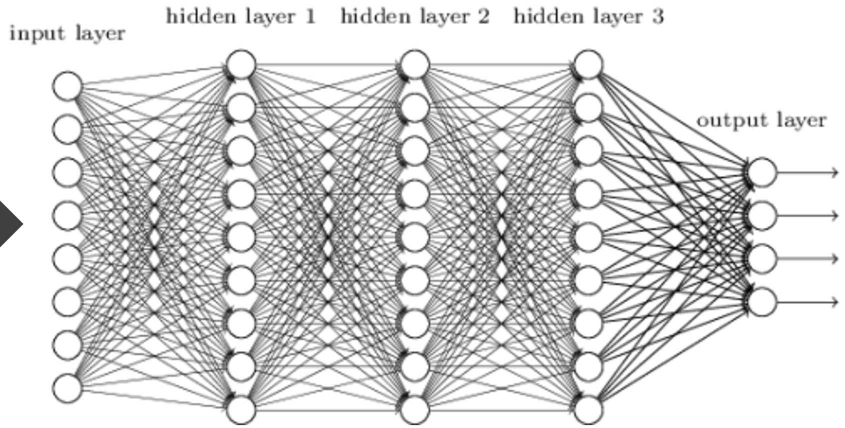
Expert Demonstrations



State/Action Pairs

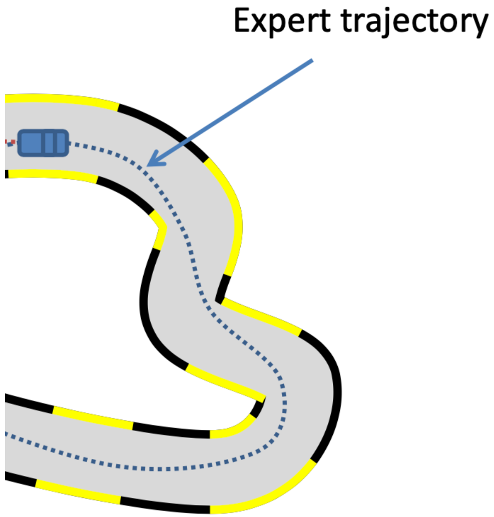


Learning



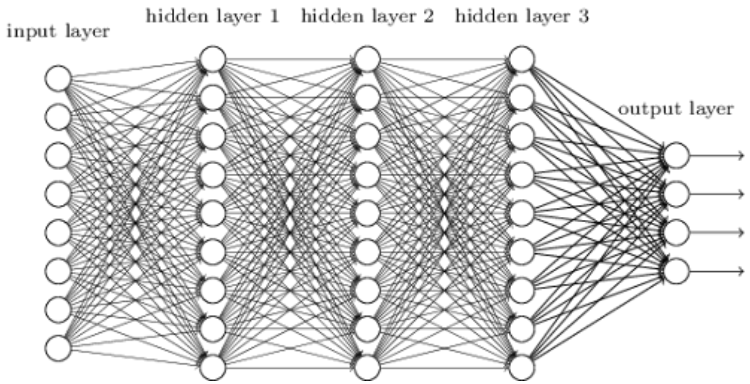
Ingredients of Imitation Learning

Demonstrator or Demonstrations



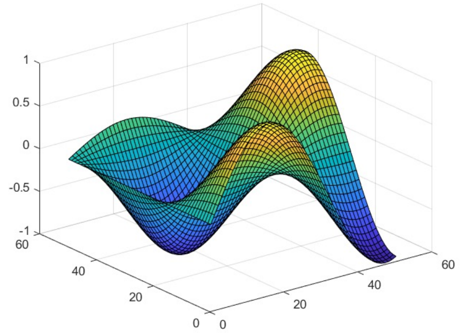
Environment/Simulator

Policy Class



Loss Function

Learning Algorithm



Problem Setup

MDP with no reward functions:

- State space, S (sometimes partially observable)
- Actions space, A
- An expert policy π^* that maps states to distributions over actions: $\pi^*(s) \rightarrow P(s)$
- Transition model $P(s_{t+1}|s_t, a_t)$: simulator or environment

Goal: Learn an imitating policy $\pi_\theta(s)$ that imitates the expert demonstrations

Problem Setup

Rollout: Sequentially execute $\pi(s_0)$ on an initial state

- produce trajectory: $\tau = (s_0, a_0, s_1, a_1, \ldots)$.

$\boldsymbol{P(\tau|\pi)}$: **Distribution of trajectories induced by a policy**

1. Sample s_0 from P_0 (distribution over initial states).
2. Initialize $t = 1$. Sample action a_i from $\pi(s_{t-1})$.
3. Sample next state s_t from applying a_t to s_{t-1} (requires access to environment).
4. Repeat form step 2 with $t = t + 1$.

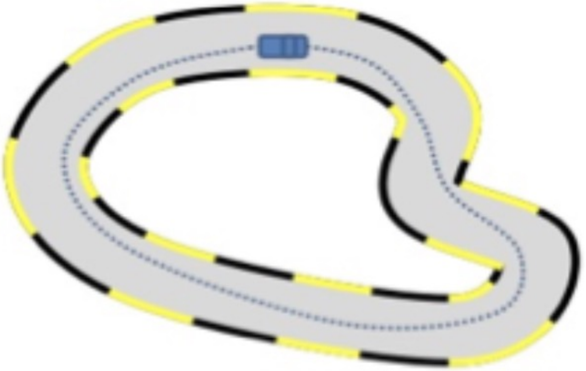
$\boldsymbol{P(s|\pi)}$: **Distribution of States induced by a policy**

- Let $P_t(s|\pi)$ denote distribution over t-th state.
- $P(s|\pi) = \frac{1}{T}\sum_t P_t(s|\pi)$

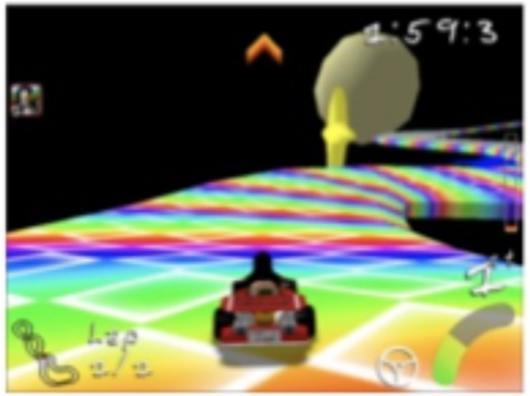
Example: Racing Game

s = game screen

a = turning angle



Training set: $D = \{\tau = \{(s_i, a_i)\}\}$ from π^*



Goal: Learn $\pi_\theta(s) \rightarrow a$

Credit: images from Stephan Ross

Today's itinerary

- Intro to Imitation Learning
- Behavioral Cloning
- Imitation Learning with Interactive Experts
- Inverse RL (MMP, Max Ent IRL)
- Learning from other sources of data (preferences, physical feedback)

Behavioral Cloning (reduction to supervised learning)

Define $P^* = P(s|\pi^*)$ (distribution of states visited by the expert)

(Recall $P(s|\pi^*) = \frac{1}{T}\sum_t P_t(s|\pi^*)$)

(sometimes abuse notation: $P^* = P(s, a^* = \pi^*(s)|\pi^*)$)

Learning Objective:

$$\arg\min_{\theta} \mathbb{E}_{(s,a^*)\sim P^*} L(a^*, \pi_\theta(s))$$

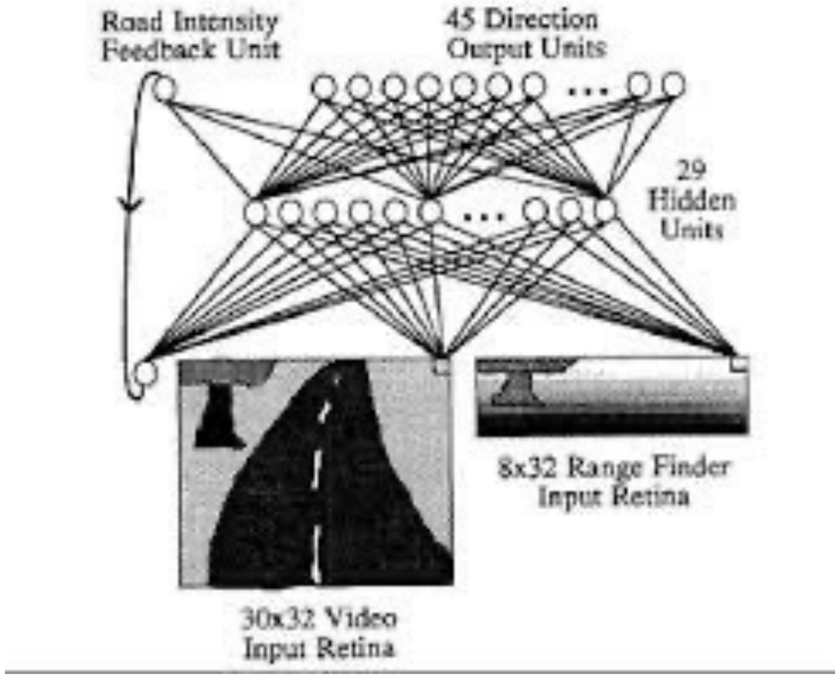
Interpretations:

1. Assuming perfect imitation so far, learn to continue imitating perfectly
2. Minimize 1-step deviation error along the expert trajectories

Behavioral Cloning: ALVINN

Learning Objective:

$$\arg\min_{\theta} \mathbb{E}_{(s,a^*)\sim P^*} L(a^*, \pi_\theta(s))$$
$$= \arg\min_{\theta} \mathbb{E}_{(s,a^*)\sim P^*} \mathrm{KL}(a^*, \pi_\theta(s))$$



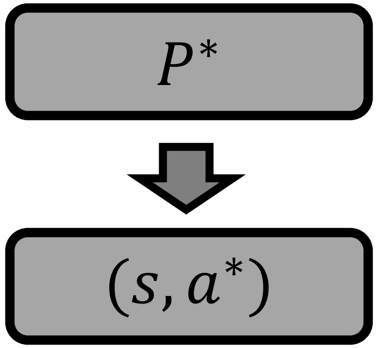
Early successes: ALVINN: NeurIPS 1989, D. Pomerleau

(General) Imitation Learning vs Behavioral Cloning

- Behavioral Cloning (supervised learning):

$$\arg\min_{\theta} \mathbb{E}_{(s,a^*)\sim P^*} L(a^*, \pi_\theta(s))$$

Distribution provided exogenously

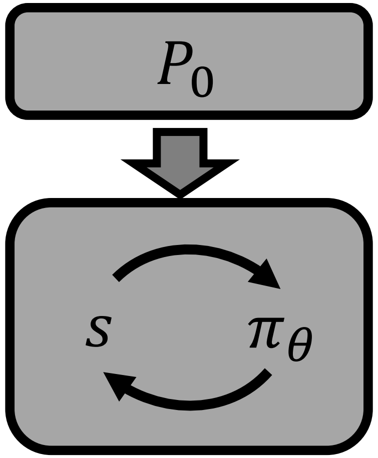


- (General) Imitation Learning:

$$\arg\min_{\theta} \mathbb{E}_{s\sim P(s|\theta)} L(\pi^*(s), \pi_\theta(s))$$

Distribution depends on the rollout

$P(s|\theta)$ = state distribution of π_θ



What can go wrong?

Errors in supervised learning:

- Assume *independent and identically distributed* (IID) state, action pairs, then if we have error at time t with probability ϵ, then over a time period the error would be bounded by ϵT in expectation.

In imitation learning, the state distribution of our data depends on the choice of actions.

End up in states that you have not seen before…

… compounding errors

During training:

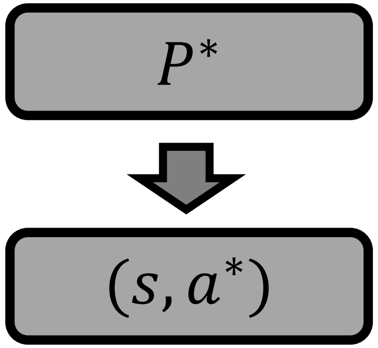
$$s \sim P^*$$

In test time:

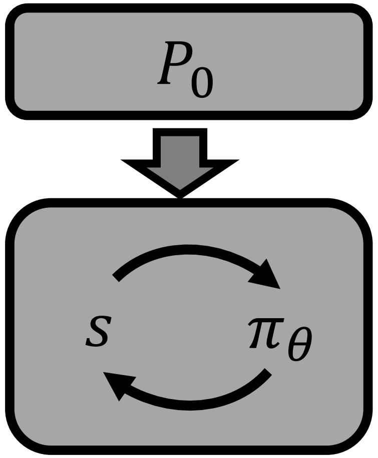
$$s \sim P(s|\pi_\theta)$$

Limitations of Behavioral Cloning: Compounding Errors

IID Assumption
(supervised learning)



Reality



π_θ makes a mistake

New state sampled not from P^*!

Worst case is catastrophic!

Cannot recover from new states

When to Use Behavioral Cloning?

Advantages:

- Simple
- Efficient

Disadvantages:

- Distribution mismatch between training and testing
- No long-term planning

Use When:

- 1-step deviations not too bad!
- Learning reactive behaviors
- Expert trajectories “cover” state space

Don’t Use When:

- 1-step deviations can lead to catastrophic error
- Optimizing long-term objective (at least not without a stronger model)

Types of Imitation Learning

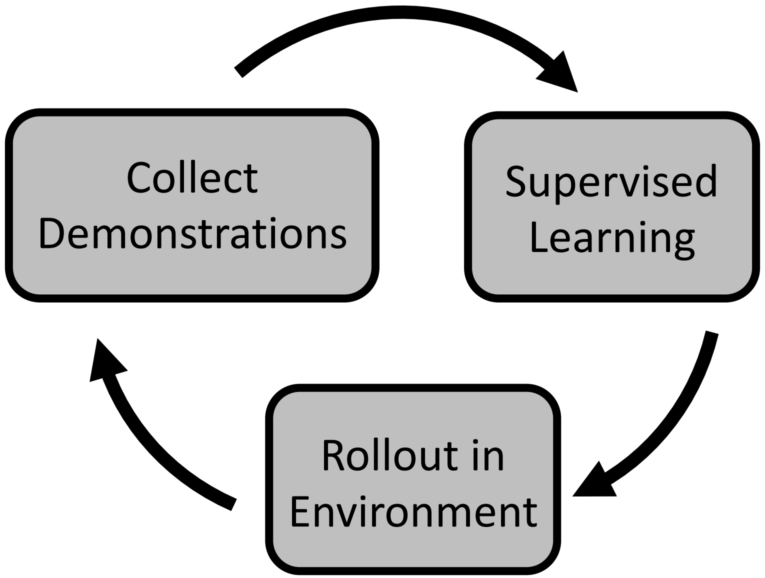
Behavioral Cloning

$$\arg\min_{\theta} \mathbb{E}_{(s,a^*)\sim P^*} L(a^*, \pi_\theta(s))$$

Works well when P^* is close to P_θ

Direct Policy Learning (via Interactive Demonstrator)

Requires Interactive Demonstrator (BC is a 1-step special case)



Inverse RL

Learn r such that:

$$\pi^* = \overbrace{\arg\max_{\theta} \mathbb{E}_{s\sim P(s|\theta)} r(s, \pi_\theta(s))}^{\text{RL problem}}$$

Assume learning r is statistically easier than directly learning π^*

Types of Imitation Learning

	Direct Policy Learning	Reward Learning	Access to Environment	Interactive Demonstrator	Pre-collected Demonstrations
Behavioral Cloning	Yes	No	No	No	Yes
Direct Policy Learning (interactive IL)	Yes	No	Yes	Yes	Optional
Inverse Reinforcement Learning	No	Yes	Yes	No	Yes

Today's itinerary

- Intro to Imitation Learning
- Behavioral Cloning
- Imitation Learning with Interactive Experts
- Inverse RL (MMP, Max Ent IRL)
- Learning from other sources of data (preferences, physical feedback)

Interactive Direct Policy Learning

Behavioral Cloning is simplest example

Beyond BC: using interactive demonstrator

Often analyzed via learning reductions

- Reduced "harder" learning problem to "easier" one
- Imitation Learning → Supervised Learning

Learning Reductions

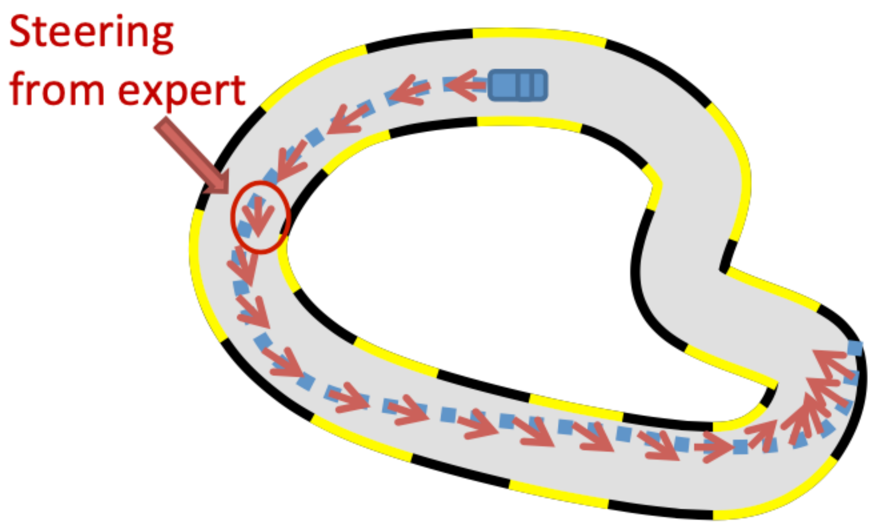
Behavioral Cloning:

$$\underbrace{\mathbb{E}_{s\sim P(s|\theta)} L\big(a^*(s), \pi_\theta(s)\big)}_{\text{A: General Imitation Learning}} \rightarrow \underbrace{\mathbb{E}_{(s,a^*)\sim P^*} L(a^*, \pi_\theta(s))}_{\text{B: Behavioral Cloning}}$$

What does learning well on B imply about A?

- e.g., can one lift PAC learning results from B to A?

Interactive Expert



Can query expert at any state

Construct loss function: $L(\pi^*(s), \pi(s))$

- Typically applied to rollout trajectories of policies we are training: $s \sim P(s|\pi)$

- Driving example: $L\left(\pi^*(s), \pi(s)\right) = \left(\pi^*(s) - \pi(s)\right)^2$

Expert provides feedback on state
visited by policy

Alternating Optimization (Naïve Attempt)

1. Fix P, estimate π
   - Solve $\arg\min_{\theta} \mathbb{E}_{s\sim P} L(\pi(s), \pi_\theta(s))$

Just behavioral cloning!

2. Fix π, estimate P
   - Empirically estimate via rolling out π

Update state distributions

3. Repeat

Not guaranteed to converge!

Sequential Learning Reductions

- Initial predictor: π_0 (initial predictor: initial expert demonstrations)
- For m sequence of predictors (initialize m=1)
  - Collect trajectories τ via rolling out π_{m-1} (typically rollout multiple times)
  - Estimate state distribution P_m using $s \in \tau$
  - Collect interactive feedback $\{\pi^*(s)|s \in \tau\}$ (requires interactive expert)
  - **Data Aggregation** (e.g., DAgger)
    - Train π_m on $P_1 \cup \cdots \cup P_m$
  - **Policy Aggregation** (e.g., SEARN & SMILe)
    - Train intermediate policy π'_m on only P_m
    - $\pi_m = \beta\pi'_m + (1-\beta)\pi_{m-1}$ (geometric blending of policies)

DAgger in Practice

Direct Policy Learning via Interactive Expert

Reduction to sequence of supervised learning problems

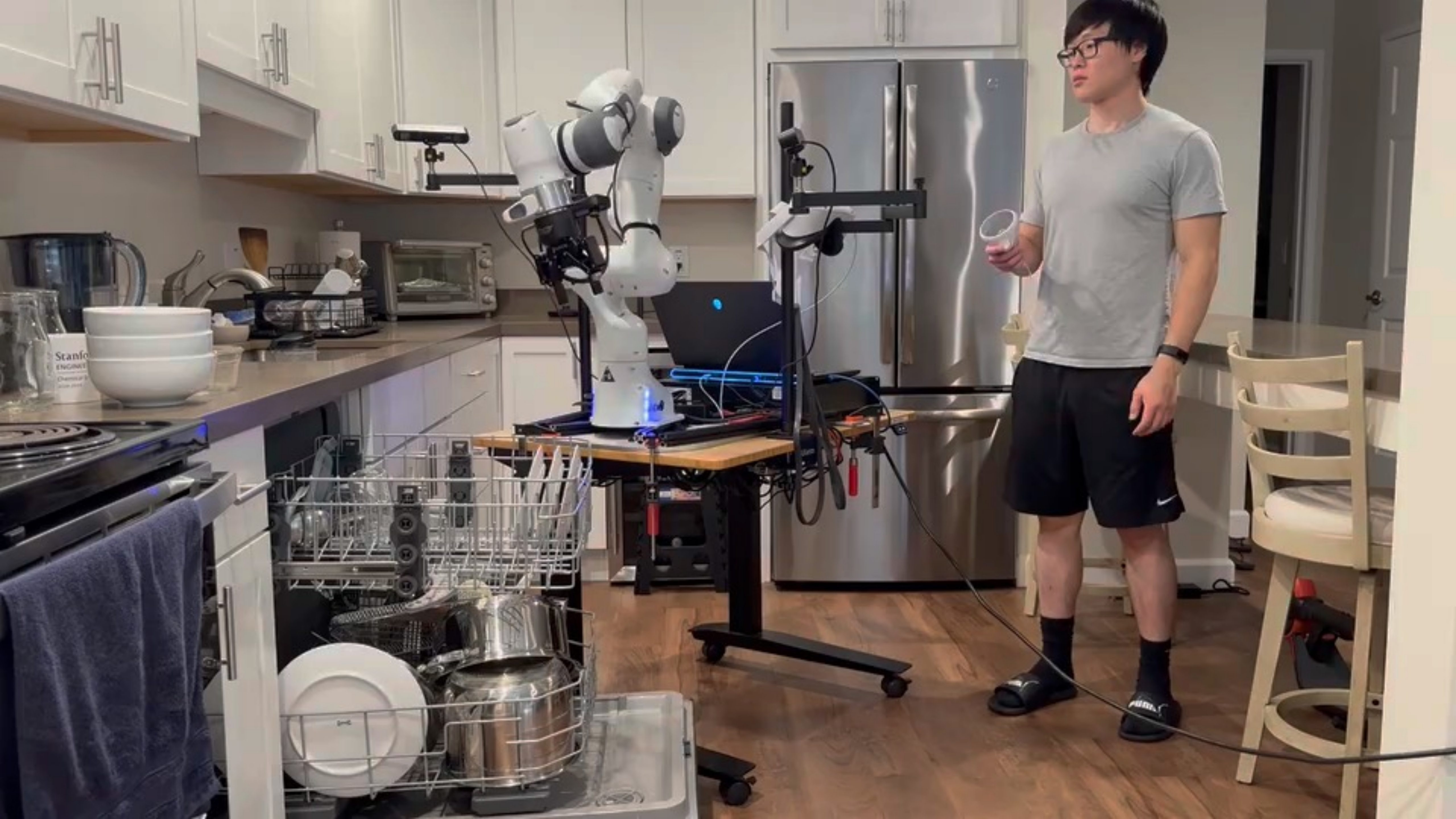
- Constructed from rollouts from previous policies
- Requires interactive expert feedback

Two approaches: Data Aggregation & Policy Aggregation

- Ensure convergence
- Motivated by different theory

Not covered:

- What is expert feedback and loss function? (depends on application)



Open X-Embodiment: Robotic Learning Datasets and RT-X Models