# CS395T: Foundations of Machine Learning for Systems Researchers

Fall 2025

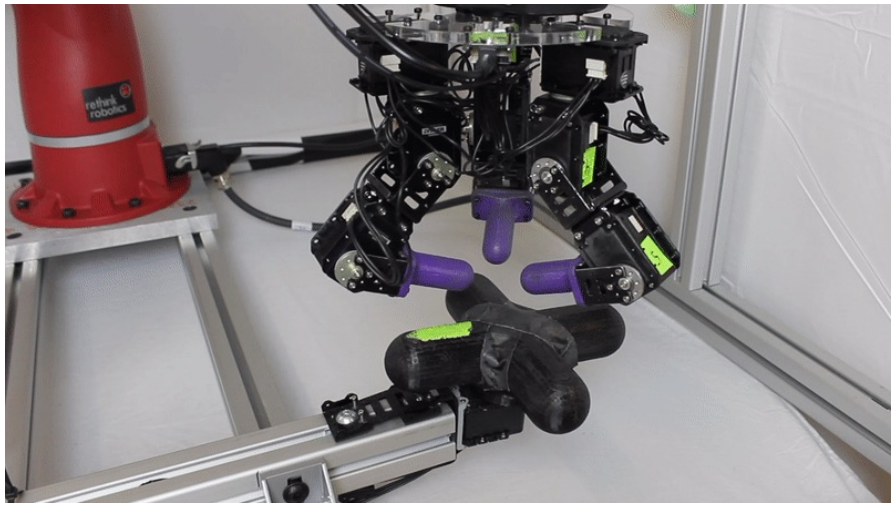
## Lecture 11:

## Learning from Humans: Demonstrations to Feedback

Lain (Zelal) Mustafaoglu

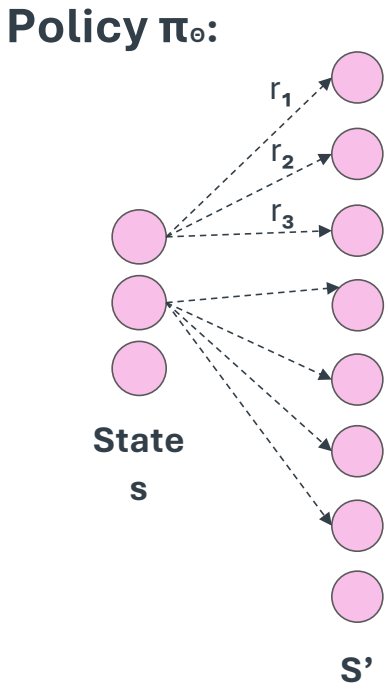
# Overview

- Learning from humans: **Demonstrations** vs. **Feedback**
- Learning from demonstrations:
  - **Imitation Learning**
    - Offline IL: BC
    - Online IL: DAgger
  - **Inverse RL**
    - GAIL
- Learning from feedback:
  - **RLHF**
    - Example: Fine-tuning LLMs
    - Example: Alignment
    - Example: Robotics
- Practical tips



Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. Rajeswaran. et al., 2018

# Optimizing policies with unknown rewards




- **Policy optimization:** so far, assumed rewards are known
- **Issue:** Real-world problems are often difficult to formalize with hard-coded rewards

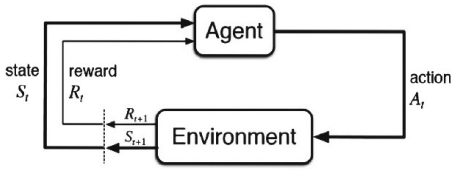
- *Can we do policy optimization if we don't have clearly defined rewards?*
  - Yes!
  - Three possible ways:
    - **Imitation learning:** Mimic expert demonstrations
    - **Inverse reinforcement learning (IRL):** Infer reward from expert demonstrations
    - **RLHF:** Use human feedback as reward signal

# Learning from experience vs. learning from humans

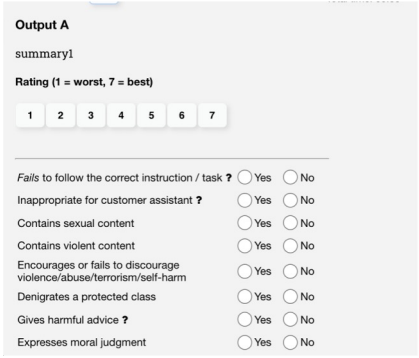
Less guidance, cheaper

More guidance, expensive

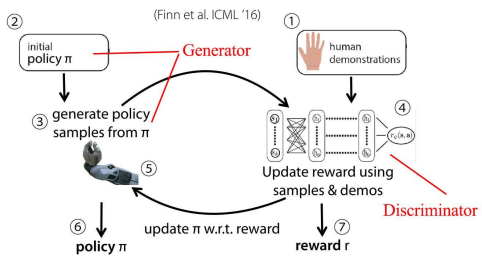
Learning from environment interactions (traditional RL)



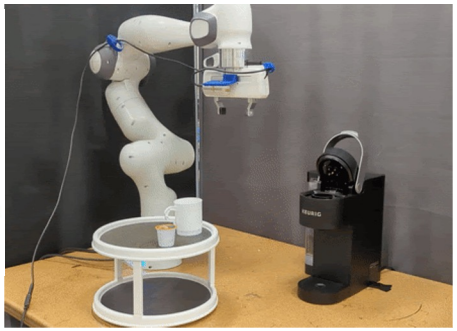

Learning from human feedback (RLHF)




Inferring reward from interactions (inverse RL)




Learning from human demonstrations (imitation learning)



# Methods: Signal vs. Interaction

**Learning Signal:** Actions / Rewards

**Interaction:** Offline / Online

**Actions × Offline**
- Offline IL, e.g. Behavior Cloning (BC)

**Rewards × Online**
- Online RL (rewards)
- RLHF (feedback as reward signal)

**Actions × Online**
- Online IL: expert corrections (e.g. DAgger)

**Rewards × Offline**
- Offline RL

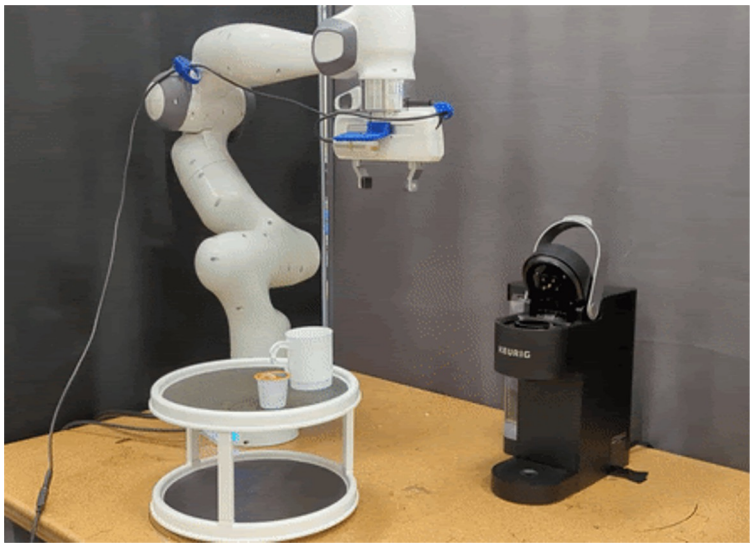
# Spectrum of methods

| Method | Human / data signal | Learns: | Online interaction? | Use when… |
|---|---|---|---|---|
| Standard RL | Given reward $r(s, a)$ | Policy/value | Usually yes | Reward known; can interact |
| IL — Offline, e.g. Behavior Cloning (BC) | Expert demos $(s, a^*)$ | Policy $\pi_\theta$ | No (offline) | Good demos available; quick warm start for RL |
| IL — Online, e.g. Dagger) | Expert corrections on (new) visited states | Policy $\pi_\theta$ | Yes | Fix BC drift; expert available online |
| Inverse RL | Expert trajectories | Reward $R_\phi$ (then policy with RL) | Demos offline; RL online | Need reusable/inspectable reward |
| RLHF | Human preferences (pairwise/ratings) | Reward $R_\phi$ and policy $\pi_\theta$ | Often iterative | No demos; humans can provide feedback |
| Offline RL | Static log with rewards | Policy/Q | No (offline) | Large logs; no interaction allowed |

# Imitation Learning

# Imitation Learning

Learn a policy by treating actions in **expert demonstrations** as *labels* and do supervised learning



What Matters in Learning from Offline Human Demonstrations for Robot Manipulation, HYDRA: Hybrid Robot Actions for Imitation Learning. Belkhale et al. 2023.

# Demonstrations

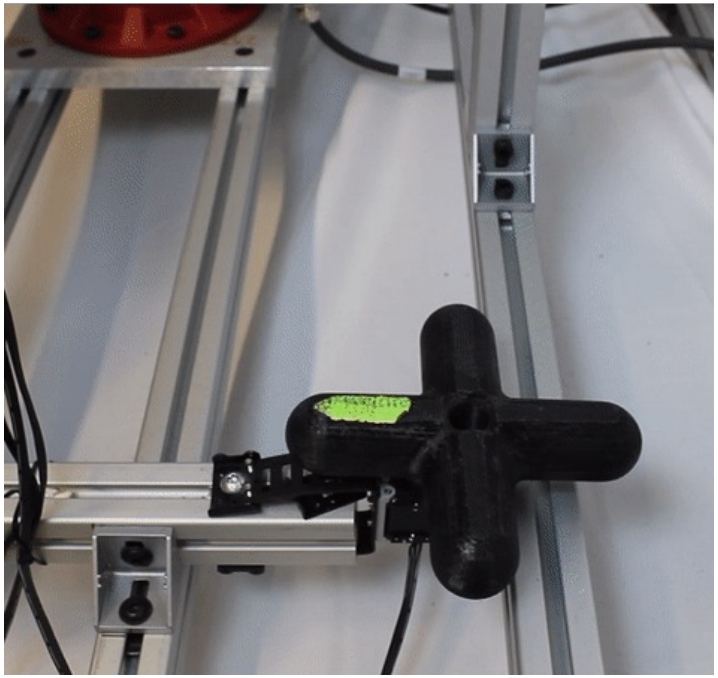
**Abstraction of demonstrations:**

$$\mathcal{D} = \{ \tau_1, \tau_2, \ldots, \tau_N \}$$

Dataset of *N* demonstrations,
where each demonstration is an expert trajectory:

$$\tau_i = \{ (s_1^{(i)}, a_1^{(i)}), (s_2^{(i)}, a_2^{(i)}), \ldots, (s_{T_i}^{(i)}, a_{T_i}^{(i)}) \}$$

We want to learn a *policy* $\pi_\theta(a \mid s)$ that behaves like the expert

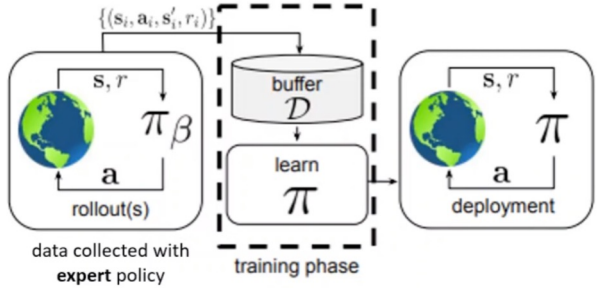


# IL approaches

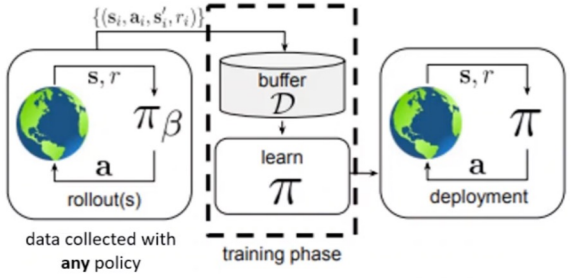
**Imitation Learning**

## Offline IL

The agent only has access to a fixed dataset of demonstrations, learning passively (e.g. **Behavior Cloning**)




**vs. offline RL:**




## Online IL

Expert demonstrations are augmented with real-time expert interactions for dynamic error correction (e.g. **DAgger**)

Learn policies directly using a static dataset *without online interaction*

# Behavior Cloning (BC)

- How do we incorporate information from demonstrations into training?
  - **Behavior Cloning:**

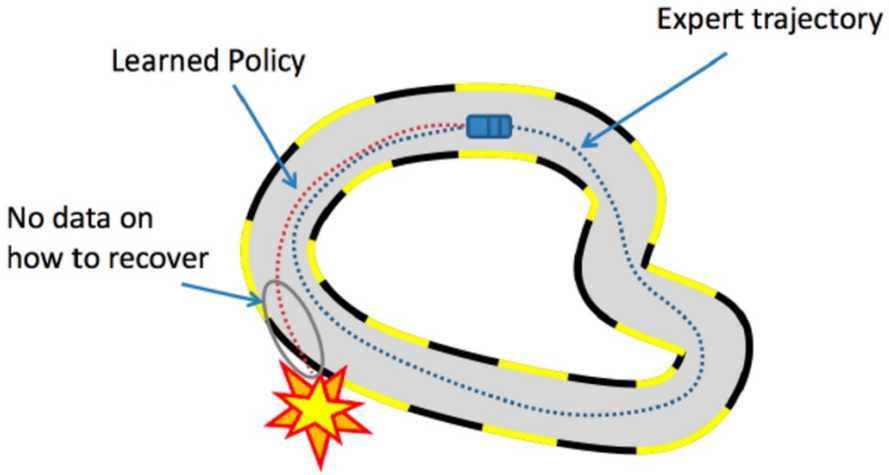
$$\underset{\theta}{\text{maximize}} \sum_{(s,a)\in\rho_D} \ln \pi_\theta(a|s)$$

- BC reduces IL to supervised learning
- *What happens when we encounter a state that wasn't in the expert distribution?*

# Compounding errors in BC

During training, BC only learns from expert data (sees only the “good” trajectories):

- **Assumption:** The agent will always stay in these states
- **Inference:** No expert data for unseen states → agent guesses randomly → errors compound → agent drifts further from expert’s behavior → *catastrophic failure*




**Two possible solutions:** more demos, or correct as you go

Online expert corrections to ”correct” agent behavior in new states

→ **Online imitation learning** (e.g. DAgger)

# DAgger (Dataset Aggregation)

- Addresses distribution shift issue in BC with **expert corrections**: expert is queried *during training* to label new states visited by agent

- Iterate: roll out current policy → query expert on visited states → aggregate → retrain BC

  ; recovery from mistakes

- **Limitation:** Querying experts for online corrections expensive & time-consuming

A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. Ross et al., 2010.

# Limitations of imitation learning

- Learned policies will only be as good as the expert since there is no ”performance measure” learned, only mimicking

*Can we use demonstrations to learn a performance measure?*

- **Inverse reinforcement learning (IRL):** infer reward function from demonstrations

# Inverse Reinforcement Learning (IRL)

# Inverse RL

Learn reward function $R_\phi$ so that expert trajectories are likely under that reward; then optimize a policy using $R_\phi$ with RL

Methods include:

- Apprenticeship Learning via Inverse Reinforcement Learning (Abbeel and Ng, 2004)
- Generative Adversarial Imitation Learning (GAIL)

# Apprenticeship Learning

- Assume linear reward:

  $r_{\mathbf{w}}(s,a) = \mathbf{w}^\top \boldsymbol{\phi}(s,a)$ Feature vector expressing task performance

- Value of a policy is a weighted sum of feature counts

$$J(\pi;\mathbf{w}) = \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty}\gamma^t r_{\mathbf{w}}(s_t,a_t)\Big] = \mathbf{w}^\top \boldsymbol{\mu}(\pi)$$

- Match **feature expectations** of learner and expert:

$$\boldsymbol{\mu}(\pi) = \mathbb{E}_\pi\Big[\sum_{t=0}^{\infty}\gamma^t\,\boldsymbol{\phi}(s_t,a_t)\Big] \approx \boldsymbol{\mu}_E = \mathbb{E}_{\pi_E}\Big[\sum_{t=0}^{\infty}\gamma^t\,\boldsymbol{\phi}(s_t,a_t)\Big]$$

- **Alternate:** find $\pi$ via RL under candidate $w$; update $w$ to separate expert vs learner

Apprenticeship Learning via Inverse Reinforcement Learning, Abbeel and Ng, 2004.

# Generative Adversarial Imitation Learning (GAIL)

Train a discriminator $D(s,a)$ to tell apart expert vs. learner actions; use discriminator as learned reward

- **Adversarial minimax objective** (discriminator vs. policy):

$$\min_{\pi} \max_{D} \mathbb{E}_{(s,a)\sim\rho^E}\left[\log D(s,a)\right] + \mathbb{E}_{(s,a)\sim\rho^\pi}\left[\log\left(1 - D(s,a)\right)\right] - \lambda\,\mathcal{H}(\pi)$$

Average over expert 's (discounted) occupancy

Average over current policy $\pi$'s (discounted) occupancy

- **Discriminator update:**

$$D \leftarrow \arg\max_{D} \mathbb{E}_{\rho^E}\left[\log D(s,a)\right] + \mathbb{E}_{\rho^\pi}\left[\log\left(1 - D(s,a)\right)\right]$$

- **Policy update** using learned reward

Learned reward

$$\pi \leftarrow \arg\max_{\pi} \mathbb{E}_{\rho^\pi}\left[r(s,a)\right] - \lambda\,\mathcal{H}(\pi)$$

$$r(s,a) = -\log\left(1 - D(s,a)\right)$$

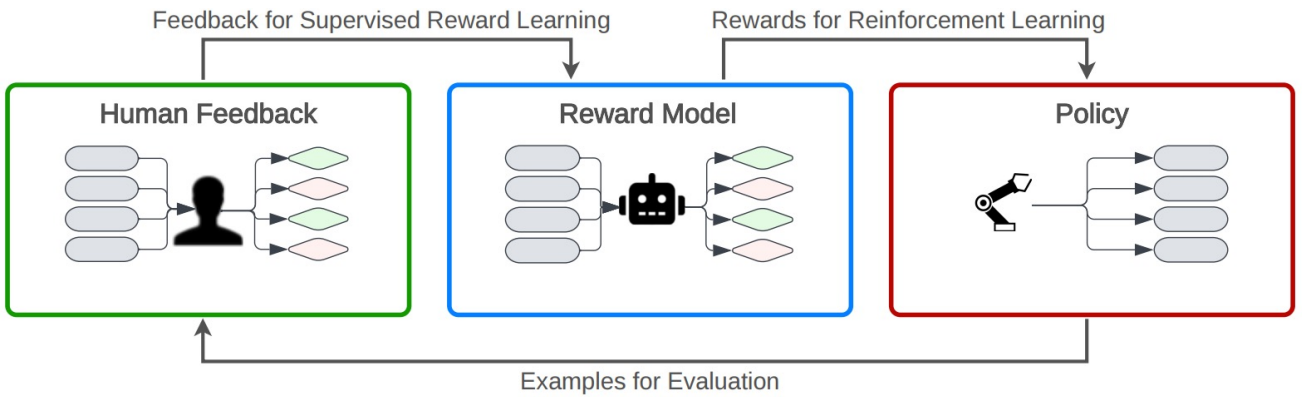
Generative Adversarial Imitation Learning, Ho and Ermon, 2016.

# What if we can't collect demonstrations?

**Problem:** Demonstrations are expensive/difficult to obtain for both IL and IRL

Collect **human feedback** (*preference data*) instead!

- **Reinforcement learning from human feedback (RLHF)**

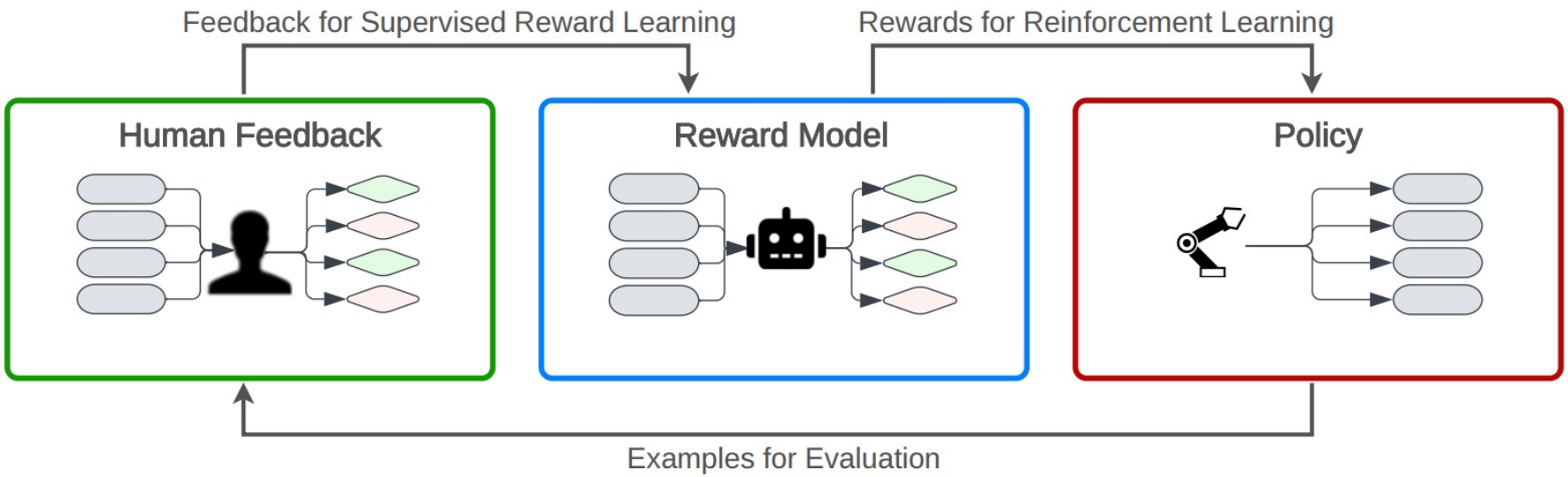



Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback, Casper et al., 2023.

# Reinforcement Learning from Human Feedback (RLHF)

# RLHF in one slide

**Key idea:** use human feedback as proxy for reward in policy optimization



# RLHF

**1. Collect human feedback**

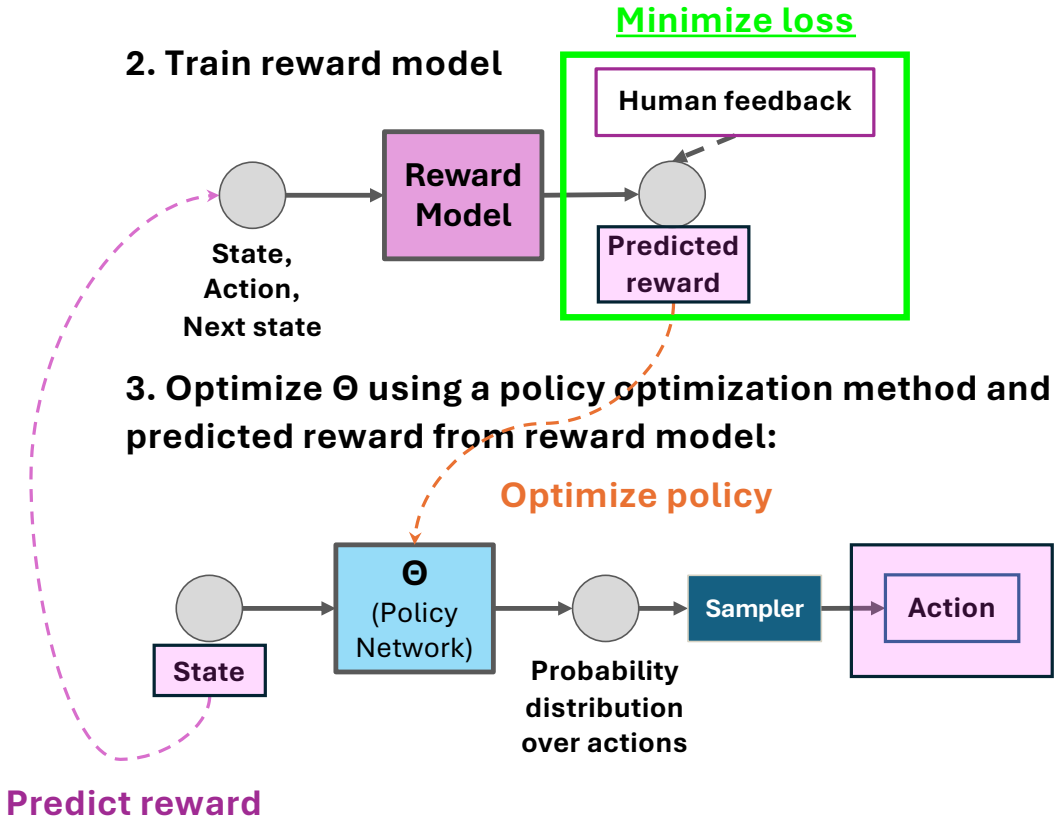
**Rating (1 = worst, 7 = best)**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|

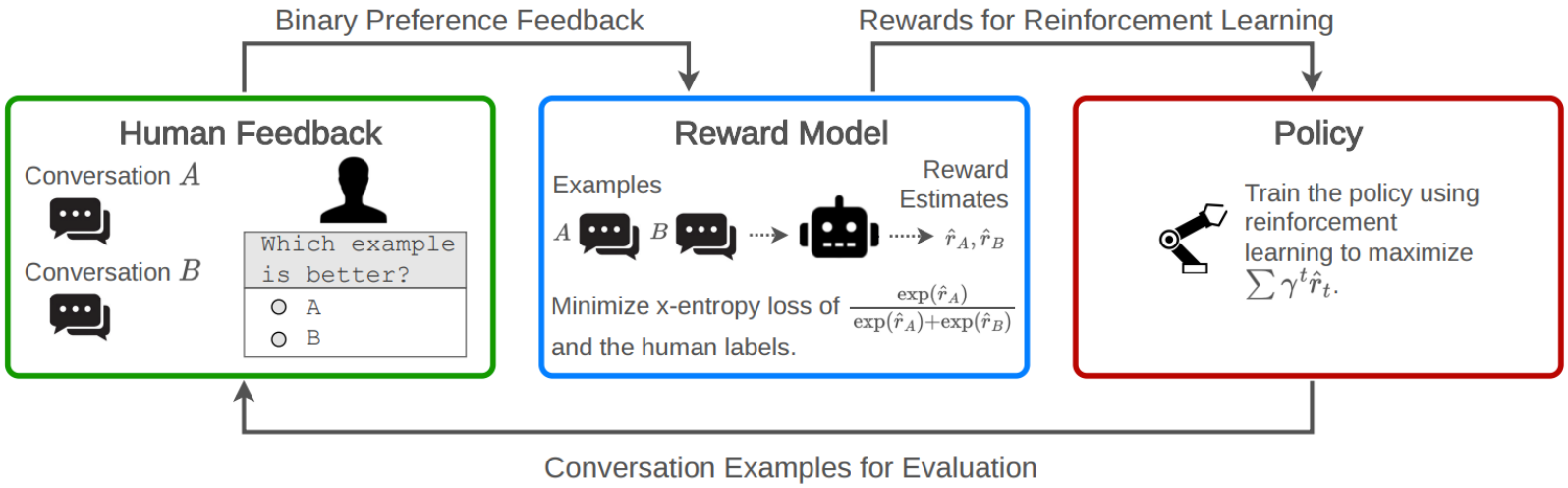
| | | |
|---|---|---|
| *Fails* to follow the correct instruction / task ? | Yes | No |
| Inappropriate for customer assistant ? | Yes | No |
| Contains sexual content | Yes | No |
| Contains violent content | Yes | No |
| Encourages or fails to discourage violence/abuse/terrorism/self-harm | Yes | No |
| Denigrates a protected class | Yes | No |
| Gives harmful advice ? | Yes | No |
| Expresses moral judgment | Yes | No |

**= Feedback score**

**2. Train reward model**



# Example 1: RLHF for fine-tuning with binary feedback




Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback Casper et al., 2023.

# Example 2: RLHF for Alignment

To be **aligned** (with human values), language models should be:

- Helpful
- Honest
- Harmless

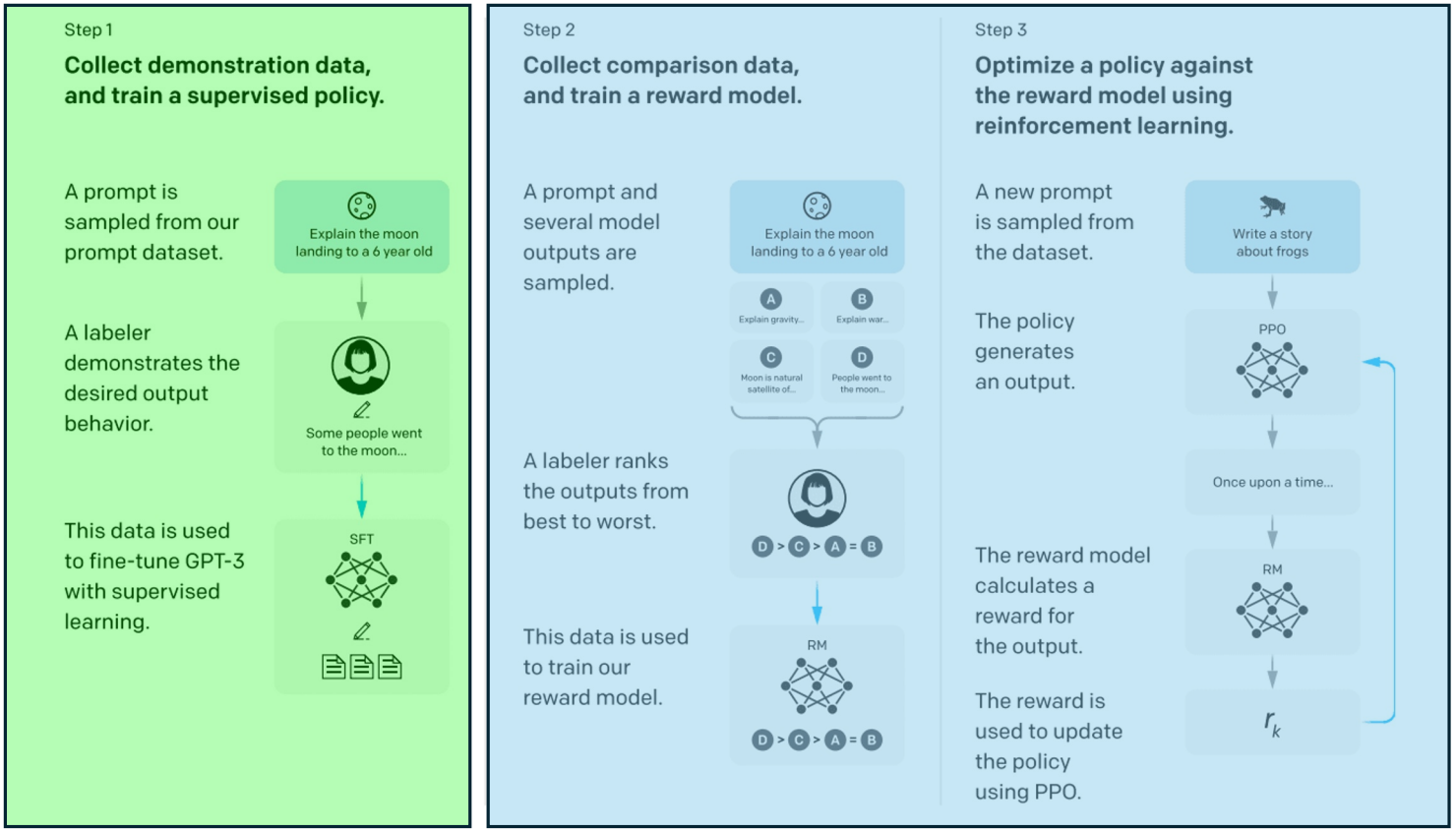
(Askell et al. 2021)

**Problem:** these objectives do not align with the language modeling objective of predicting the next token

**Solution:** RLHF for alignment

# RLHF for Alignment: InstructGPT

**Supervised pre-training**
(= imitation learning)



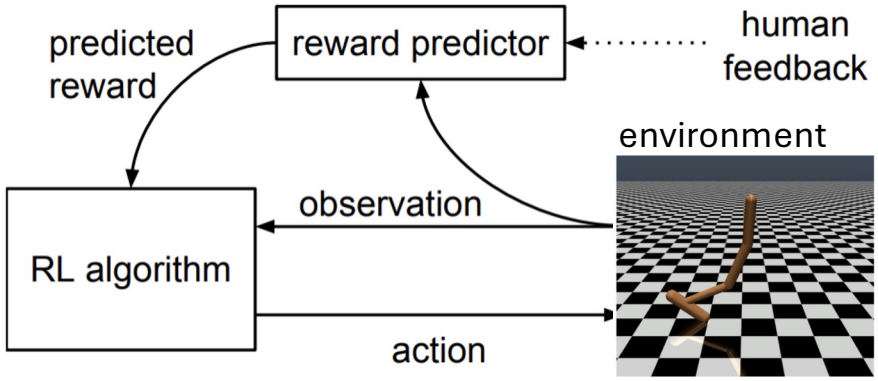

**RLHF:**

- Reward model training
- Policy optimization

Training language models to follow instructions with human feedback. Ouyang et al. 2022.

# Example 3: RLHF for Robotics

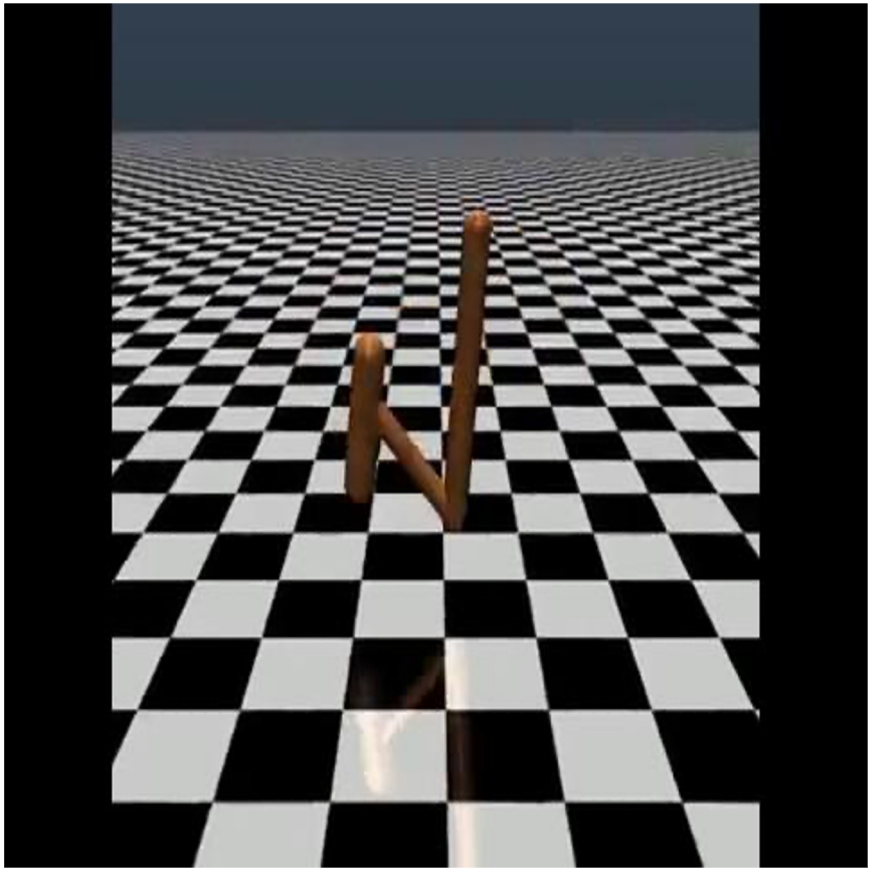
- **Human preference data collection:** Labelers are provided with a visualization of two trajectory segments, in the form of 1-2 second clips
- Labelers are asked the segment they prefer, or if the two segments are equally good or not comparable




Deep Reinforcement Learning from Human Preferences. Christiano et al. 2023.

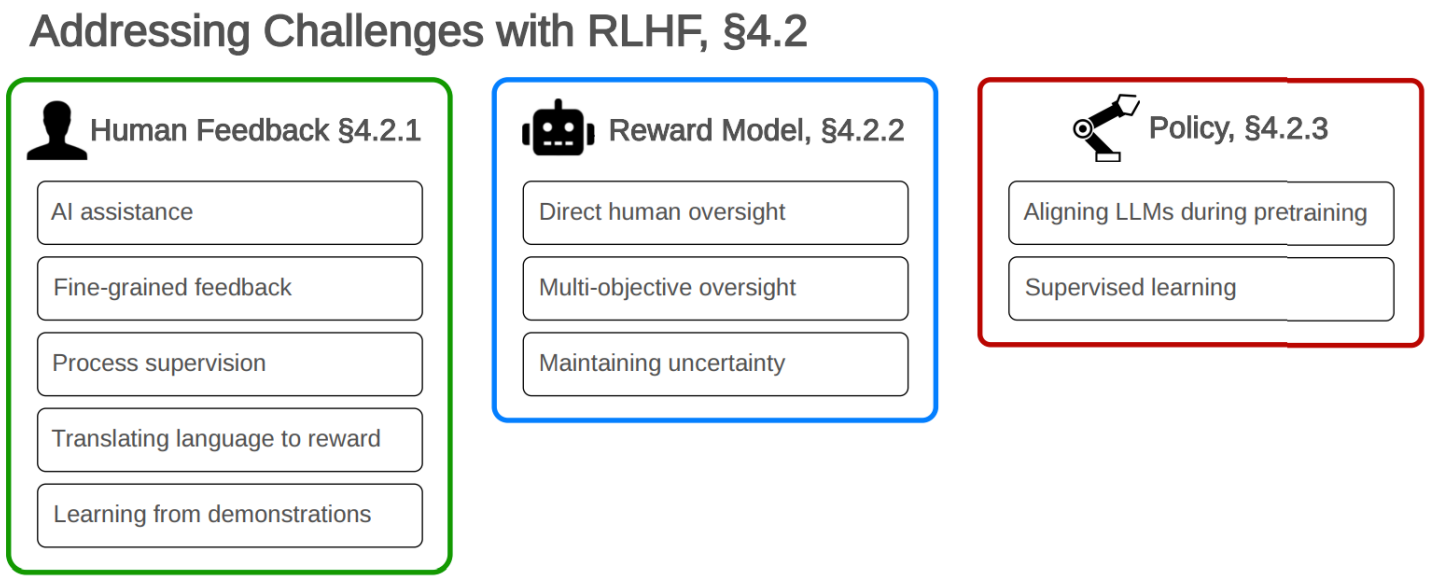
# Learning novel behavior

- Hopper was successfully trained to do a backflip in one hour with 900 human labels (from researchers)



Deep Reinforcement Learning from Human Preferences. Christiano et al. 2023.

# Challenges with RLHF

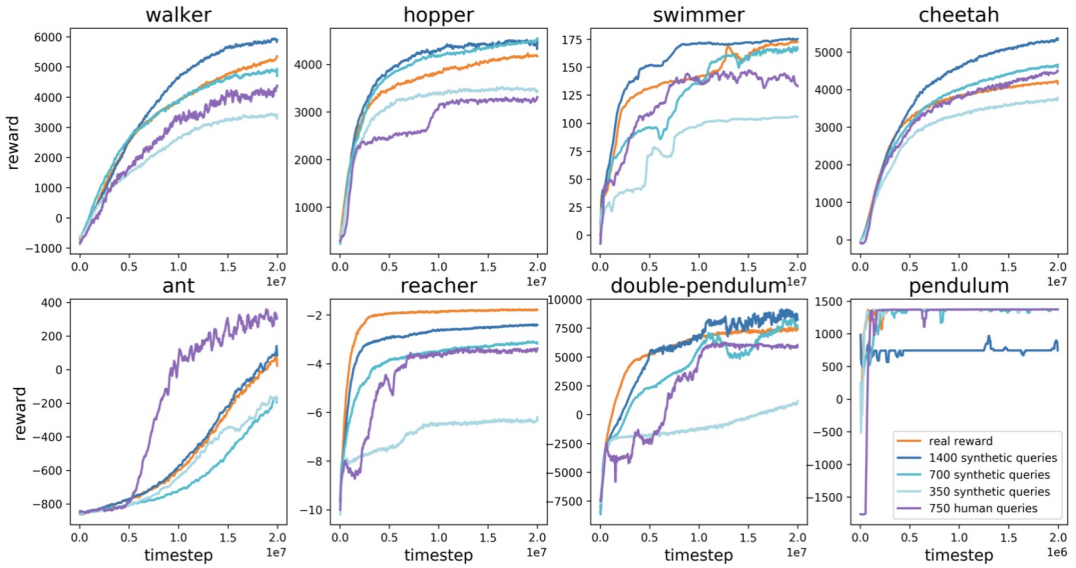



Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback Casper et al., 2023.

# Practical Tips

# Synthetic feedback

- Evaluated on **simulated robotics tasks** on MuJoco
- **RL Method:** TRPO with 750 human queries
- **Baselines:** TRPO with real reward, TRPO with 350/700/1400 synthetic queries
- Synthetic feedback can be almost as good as human feedback, if not better!




Deep Reinforcement Learning from Human Preferences. Christiano et al. 2023.

# Reward modeling

- **In practice:** use same kind of base model for reward model and policy

- *What size should your reward model be?*
  - 6B reward model was selected over 175B model for stability in InstructGPT

# Two common approaches you might encounter

- **RLHF (general):** SFT/BC → train reward model Rϕ → PPO with KL to SFT/BC using Rϕ

- **RL for robotics:** bootstrapping RL methods with demonstrations
  - *Shaped rewards* from pre-training on demonstrations accelerate convergence
  - Two approaches:
    - On-policy: Demo Augmented Policy Gradients (DAPG)
    - Off-policy: Demos and Off-Policy Actor Critics (DDPGfD) which is DDPG augmented with demonstrations

# Demo Augmented Policy Gradients (DAPG)

**Key Idea**: use BC to bootstrap RL

- Augment the original objective with a weighted Behavior Cloning term

$$g_{aug} = \underbrace{\sum_{(s,a)\in\rho_\pi} \nabla_\theta \ln \pi_\theta(a|s) A^\pi(s,a)}_{\text{Policy Gradient}} + \underbrace{\sum_{(s,a)\in\rho_D} \nabla_\theta \ln \pi_\theta(a|s) w(s,a)}_{\text{Behavior Cloning Gradient}}$$

Data collected by the policy

Demonstration data

- **Heuristic weighting scheme** $w(s,a)$ to decay BC contributions over time as our policy improves

$$w(s,a) = \lambda_0 \lambda_1^k \max_{(s',a')\in\rho_\pi} A^\pi(s',a') \quad \forall (s,a) \in \rho_D$$

k = number of PG iterations

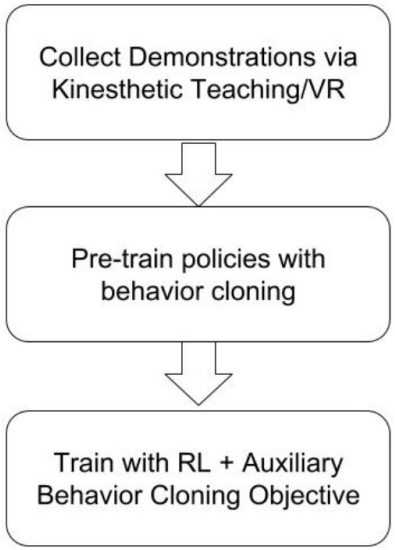
Highest advantage in data collected by PG → *approximation* to $A^\pi(s',a')$ for $\rho_D$

- $\lambda_0 = 0, w(s,a) = 1 \rightarrow$ **Behavior Cloning**; $\lambda_0 > 0, w(s,a) = 0 \rightarrow$ **RL**

Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. Rajeswaran. et al., 2018

# Why DAPG?

- Why bootstrap RL with BC?
  - Eliminates reward shaping
  - Discovers more natural looking behaviors
  - Guides exploration
  - Decrease sample complexity
- Demonstrations are collected with VR in simulation
  - Only 25 demonstrations per task to achieve 30x sample efficiency and 30x training speed
- Demonstrations incorporate human *priors* to "kickstart" learning
  - Alternative to reward shaping (instead of sparse task completion rewards)
    - Manual and labor intensive




Demonstration Augmented Policy Gradient

Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. Rajeswaran. et al., 2018

# Demos and Off-Policy Actor Critics (DDPGfD)

**Key idea**: use demonstrations as initial samples to *pre-train* our policy before doing RL with deep deterministic policy gradients (DDPG)

- DDPG is a policy gradient algorithm that picks actions deterministically
  - DDPG adds noise to the best action for exploration instead of relying on stochastic action selection for exploration

- Learning the value of expert states does not necessarily push the policy towards expert behavior

"*Leveraging Demonstrations for Deep RL on Robotics…*", Vecerik et al., 2017

# Spectrum of methods

| Method | Human / data signal | Learns: | Online interaction? | Use when… |
|---|---|---|---|---|
| Standard RL | Given reward $r(s, a)$ | Policy/value | Usually yes | Reward known; can interact |
| IL — Offline, e.g. Behavior Cloning (BC) | Expert demos $(s, a^*)$ | Policy $\pi_\theta$ | No (offline) | Good demos available; quick warm start for RL |
| IL — Online, e.g. Dagger) | Expert corrections on (new) visited states | Policy $\pi_\theta$ | Yes | Fix BC drift; expert available online |
| Inverse RL | Expert trajectories | Reward $R_\phi$ (then policy with RL) | Demos offline; RL online | Need reusable/inspectable reward |
| RLHF | Human preferences (pairwise/ratings) | Reward $R_\phi$ and policy $\pi_\theta$ | Often iterative | No demos; humans can provide feedback |
| Offline RL | Static log with rewards | Policy/Q | No (offline) | Large logs; no interaction allowed |