Learning Rewards

CS 224R

Course reminders

- Project proposal due **tonight.**
  (graded fairly lightly — really for your benefit!)
- Homework 2 due next Wednesday (**start early!**)

The plan for today

1. Where do rewards come from?
2. Learning rewards from example goals, behaviors
3. Learning rewards from human preferences

Key learning goals:

- why task specification is hard (& why naïve methods fail)
- methods for learning rewards from human supervision

Where does the reward come from?

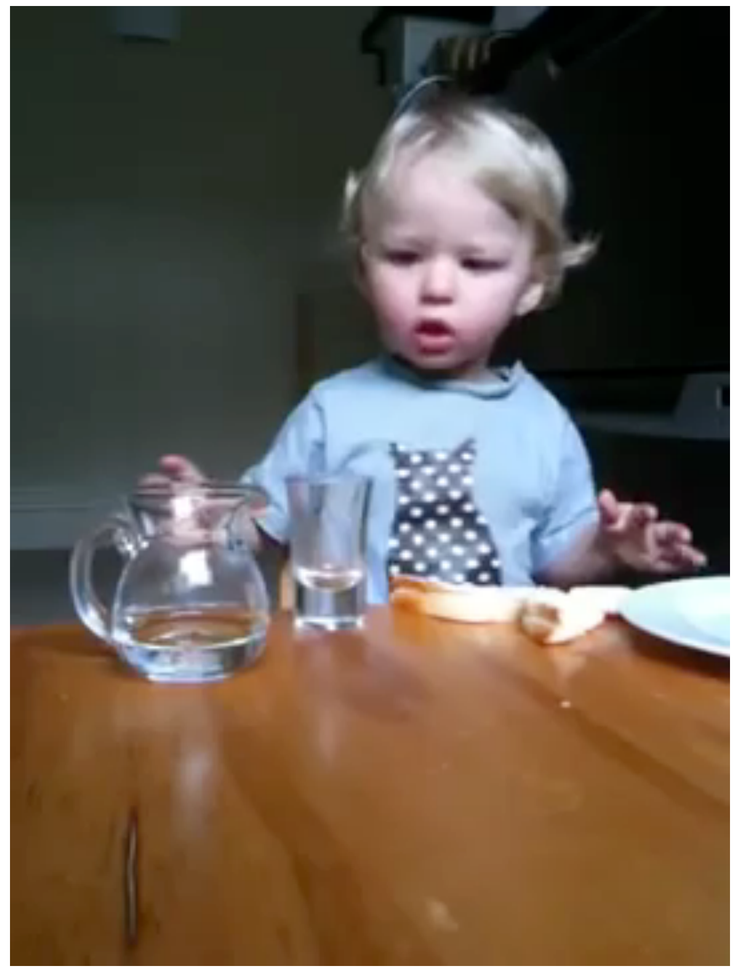
Computer Games

reward

Mnih et al. '15

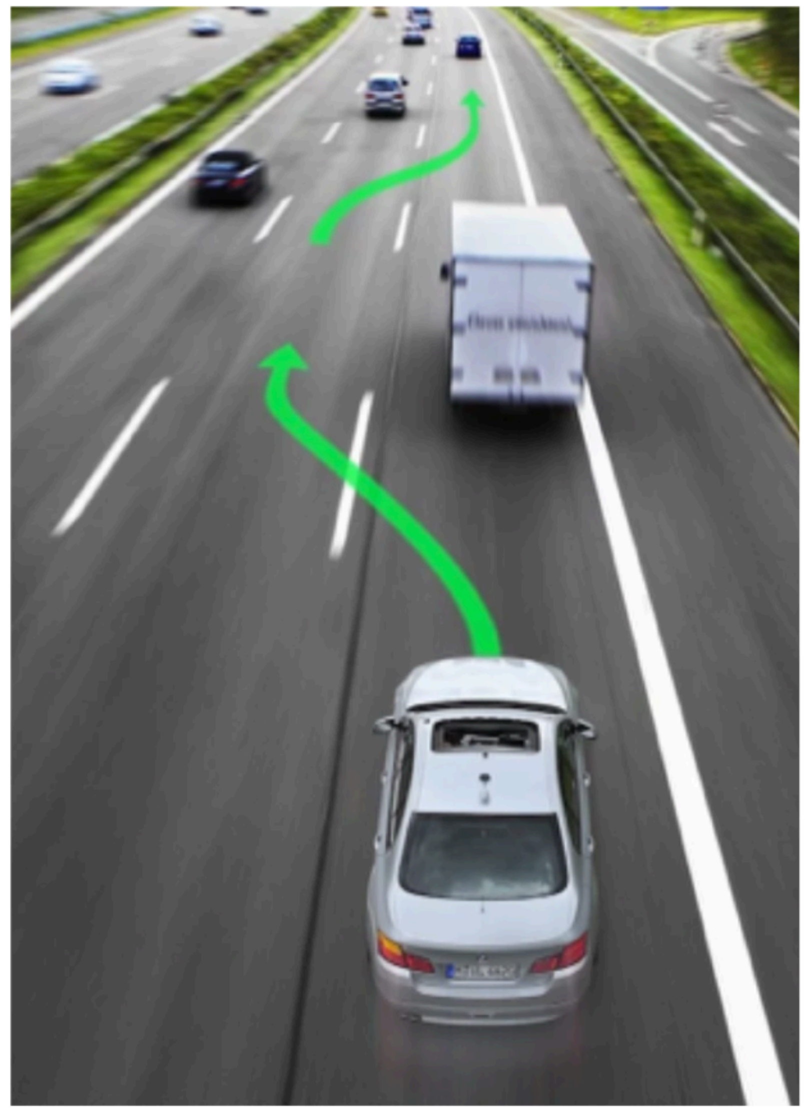
Real World Scenarios

robotics



dialog

autonomous driving



what is the reward?

often use a proxy

Are there other easier ways of providing task supervision?

We've seen one alternative approach so far.

Where does the reward come from?

Direct imitation learning: Mimic actions of expert

- but no reasoning about outcomes or dynamics
- the expert might have different degrees of freedom
- might not be possible to provide demonstrations

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

The plan for today

1. Where do rewards come from?
2. **Learning rewards from example goals, behaviors**
3. Learning rewards from human preferences

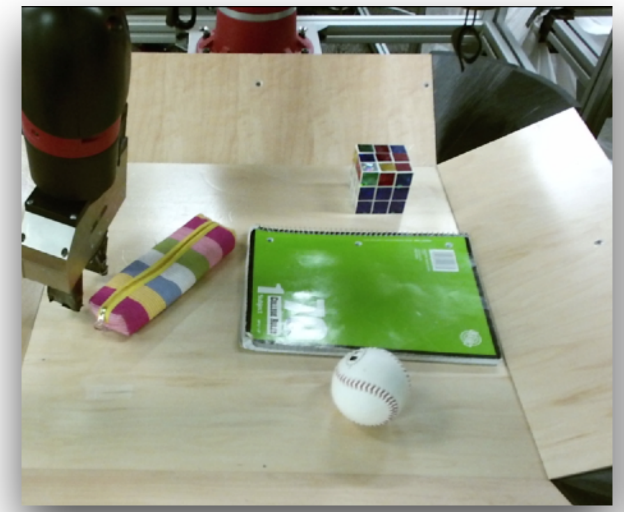
Goal classifiers

Key idea: Learn to discern goal states from other states

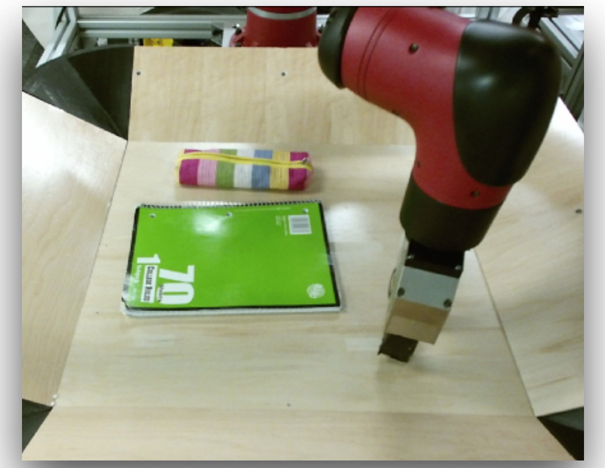
Example task: put pencil case behind notebook

Positive examples


Negative examples




Trained binary classifier

 $\mapsto 1$  $\mapsto 0$

Use output as reward signal.

1. Collect examples of successful & unsuccessful states (states inside and outside of goal set G)
2. Train binary classifier (with inputs s_i and labels $\mathbf{1}(s_i \in G)$)
3. Run RL with classifier as reward.

Goal classifiers

Key idea: Learn to discern goal states from other states

1. **Collect examples of successful & unsuccessful states**
   (states inside and outside of goal set G)
2. **Train binary classifier**
   (with inputs s_i and labels $\mathbf{1}(s_i \in G)$)
3. **Run RL with classifier as reward.**

What can go wrong?

The RL algorithm will seek out states that the classifier thinks are good.

It may simply find states that the classifier wasn't trained on!

—> exploiting the classifiers weaknesses

Goal classifiers

Can we prevent the RL algorithm from exploiting the classifier's weaknesses?

A proposition: Add states that RL visits as negative examples for the classifier

Specifically:

1. Collect initial set of successful states D_+ and unsuccessful states D_-.
2. Update classifier using D_+ and D_-
3. Collect experience $s_t, a_t, \ldots$ using policy π
4. Update policy π using classifier-based reward.
5. Add visited states to negatives: $D_- \leftarrow D_- \cup \{s_t\}$

Why might this work or not work?

Will it learn an accurate classifier?

Do we expect the policy to work?

What will the classifier output for successful states?

Goal classifiers

A proposition: Add states that RL visits as negative examples for the classifier

Specifically:

1. Collect initial set of successful states D_+ and unsuccessful states D_-.
2. Update classifier using D_+ and D_-
3. Collect experience $s_t, a_t, \ldots$ using policy π
4. Update policy π using classifier-based reward.
5. Add visited states to negatives: $D_- \leftarrow D_- \cup \{s_t\}$

Why might this work or not work?

Classifier can't be exploited. 😌

But, what if some of the visited states *are* successful? 😳

As long as batches are balanced, classifier will output p≥0.5 for successful states. 😅

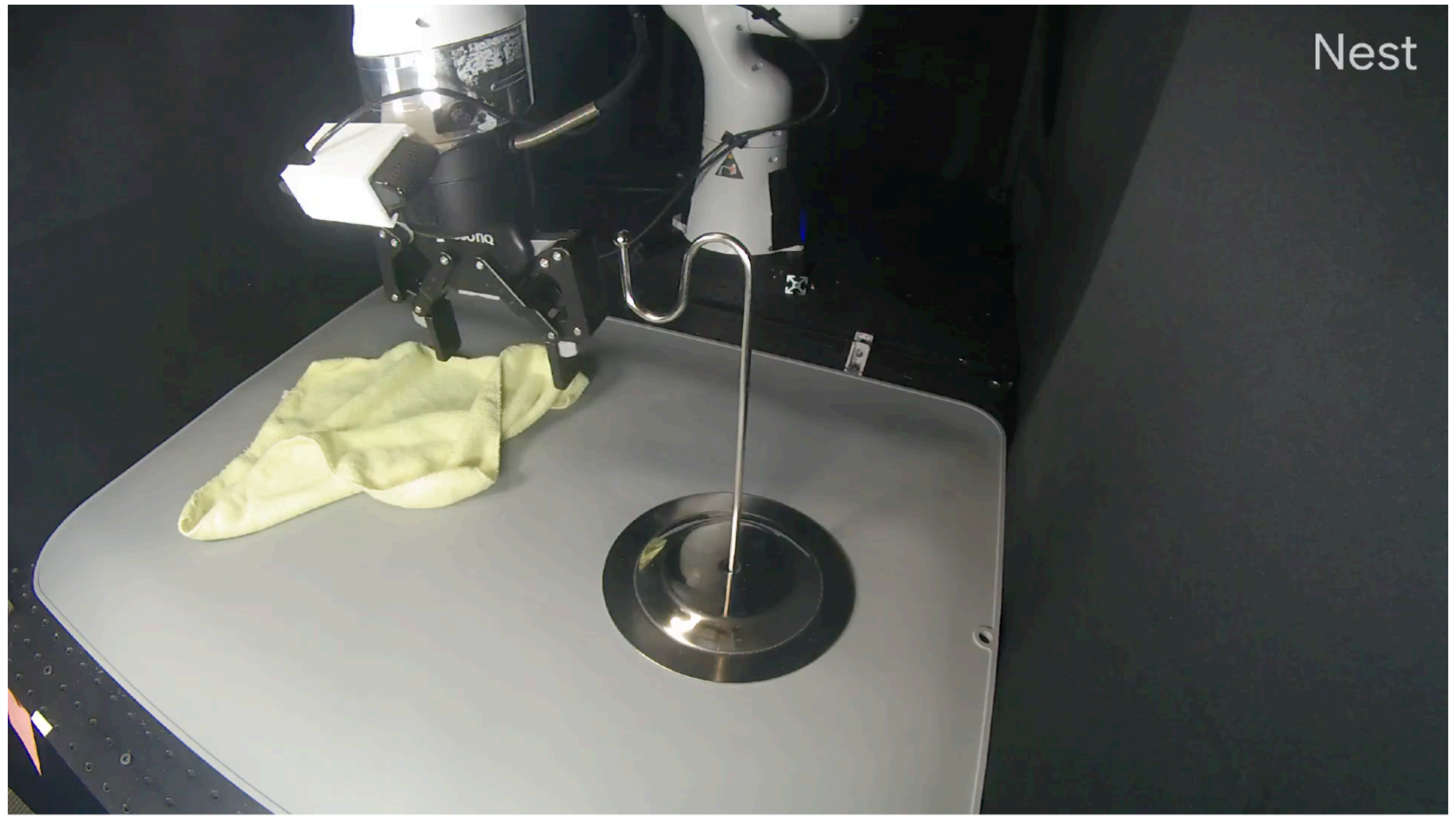
Goal classifiers for robotic reinforcement learning

Collect 50 demonstrations:

- use final states as success state examples
- initialize RL replay buffer with demos

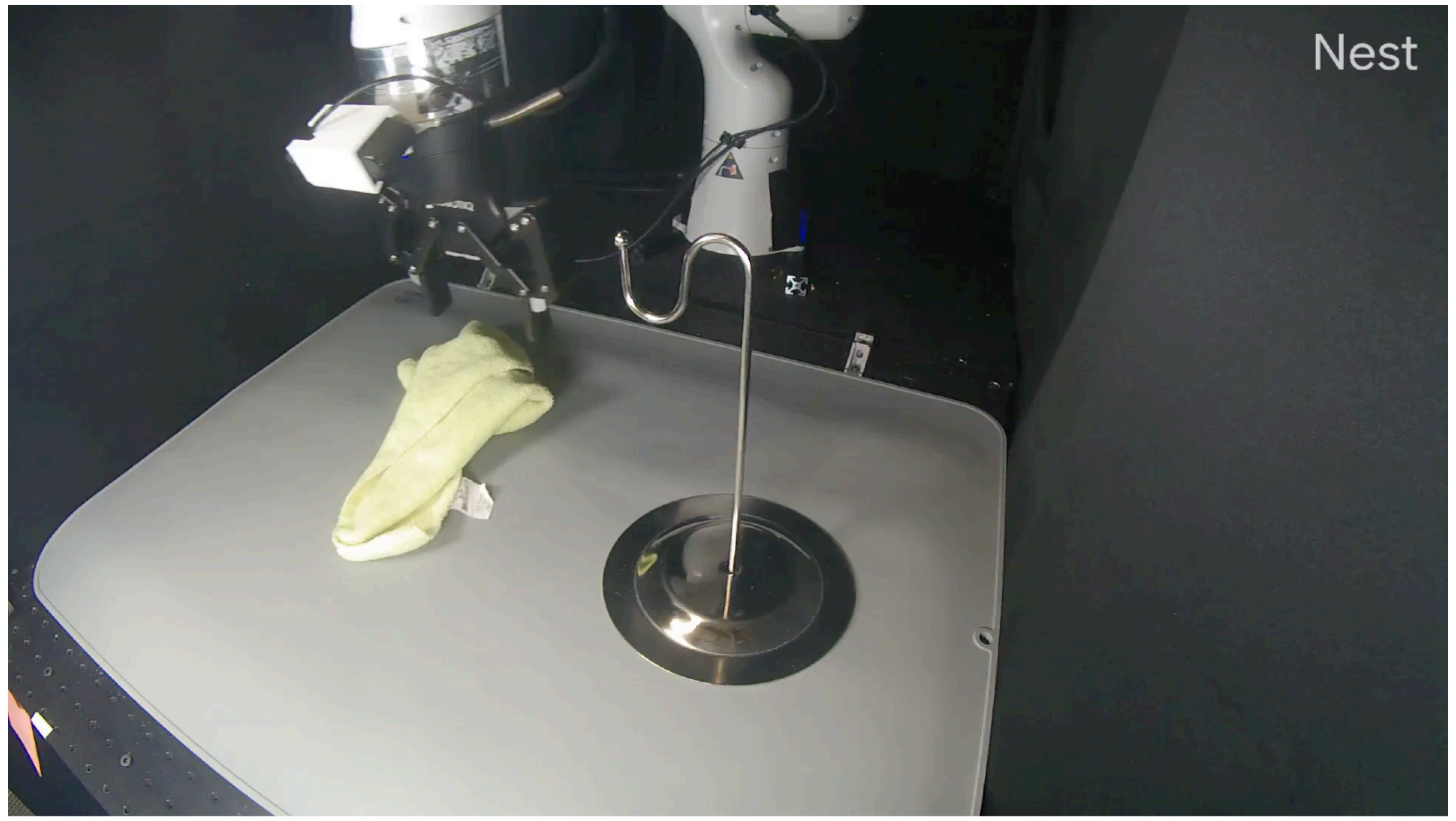
Important to regularize the classifier!

Directly imitating demos:



26% success rate

RL policy trained with learned classifier:



62% success rate

Sharma, Ahmed, Ahmad, Finn. Self-Improving Robots: End-to-End Autonomous Visuomotor Reinforcement Learning. ‘23

Aside: Generative adversarial networks

This is also how GANs work!

1. Train classifier to discriminate between real data and generated data
2. Train generator to generate data that the classifier thinks is real.

"Yorkshire terrier" generated by VQ-GAN

Yu et al. ViT-VQGAN. ICLR '22.

Phenaki video generation (uses GAN-loss)

Side view of an astronaut is walking through a puddle on mars.
The astronaut is dancing on mars.
The astronaut walks his dog on mars.
The astronaut and his dog watch fireworks.

Villegas et al. Phenaki. ICLR '23.

At convergence: generator will match data distribution p(x)

From goal classifiers to more general rewards

What if you aren't trying to reach a goal?

Given: Demonstration trajectories $D := \{(s_1, a_1, \ldots)\}$ from expert policy π_{exp}

Goal: Match expert state-action distribution, $d^{\pi_{exp}}(s, a)$

Can use the same algorithm as before!

Positive examples: all (s_t, a_t) from demos D

Negative examples: (s_t, a_t) from policy collected during RL

(similar to before)

"*generative adversarial imitation learning*"

Recap of reward classifiers

- Pre-trained classifiers can be exploited when optimized against.
- Solution: Update the classifier *during RL*, using policy data as negatives
- Can learn goal classifier with success examples, full reward with demos.

+ practical framework for task specification

~ adversarial training can be unstable

(though variety of regularization tricks from GAN literature)

- requires examples of desired behavior or outcomes

The plan for today

1. Where do rewards come from?
2. Learning rewards from example goals, behaviors
3. **Learning rewards from human preferences**

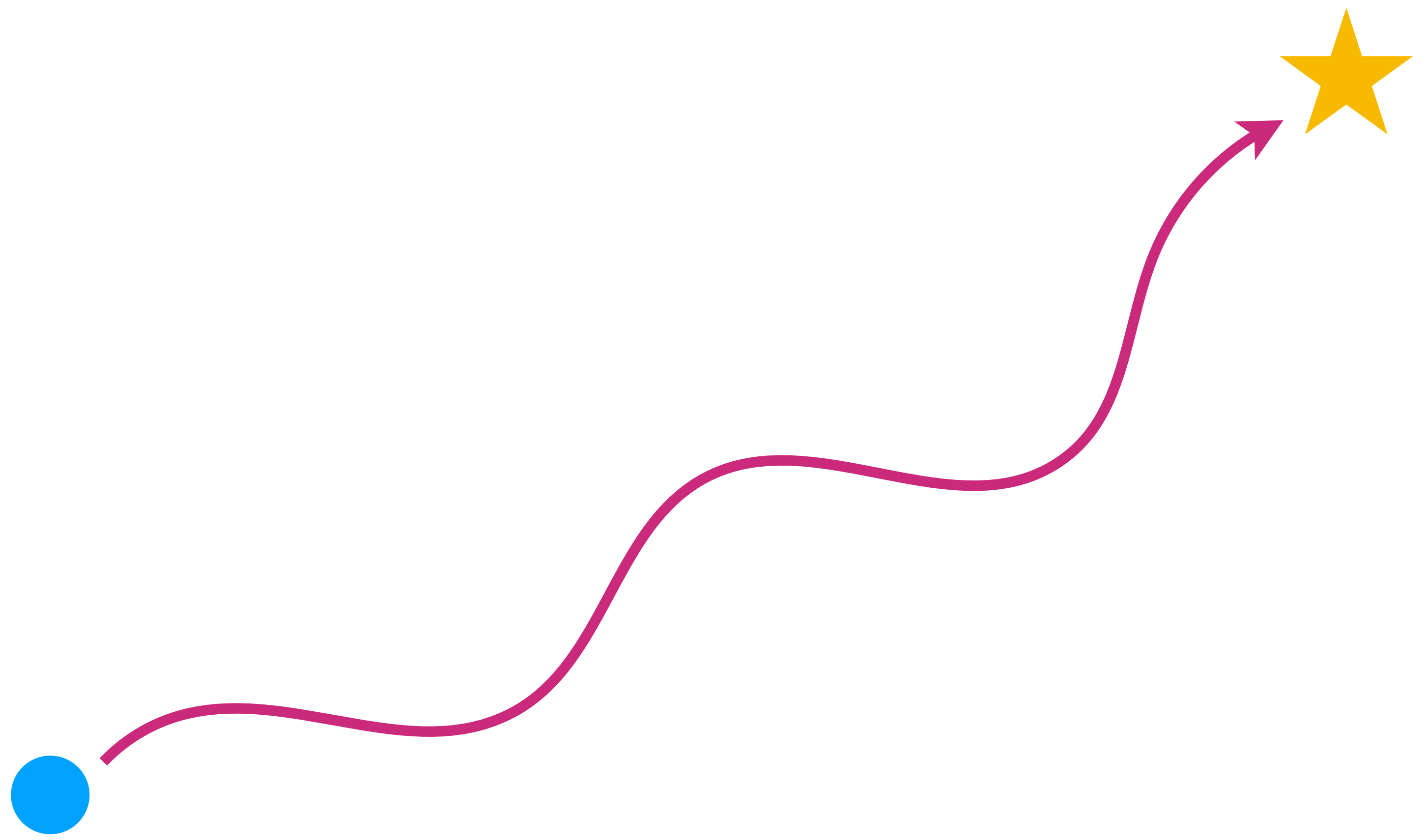
Can humans provide feedback on policy roll-outs?

instead of requiring demos or example goals

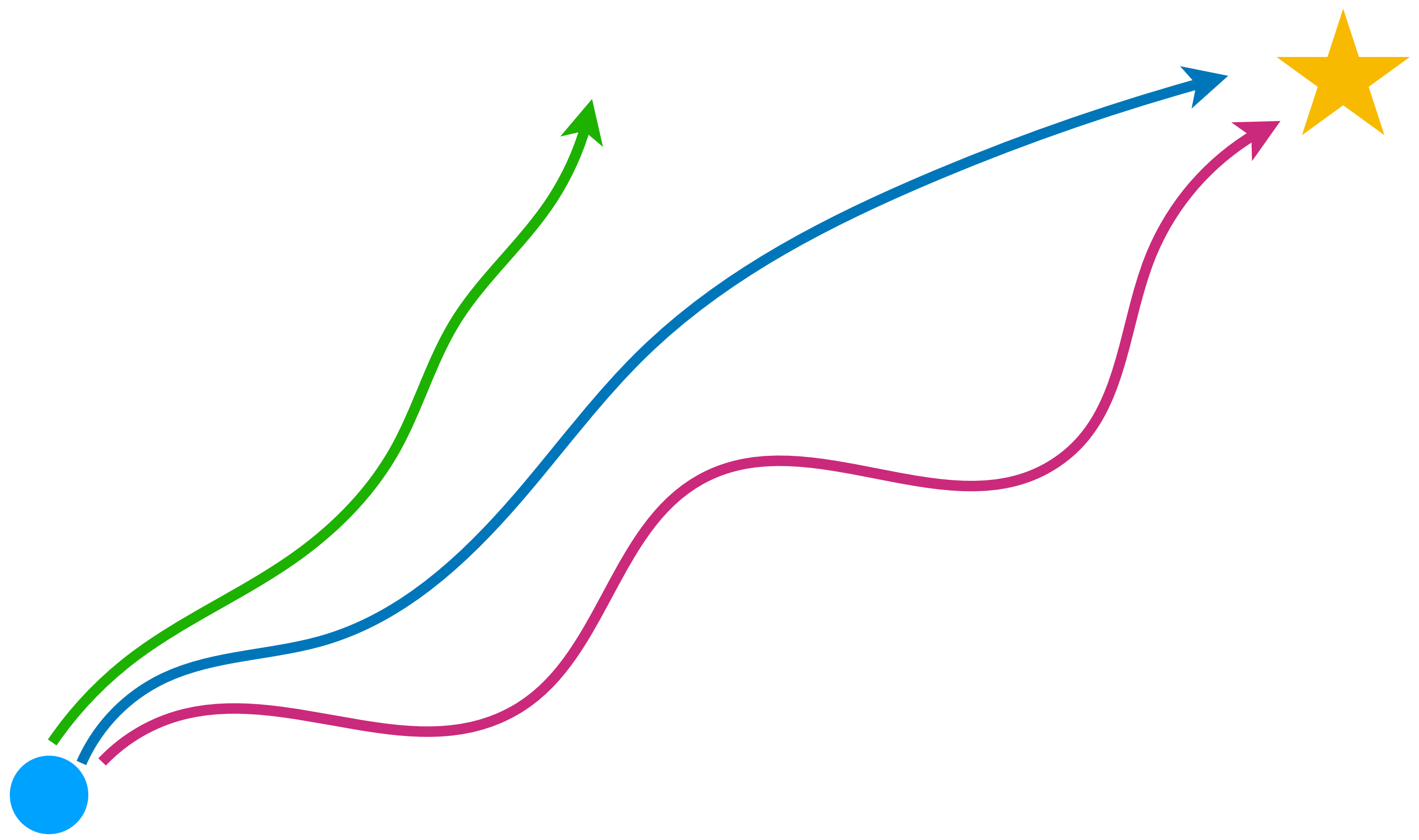
(or in addition to!)

A couple options

"How good is this trajectory?"



"Which trajectory is better?"



Relative **preferences** are easier to provide!

How to learn a reward function from human preferences?

Human says τ_w is better than τ_l. (can write it as $\tau_w \succ \tau_l$)

Note: τ could be a *full* or *partial* roll-out!

We want a reward r_θ such that $$\underbrace{\sum_{(s,a)\in\tau_w} r_\theta(s,a)}_{\text{shorthand } r_\theta(\tau_w)} > \sum_{(s,a)\in\tau_l} r_\theta(s,a).$$

Humans are classifying which trajectory is better.

Reward should be discriminative as well.

How to learn a reward function from human preferences?

Human says τ_w is better than τ_l. (can write it as $\tau_w \succ \tau_l$)

Note: τ could be a *full* or *partial* roll-out!

We want a reward r_θ such that $\underbrace{\sum_{(s,a)\in\tau_w} r_\theta(s,a)}_{\text{shorthand } r_\theta(\tau_w)} > \sum_{(s,a)\in\tau_l} r_\theta(s,a)$.

We can define $\sigma\left(r_\theta(\tau_a) - r_\theta(\tau_b)\right)$ to be the estimated probability that $\tau_a \succ \tau_b$.

Then, we can maximize log probability: $\max_\theta \mathbb{E}_{\tau_w,\tau_l}\left[\log \sigma\left(r_\theta(\tau_w) - r_\theta(\tau_l)\right)\right]$

How to learn a reward function from human preferences?

Complete reward learning algorithm

1. Given dataset $\{\tau_i\}$, sample batches of k trajectories and ask humans to rank.

(for LLMs, these k trajectories all have the same prompt)

2. Compute $r_\theta(\tau_1), \ldots, r_\theta(\tau_k)$ under current reward model r_θ

3. For all $\binom{k}{2}$ pairs per batch, compute $\nabla_\theta \mathbb{E}_{\tau_w, \tau_l}\left[\log \sigma\left(r_\theta(\tau_w) - r_\theta(\tau_l)\right)\right]$ where $\tau_w \succ \tau_l$

4. Update θ using computed gradient.

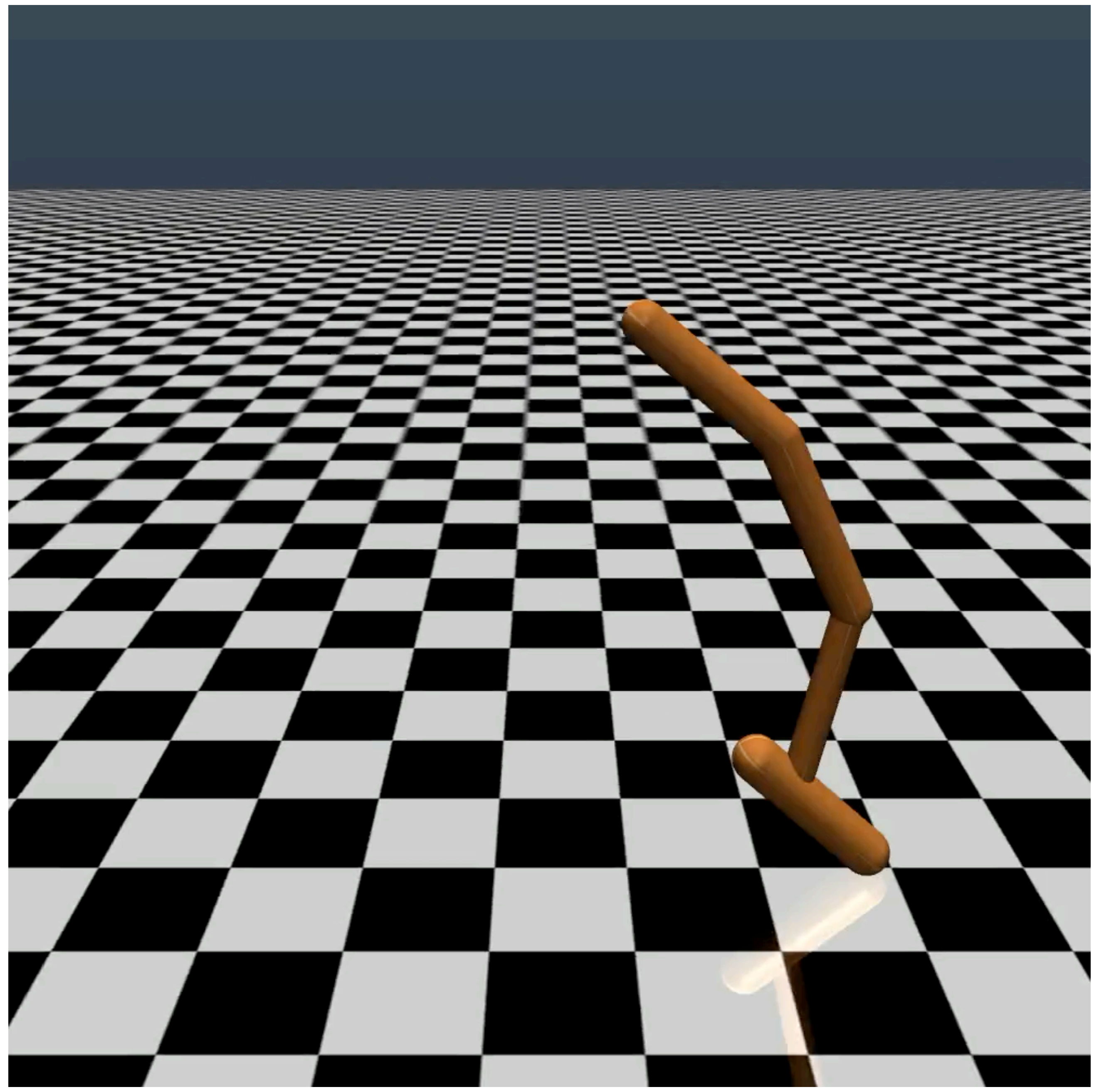
Some notes: This can be done in the loop of online RL.

The preferences could be provided by another model!

Learning rewards from human feedback

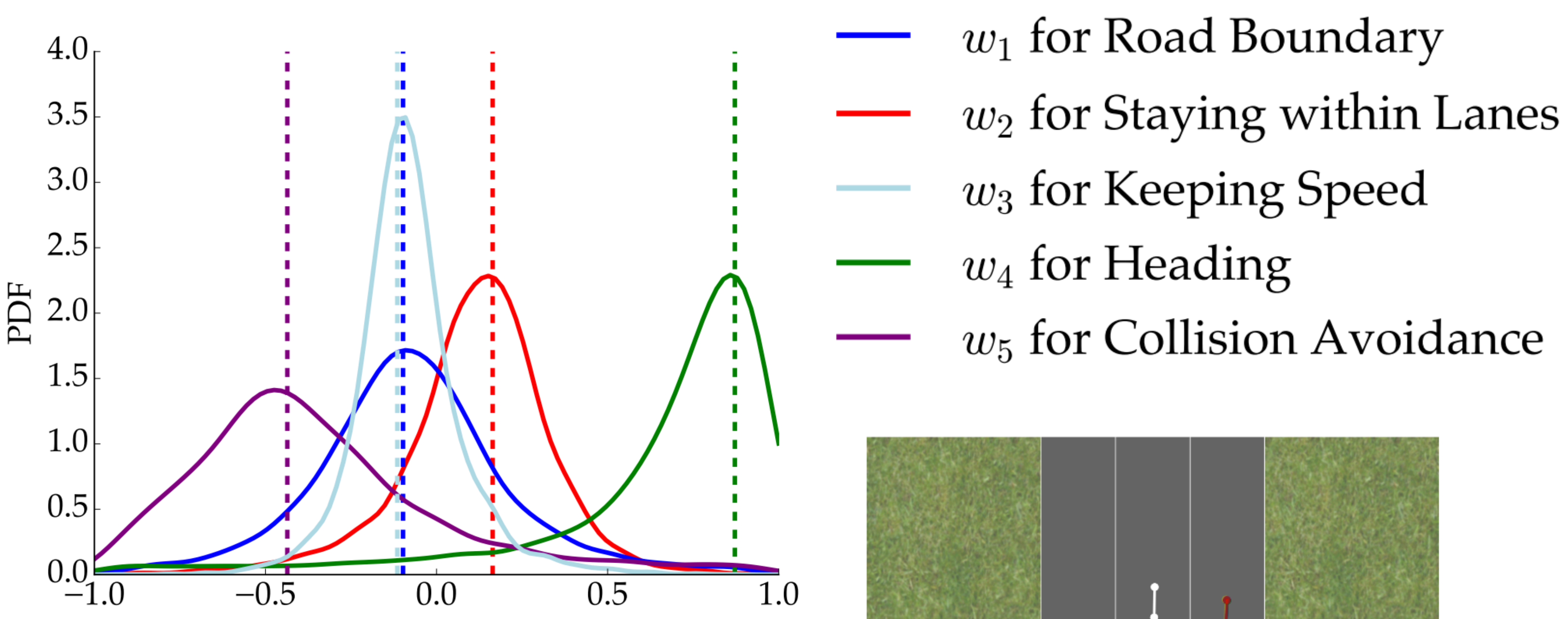
Learning rewards in the loop of online RL

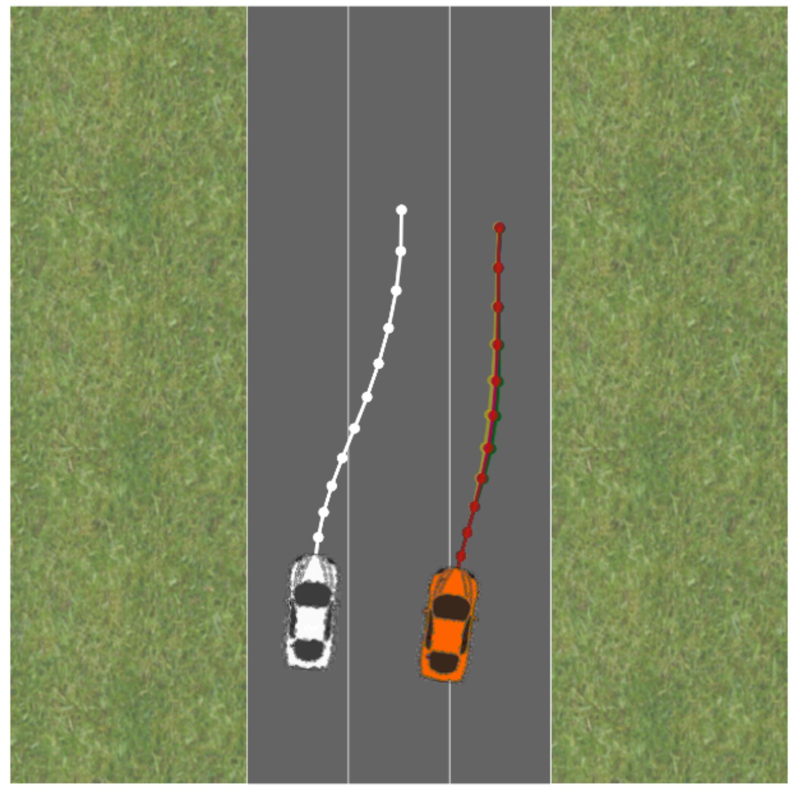
uses 900 human preference queries



Learning rewards for driving

to weight different factors





Christiano et al. Deep Reinforcement Learning from Human Preferences. NeurIPS 2017.

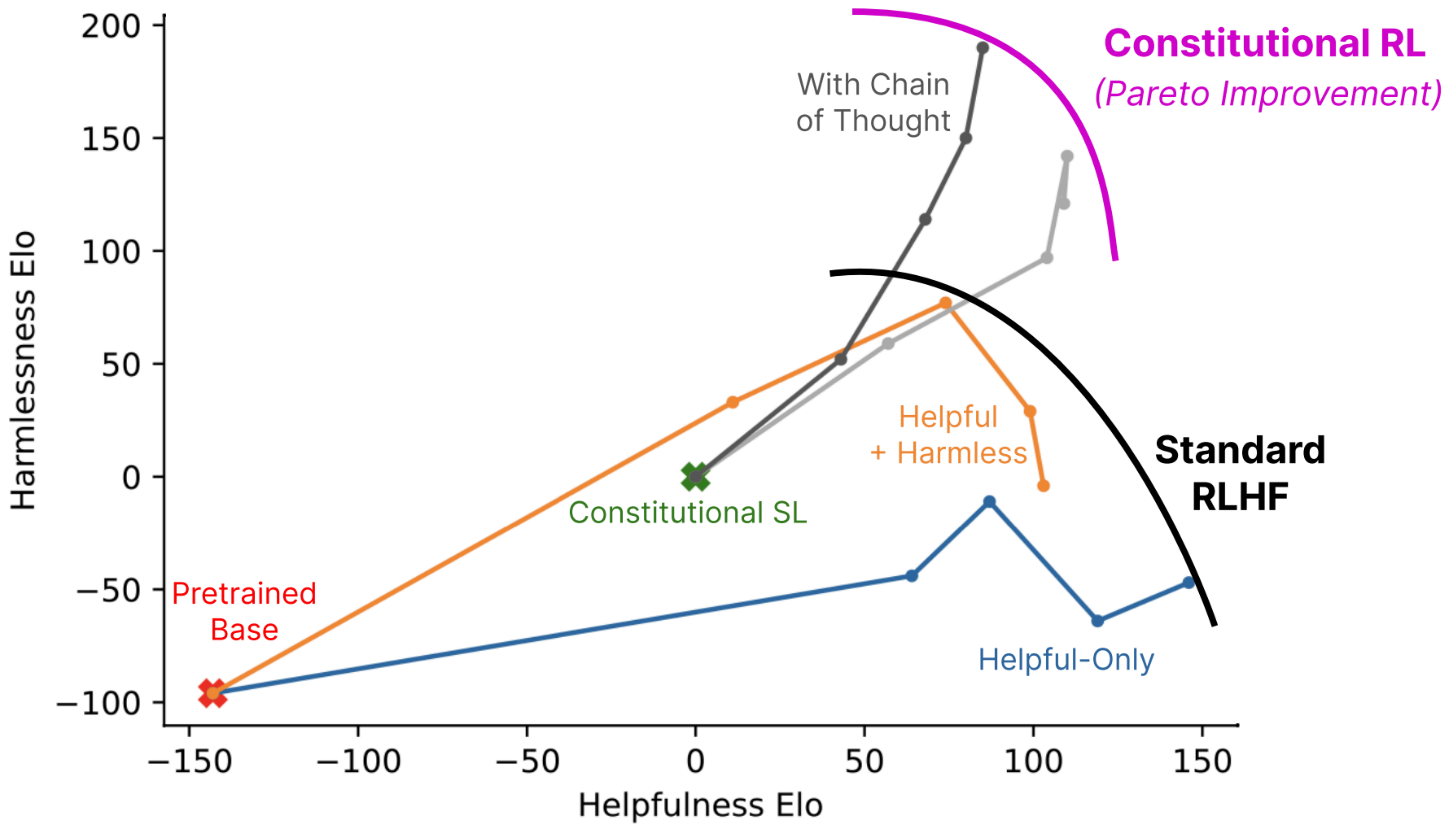
Sadigh et al. Active Preference-Based Learning of Reward Functions. RSS 2017.

Learning rewards from ~~human~~ AI feedback

Reinforcement learning with AI feedback (RLAIF)

Ask another language model “which of these responses is less harmful?”

Key insight: critique is easier than generation!



Bai et al. Constitutional AI: Harmlessness from AI Feedback. 2022.

Summary of Reward Learning

#1 Takeaway: Rewards can't be taken for granted!

Learning rewards from goals, demos

+ practical framework for task specification

~ adversarial training can be unstable
(though variety of regularization tricks from GAN literature)

- requires examples of desired behavior or outcomes

Learning rewards from human preferences

+ pairwise preferences easy to provide
(doesn't require example goals, demos!)

+ has been deployed at scale!

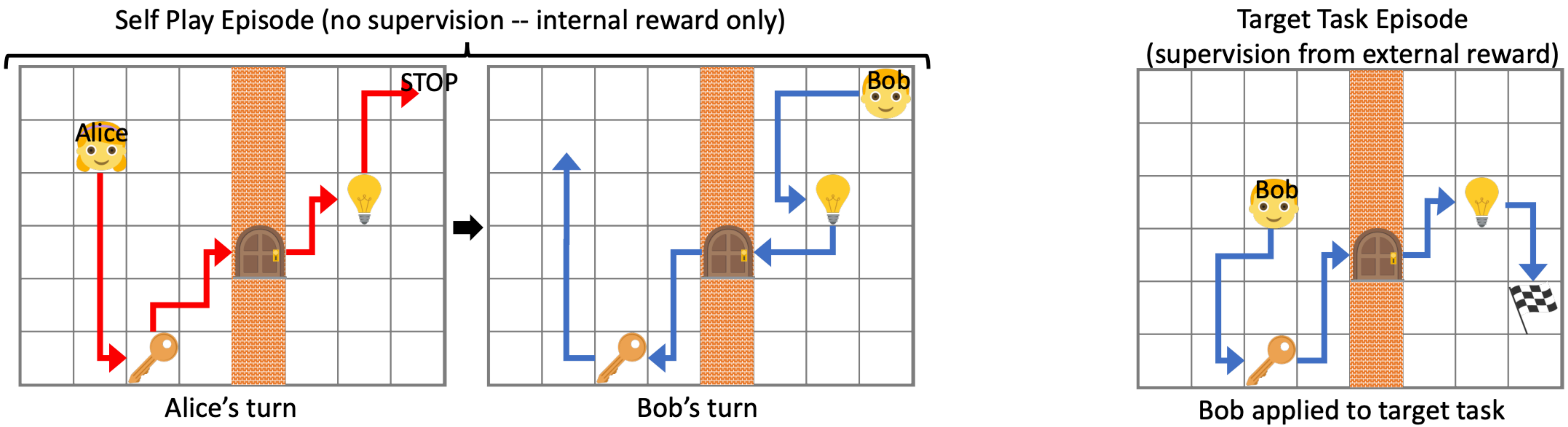
- may require supervision in the loop of RL
(usually requires more human time)

Thought exercise: Are there other forms of feedback, supervision that might be helpful?

Can RL agents propose their own goals?

Entire area of "unsupervised" RL

One example: Formulate two-player game, with a goal-setter and a goal reacher



Sukhbaatar et al. Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play. ICLR 2018

The plan for today

1. Where do rewards come from?
2. Learning rewards from example goals, behaviors
3. Learning rewards from human preferences

Key learning goals:

- why task specification is hard (& why naïve methods fail)
- methods for learning rewards from human supervision

Course reminders

- Project proposal due **tonight**.
  (graded fairly lightly — really for your benefit!)
- Homework 2 due next Wednesday (**start early!**)

Next week: Reinforcement learning from offline datasets