Alignment and Active Learning in HRI

Michelle Zhao,

October 29, 2024

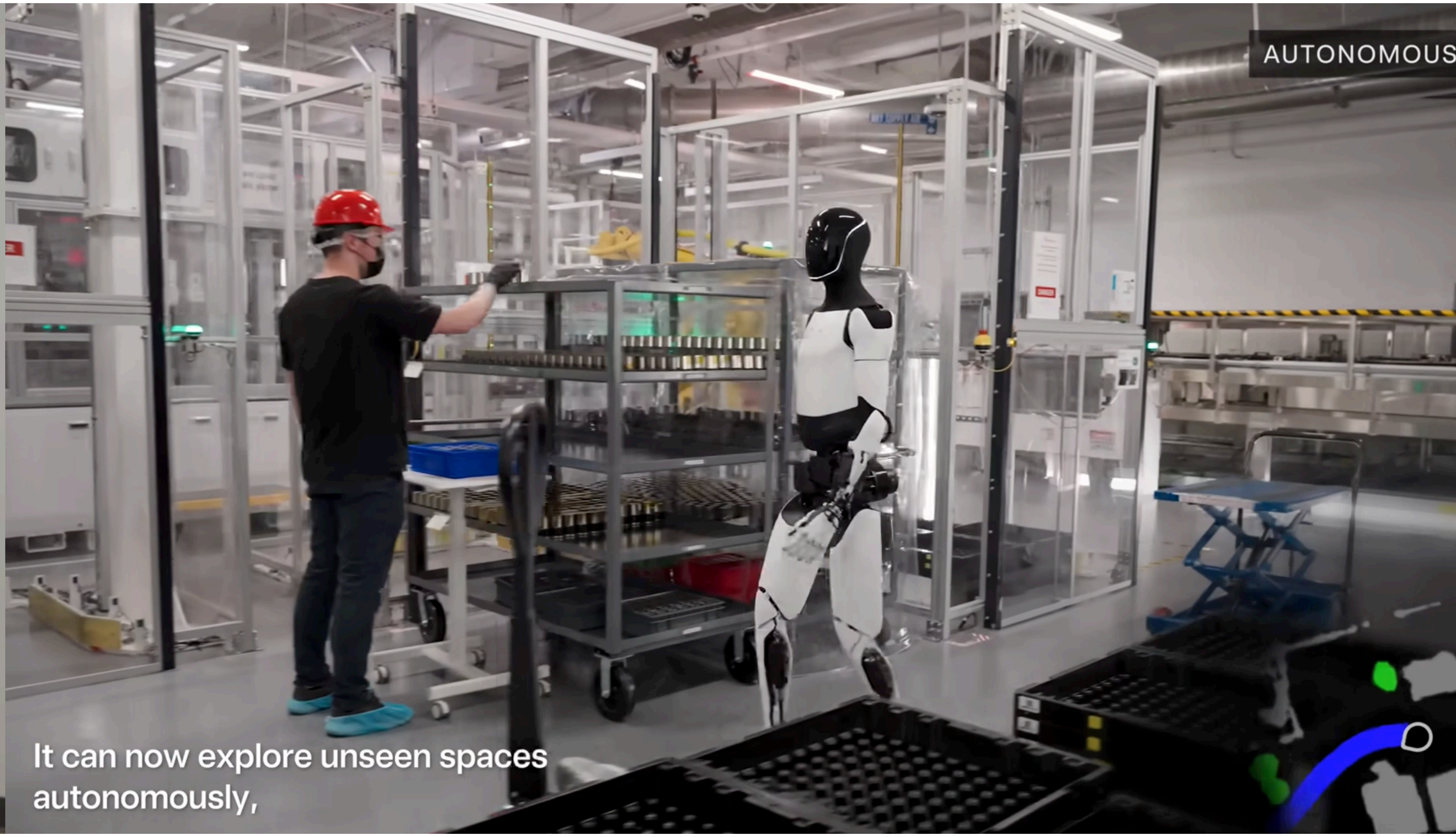
Outline

- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference ***active***
- Case Study 2: Active learning for black-box policies

Outline

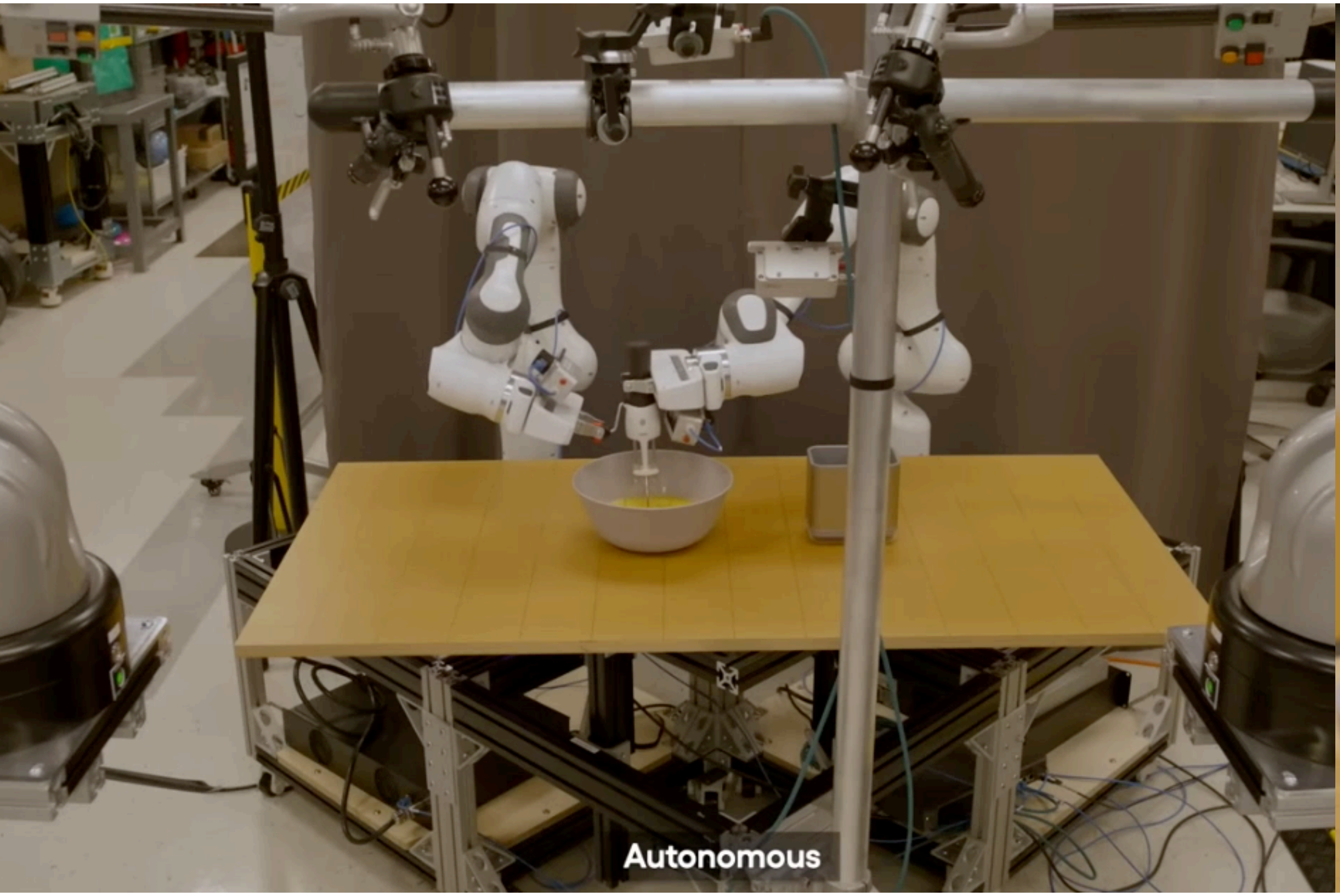
- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference *active*
- Case Study 2: Active learning for black-box policies

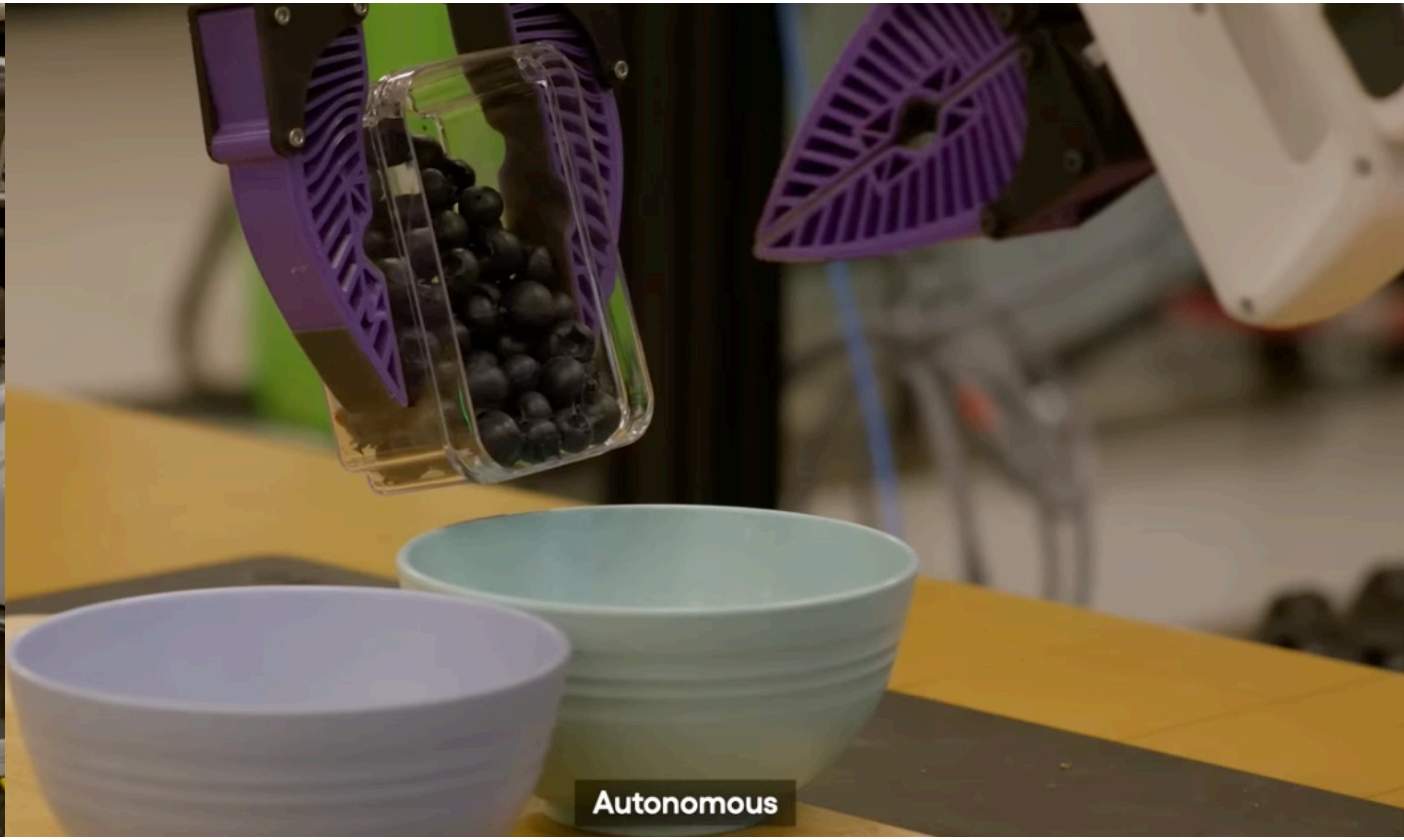
We're starting to see remarkable strides in learning for robotics



1X, Tesla

We're starting to see remarkable strides in learning for robotics





Underlying Aim: Robots that behave as we want them to!

Source

Source

Alignment in Robotics

How can we get robots to do what we want them to?

Leike, Jan, et al. "Scalable agent alignment via reward modeling: a research direction."

Robots don’t know what we want

BBC

Home News Sport Business Innovation Culture Travel Earth Video Live

Burger-flipping robot taken offline after one day

9 March 2018

Share

WATCH: Flippy the burger robot gets to work

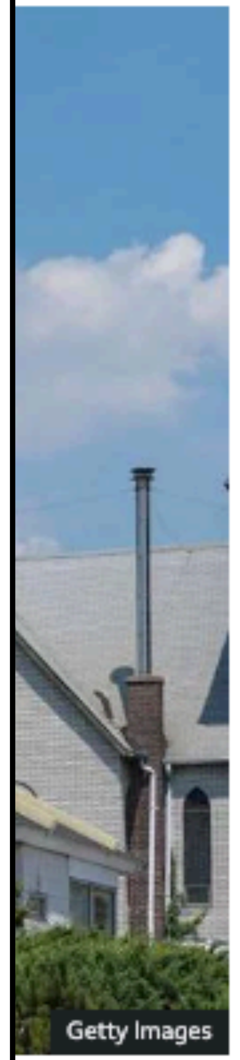
Flippy the burger-flipping robot that started work this week in a California restaurant has been forced to take a break because it was too slow.

The robot was installed at a Cali Burger outlet in Pasadena and replaced human cooks.

BBC

Home News Sport Business Innovation Culture Travel Earth Video Live

Bacon ice cream and nugget overload sees misfiring McDonald's AI withdrawn



Asia / East Asia

AI fail: Japan’s Henn-na Hotel dumps ‘annoying’ robot staff, hires humans

- Dinosaur receptionists are a thing of the past as Japan’s first robot hotel concludes there “are places where they are just not needed”

Listen to this article

Julian Ryall + FOLLOW

Published: 12:32pm, 16 Jan 2019

Why you can trust SCMP

Toyota Research Institute

Outline

- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference *active*
- Case Study 2: Active learning for black-box policies

Outline

- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference ***active***
- Case Study 2: Active learning for black-box policies

Alignment in Robotics

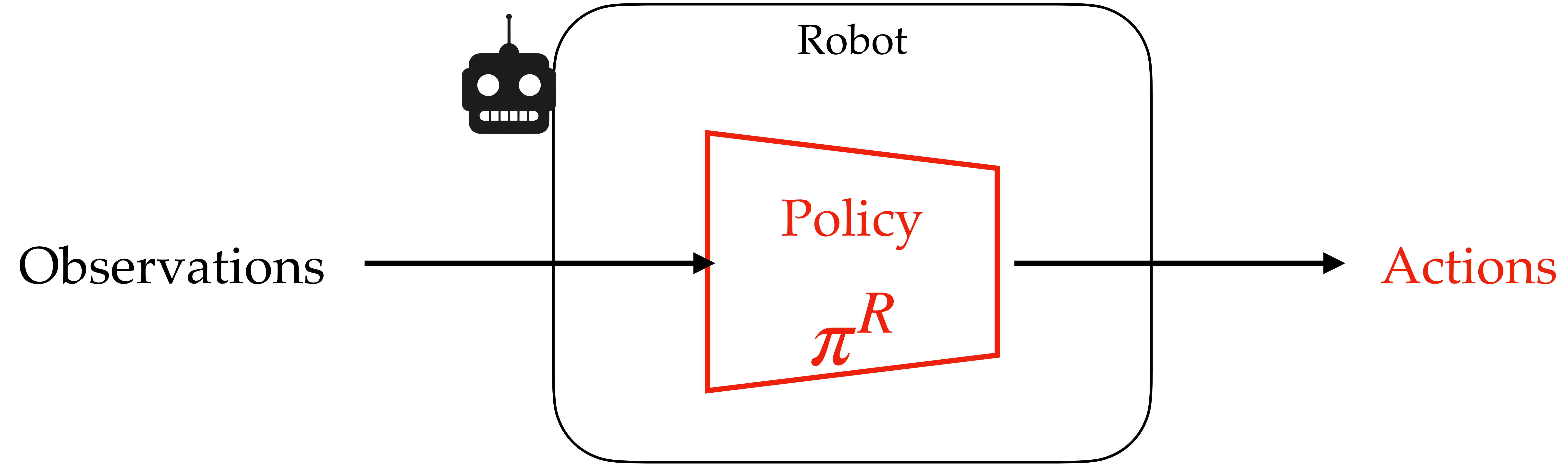
How can we get robots to do what we want them to?

Leike, Jan, et al. "Scalable agent alignment via reward modeling: a research direction."

Alignment in Robotics

How can we get robots to do what we want them to?

Take actions

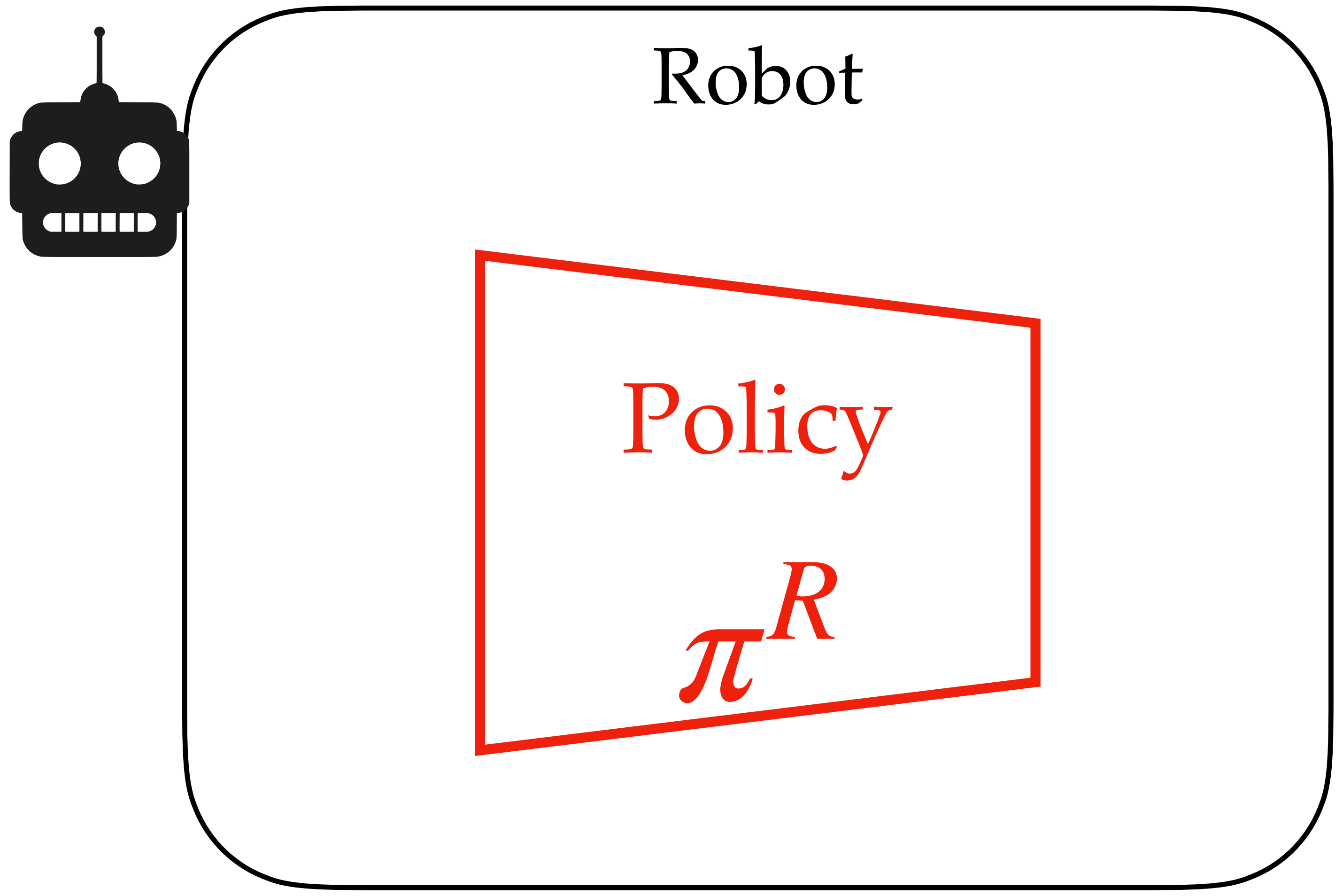


Alignment in Robotics

How can we get robots to do what we want them to?

Take actions

Achieve human objectives



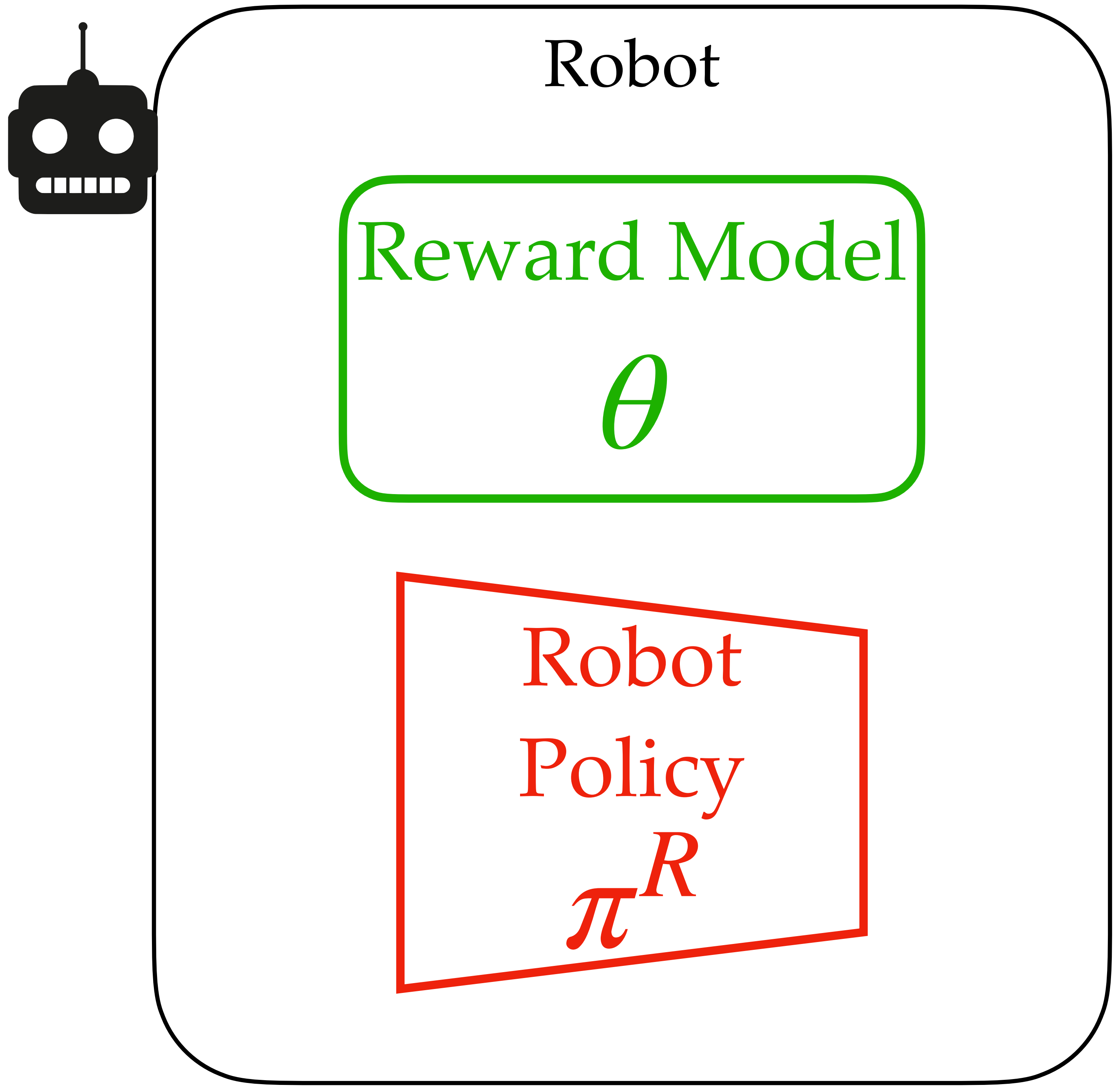
Alignment in Robotics

How can we get robots to do what we want them to?

Learning from human feedback

Take actions

Achieve human objectives



Alignment in Robotics

How can we get robots to do what we want them to?

Learning from human feedback

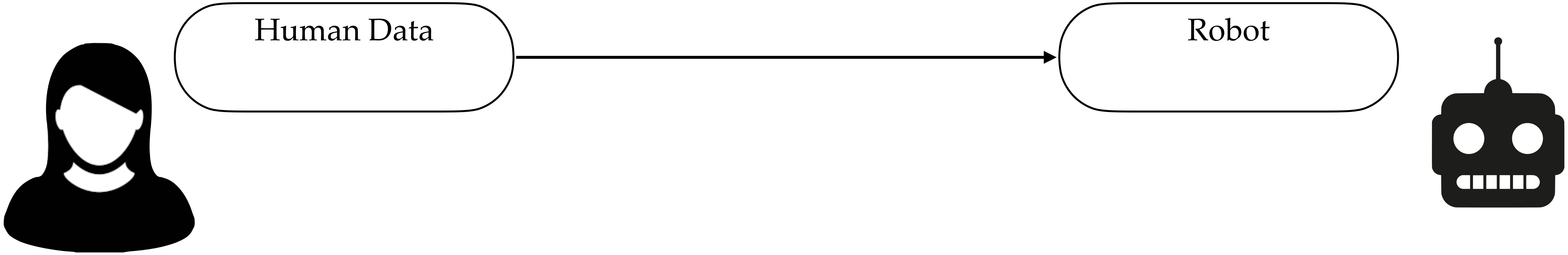
to have robots

Take actions

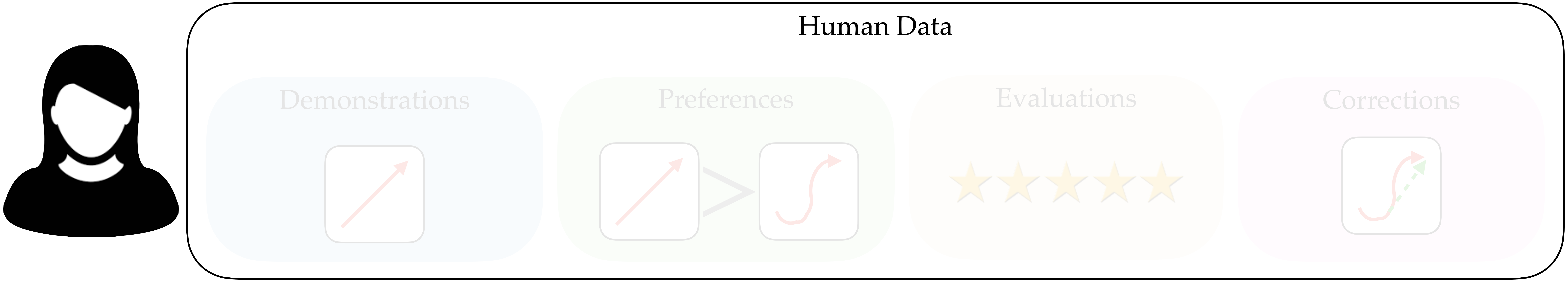
that

Achieve human objectives

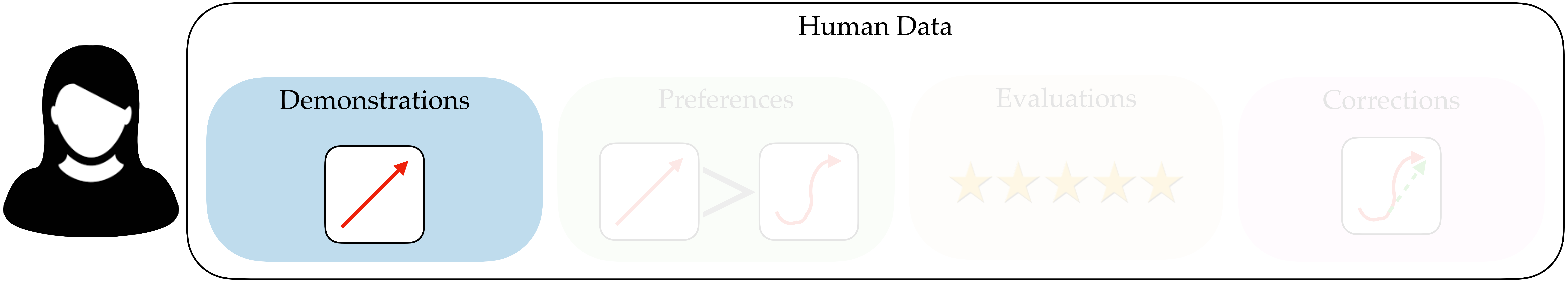
Learning from human feedback

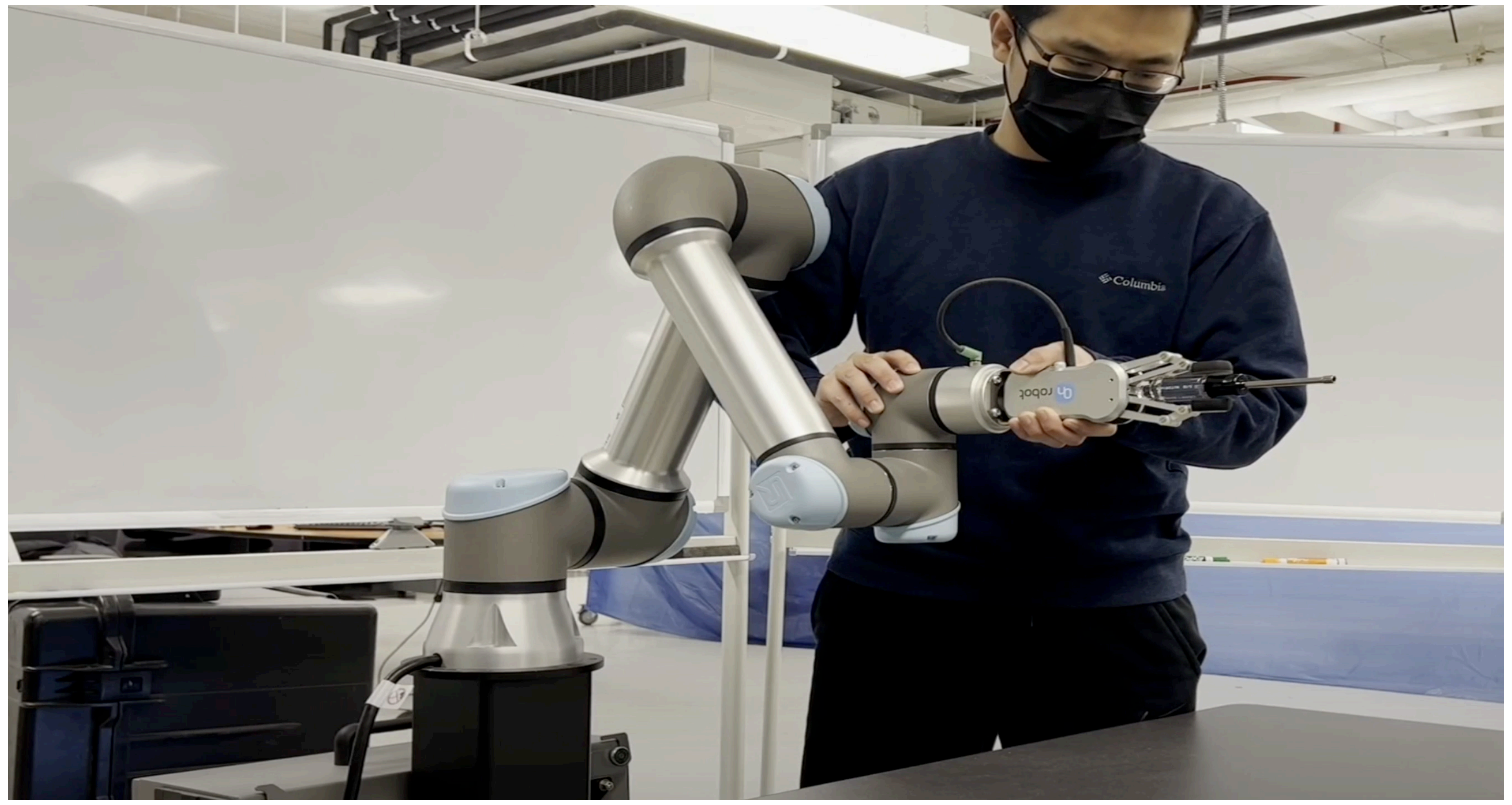


Learning from human feedback



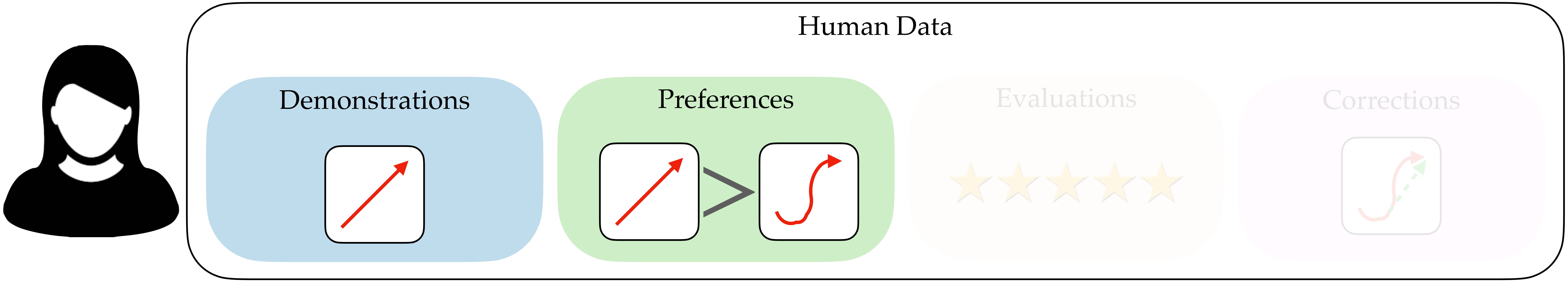
Learning from human feedback

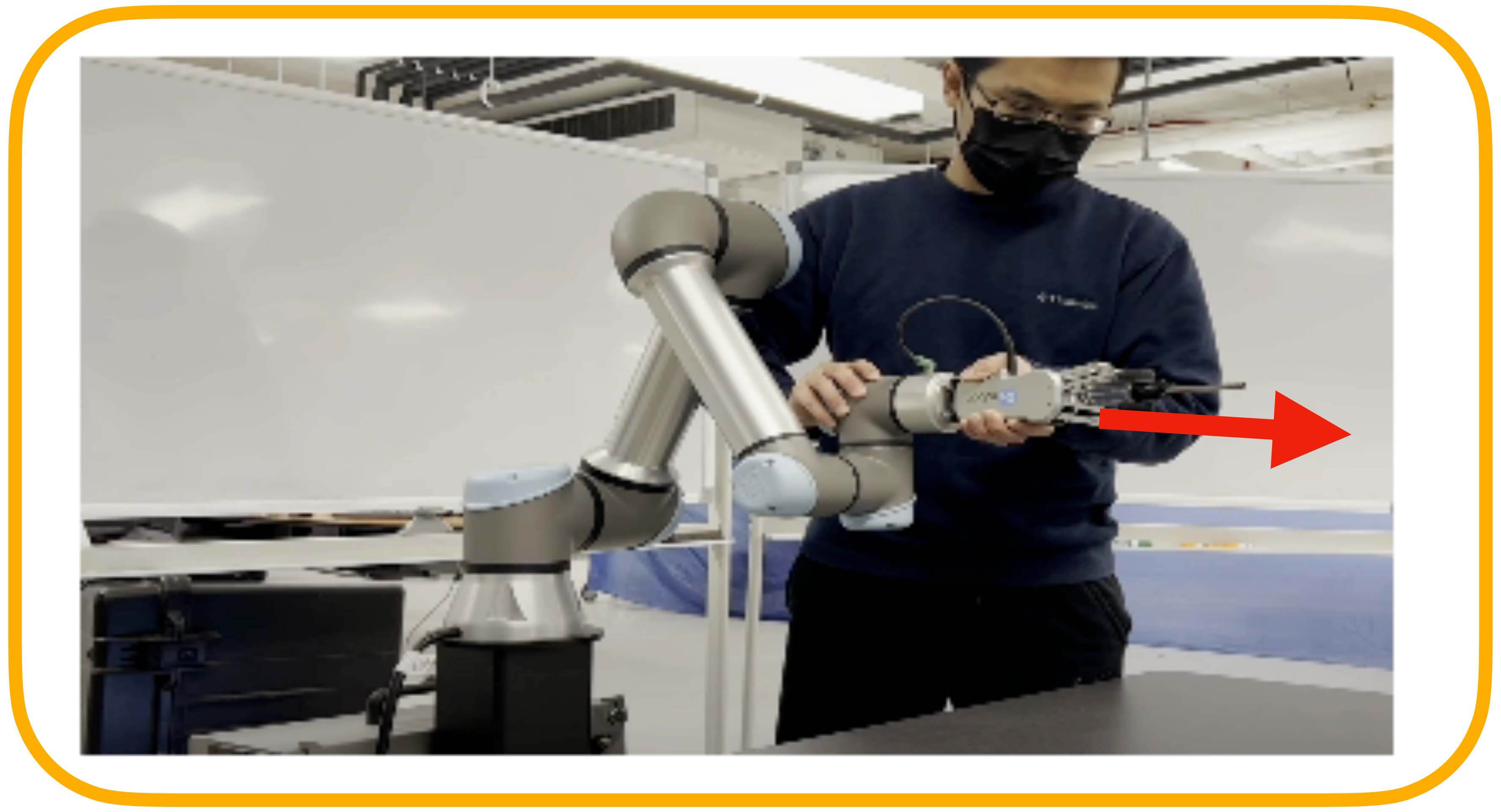


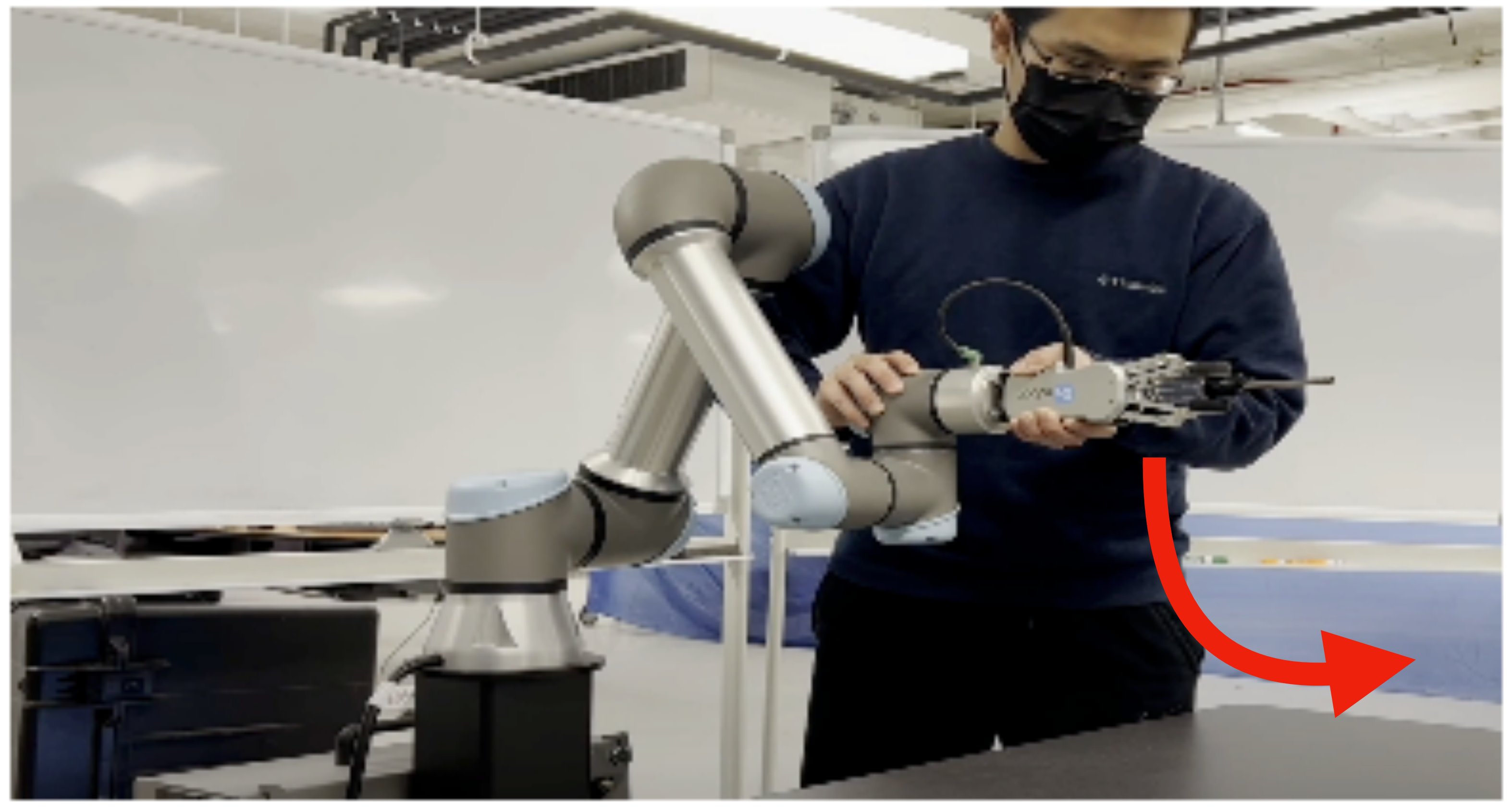


Robot Learning from Demonstration

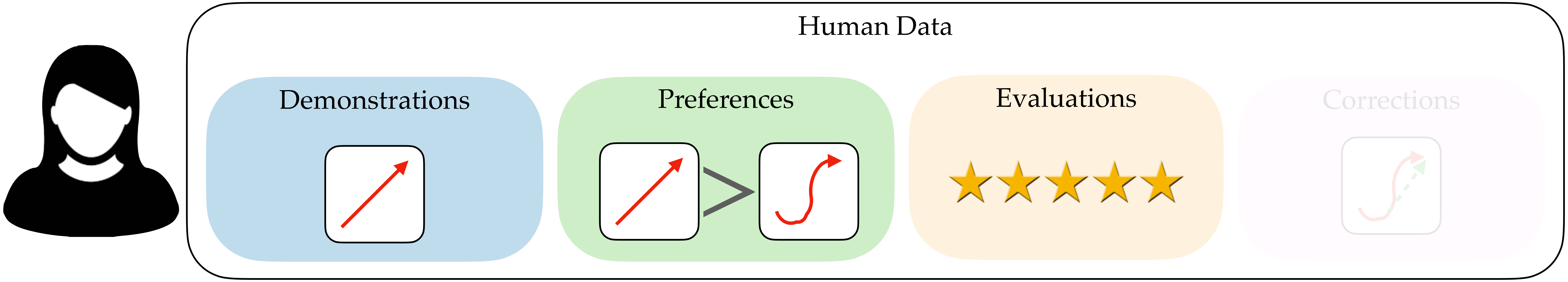
Learning from human feedback

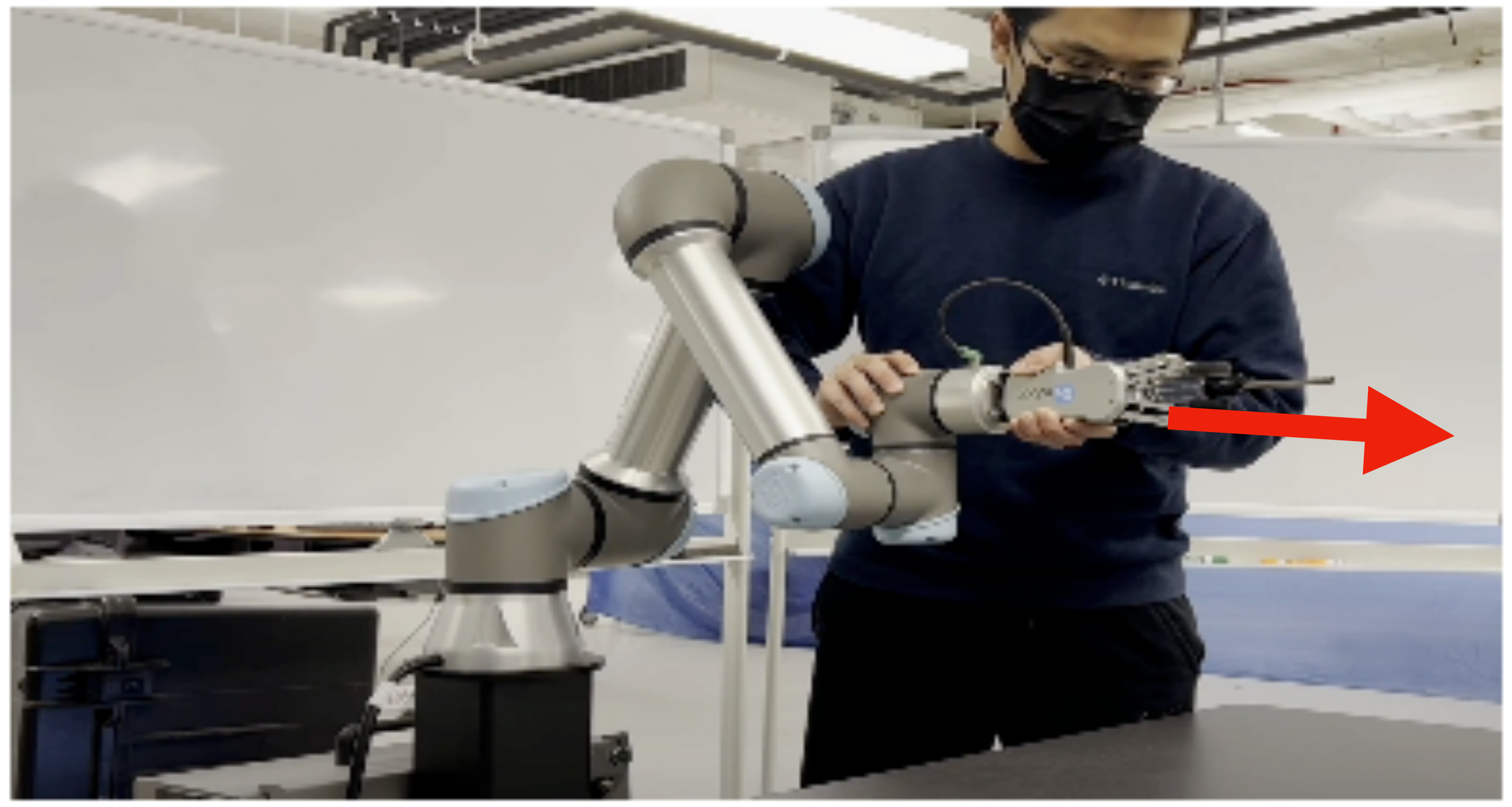




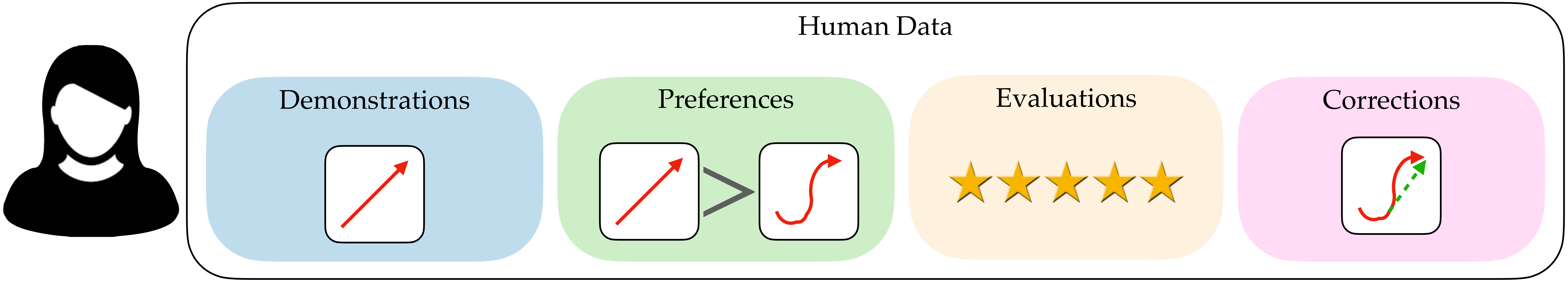


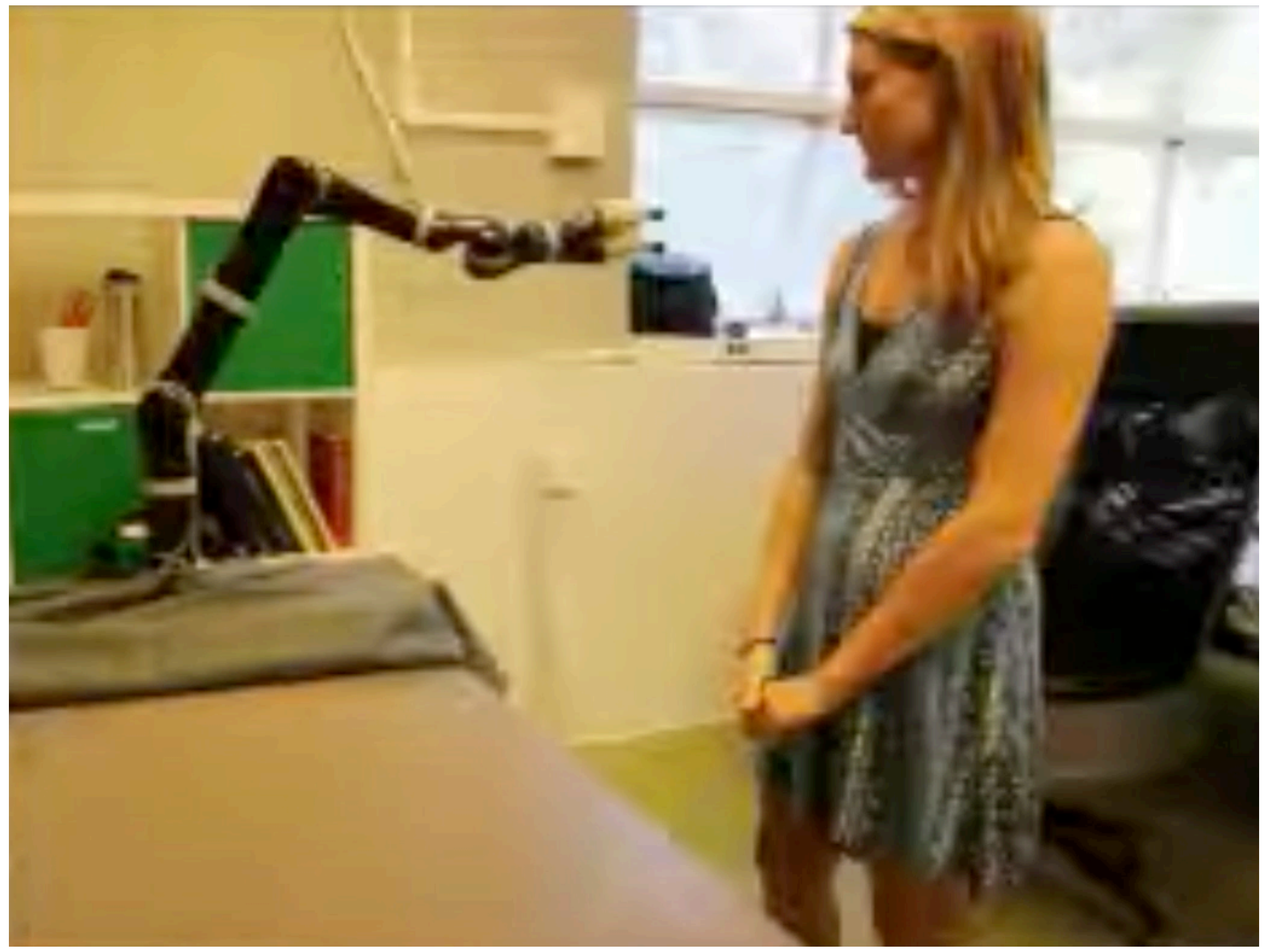
Learning from human feedback





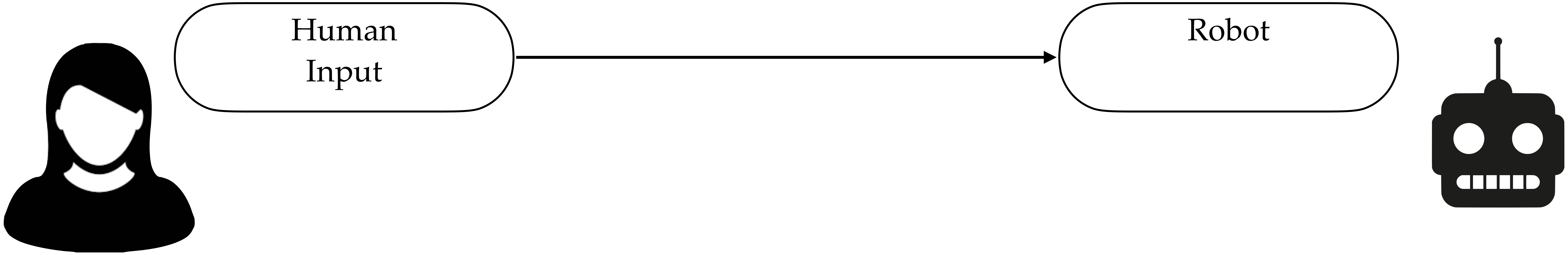
Learning from human feedback



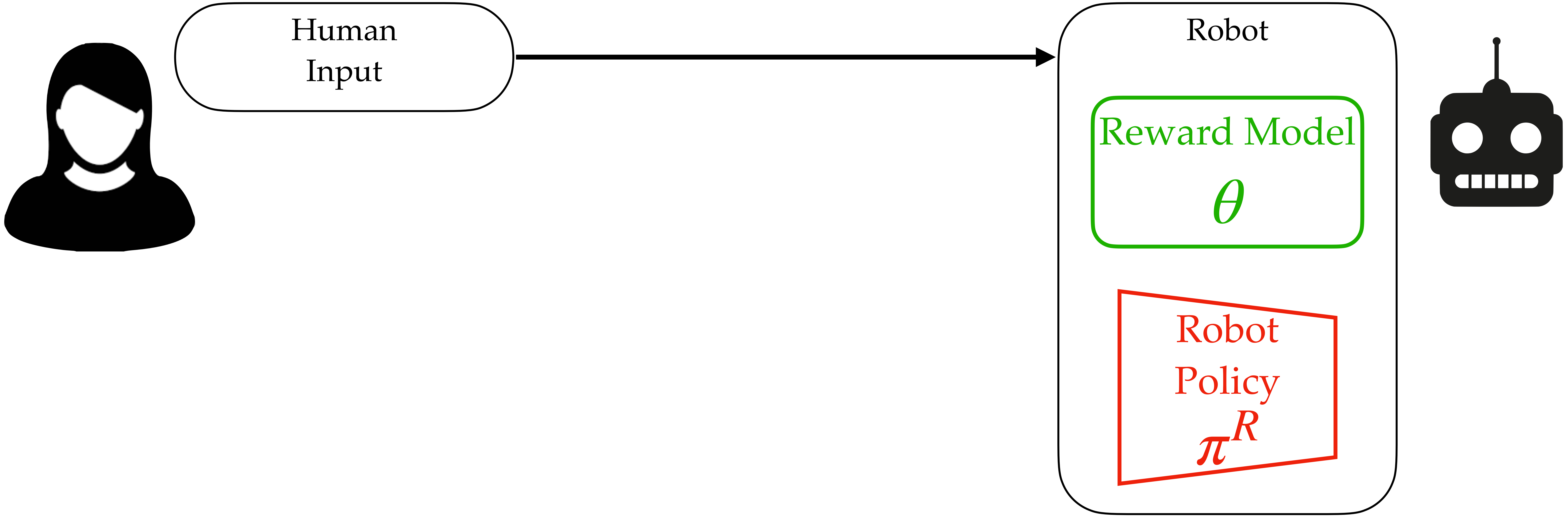


Learning from Physical Human Corrections, One Feature at a Time

Learning from human feedback



Learning from human feedback

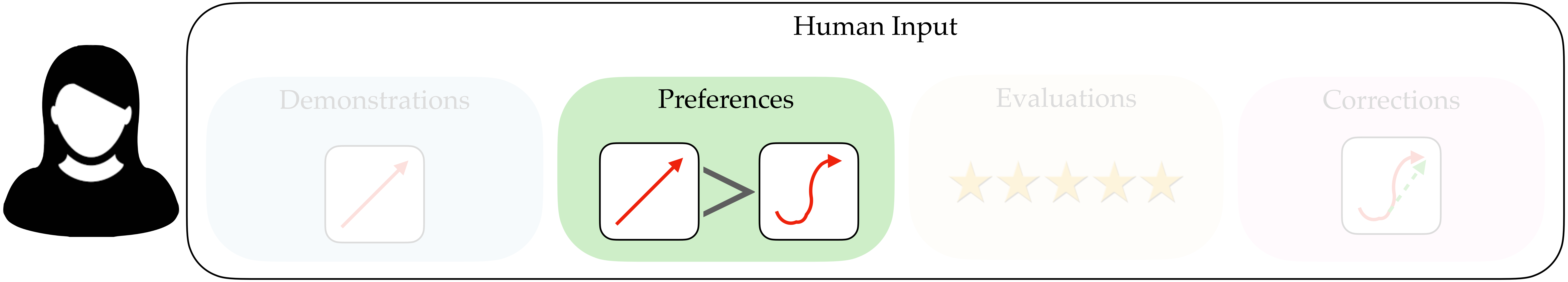


Outline

- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference ***active***
- Case Study 2: Active learning for black-box policies

Outline

- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference *active*
- Case Study 2: Active learning for black-box policies

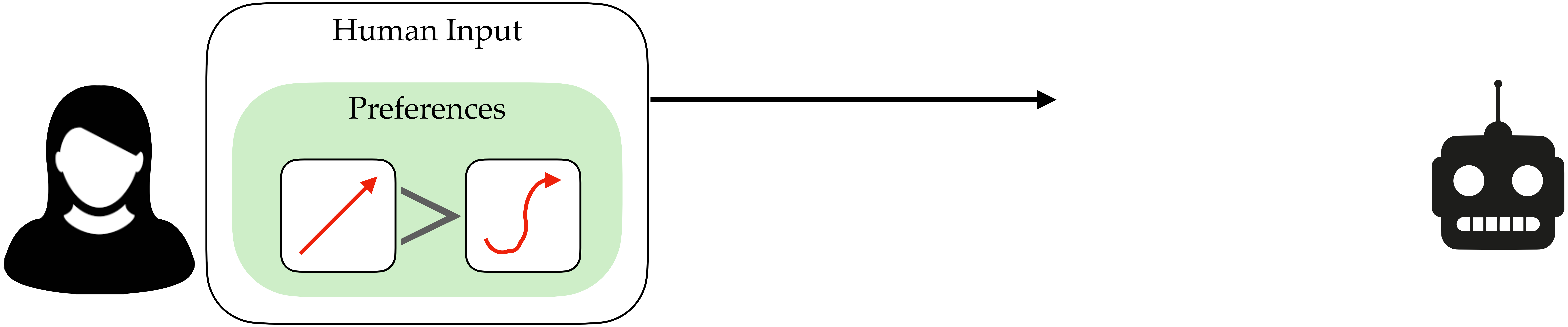


Let’s take a closer look 👀 at

Active **Preference-Based Learning of Reward Functions**

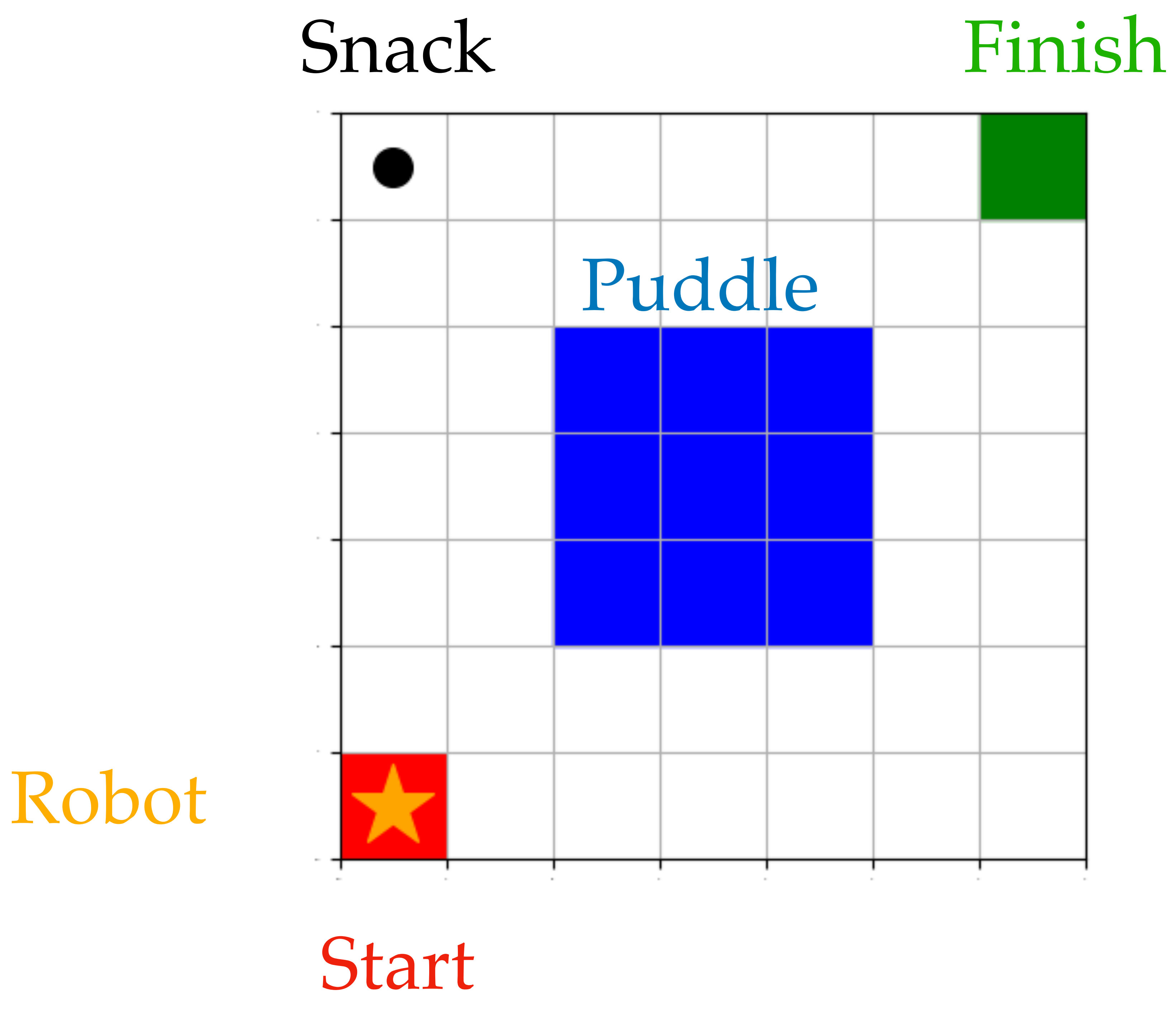
Sadigh, Dorsa, et al. *Active preference-based learning of reward functions*. 2017.
Biyik, Erdem, and Dorsa Sadigh. "Batch active preference-based learning of reward functions." *Conference on robot learning*. PMLR, 2018.
Bıyık, Erdem, et al. "Asking easy questions: A user-friendly approach to active reward learning." *arXiv preprint arXiv:1910.04365* (2019).

Preference-based learning: Interaction Setup



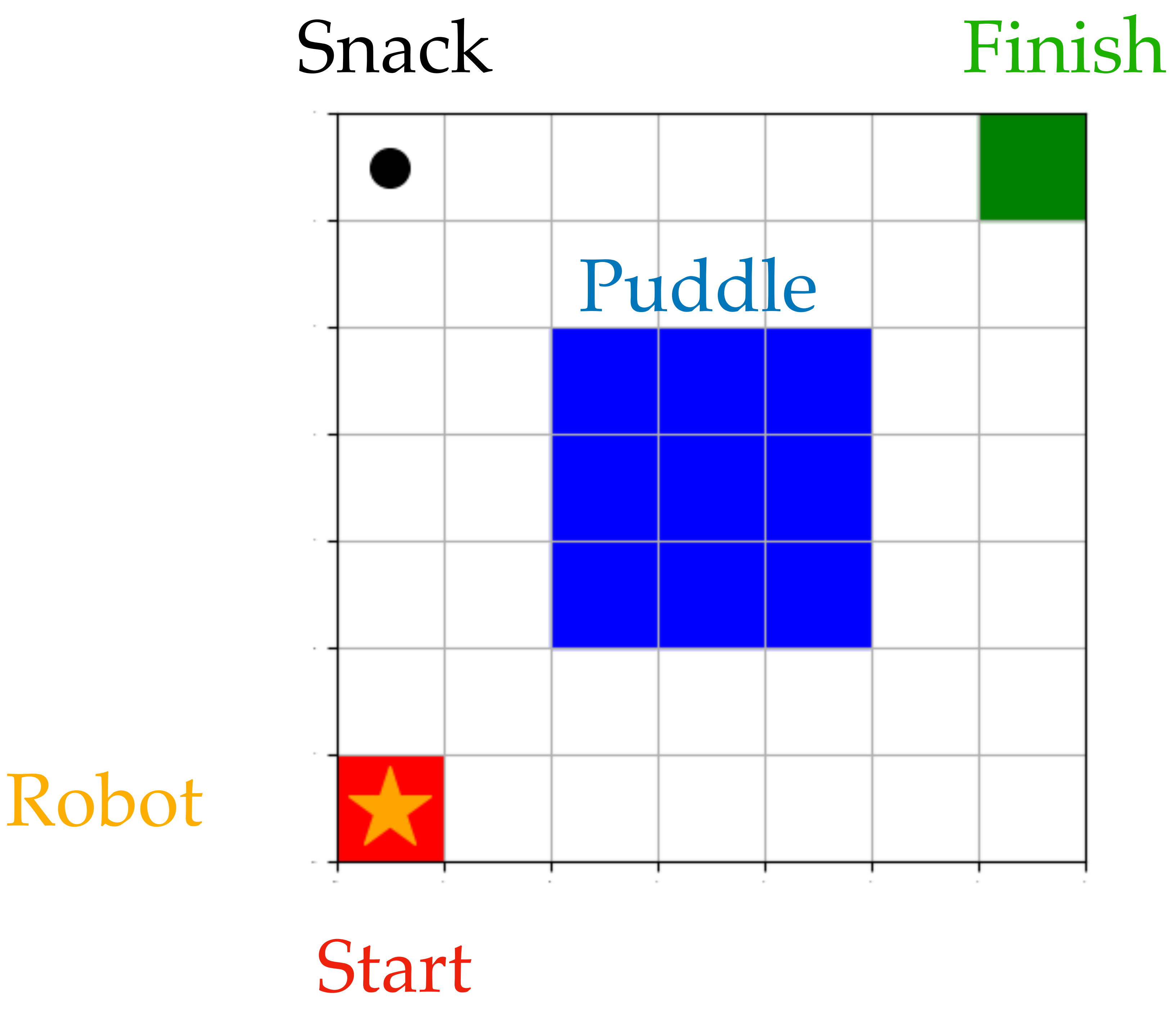
Step 1: Formalizing the Objective

Let's decide what we want



Step 1: Formalizing the Objective

Let's decide what we want



Actions $a \in A$

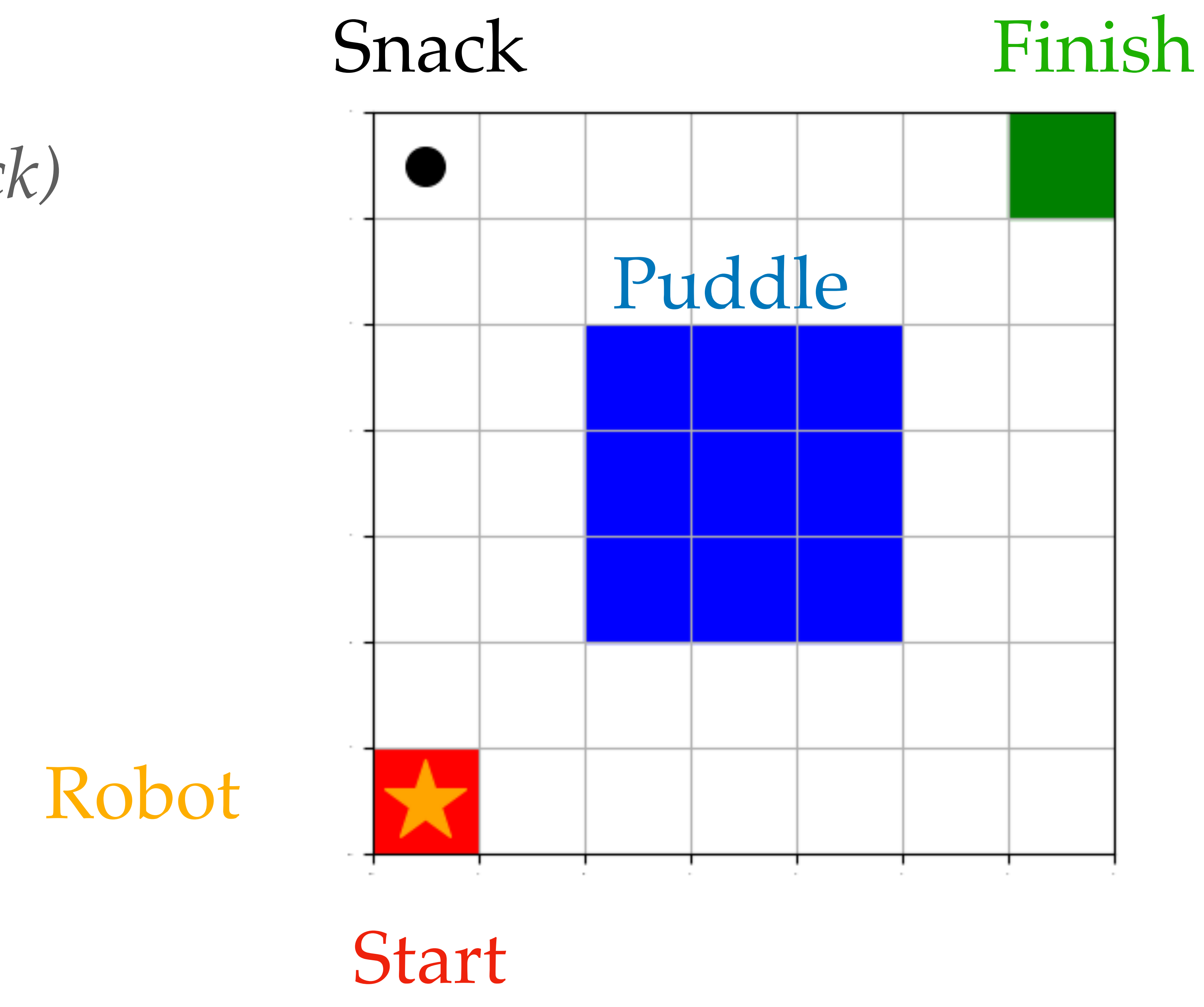
↑ ↓ → ←

Step 1: Formalizing the Objective

Let's decide what we want

Task Objectives I want to teach the robot:

1. Snack: Good *(want to eat a snack)*



Step 1: Formalizing the Objective

Let's decide what we want

Task Objectives I want to teach the robot:

1. Snack: Good *(want to eat a snack)*
2. Puddle: Bad *(want to avoid puddles)*
3. Finish: Good *(want to get to the finish)*
4. Steps: Bad *(want to take as few steps as possible)*

Step 1: Formalizing the Objective

Let's decide what we want

Task Objectives I want to teach the robot:

1. Snack: Good *(want to eat a snack)*
2. Puddle: Bad *(want to avoid puddles)*
3. Finish: Good *(want to get to the finish)*
4. Steps: Bad *(want to take as few steps as possible)*

What matters?

How does it matter?

Reward function

$$R(s) = \underbrace{\theta^T}_{\text{Weights}} \underbrace{\phi(s)}_{\text{Set of selected features}}$$

Step 1: Formalizing the Objective

Let's decide what we want

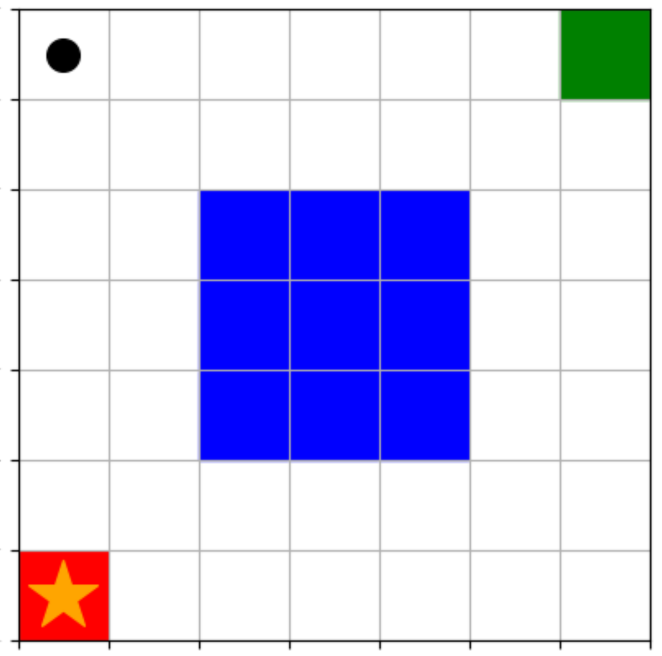
Task Objectives I want to teach the robot:

$$R(s) = \underbrace{\theta^T}_{\text{Weights}} \underbrace{\phi(s)}_{\text{Set of selected features}}$$

1. Snack: Good *(want to eat a snack)*
2. Puddle: Bad *(want to avoid puddles)*
3. Finish: Good *(want to get to the finish)*
4. Steps: Bad *(want to take as few steps as possible)*

What matters?

$\phi(s)$: [# snacks,
distance from puddle,
distance from finish,
timesteps occurred]



Step 1: Formalizing the Objective

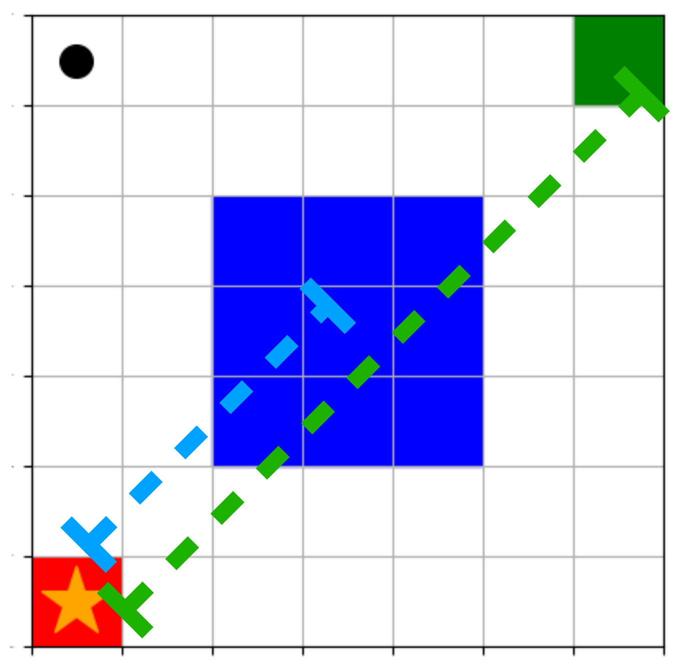
Let's decide what we want

Task Objectives I want to teach the robot:

1. Snack: Good *(want to eat a snack)*
2. Puddle: Bad *(want to avoid puddles)*
3. Finish: Good *(want to get to the finish)*
4. Steps: Bad *(want to take as few steps as possible)*

$$R(s) = \underbrace{\theta^T}_{\text{Weights}} \underbrace{\phi(s)}_{\text{Set of selected features}}$$

$\phi(s)$: [# snacks,
distance from puddle,
distance from finish,
timesteps occurred]



$$\phi(s) = [\ 0\ ,\ \ 3.6\ ,\ 7.8,\ \ \ 1\ \ ]$$

Step 1: Formalizing the Objective

Let's decide what we want

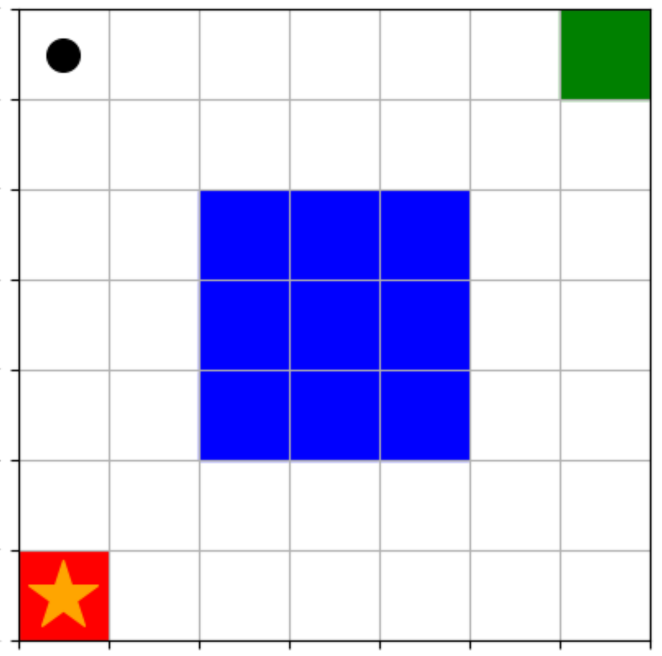
Task Objectives I want to teach the robot:

1. Snack: Good *(want to eat a snack)*
2. Puddle: Bad *(want to avoid puddles)*
3. Finish: Good *(want to get to the finish)*
4. Steps: Bad *(want to take as few steps as possible)*

How does it matter?

$$R(s) = \underbrace{\theta^T}_{\text{Weights}} \underbrace{\phi(s)}_{\text{Set of selected features}}$$

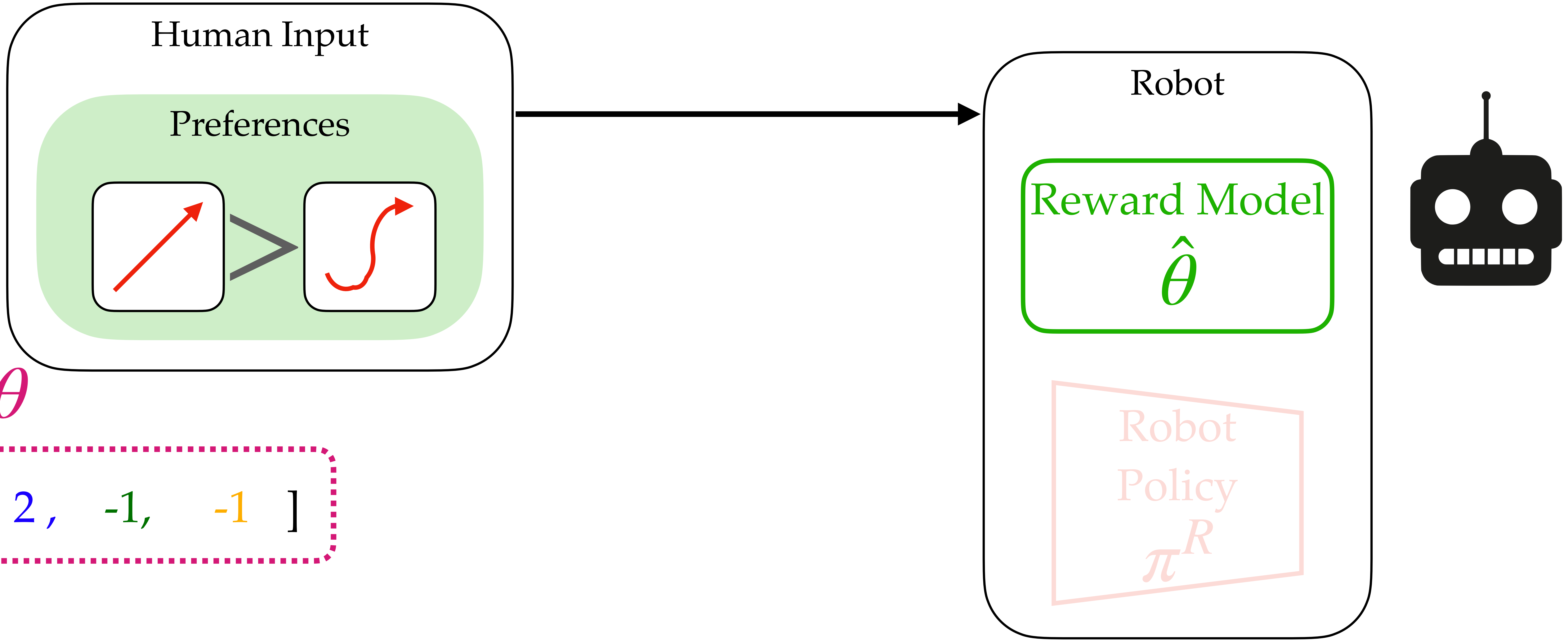
$\phi(s)$: [# snacks, distance from puddle, distance from finish, # timesteps occurred]



$\phi(s)$ = [0 , 3.6 , 7.8, 1]

Weights θ = [5 , 2 , -1, -1]

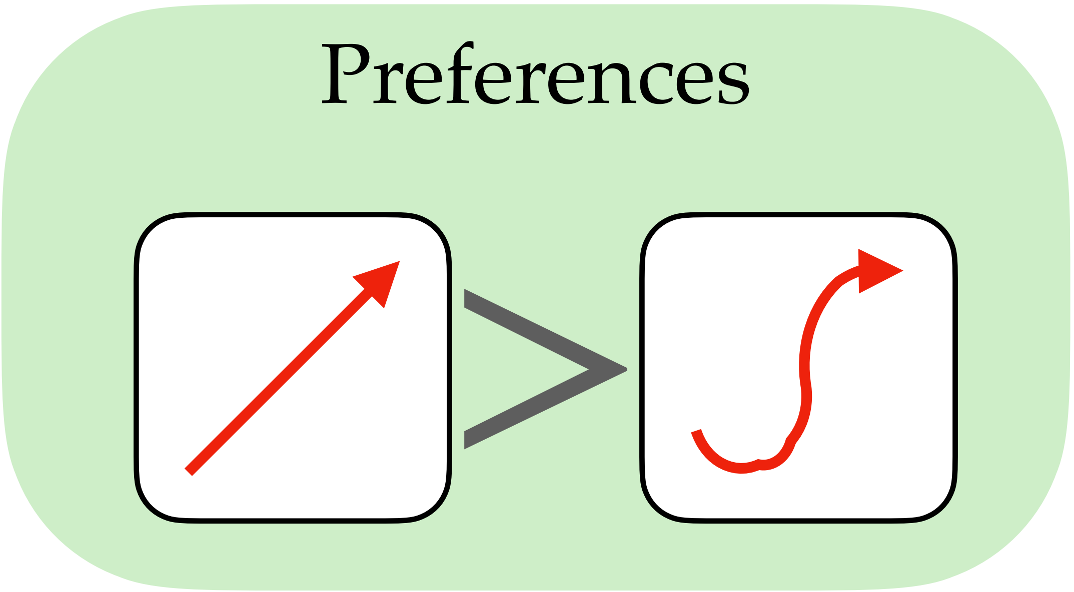
Preference-based learning: Interaction Setup

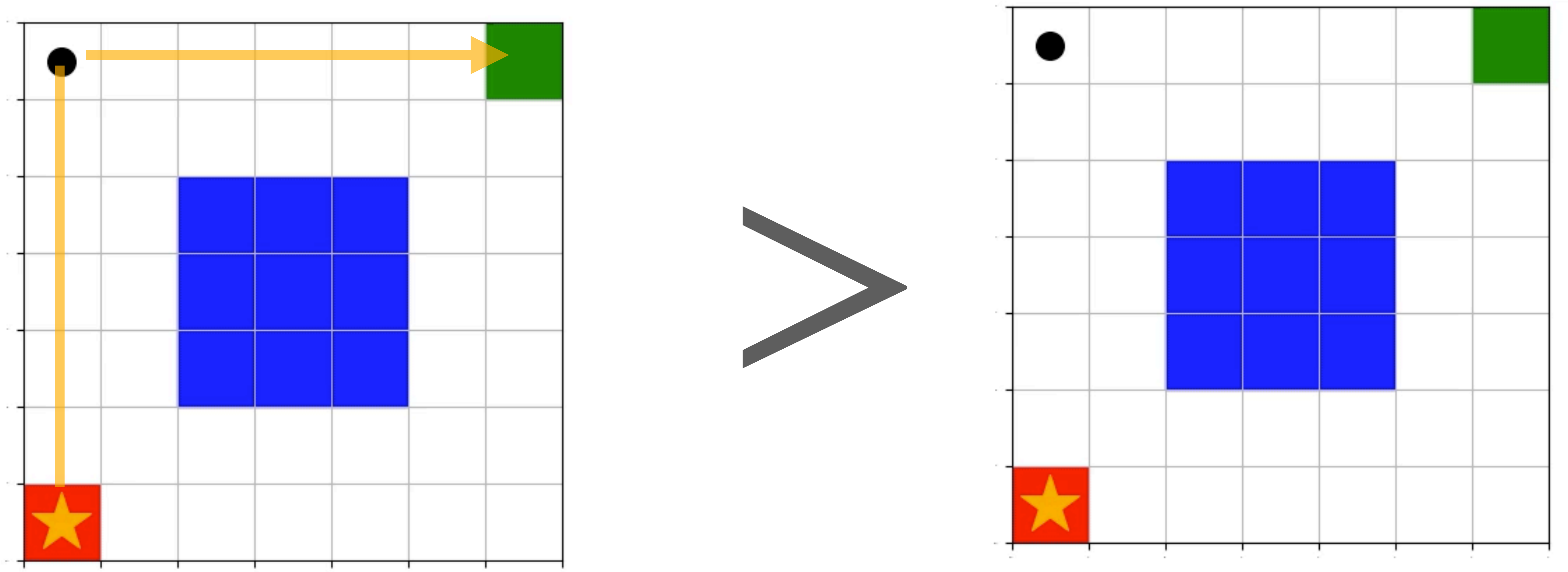


Reward θ

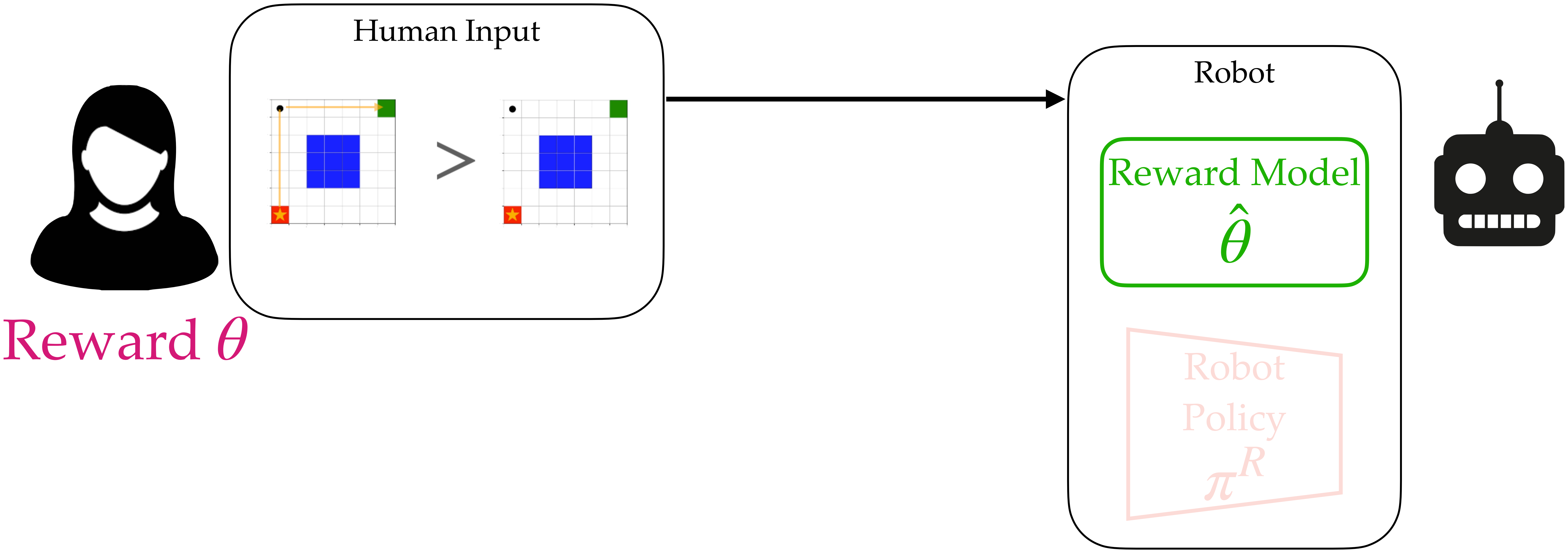
$\theta = [\ 5,\ 2,\ -1,\ -1\ ]$

Let's try giving the robot a preference together!



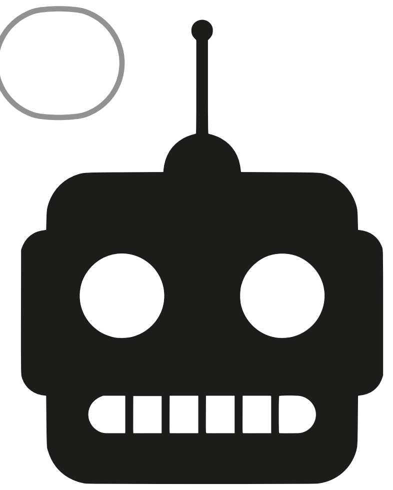


Step 2: What does the robot do with this information?



Step 2: Bayes update to learn from preference

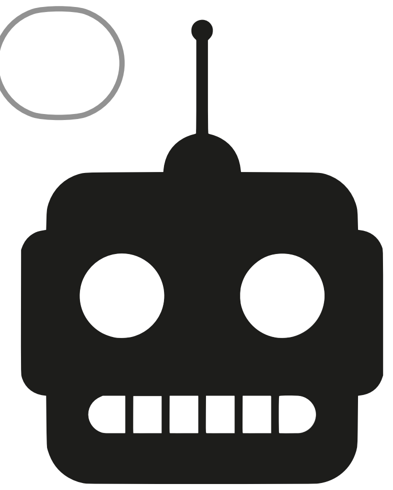
I have no idea what θ might be!
It could be anything in $\mathbb{R}^4$



Step 2: Bayes update to learn from preference

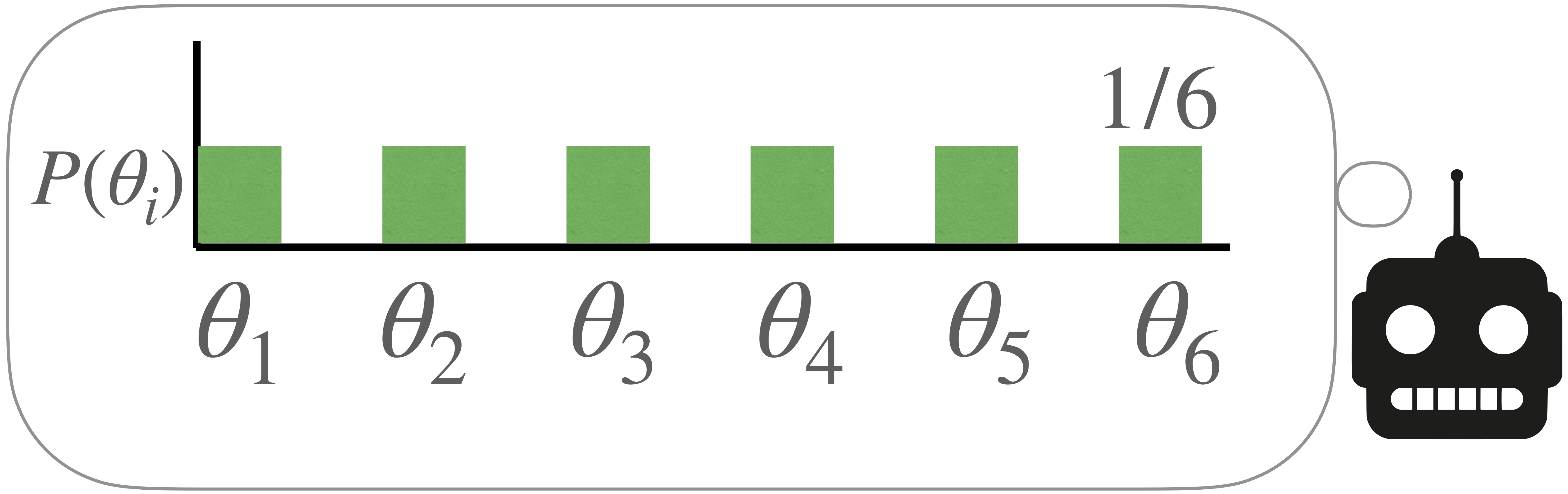
We first initialize a distribution over Θ

$\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6$

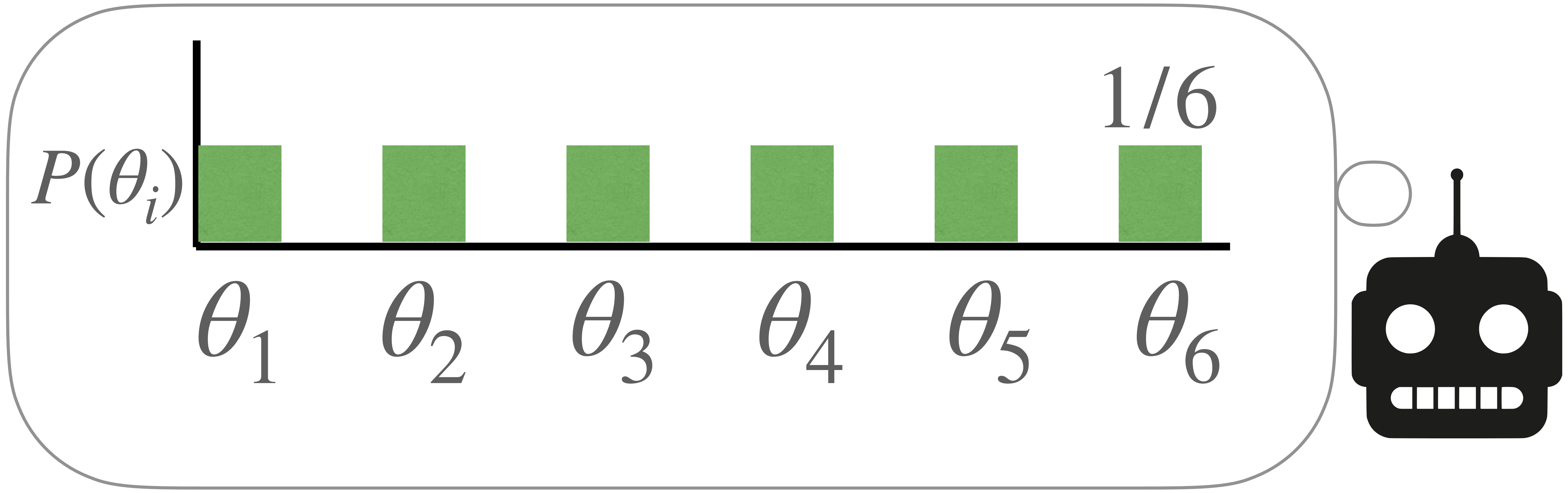


Step 2: Bayes update to learn from preference

We first initialize a distribution over Θ

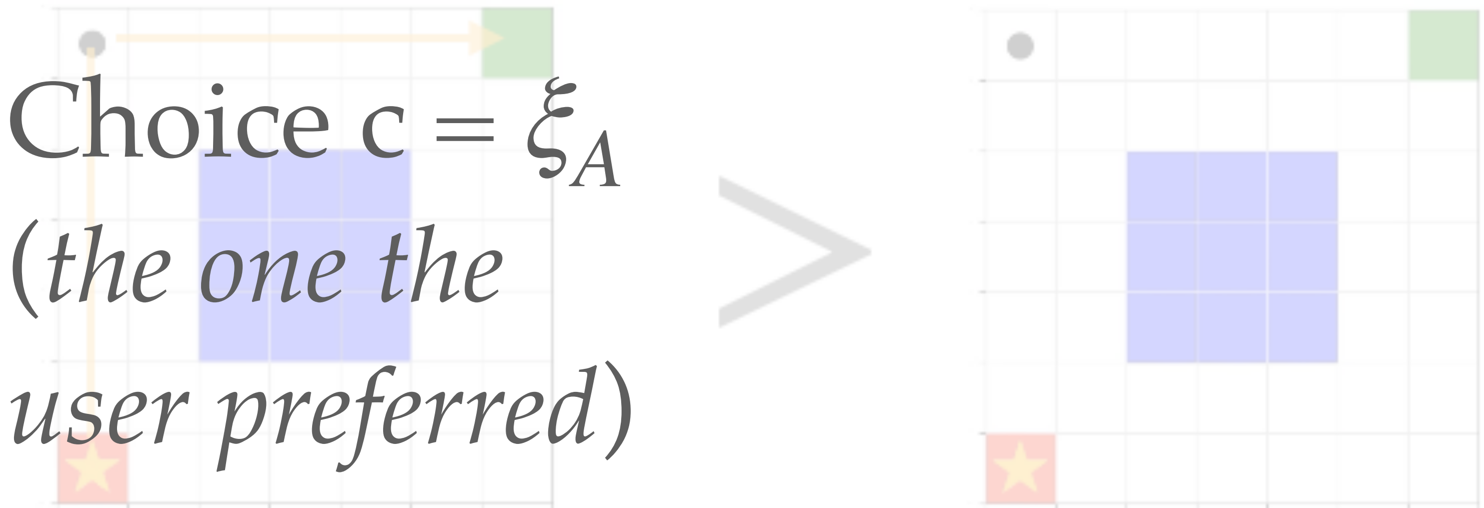


Step 2: Bayes update to learn from preference



We just received data from the user in the form of a preference

Query $Q = \{\xi_A, \xi_B\}$



Choice c = ξ_A
(*the one the user preferred*)

We will use Bayes Rule to obtain a posterior probability

Building up: Bayes Update

$$P(Y)\,P(X\,|\,Y) \quad = \quad P(X,Y)$$

Chain Rule: $P(c,Q)\,P(\theta\,|\,c,Q) = P(c,Q,\theta)$

Building up: Bayes Update

$$P(Y)\,P(X\,|\,Y) \quad = \quad P(X, Y)$$

Chain Rule: $P(c, Q)\,P(\theta\,|\,c, Q) = P(c, Q, \theta)$

$$P(\theta\,|\,c, Q) = \frac{P(c, Q, \theta)}{P(c, Q)}$$

Building up: Bayes Update

$$P(Y)\,P(X\,|\,Y) \quad = \quad P(X, Y)$$

Chain Rule: $P(c, Q)\,P(\theta\,|\,c, Q) = P(c, Q, \theta)$

$$P(\theta\,|\,c, Q) = \frac{\boxed{P(c, Q, \theta)}}{P(c, Q)}$$

$$= \frac{\boxed{P(c\,|\,Q, \theta)\,P(Q, \theta)}}{P(c, Q)}$$

Building up: Bayes Update

$$P(Y)\,P(X \mid Y) \quad = \quad P(X, Y)$$

Chain Rule: $P(c, Q)\,P(\theta \mid c, Q) = P(c, Q, \theta)$

$$P(\theta \mid c, Q) = \frac{P(c, Q, \theta)}{P(c, Q)}$$

$$= \frac{P(c \mid Q, \theta)\,\boxed{P(Q, \theta)}}{\boxed{P(c, Q)}}$$

$$= \frac{P(c \mid Q, \theta)\,\boxed{P(Q)\,P(\theta)}}{\boxed{P(c \mid Q)\,P(Q)}}$$

Building up: Bayes Update

$$P(Y)\,P(X \mid Y) \quad = \quad P(X, Y)$$

Chain Rule: $P(c, Q)\,P(\theta \mid c, Q) = P(c, Q, \theta)$

$$P(\theta \mid c, Q) = \frac{P(c, Q, \theta)}{P(c, Q)}$$

$$= \frac{P(c \mid Q, \theta)\,P(Q, \theta)}{P(c, Q)}$$

$$= \frac{P(c \mid Q, \theta)\,\cancel{P(Q)}\,P(\theta)}{P(c \mid Q)\,\cancel{P(Q)}}$$

$$= \frac{P(c \mid Q, \theta)\;P(\theta)}{P(c \mid Q)}$$

Building up: Bayes Update

Bayes Rule: $$\overbrace{P(\theta \mid c, Q)}^{\text{P(rew | query choice)}} = \frac{\overbrace{P(c \mid Q, \theta)}^{\text{P(choice | query, rew)}} \; \overbrace{P(\theta)}^{\text{P(rew)}}}{\underbrace{P(c \mid Q)}_{\text{P(choice | query)}}}$$

Uniform prior

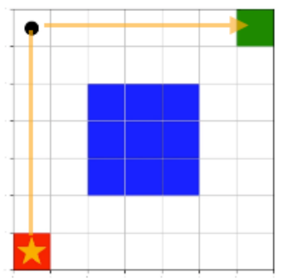
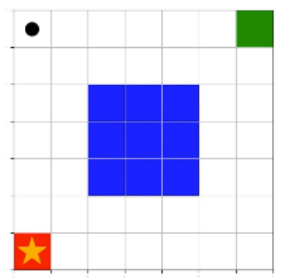
Normalization

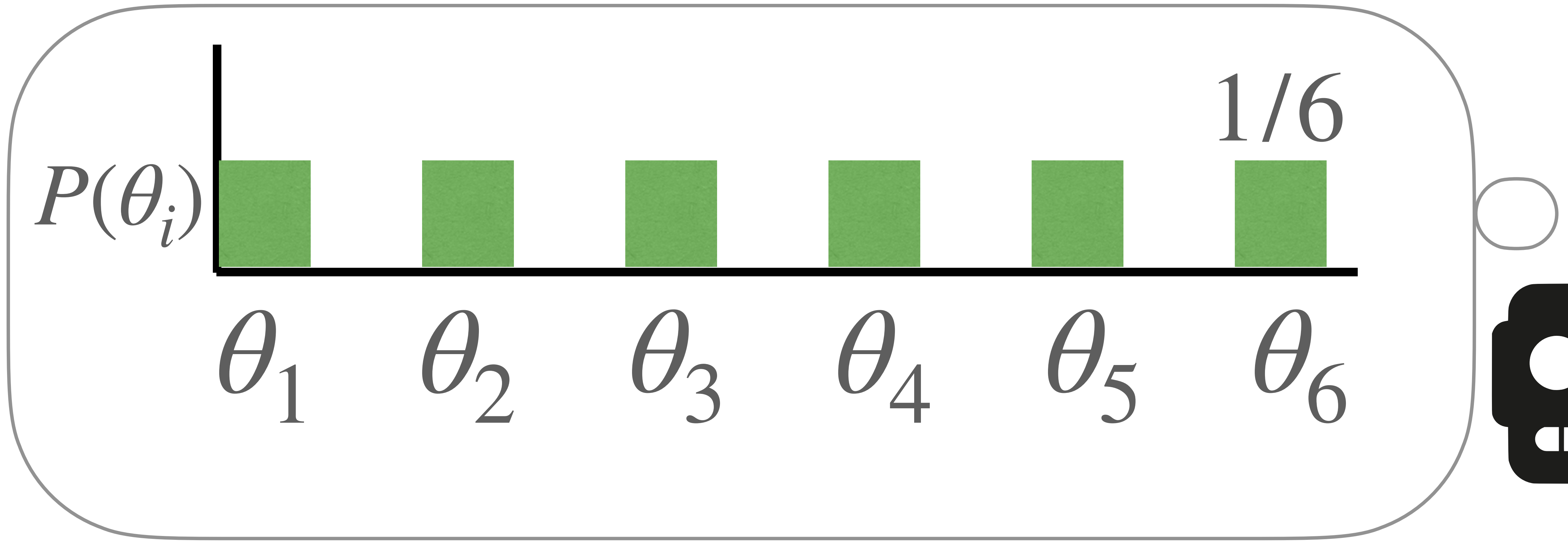
Boltzmann: Likelihood of Human Decision | Model

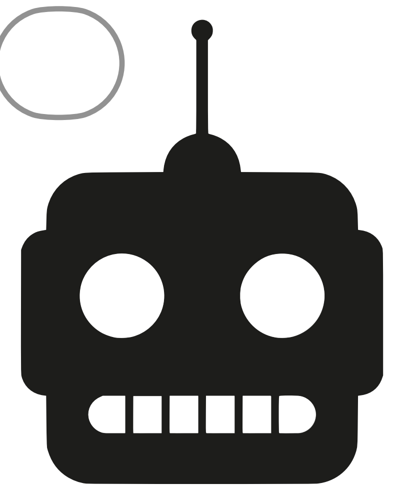
$$\underbrace{P(c \mid Q, \theta)}_{\text{P(choice | query, rew)}} = \frac{e^{R(c)}}{\sum_{q \in Q} e^{R(q)}}$$

Boltzmann Rational Model
(Might also see this as Bradley-Terry model of preferences)

Step 2: Bayes update to learn from preference

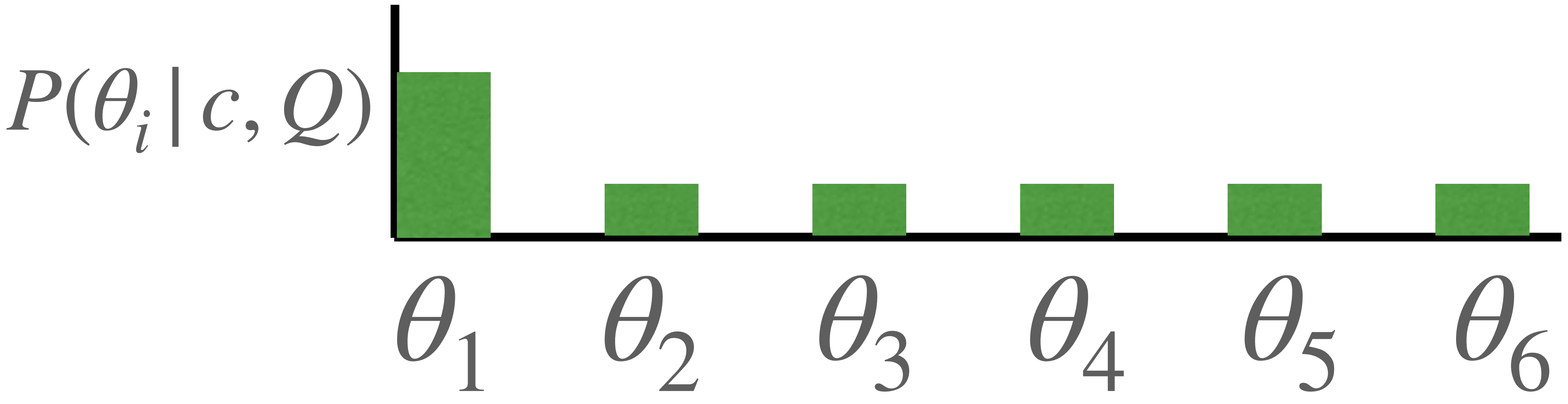
Q, c  $>$ 





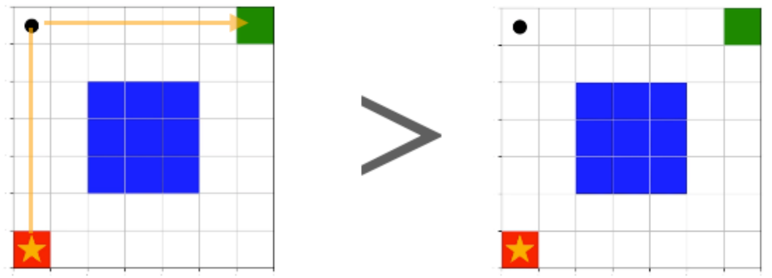
Use Bayes to compute prob. model given data

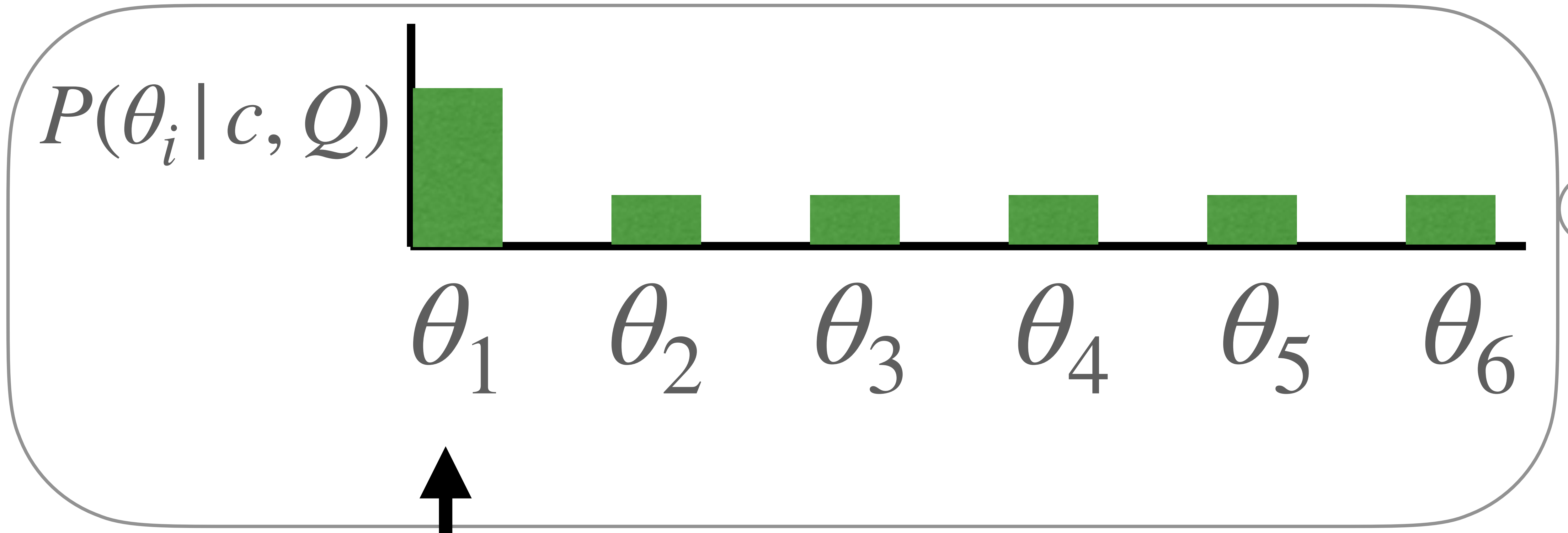
$$P(\theta \mid c, Q) = \frac{P(c \mid Q, \theta) P(\theta)}{P(c \mid Q)}$$

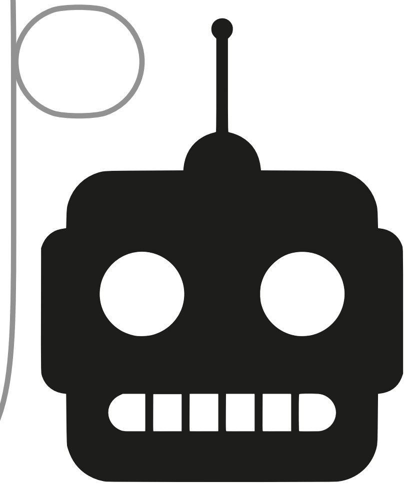


Step 2: Bayes update to learn from preference

Q, c





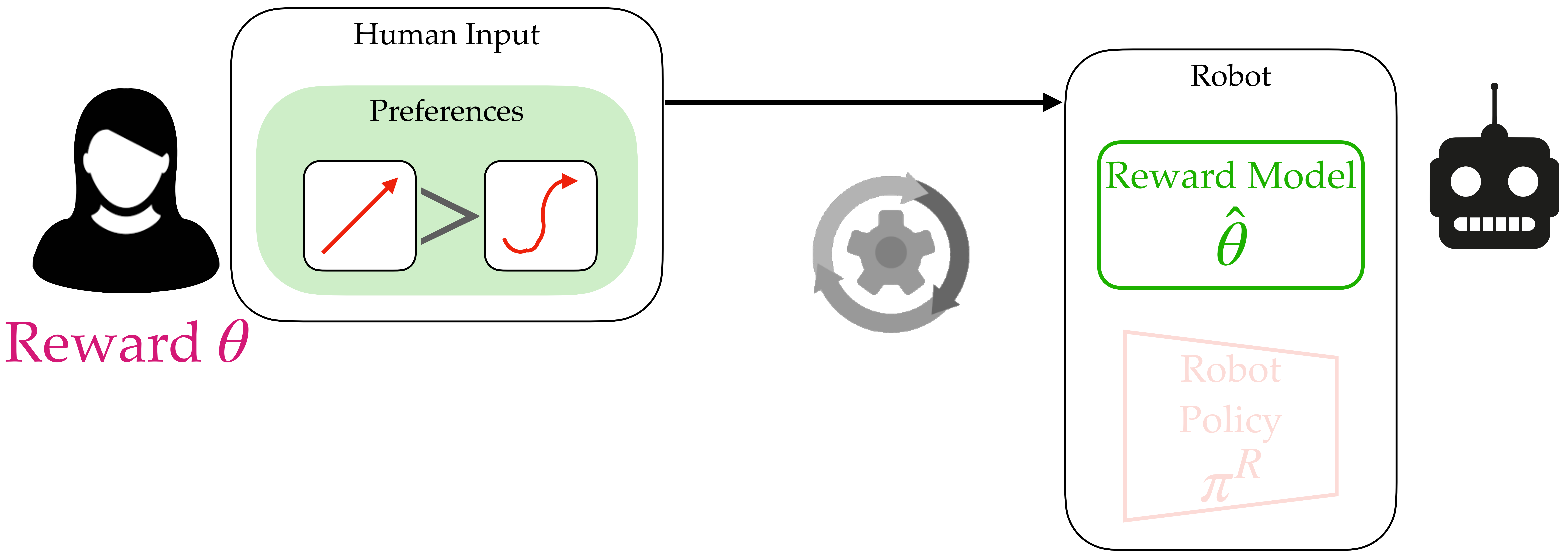


MAP (maximum a posteriori) estimate serves as $\hat{\theta}$

Reward Model

$\hat{\theta}$

Preference-Based Learning of Reward Functions



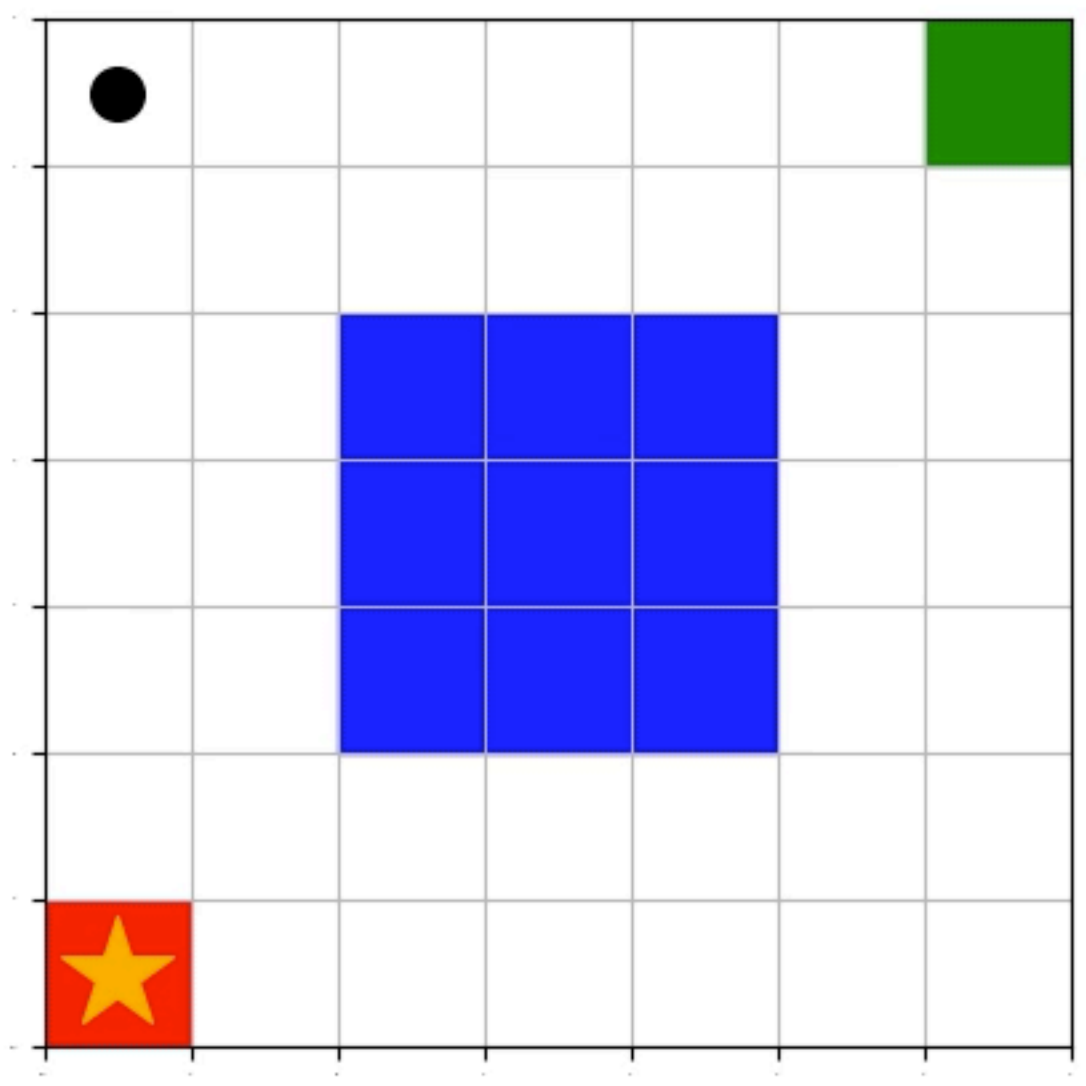
Outline

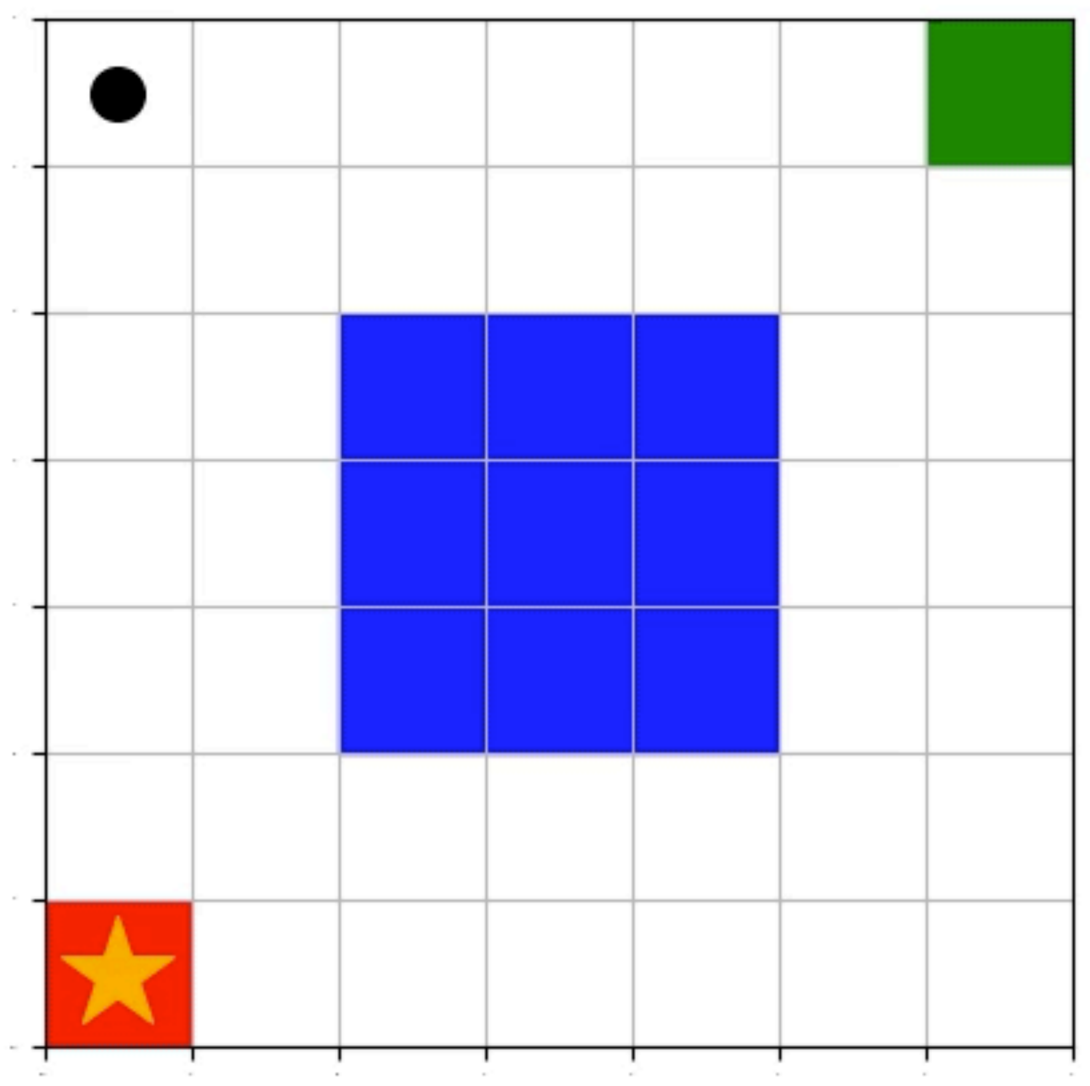
- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference *active*
- Case Study 2: Active learning for black-box policies

Challenges in the [Passive] Learning from Feedback Paradigm

- The agent’s ability to learn relies on good training data.

Let's consider another pair of trajectories





Challenges in the [Passive] Learning from Feedback Paradigm

- The agent's ability to learn relies on good training data.
- The onus to provide the good training data falls completely on the user to know what the robot needs.
- What else?

Challenges in the [Passive] Learning from Feedback Paradigm

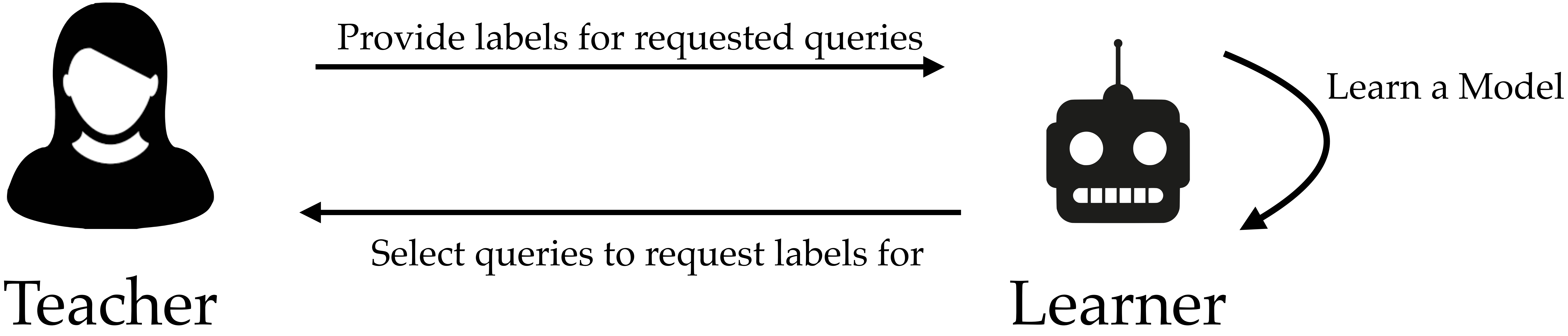
- The agent's ability to learn relies on good training data.
- The onus to provide the good training data falls completely on the user to know what the robot needs.
- What else?
- At scale, it can require fleets of highly trained users.

Toyota Research Institute

Outline

- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference ***active***
- Case Study 2: Active learning for black-box policies

Active Learning



The learner (robot) remains in control and requests annotated data from the human teacher.

The learner can be curious and request information from the teacher based on different query strategies.

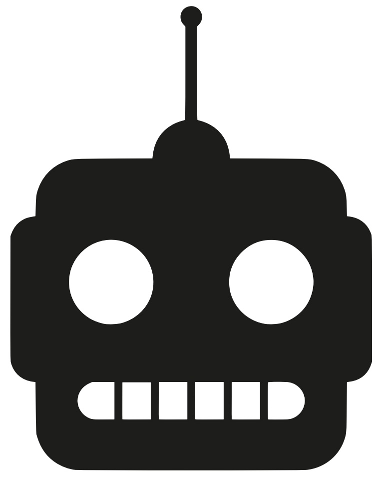
The key decision in active learning: Query Strategy

To design: an active robot learner who asks for help

Teacher

Provide labels for requested queries

Select queries to request labels for



Learner

I know that I need help

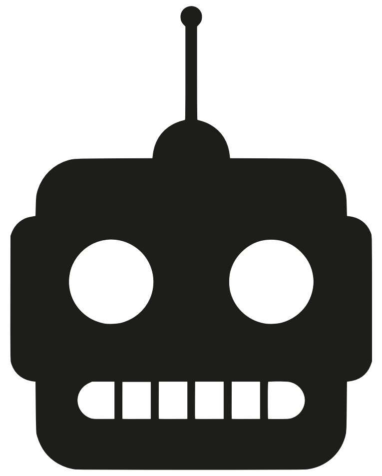
How do I know that I need help?

To design: an active robot learner who asks for help

Teacher

Provide labels for requested queries

Select queries to request labels for



Learner

I know what help to ask for

How do I ask for help?

I know that I need help

How do I know that I need help?

Query Strategy: how do I ask for help?

Uncertainty Minimization (Gaining Information)

Selects unlabeled items whose labels (once received) will reduce the robot's uncertainty over the model.

✓ Volume Removal

✓ Information Gain

Diversity Sampling (Exploration)

Selects unlabeled items that differ from or are unseen in the data the robot has already seen.

✓ Variety of diversity metrics

✓ Different exploration objectives

Random

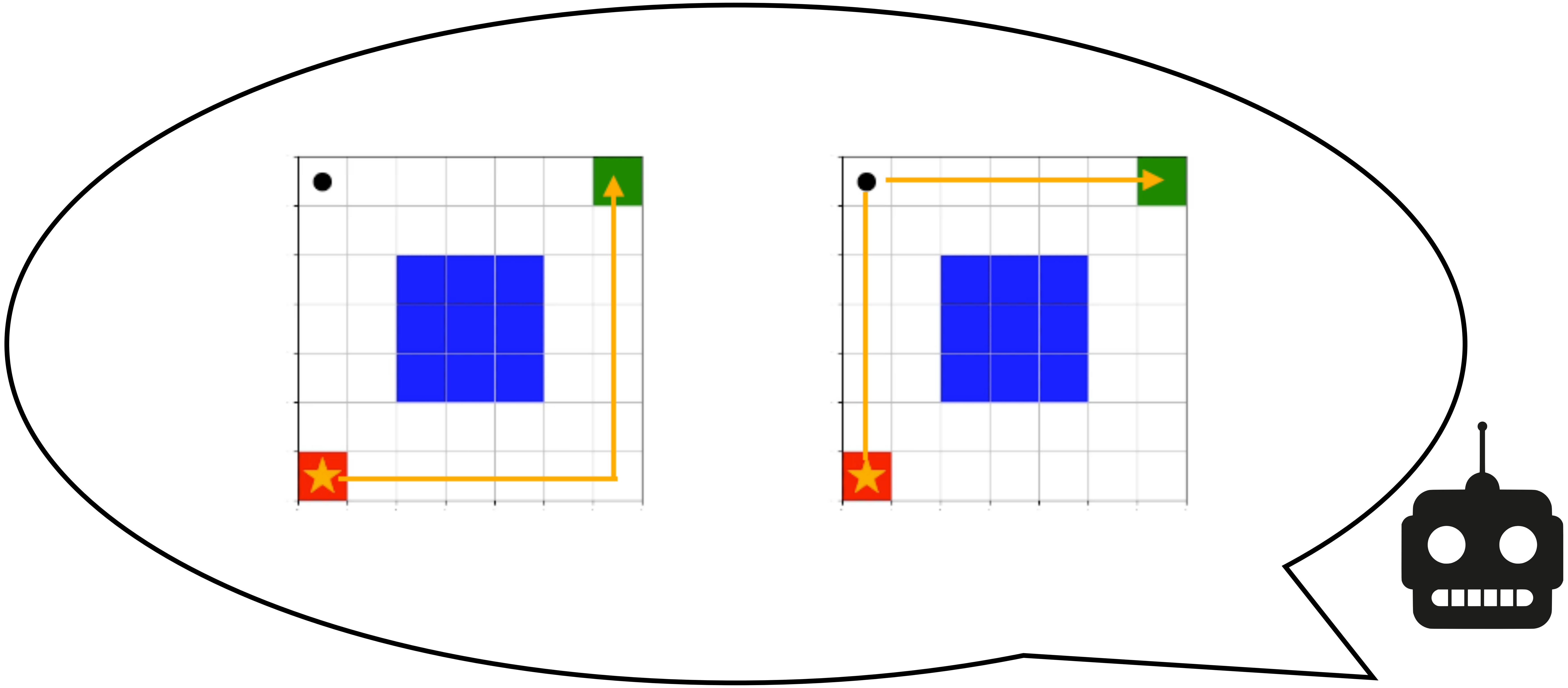
Outline

- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference ***active***
- Case Study 2: Active learning for black-box policies

Outline

- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference ***active***
- Case Study 2: Active learning for black-box policies

Let's take a closer look 👀 at **Active** Preference-Based Learning of Reward Functions



Let’s take a closer look 👀 at
Active Preference-Based Learning of Reward Functions

I know what help to ask for

How do I ask for help?

I know that I need help

How do I know that I need help?

Let's take a closer look 👀 at
Active Preference-Based Learning of Reward Functions

I know what help to ask for

How do I ask for help?

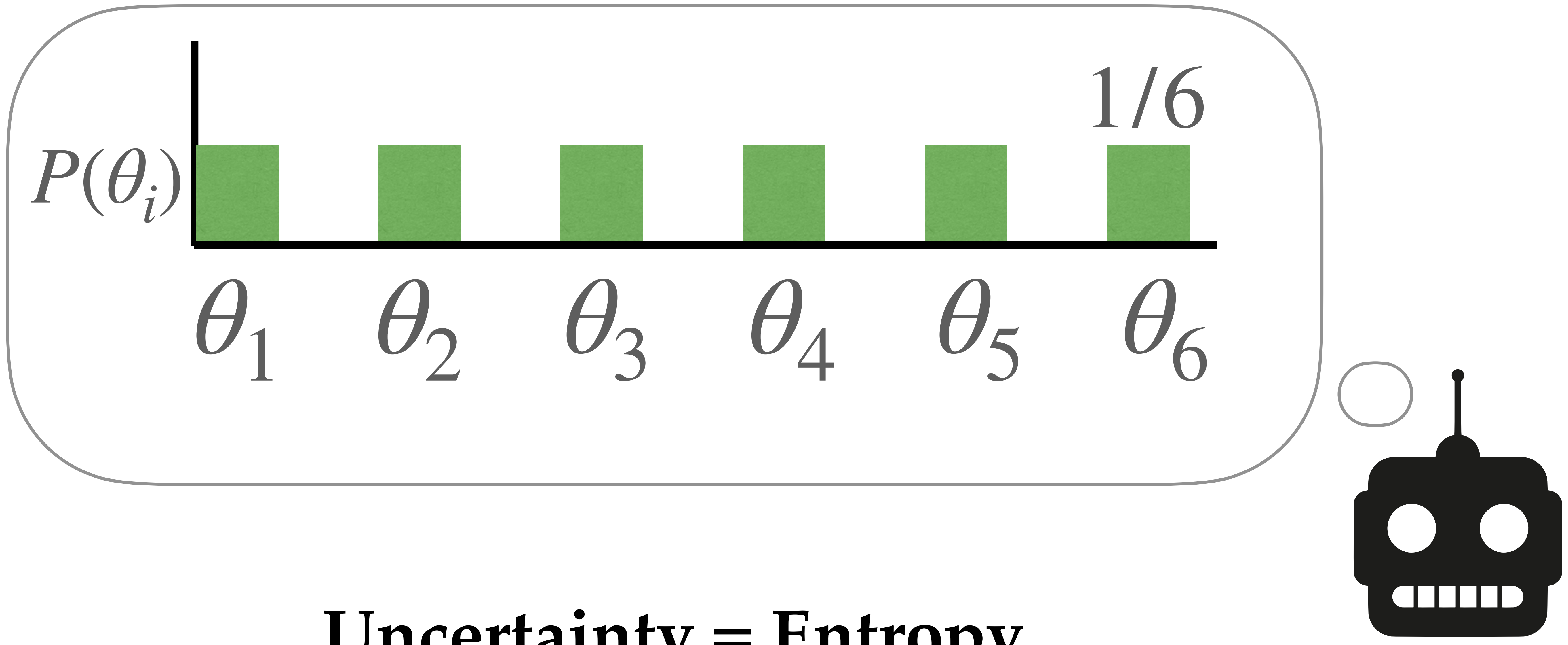
I know that I need help

How do I know that I need help?

Pick query that maximally reduces uncertainty

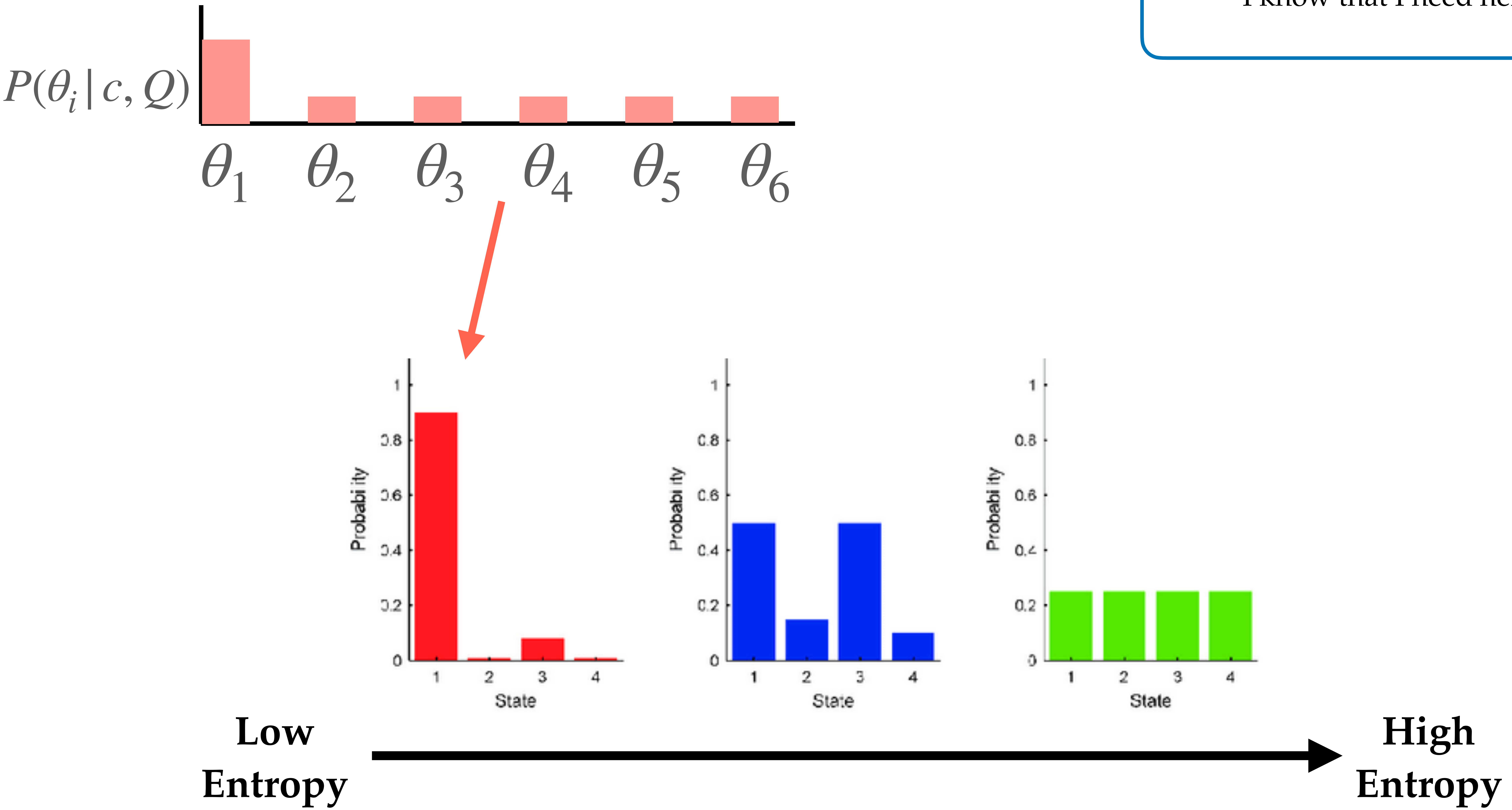
High uncertainty

I know that I need help



Uncertainty = Entropy

I know that I need help



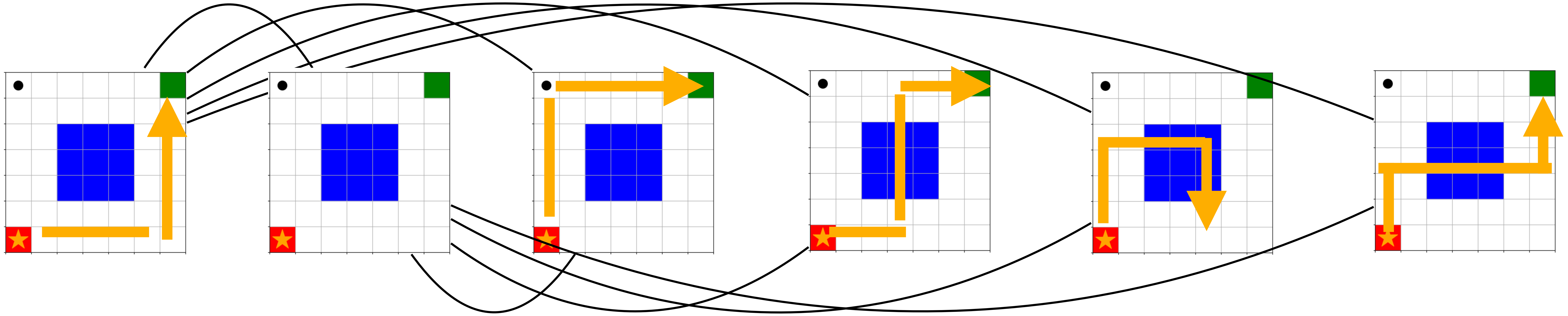
I know what help to ask for

The robot wants to find the query that will reduce the its uncertainty the most

$$\underset{Q \in \text{Possible Queries}}{\arg\max} \ \text{Uncertainty Reduction}\,(Q)$$

The robot wants to find the query that will reduce the its uncertainty the most

$$\underset{Q \in \text{Possible Queries}}{\arg\max} \ \text{Uncertainty Reduction}(Q)$$



The robot wants to find the query that will reduce the its uncertainty the most

$$\underset{Q \in \text{Possible Queries}}{\arg\max} \; \underbrace{\overbrace{H(\theta)}^{\text{Uncertainty prior to query}} - \overbrace{\mathbb{E}_c[H(\theta \mid c, Q)]}^{\text{Uncertainty after human response}}}_{\text{Uncertainty Reduction } (Q)}$$

The robot wants to find the query that will reduce the its uncertainty the most

$$\underset{Q \in \text{Possible Queries}}{\arg\max} \; \underbrace{\overbrace{H(\theta)}^{\text{Uncertainty prior to query}} - \overbrace{\mathbb{E}_c[H(\theta \mid c, Q)]}^{\text{Uncertainty after human response}}}_{\text{Uncertainty Reduction }(Q)}$$

Sadigh, Dorsa, et al. *Active preference-based learning of reward functions*. 2017.
Biyik, Erdem, and Dorsa Sadigh. "Batch active preference-based learning of reward functions." *Conference on robot learning*. PMLR, 2018.
Bıyık, Erdem, et al. "Asking easy questions: A user-friendly approach to active reward learning." *arXiv preprint arXiv:1910.04365* (2019).

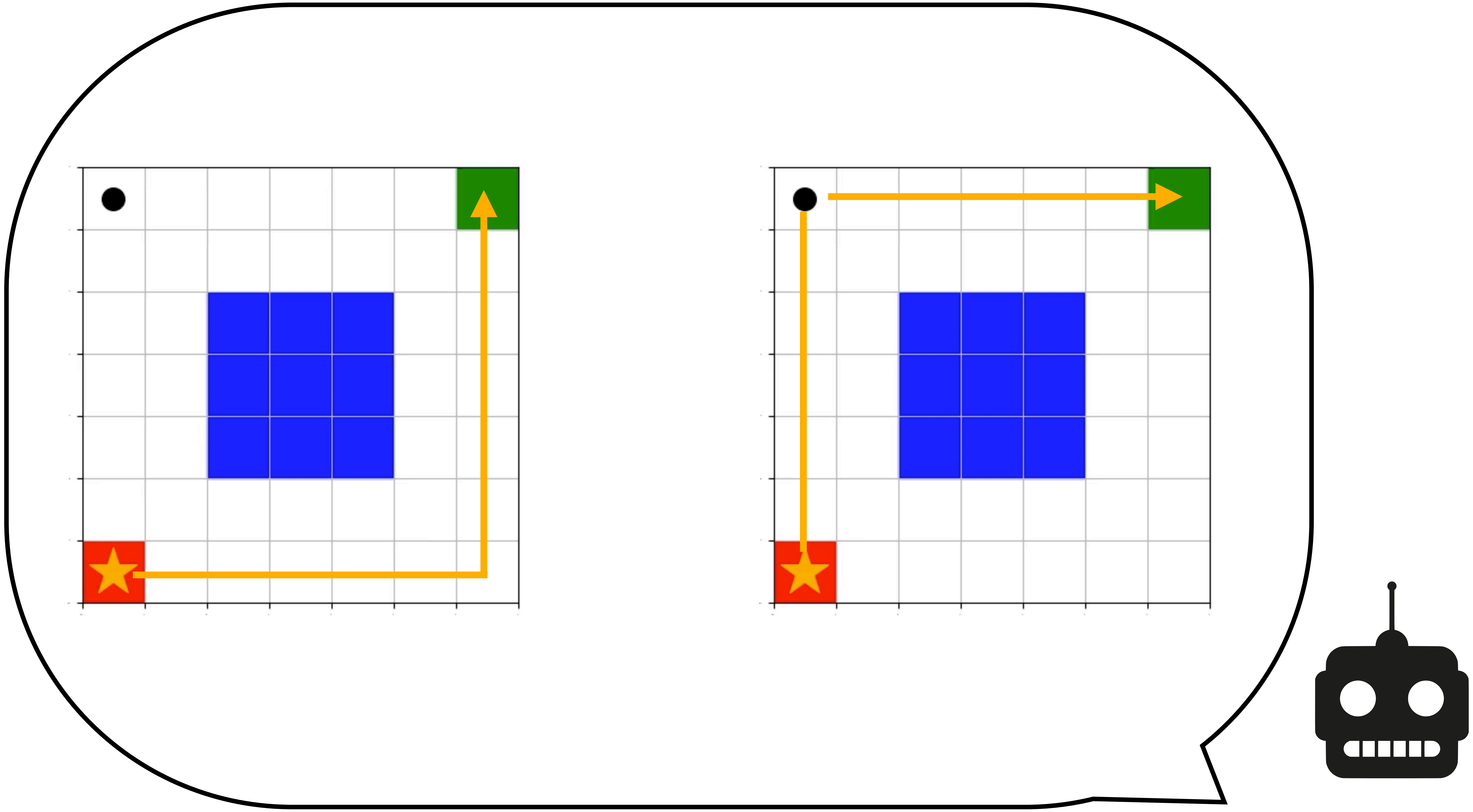
Choosing the query that will reduce the its uncertainty the most maximizes information gain

$$\underset{Q \in \text{Possible Queries}}{\arg\max} \quad \text{Information Gain}(Q)$$

When we optimize for information gain, we simultaneously produce queries that seek to be easy for the human to answer.

Bıyık, Erdem, et al. "Asking easy questions: A user-friendly approach to active reward learning." *arXiv preprint arXiv:1910.04365* (2019).

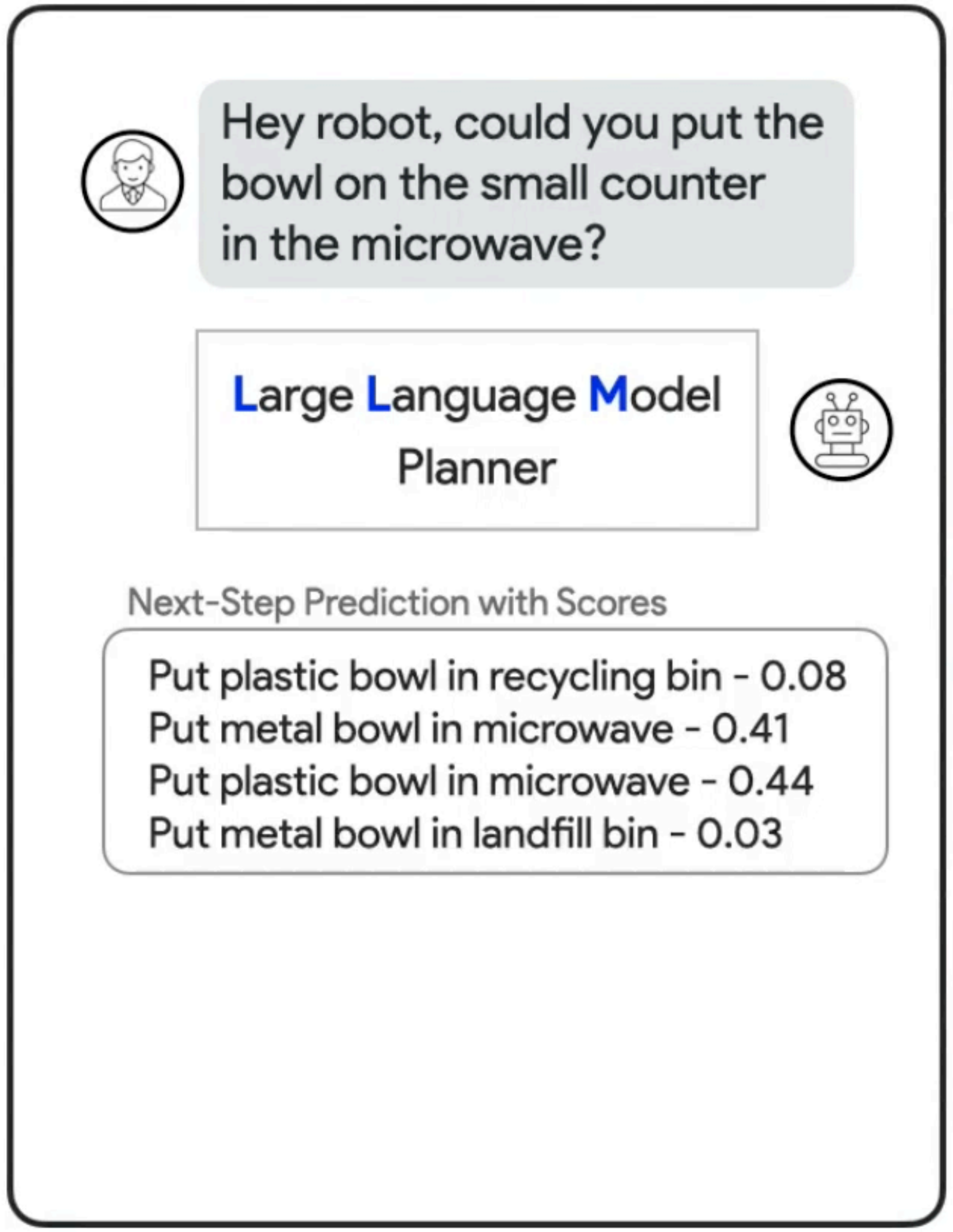
Applying information gain, We have for our most informative query:



These preference learning techniques are key in training models like ChatGPT

ChatGPT

More robots that ask for help



Ren, Allen Z., et al. "Robots that ask for help: Uncertainty alignment for large language model planners." *CoRL* (2023).

- Robots capable of self-assessments ability and a priori competency predictions can help improve overall team performance and trust.

Bridgwater, Tom, et al. "Examining profiles for robotic risk assessment: Does a robot's approach to risk affect user trust?." Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction. 2020.

Outline

- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference ***active***
- Case Study 2: Active learning for black-box policies

Summary thus far:

- Learning from human feedback seeks to align robot behaviors to human intentions
    - Human feedback can occur through multiple types (i.e. preferences)

Summary thus far:

- Learning from human feedback seeks to align robot behaviors to human intentions
    - Human feedback can occur through multiple types (i.e. preferences)
- Active learning shifts the interaction paradigm by allowing the learner to *actively* request feedback (labels) from the human.
    - Effective learning occurs by coming up with a good strategy for querying the human (*strategically asking for help*)

Summary thus far and Questions?

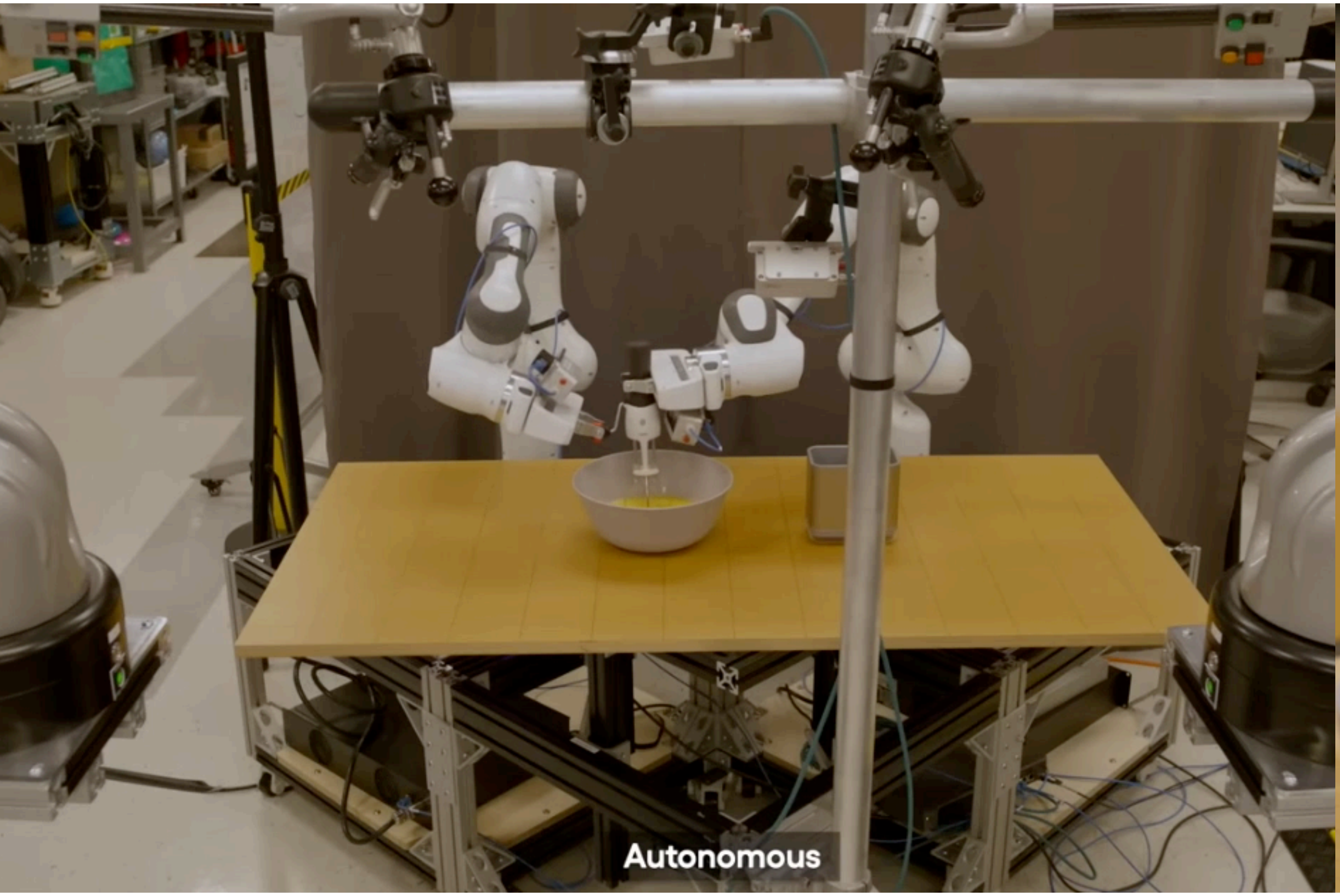
- Learning from human feedback seeks to align robot behaviors to human intentions
    - Human feedback can occur through multiple types (i.e. preferences)
- Active learning shifts the interaction paradigm by allowing the learner to *actively* request feedback (labels) from the human.
    - Effective learning occurs by coming up with a good strategy for querying the human (*strategically asking for help*)
- To enable an effective active robot learner, 2 necessary design decisions:

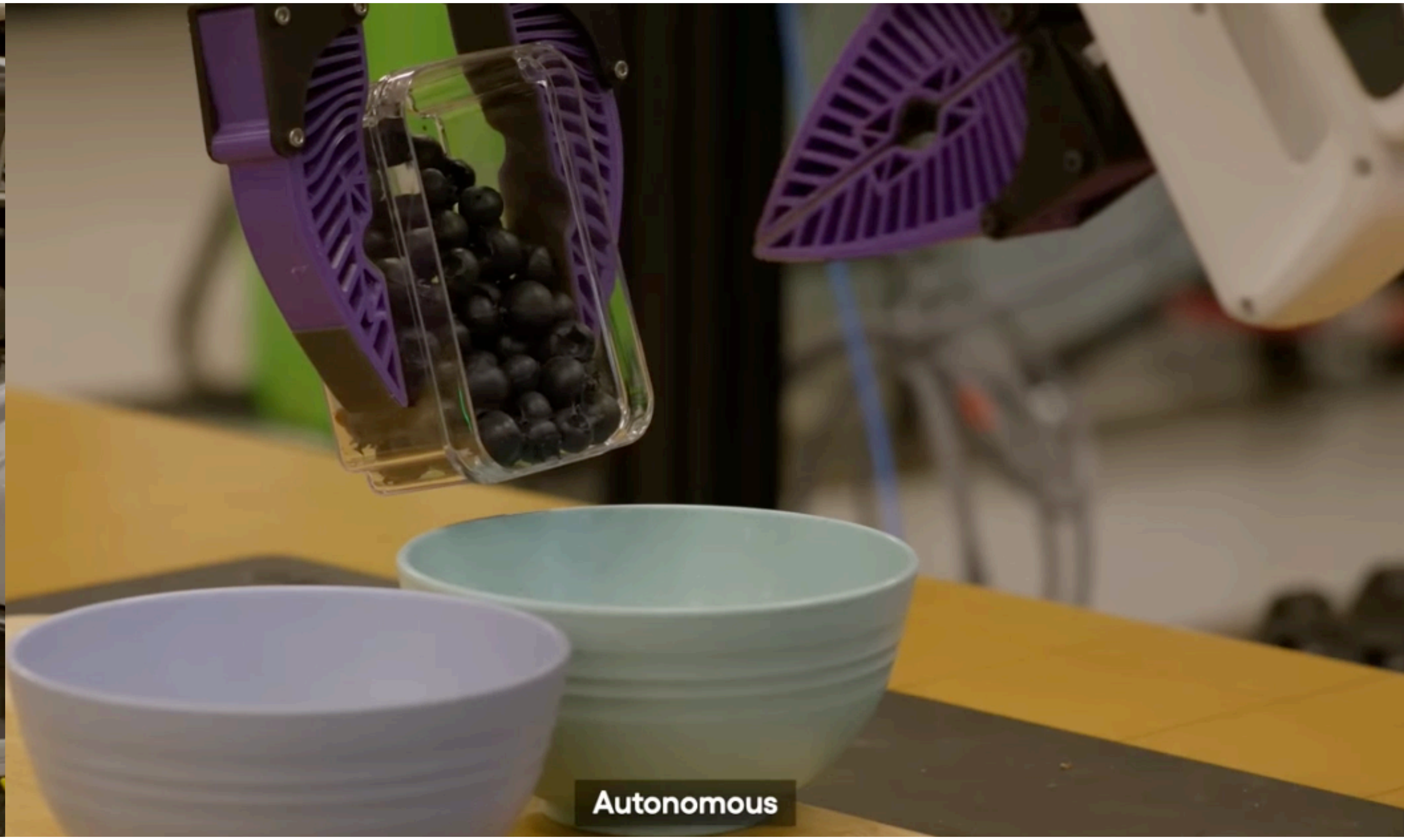
I know what help to ask for	I know that I need help
How do I ask for help?	*How do I know that I need help?*

Outline

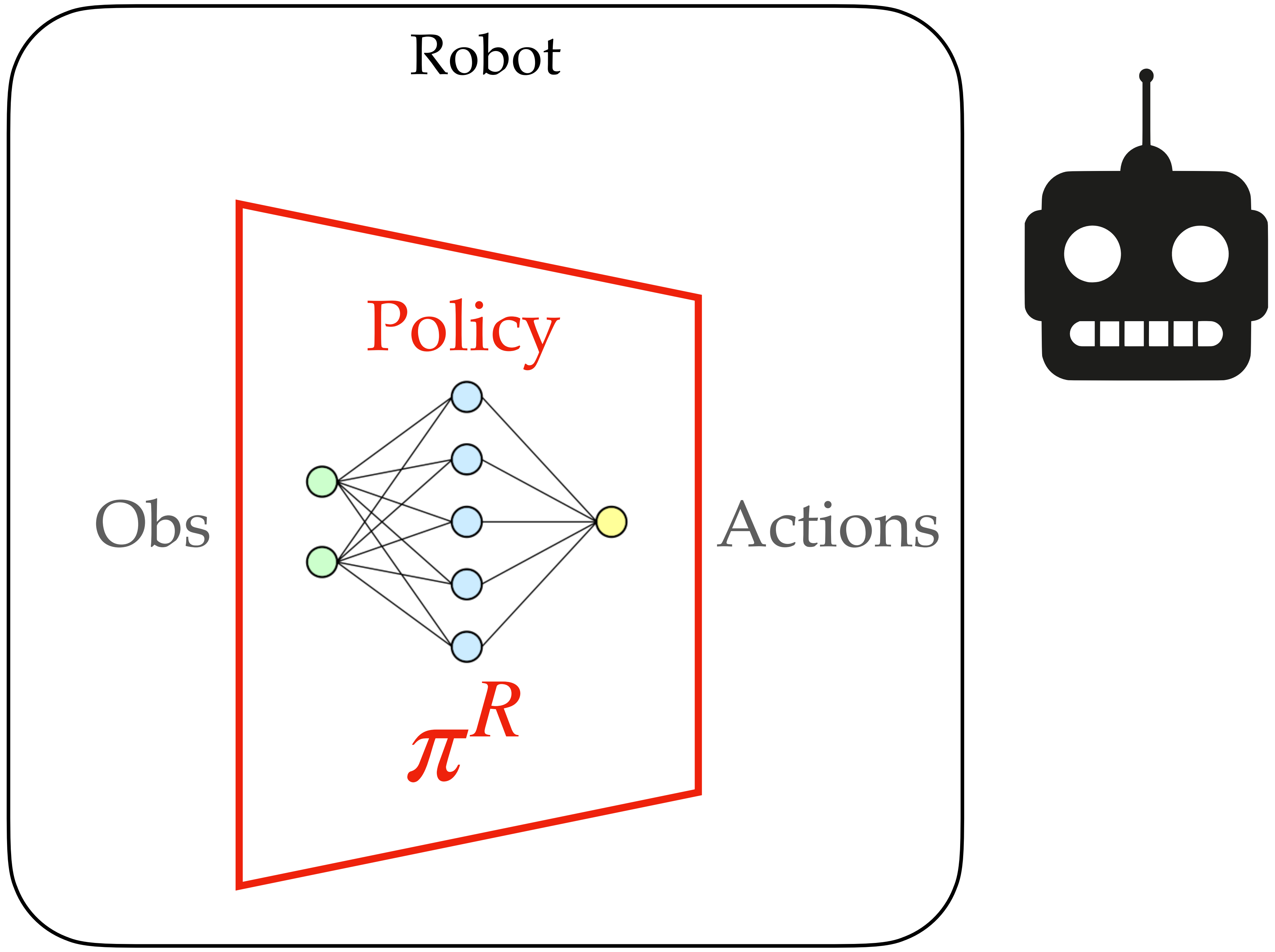
- Alignment problem
- Alignment process: Learning from human feedback
- Case Study 1: Learning from preferences
- Active Learning: Why and How?
- Revisiting Case Study 1: Making learning from preference *active*
- Case Study 2: Active learning for black-box policies

What about black-box policies?

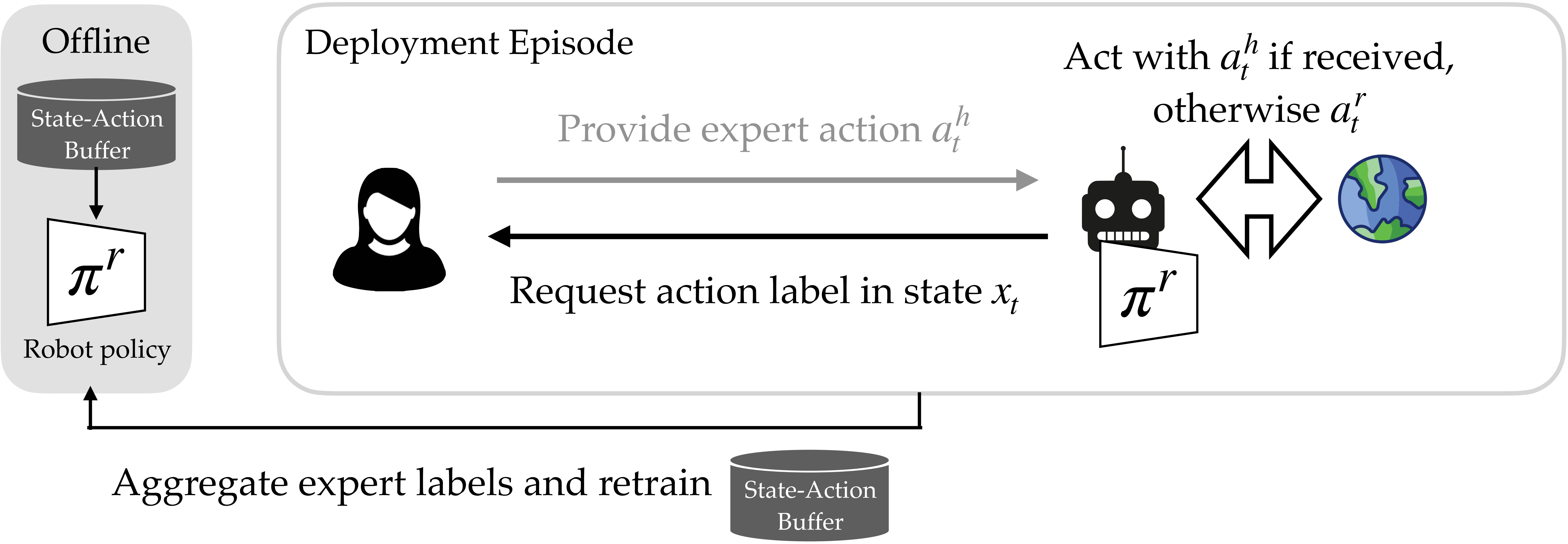


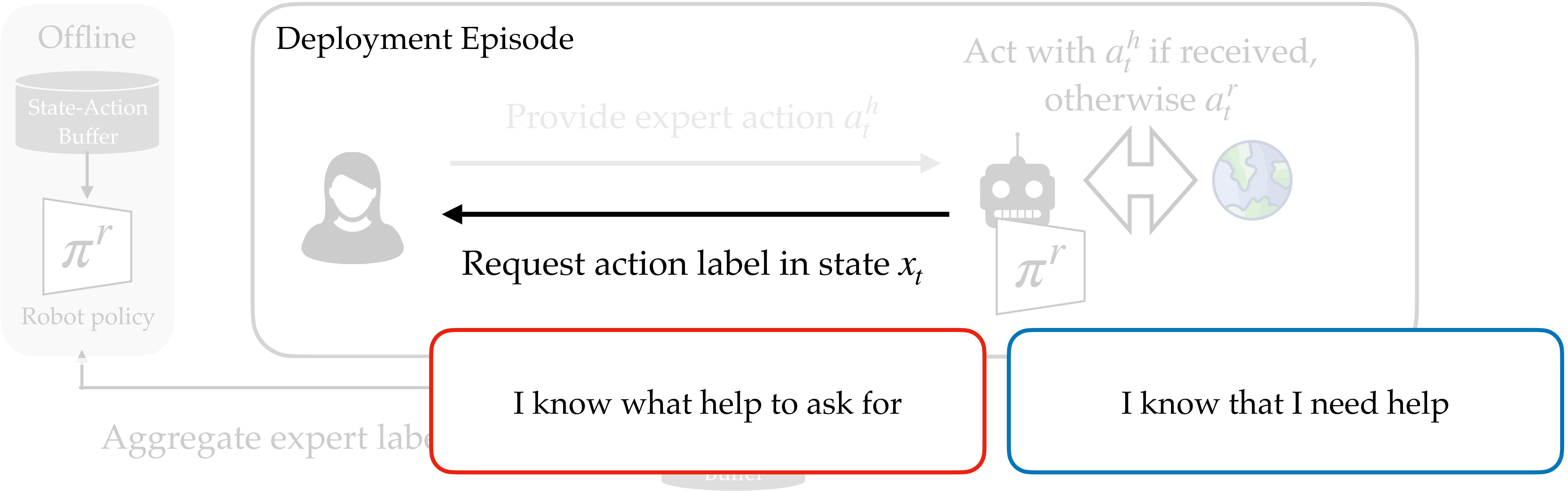


We benefited from our robot representing the reward distribution



Online Interactive Imitation Learning





How do I ask for help?

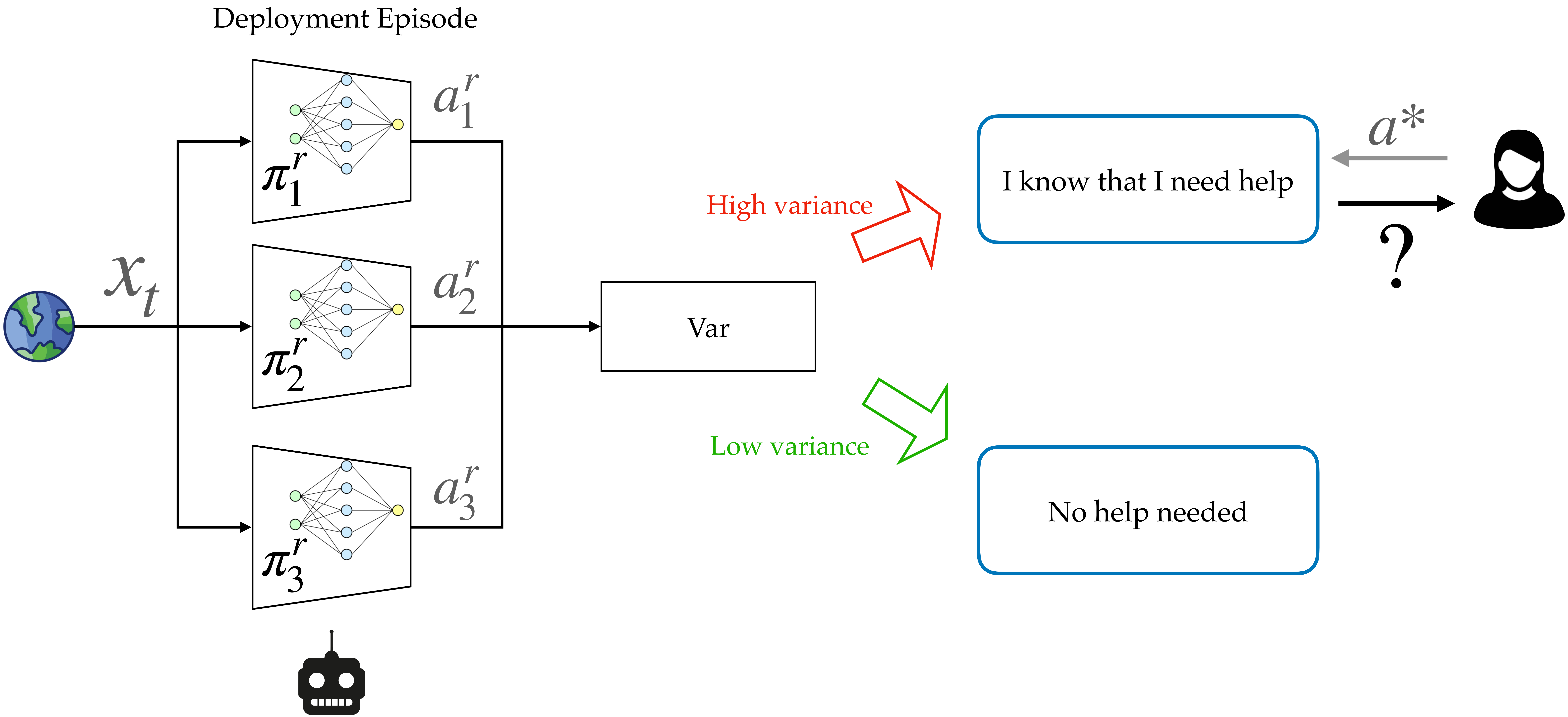
How do I know that I need help?

Expert Teleoperation

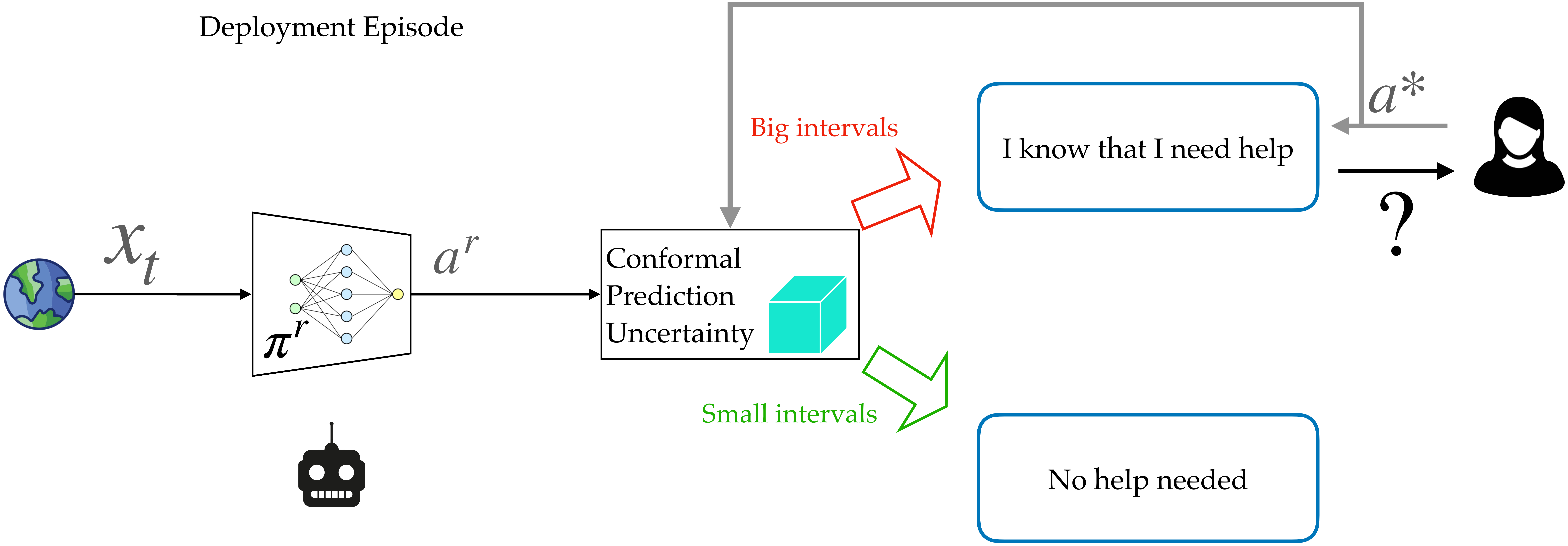
✓ Ensembles

✓ Conformal prediction

Uncertainty Quantification: Ensemble Disagreement



Uncertainty Quantification: Conformal Prediction

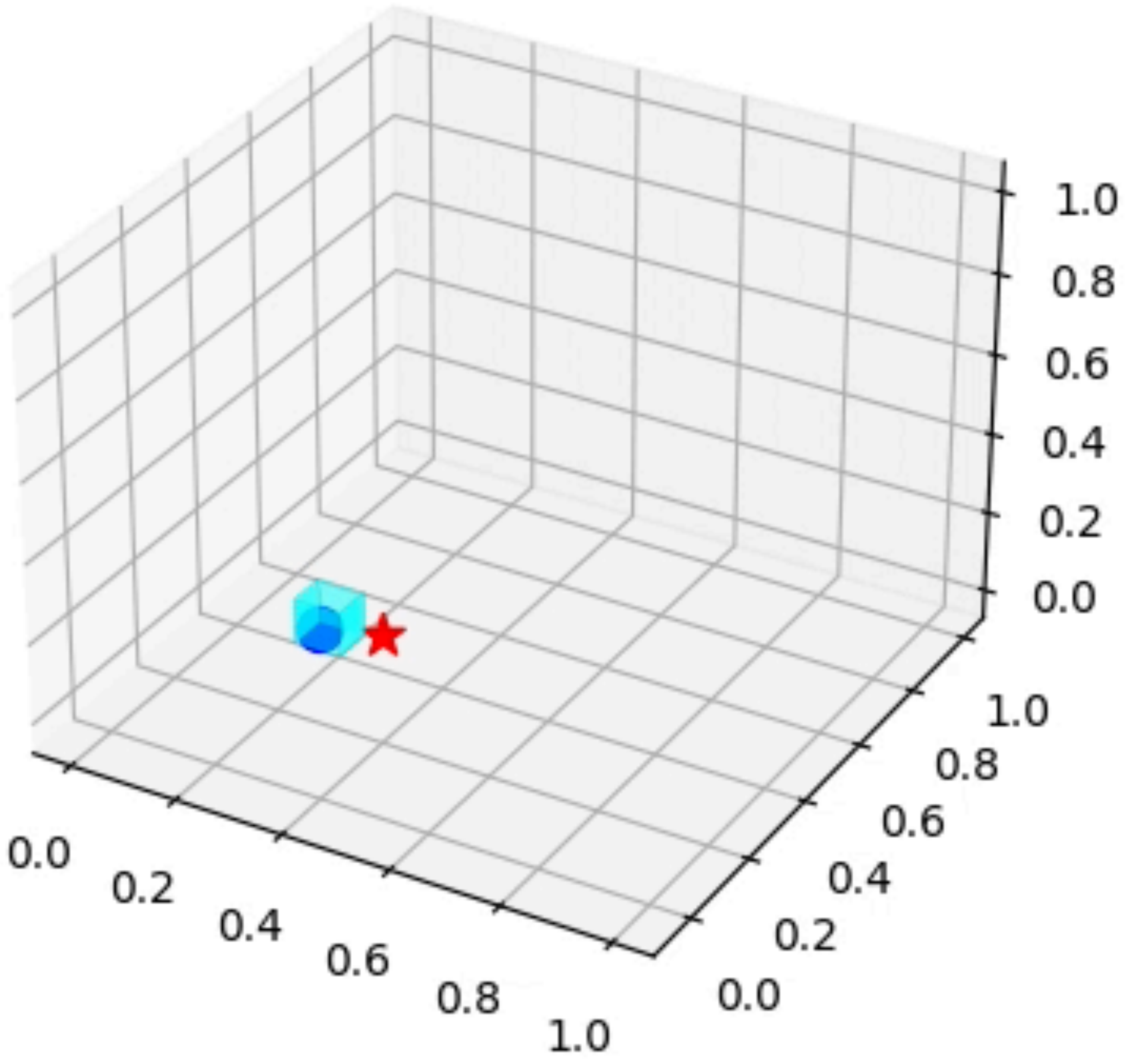


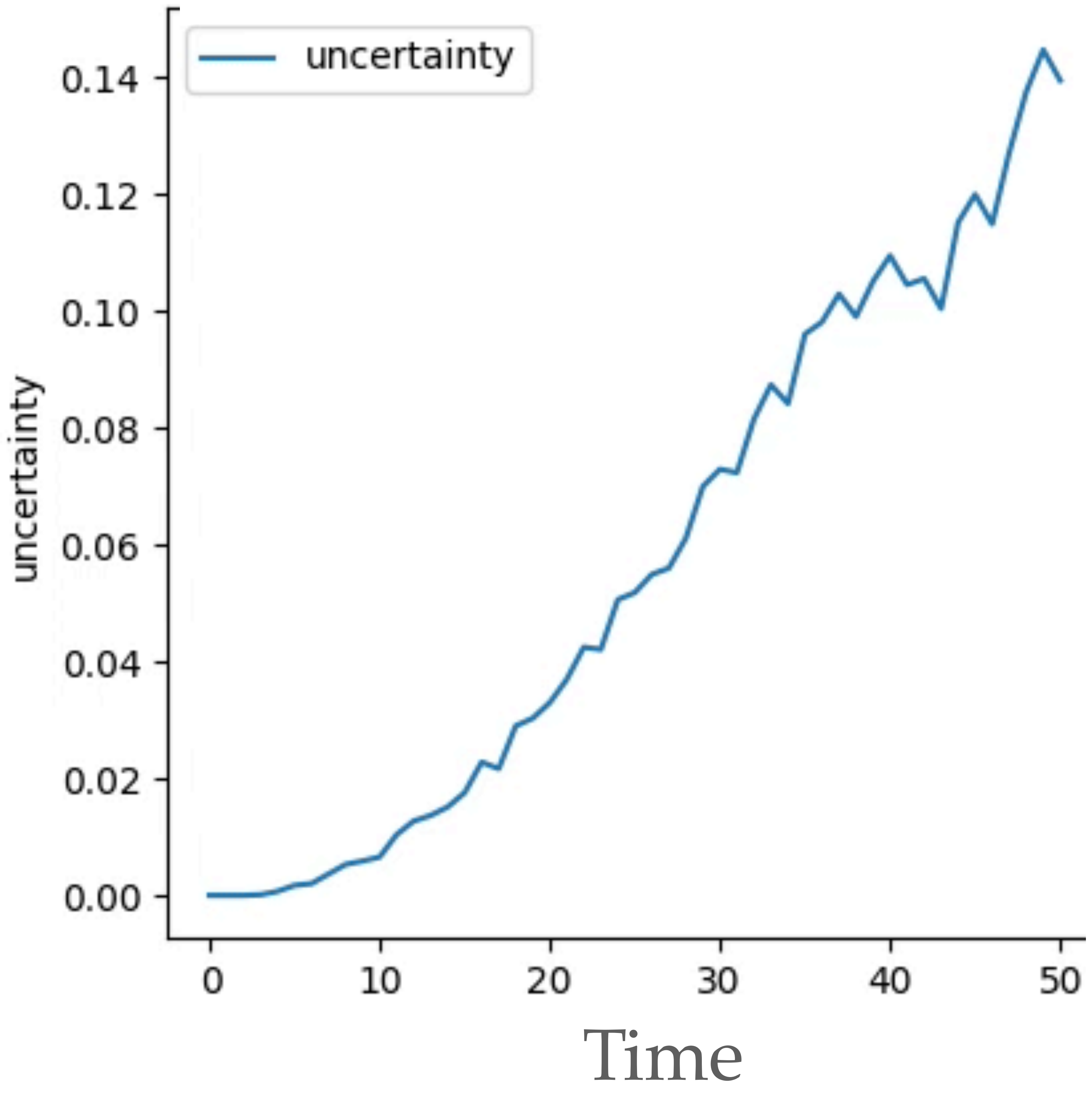
Online Conformal: Whenever we observe the human's ground truth action, use the prediction error to adaptively adjust uncertainty estimate.

Uncertainty Quantification: Online Conformal Prediction

Model predictions

Ground truth (human feedback)





Prediction interval = uncertainty

Summary and Questions?

- Learning from human feedback seeks to align robot behaviors to human intentions
    - Human feedback can occur through multiple types (i.e. preferences)
- Active learning shifts the interaction paradigm by allowing the learner to *actively* request feedback (labels) from the human.
    - Effective learning occurs by coming up with a good strategy for querying the human (*strategically asking for help*)
- To enable an effective active robot learner, 2 necessary design decisions:

✓ Preferences

✓ Expert control

I know what help to ask for

How do I ask for help?

I know that I need help

How do I know that I need help?

✓ Information gain

✓ Ensembles

✓ Conformal prediction