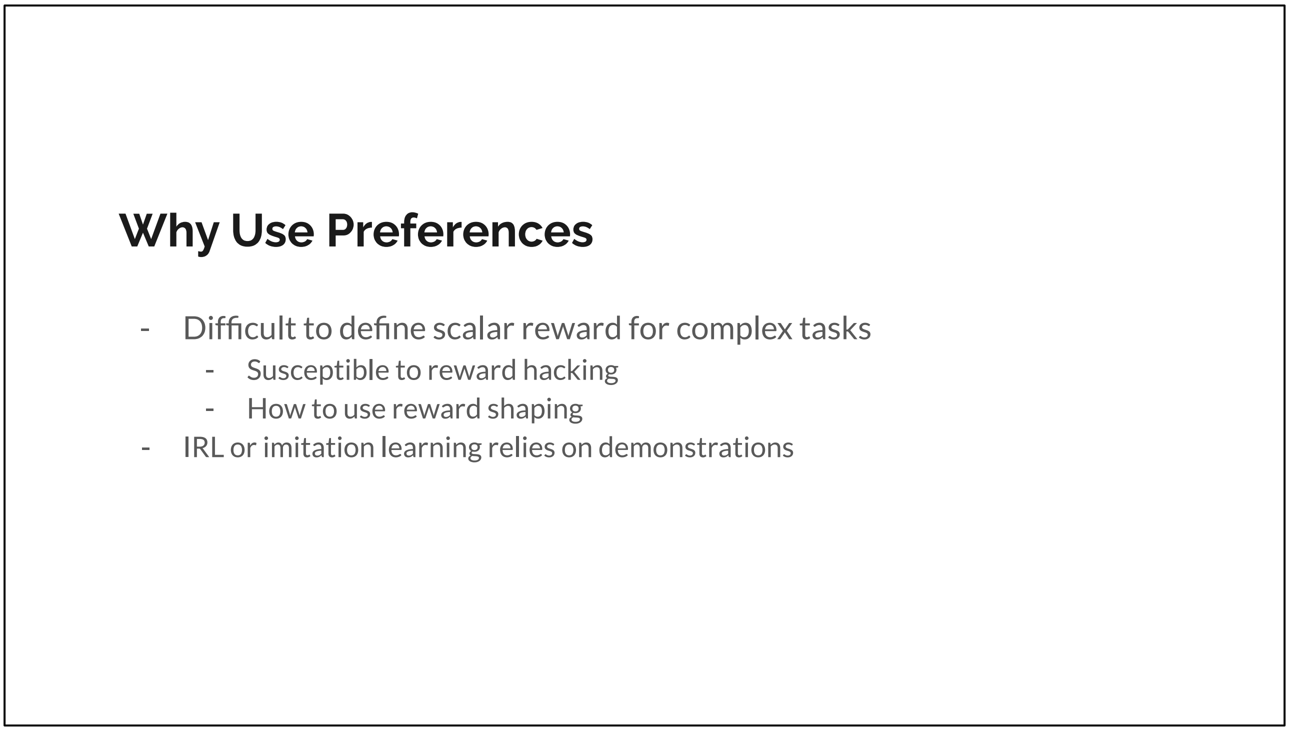
Preference-Based Reinforcement Learning

Myra Cheng



Why Use Preferences

- Difficult to define scalar reward for complex tasks
    - Susceptible to reward hacking
    - How to use reward shaping
- IRL or imitation learning relies on demonstrations

misalignment between our values and the objectives of our RL systems

Reward hacking: maximize given reward but reward doesnt align with the actual task

Reward shaping: reward also helps guide the agent to learn the correct thing

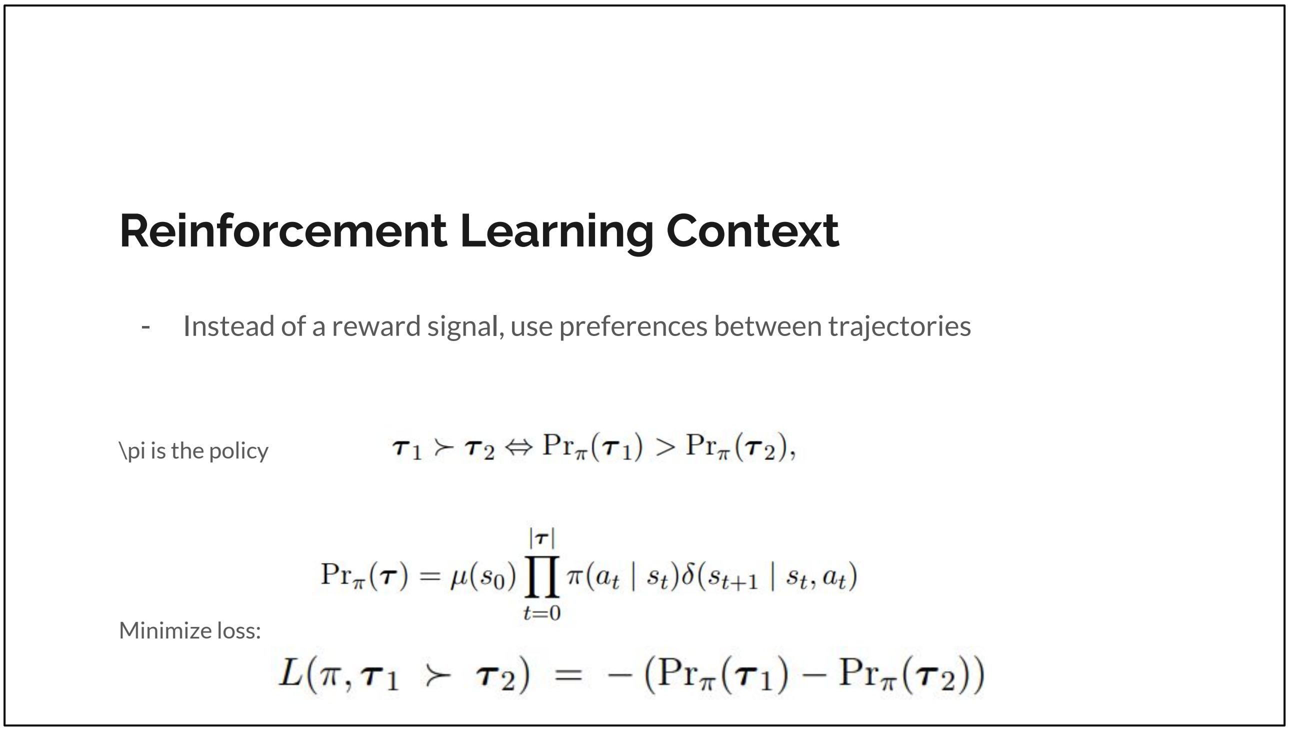
Preference Learning

- Learn from pairwise preferences
  - Each sample in the training set is (a, b, ρ) where
    - a and b are two comparable objects
    - ρ is the preference (either a<b or b<a)

$$\rho(\boldsymbol{\tau}_i \succ \boldsymbol{\tau}_j) = 1 - \rho(\boldsymbol{\tau}_j \succ \boldsymbol{\tau}_i)$$

Assume that preferences are generated by underlying "true" reward fn

In RL context, objects can be states, actions, or trajectories



Reinforcement Learning Context

- Instead of a reward signal, use preferences between trajectories

\pi is the policy

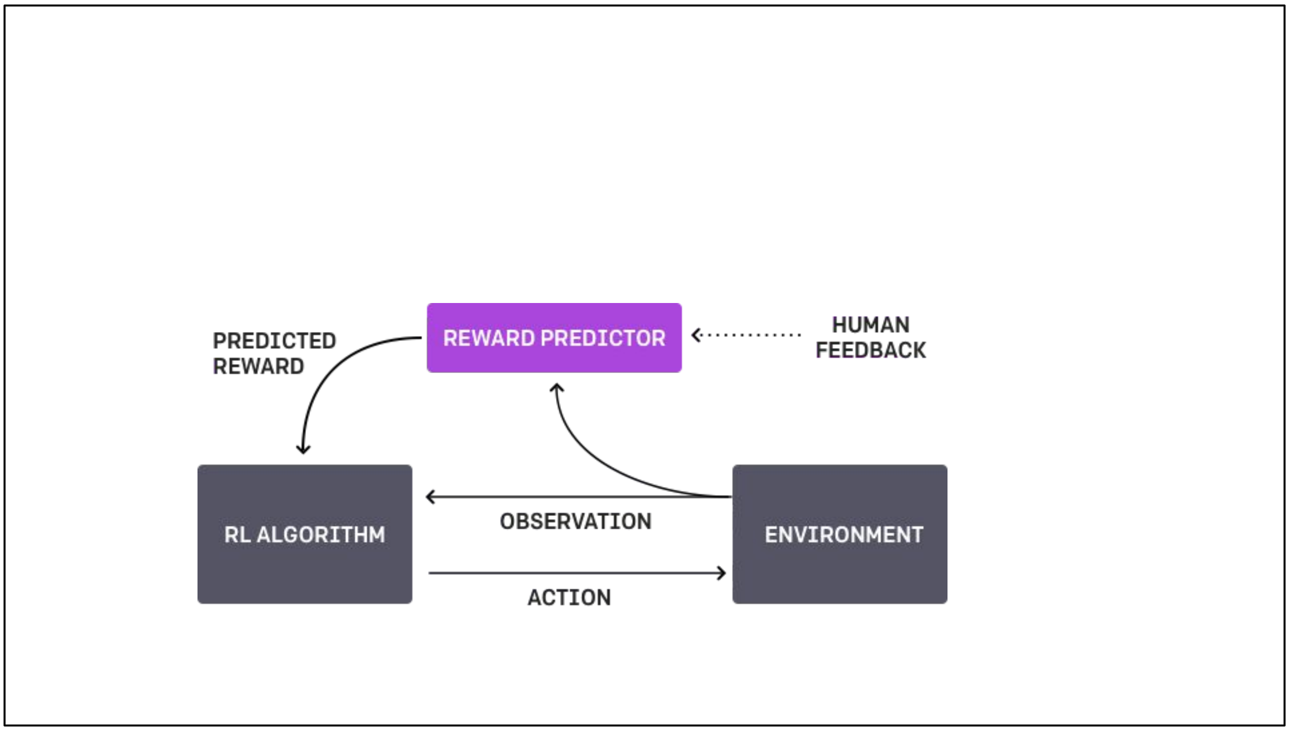
$$\boldsymbol{\tau}_1 \succ \boldsymbol{\tau}_2 \Leftrightarrow \Pr_\pi(\boldsymbol{\tau}_1) > \Pr_\pi(\boldsymbol{\tau}_2),$$

$$\Pr_\pi(\boldsymbol{\tau}) = \mu(s_0) \prod_{t=0}^{|\boldsymbol{\tau}|} \pi(a_t \mid s_t) \delta(s_{t+1} \mid s_t, a_t)$$

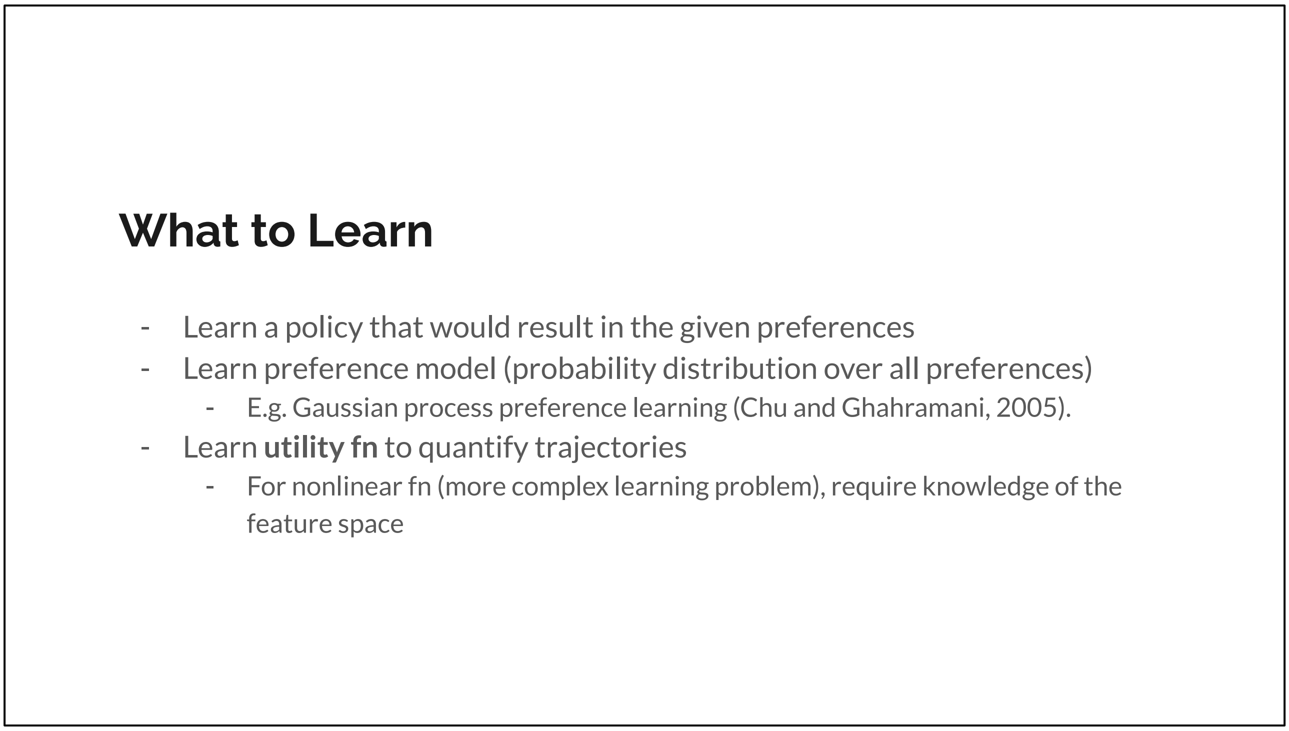
Minimize loss:

$$L(\pi, \boldsymbol{\tau}_1 \succ \boldsymbol{\tau}_2) = -(\Pr_\pi(\boldsymbol{\tau}_1) - \Pr_\pi(\boldsymbol{\tau}_2))$$

Trajectories are most general (can consist of single state)



Offline for nuro context



What to Learn

- Learn a policy that would result in the given preferences
- Learn preference model (probability distribution over all preferences)
  - E.g. Gaussian process preference learning (Chu and Ghahramani, 2005).
- Learn **utility fn** to quantify trajectories
  - For nonlinear fn (more complex learning problem), require knowledge of the feature space

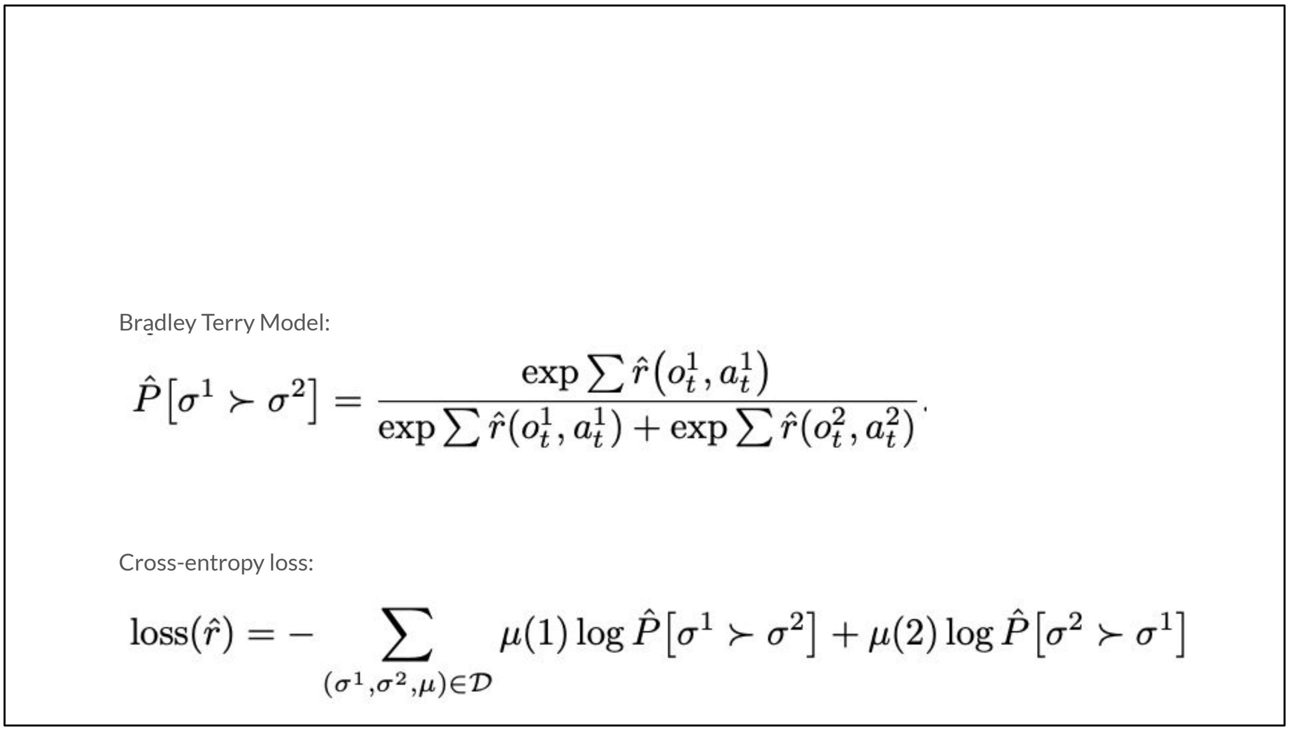
In RL, we want a utility fn hat induces the same, optimal policy as the true reward function. (but may not have markov property)

Christiano et al.

1. Maintain a policy \pi and reward function estimate r’
   a. The policy is updated by some RL algorithm and produces sets of trajectories
2. Select pairs of trajectories
3. Parameters of r’ optimized using supervised learning

Each step is asynchronous/offline from the other ones (many other approaches are interactive)

Reward function might be non-stationary



Bradley Terry Model:

$$\hat{P}\left[\sigma^1 \succ \sigma^2\right] = \frac{\exp \sum \hat{r}\left(o_t^1, a_t^1\right)}{\exp \sum \hat{r}(o_t^1, a_t^1) + \exp \sum \hat{r}(o_t^2, a_t^2)}.$$

Cross-entropy loss:

$$\operatorname{loss}(\hat{r}) = -\sum_{(\sigma^1,\sigma^2,\mu)\in\mathcal{D}} \mu(1) \log \hat{P}\left[\sigma^1 \succ \sigma^2\right] + \mu(2) \log \hat{P}\left[\sigma^2 \succ \sigma^1\right]$$

assume that the human’s probability of preferring a segment i depends exponentially on the value of the latent reward summed over the length of the clip

minimize the cross-entropy loss between these predictions and the actual human labels

Step 3 (learning r’)

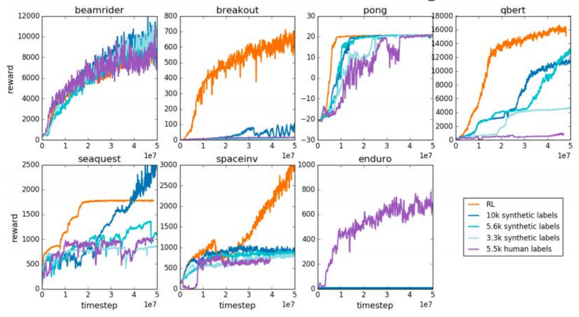
- Average of ensemble of predictors

Selecting queries: maximize uncertainty by asking for preferences on the pairs that have most variance across the ensemble

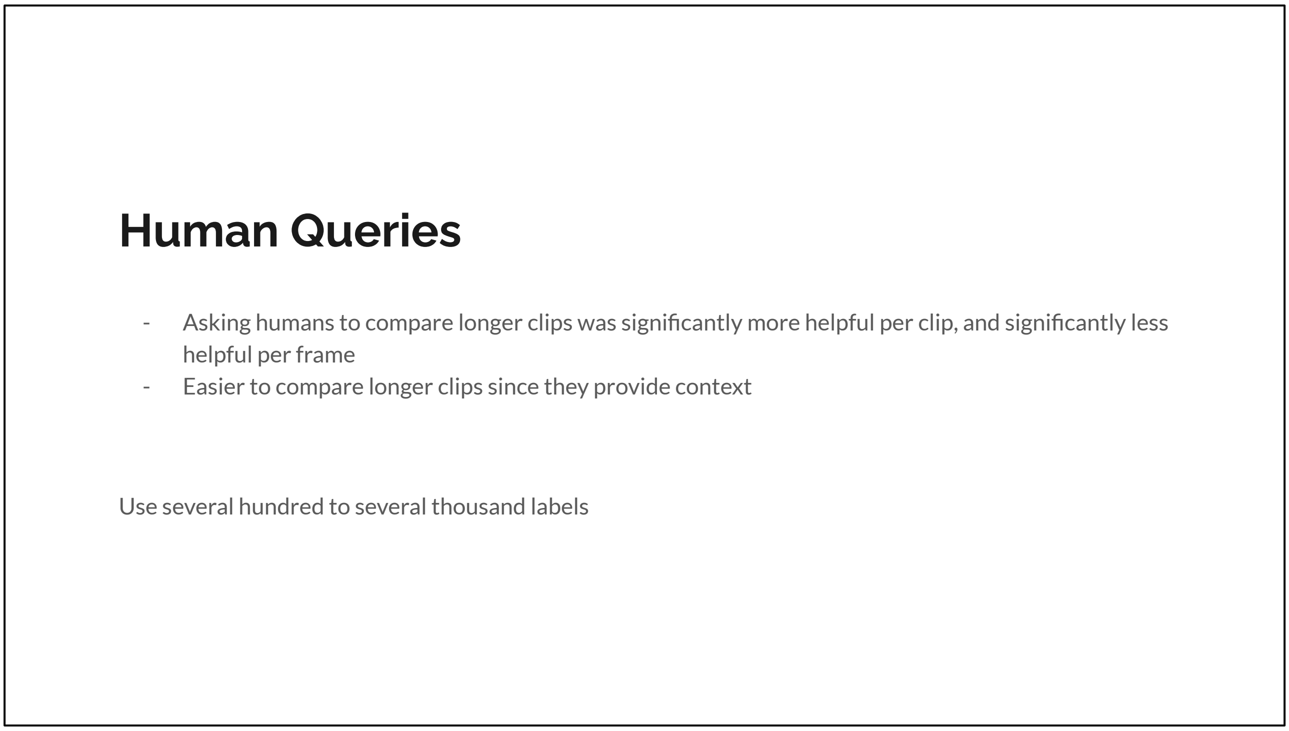
- Want to maximize expected value of information of the query

Empirical Results (mixed)

- Tested in simulated robotic tasks in MuJoCo and OpenAI gym
- After 1400 labels, preference-based learning performs better than using tobjectie reward
- Depends on how the human is asked for feedback, e.g. told that the robot should “stand upright”



On the Ant task the human feedback significantly outperformed the synthetic feedback, apparently because we asked humans to prefer trajectories where the robot was “standing upright,” which proved to be useful reward shaping



Human Queries

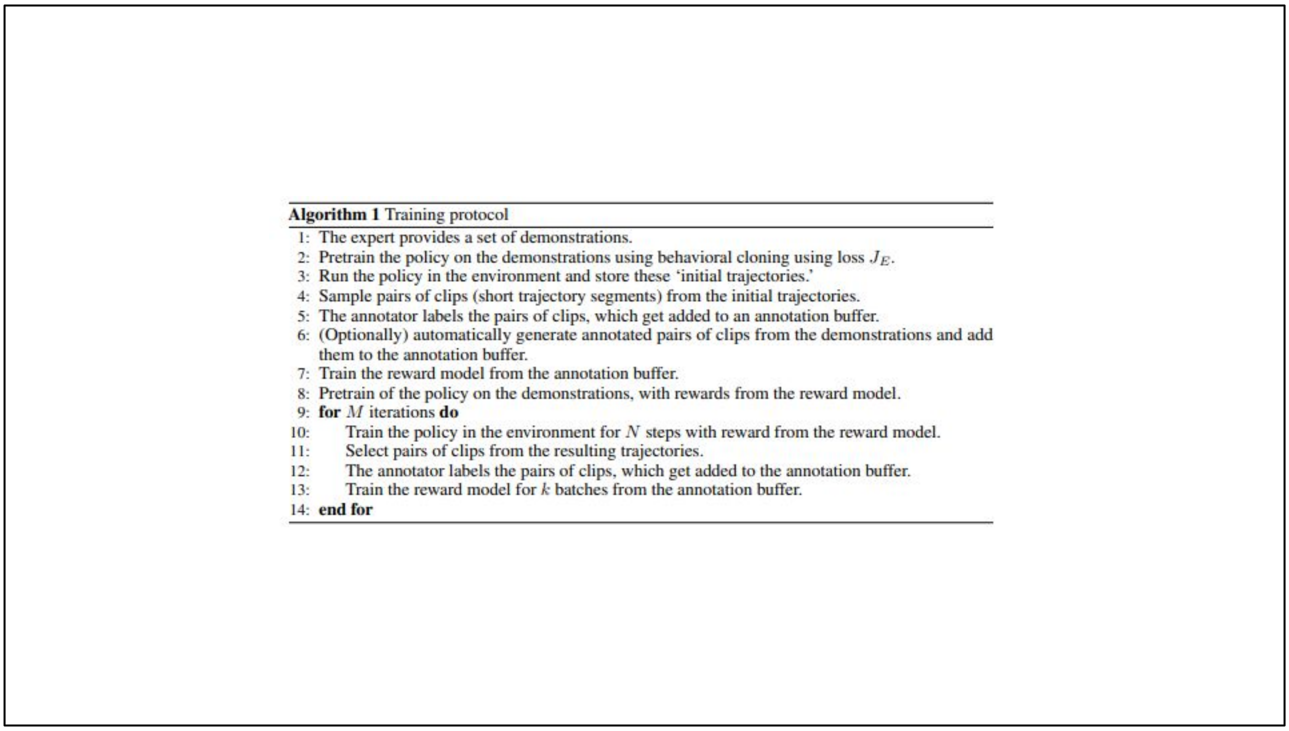
- Asking humans to compare longer clips was significantly more helpful per clip, and significantly less helpful per frame
- Easier to compare longer clips since they provide context

Use several hundred to several thousand labels

Requires the same amount of time

Ibarz et al.

- Combine this supervised approach with imitation learning
  - Initialize the agent's policy with imitation learning
  - Use trajectory preferences + expert demonstrations
    - Pretrain the policy from demonstrations using behavioral cloning + reward as defined by learned r'
- Use CNN for the supervised model
- Autolabels and synthetic feedback help

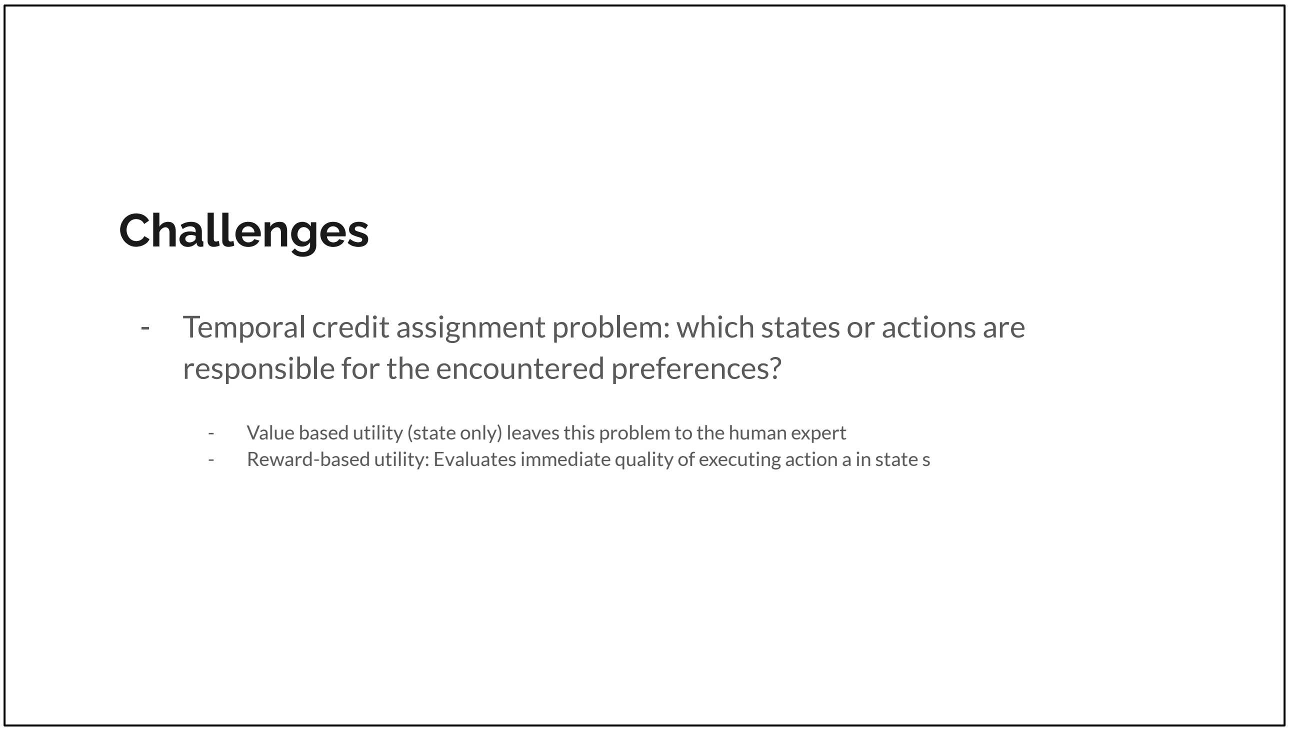


Algorithm 1 Training protocol

```
1: The expert provides a set of demonstrations.
2: Pretrain the policy on the demonstrations using behavioral cloning using loss $J_E$.
3: Run the policy in the environment and store these ‘initial trajectories.’
4: Sample pairs of clips (short trajectory segments) from the initial trajectories.
5: The annotator labels the pairs of clips, which get added to an annotation buffer.
6: (Optionally) automatically generate annotated pairs of clips from the demonstrations and add
   them to the annotation buffer.
7: Train the reward model from the annotation buffer.
8: Pretrain of the policy on the demonstrations, with rewards from the reward model.
9: for $M$ iterations do
10:     Train the policy in the environment for $N$ steps with reward from the reward model.
11:     Select pairs of clips from the resulting trajectories.
12:     The annotator labels the pairs of clips, which get added to the annotation buffer.
13:     Train the reward model for $k$ batches from the annotation buffer.
14: end for
```

Preference Elicitation/Generation

- What trajectories to consider, and how to pair them together?
- When to ask for new queries?
- Minimize expert queries (costly, human interaction), while generating new trajectories is much cheaper



Challenges

- Temporal credit assignment problem: which states or actions are responsible for the encountered preferences?
  - Value based utility (state only) leaves this problem to the human expert
  - Reward-based utility: Evaluates immediate quality of executing action a in state s

todo

Papers

- Deep Reinforcement Learning from Human Preferences Christiano et al. 2017
    - Follow up: Reward learning from human preferences and demonstrations in Atari
- A survey of preference-based reinforcement learning methods Wirth et al. 2017
- More papers: Human-in-the-loop reward papers