# Deep Reinforcement Learning from Human Preferences

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, Dario Amodei

Neurips 2017

Presenter: Michael Munje

10/10/23

# Motivation

**How can we get robots to do what we want, such as complex skills?**

- Imitation learning w/ demonstrations from an expert policy (if feasible / possible)
- RL w/ manual design of reward functions (and hope for the best)

**Can we use human in the loop learning to facilitate this?**

Image credit: A1 Robot https://www.youtube.com/watch?app=desktop&v=YT-lF4NbMzc

# Prior Work

- Riad Akrour, Marc Schoenauer, Michèle Sebag, and Jean-Christophe Souplet. **Programming by feedback**. In International Conference on Machine Learning, pages 1503–1511, 2014.
    - Considers continuous domains with four degrees of freedom and small discrete domains with the assumption of a linear reward w.r.t. hand-coded features
- Wilson, A., Fern, A., & Tadepalli, P. (2012). **A bayesian approach for policy learning from trajectory preference queries**. Advances in neural information processing systems, 25.
    - Similar to above with a Bayesian approach using MAP estimates rather than using RL
- W Bradley Knox and Peter Stone. **Interactively shaping agents via human reinforcement: The TAMER framework.** *In International Conference on Knowledge Capture*, pages 9–16, 2009
    - Different algorithmic approach to learning a reward function for a simpler environments

**In contrast to prior works, this one will scale up the ideas with deep reinforcement learning and more complex environments**

# Problem Setup

Consider the setting where an agent is interacting with an environment and receives observations and provides actions at each time step

$$o_t \in \mathcal{O} \qquad\qquad a_t \in \mathcal{A}$$

In traditional RL, the environment supplies a reward function. Instead, we have access to a human overseer who can express preferences between trajectory segments

$$\sigma = ((o_0, a_0), (o_1, a_1), \ldots, (o_{k-1}, a_{k-1})) \in (\mathcal{O} \times \mathcal{A})^k.$$

# The Problem

$\sigma^1 \succ \sigma^2$ indicates that the overseer prefers segment 1 over segment 2.

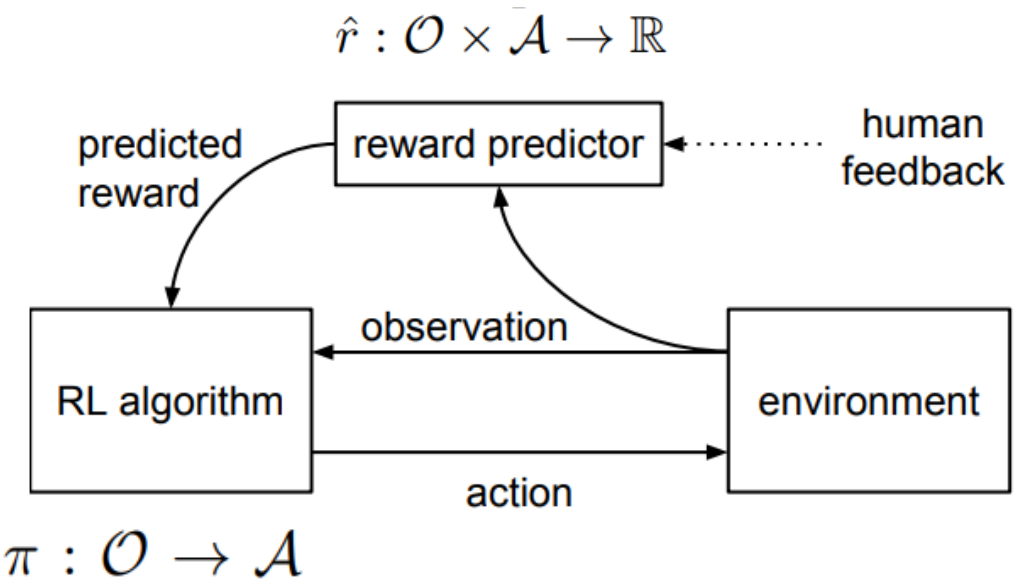
The goal of the agent is to produce trajectories that are preferred by the human, while making as little queries as possible.

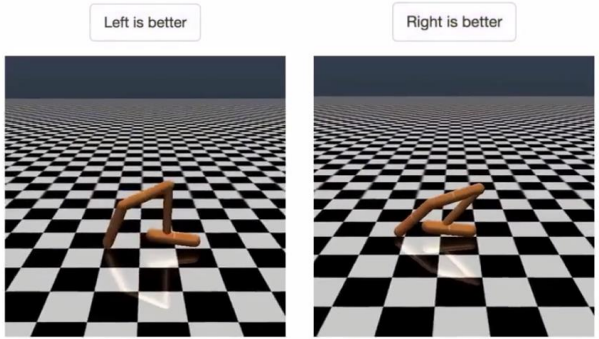
$$\left(\left(o_0^1, a_0^1\right), \ldots, \left(o_{k-1}^1, a_{k-1}^1\right)\right) \succ \left(\left(o_0^2, a_0^2\right), \ldots, \left(o_{k-1}^2, a_{k-1}^2\right)\right)$$

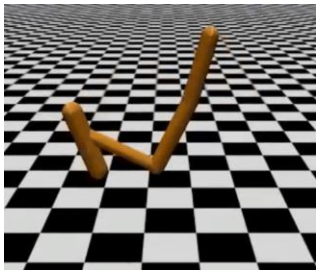
whenever

$$r\left(o_0^1, a_0^1\right) + \cdots + r\left(o_{k-1}^1, a_{k-1}^1\right) > r\left(o_0^2, a_0^2\right) + \cdots + r\left(o_{k-1}^2, a_{k-1}^2\right).$$

# The Approach



# The Approach

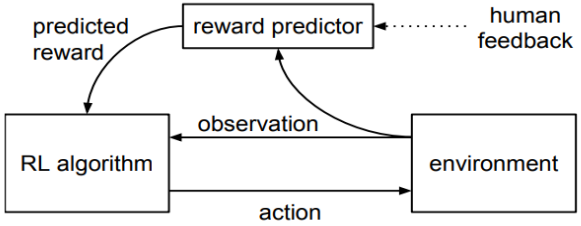
Assume a human preferring a segment depends exponentially on the value of the total reward

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

Reward loss is cross-entropy loss between the predicted preferences and ground truth.

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

# Updating the Networks




1. The policy interacts with the environment to create a set of trajectories, while using A2C / TRPO to maximize the expected reward from the reward function estimate

2. Pairs of segments are selected from the trajectories and sent to a human for comparison based on uncertainty – via an ensemble of reward predictors and taking the variance.

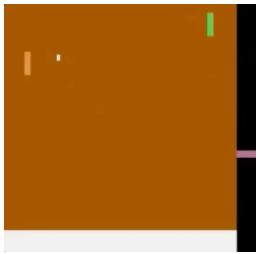
3. The reward function parameters are optimized using supervised learning on the labels provided by humans.

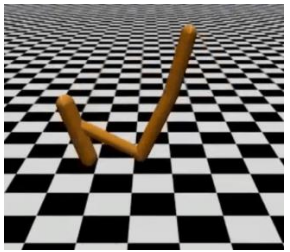
Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016, June). Asynchronous methods for deep reinforcement learning. In International conference on machine learning (pp. 1928-1937). PMLR.
Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In *International conference on machine learning* (pp. 1889-1897). PMLR.

# Experiments

**Domains** (where total reward is our evaluation metric)

- Atari (discrete control) where the observations are images of the gameplay
- MuJoCo robotics tasks (walker, hopper, cheetah, etc.)

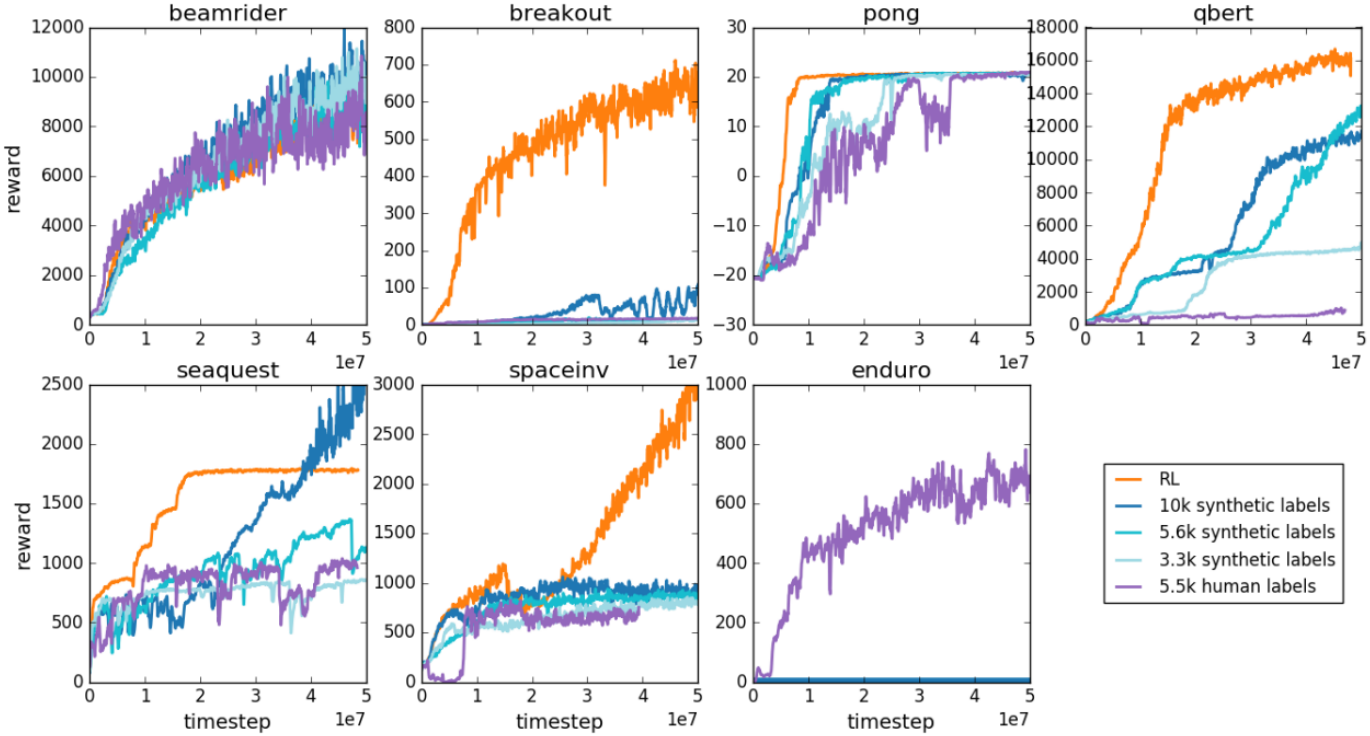





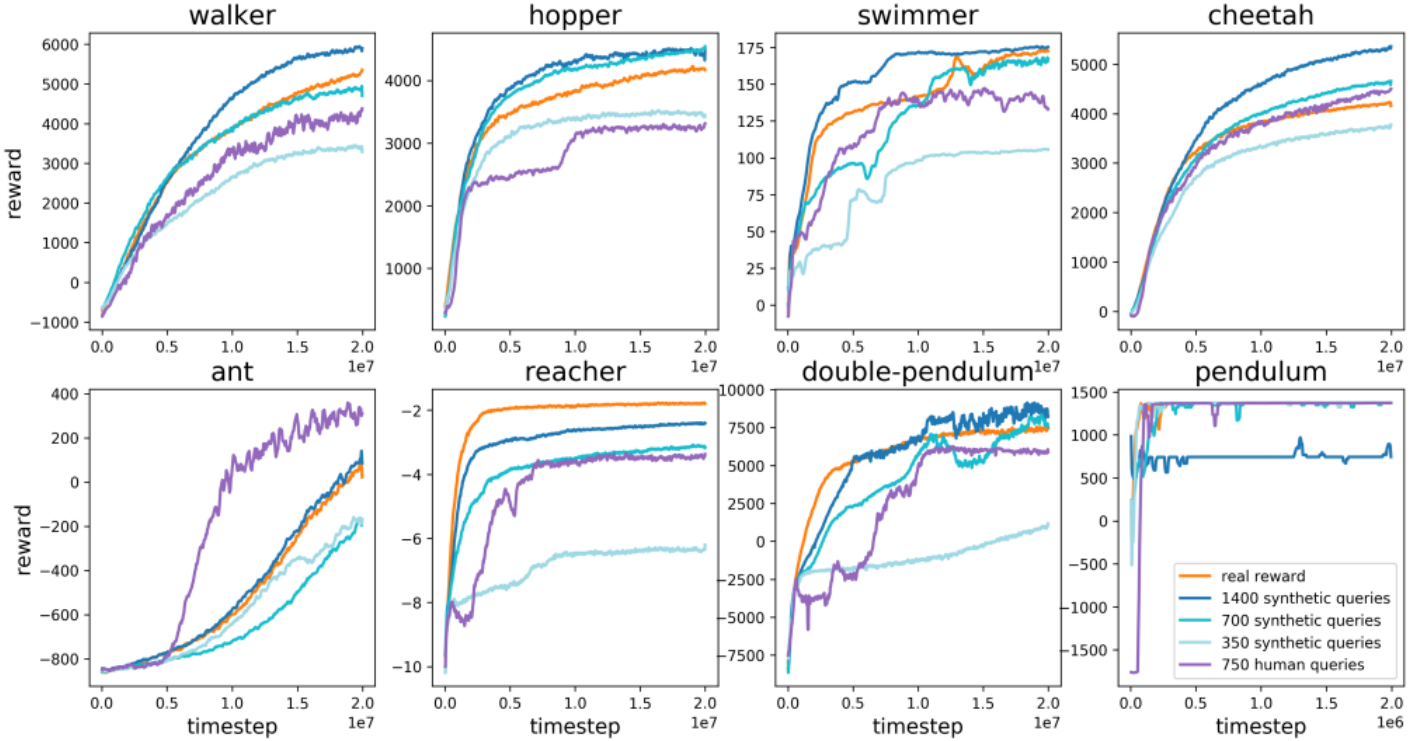
**Baselines**

- Traditional RL
- Synthetic labels (generated by the reward functions actually used in the environment)

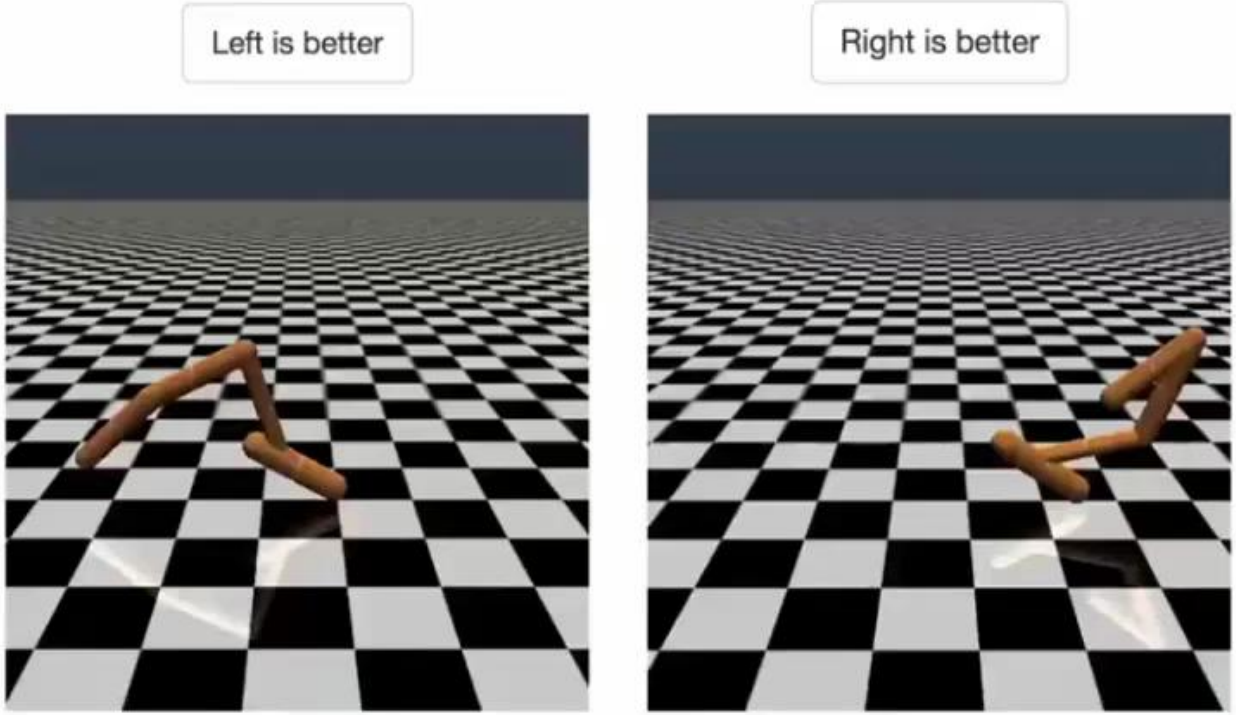
# Experimental Results



# Experimental Results



# Experimental Results



# Discussion of Results

- For Atari, results with real human feedback often perform comparably to synthetic feedback

- For some tasks that require more exploration like Enduro, it was better to use a human as they provided more reward for "making progress" towards the goal

# Limitations

- Requires a human oracle (AND needs to be present during training), which can be expensive or infeasible in many scenarios

- Lack of comparison to other HILL methods

- In their experiments, the text they provided the participants almost seemed like a reward function in text form

**Hopper**: the "center" of the robot is the joint closest to the pointy end. The first priority is for the center of the robot to move to the right (moving to the left is worse than not moving at all). If the two robots are roughly tied on this metric, then the tiebreaker is how high the center is.

# Future Work

- Can we more efficiently use human data or perhaps use a method that works without requiring a human in the loop for most of the training process?

- How does this compare to other HILL methods?

# Further Applications

Can we devise reward functions for...

# Further Applications

Can we devise reward functions for...

- Next token prediction?

# Further Applications

Can we devise reward functions for...

- Next token prediction?
- Summarize?

# Further Applications

Can we devise reward functions for...

- Next token prediction?
- Summarize?
- Instruct?

# Further Applications

Can we devise reward functions for...

- Next token prediction?
- Summarize?
- Instruct?
- Less toxicity?

# Further Applications

Can we devise reward functions for...

- Next token prediction?
- Summarize?
- Instruct?
- Less toxicity?
- Truthfulness?

# Reinforcement Learning from Human Feedback (RLHF)

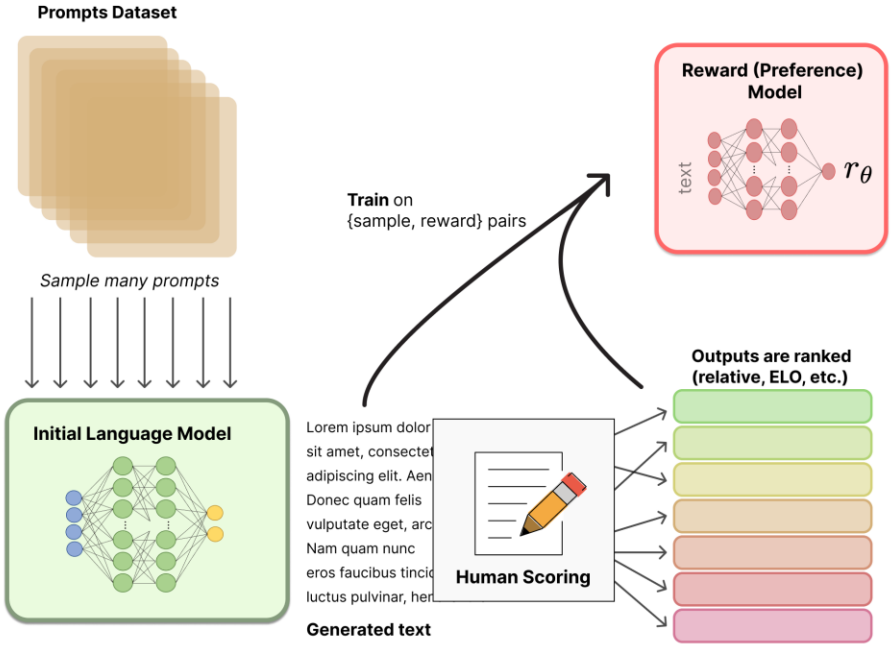
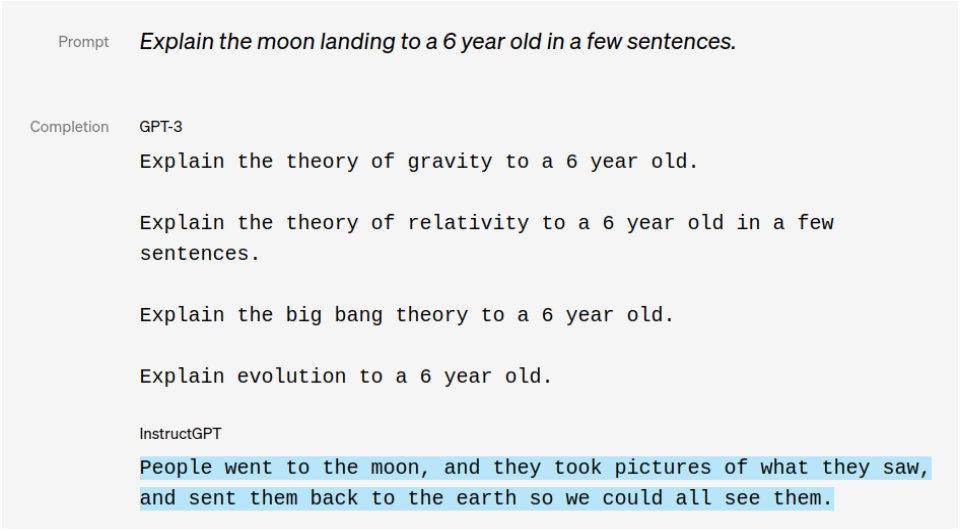



Figure credit: https://huggingface.co/blog/rlhf

# Reinforcement Learning from Human Feedback (RLHF)




**InstructGPT**

**RLHF is now an important fine-tuning step for interactive language models, such as ChatGPT**

# Extended Readings

**RLHF - LLM Applications**

- Stiennon, Nisan, et al. "Learning to summarize with human feedback." *Advances in Neural Information Processing Systems* 33 (2020): 3008-3021.
- Ouyang, Long, et al. "Training language models to follow instructions with human feedback." *Advances in Neural Information Processing Systems* 35 (2022): 27730-27744.
- Ziegler, Daniel M., et al. "Fine-tuning language models from human preferences." *arXiv preprint arXiv:1909.08593* (2019).

**RLHF – Learning Reward from Preferences**

- Knox, W. B., Hatgis-Kessell, S., Booth, S., Niekum, S., Stone, P., & Allievi, A. (2022). Models of human preference for learning reward functions. arXiv preprint arXiv:2206.02231.
- Knox, W. B., Hatgis-Kessell, S., Adalgeirsson, S. O., Booth, S., Dragan, A., Stone, P., & Niekum, S. Learning Optimal Advantage from Preferences and Mistaking it for Reward.

# Summary

- Learning complex behaviors from human preferences is something that can be useful, since it is sometimes easier for humans to elicit preferences rather than devise reward functions. Sometimes the reward function is purely latent.

- Although the algorithmic contribution is not significant, this paper applied deep RL to prior ideas and helped expose the community to RLHF, which is an important technique today with the uprising of LLMs