# Lecture 9: RLHF and Guest Lecture on DPO

Emma Brunskill

CS234 Reinforcement Learning.

Spring 2024

# Midterm

- In class on Wednesday
- You are allowed 1 side of 1 8.5" x 11" sheet of notes
- All material through today's lecture (Monday) is eligible for the exam
- See Ed post for additional details and past related midterm/quizzes
- Good luck!

# Refresh Your Understanding L9N1

Select all that are true

- (a) The Bradley Terry model expresses the probability that someone will select option $b_i$ over $b_j$
- (b) Using preference tuples and the Bradley Terry model, one can learn a model of the reward function
- (c) The resulting reward function can be shifted by any constant and will not change the resulting preferences
- (d) The resulting reward function can be multiplied by any constant and will not change the resulting preferences
- (e) In RLHF we update the reward model after each PPO roll out
- (f) Not sure

# Refresh Your Understanding L9N1 Solutions

Select all that are true

1. The Bradley Terry model expresses the probability that someone will select option $b_i$ over $b_j$
2. Using preference tuples and the Bradley Terry model, one can learn a model of the reward function
3. The resulting reward function can be shifted by any constant and will not change the resulting preferences
4. The resulting reward function can be multiplied by any constant and will not change the resulting preferences
5. In RLHF we update the reward model after each PPO roll out
6. Not sure

# Class Structure

- Last time: Imitation Learning (Max Entropy IRL) and RLHF
- This time: RLHF and Direct Preference Optimization (best paper runner up at top ML conference) guest lecture
- Next time: Midterm

# Today

- RLHF for LLM
- Direct Preference Optimization

# Pairwise Comparisons

- Often easier for people to make than hand writing a reward function
- Often easier than providing scalar reward (how much do you like this ad?)

# Fitting the Parameters of a Bradley-Terry Model

- Consider $k$-armed bandits[1]: $K$ actions $b_1, b_2, \ldots b_k$. No state/context.
- Assume a human makes noisy pairwise comparisons, where the probability she prefers $b_i \succ b_j$ is

$$P(b_i \succ b_j) = \frac{\exp(r(b_i))}{\exp(r(b_i)) + \exp(r(b_j))} = p_{ij} \tag{1}$$

- Assume have $N$ tuples of form $(b_i, b_j, \mu)$ where $\mu(1) = 1$ if the human marked $b_i \succ b_j$, $\mu(1) = 0.5$ if the human marked $b_i = b_j$, else 0 if $b_j \succ b_i$
- Maximize likelihood with cross entropy

$$loss = - \sum_{(b_i, b_j, \mu) \in \mathcal{D}} \mu(1) \log P(b_i \succ b_j) + \mu(2) \log P(b_j \succ b_j) \tag{2}$$

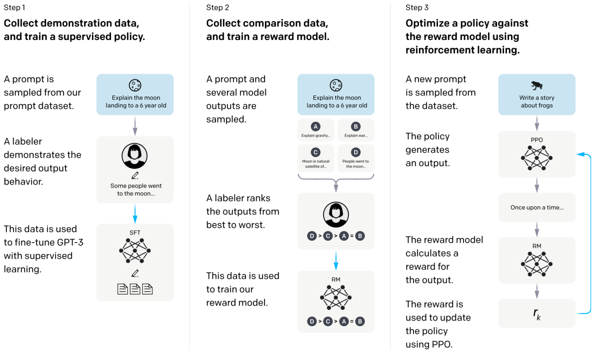
- Use learned reward model, and do PPO with this model
- See prior lecture for notes on doing this over trajectories

[1] We will see more on bandits later in the course

# From Backflips to ChatGPT

- How is this used in ChatGPT?
- Next set of slides are from part of Tatsu Hashimoto's Lecture 11 in CS224N

# High-level instantiation: ‘RLHF’ pipeline

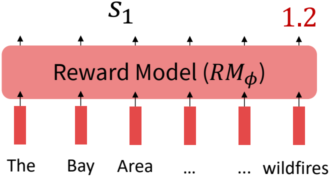



- First step: instruction tuning!
- Second + third steps: maximize reward (but how??)

# How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

| | | | | |
|---|---|---|---|---|
| `An earthquake hit San Francisco. There was minor property damage, but no injuries.` | > | `A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.` | > | `The Bay Area has good weather but is prone to earthquakes and wildfires.` |
| $s_1$ | | $s_3$ | | $s_2$ |




Bradley-Terry [1952] paired comparison model

$$J_{RM}(\phi) = -\mathbb{E}_{(s^w, s^l) \sim D}\left[\log \sigma(RM_\phi(s^w) - RM_\phi(s^l))\right]$$

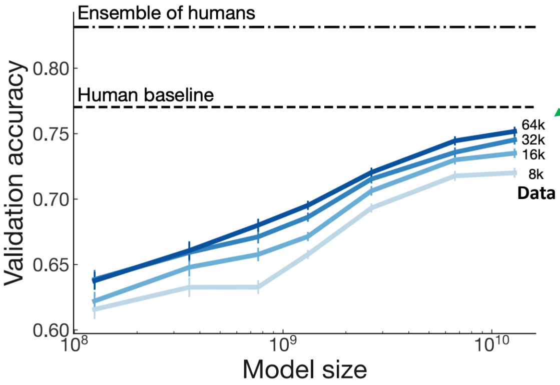
"winning" sample

"losing" sample

$s^w$ should score higher than $s^l$

# Make sure your reward model works first!

Evaluate RM on predicting outcome of held-out human judgments




**Large enough RM trained on enough data approaching single human perf**

[Stiennon et al., 2020]

## RLHF: Putting it all together [Christiano et al., 2017; Stiennon et al., 2020]

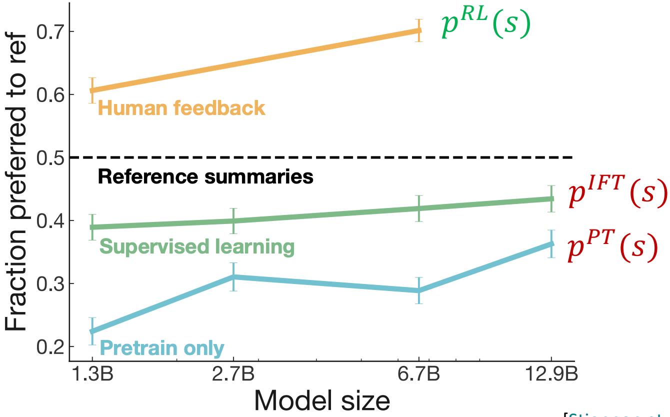
- Finally, we have everything we need:
  - A pretrained (possibly instruction-finetuned) LM $p^{PT}(s)$
  - A reward model $RM_\phi(s)$ that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
  - A method for optimizing LM parameters towards an arbitrary reward function.
- Now to do RLHF:
  - Initialize a copy of the model $p_\theta^{RL}(s)$ , with parameters $\theta$ we would like to optimize
  - Optimize the following reward with RL:

$$R(s) = RM_\phi(s) - \underbrace{\beta \log \left( \frac{p_\theta^{RL}(s)}{p^{PT}(s)} \right)}$$

Pay a price when $p_\theta^{RL}(s) > p^{PT}(s)$

This is a penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler** (**KL**) divergence between $p_\theta^{RL}(s)$ and $p^{PT}(s)$.

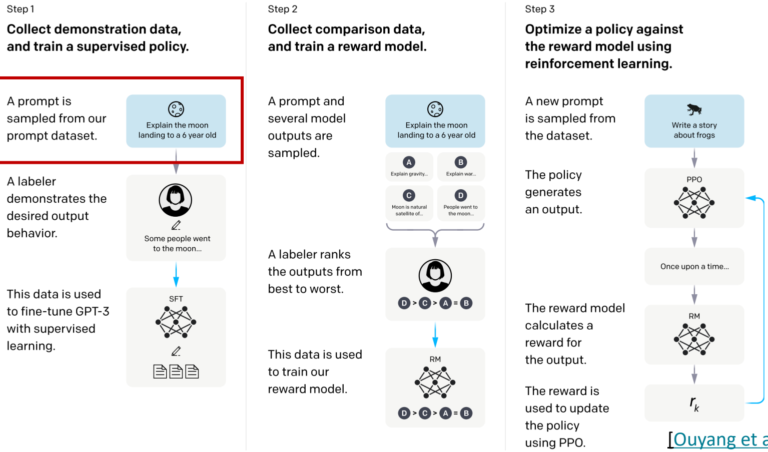
# RLHF provides gains over pretraining + finetuning




[Stiennon et al., 2020]

# InstructGPT: scaling up RLHF to tens of thousands of tasks

**30k tasks!**




[Ouyang et al., 2022]

## Controlled comparisons of “RLHF” style algorithms

| Method | Simulated win-rate (%) | Human win-rate (%) |
|---|---|---|
| GPT-4 | $79.0 \pm 1.4$ | $69.8 \pm 1.6$ |
| ChatGPT | $61.4 \pm 1.7$ | $52.9 \pm 1.7$ |
| PPO | $46.8 \pm 1.8$ | $55.1 \pm 1.7$ |
| Best-of-$n$ | $45.0 \pm 1.7$ | $50.7 \pm 1.8$ |
| Expert Iteration | $41.9 \pm 1.7$ | $45.7 \pm 1.7$ |
| SFT 52k (Alpaca 7B) | $39.2 \pm 1.7$ | $40.7 \pm 1.7$ |
| SFT 10k | $36.7 \pm 1.7$ | $44.3 \pm 1.7$ |
| Binary FeedME | $36.6 \pm 1.7$ | $37.9 \pm 1.7$ |
| Quark | $35.6 \pm 1.7$ | - |
| Binary Reward Conditioning | $32.4 \pm 1.6$ | - |
| Davinci001 | $24.4 \pm 1.5$ | $32.5 \pm 1.6$ |
| LLaMA 7B | $11.3 \pm 1.1$ | $6.5 \pm 0.9$ |

- Many works study RLHF behaviors using GPT-4 feedback (**Simulated**) as a surrogate for **Human** feedback.
- PPO (method in InstructGPT) does work
- Simple baselines (Best-of-n, Training on ‘good’ outputs) works well too

[Dubois et al 2023]

# Today

- RLHF for LLM
- **Direct Preference Optimization**

# Learning More

- Learning and making decisions from human preferences is a rich area intersecting social choice, computational economics and AI
- New course at Stanford on this topic: Koyejo's CS329H: Machine Learning from Human Preferences

# Class Structure

- Last time: Imitation Learning (Max Entropy IRL) and RLHF
- This time: RLHF and Direct Preference Optimization (best paper runner up at top ML conference) guest lecture
- Next time: Midterm