Reinforcement Learning from Human Feedback

- Instruction tuning uses labeled data. This has limitations:
  - As models get larger, low-quality datasets limit capabilities
  - Model can't generalize to tasks beyond those in tuning datasets

* Alternative: Generate outputs from state-of-the-art models on new problems, then get reward signal from humans

* RLFH is a key component of ChatGPT and similar systems

- RLFH
  - Base language model $p(y|x)$ assigns probabilities to completions. Train offline.
  - Reward model $r(x,y)$ maps completions y to real-valued scores
  - Data for reward model: Collect 2 LM completions (y_1, y_2) for a single input x. x can be anything as long as people will have preferences over what comes next
  - Annotators label $y_1 \succ y_2$ (prefer 1 to 2) or vice versa
  - Learn r using a Bradley-Terry model over human preference:

$$\boxed{P(y_1 \succ y_2) = \frac{e^{r(x,y_1)}}{e^{r(x,y_1)} + e^{r(x,y_2)}}}$$

* Turns scores into log probabilities. Same as logistic regression, but we learn a continuous scoring function, not a classifier

- RL Phase: do RL with PPO, optimize expected reward

$$E_{x \sim D,\, y \sim p(\cdot|x)} \left[r(x,y) \right]$$

* subject to additional KL penalty that p not deviate too far from base LM p.

- Ideal Scenario: p continuously gets better and better. Reward model can now judge those new, better completions and drive it to get better.

Instruction Tuning

- Want to optimize models for P(answer | prompt), but they're learned on basic LM objective P(word | context)
- One solution: Fine tune these models to do what we care about
- Two ways of doing this in 2023:
    1. Instruction Tuning: Supervised fine-tuning on data derived from many NLP tasks
    2. **RLHF**: RL to improve human judgements of how good outputs are