RLHF: aligning LLMs with human intents

→ Start from a large pretrained LLM
→ Step 1: instruction tuning (SFT)

Limitations of instruction tuning:

→ Don't learn from negative feedback
→ Some prompts (e.g., creative) have many acceptable outputs, we only train on one of them
→ Hard to encourage abstaining when the model doesn't know something
→ Does not directly involve human prefs

How do we incorporate human prefs to address the above issues?

prompt X → Instruction-tuned LLM → Sampling → Y₁

Preferences:

Rank which output is most appropriate given this prompt \( Y_1 > Y_3 > Y_2 \)
limitation: extremely expensive to collect

idea: can we train a model to predict human pref judgment

reward model

input: prompt $x$, output $y$;
output: scalar score

Bradley-Terry pairwise preference model

$y_w \Rightarrow$ preferred by humans

$y_L \Rightarrow$ not preferred

$r(x,y) \Rightarrow$ reward for output $y$ given $x$;
scalar score

$$P(y_w > y_L | x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_L))}$$

$$L = -\log \left( \frac{1}{1 + e^{-x}} \right)$$

simplify

$$o(x) = \frac{1}{1 + e^{-x}}$$

$$L = -\log o(r(x, y_w) - r(x, y_L))$$
→ Intuition: good sample $y_w$‘s reward should be greater than $y_0$‘s reward.

Using the reward model:

```
prompt $x$ $\rightarrow$ SPT LLM $\rightarrow$ SPT LLM $\rightarrow$ $y_1$ $\rightarrow$ [reward model] $\rightarrow$ $y_1$
```

```
$y_2$ $\rightarrow$ [reward model] $\rightarrow$ $y_2$
```

How do we use this to align LLMs to human prefs?

1. "best-of-n" sampling (rejection sampling)
   - generate n samples for a given prompt, score each w/ reward model, choose sample with highest reward
   - very expensive
2. Just fine-tune the LM to maximize \( P(y_w | x) \)
   \[ \Rightarrow \text{RAFT} \]

3. Use reinforcement learning to increase \( P(y_w | x) \) by a small amount
   and decrease \( P(y_L | x) \) by a small amount, where amounts are functions of
   \( R(x, y_w), R(x, y_L) \)

**RLHF step 3:**

- We observe a reward only after generating a complete sequence

\[ \text{TT}_{\text{ref}} \equiv \text{SFT LM checkpoint} \]

\[ \text{TT} \equiv \text{current policy model} \]

\[ \Rightarrow \text{init. to } \text{TT}_{\text{ref}} \]

**Final aligned model**

\[ \max_{\text{TT}} \mathbb{E}_{y_{1:L}} \left[ r(x, y_{1:L}) - \beta \text{KL}(\text{TT}(y_{1:L}) || \text{TT}_{\text{ref}}(y_{1:L})) \right] \]

**KL penalty to prevent huge deviations from \( \text{TT}_{\text{ref}} \)**
\[
D_{KL}(\pi(y|x) \| \pi_{\text{ref}}(y|x)) = \frac{\prod_{i} \pi(w_i | w_1 \ldots w_{i-1}, x)}{\prod_{i} \pi_{\text{ref}}(w_i | w_1 \ldots w_{i-1}, x)}
\]

- Optimize using PPO algorithm
  - Schulman, 2016 → ChatGPT, GPT4
  - Can also use REINFORCE
  - Williams 1992
  - Gemini

**Final RLHF pipeline:**

1. Base pre-trained LM
2. Instruction tuning
   - Gathered dataset of (last, input, output)
3. Instruction tuned LLM (SFT LLM)
4. Reward Model
   - Train with PPO to maximize expected reward
   - Demonstrate to humans to label them
5. RLHF aligned model

\( \text{gen. } \{x_1, y_1, y_2\} \)