Reinforcement Learning from Human Feedback

RL from human preferences

Why preferences, not ratings

Humans are often better at making **relative judgments** than **absolute judgments**.

- "Which of the football teams will be more competitive next season? Michigan or Michigan State?"
- "On a scale from 0 to 10, how competitive will Michigan football team be? how competitive will Michigan State be?"

More issues

- annotators may have different interpretations of what a score like "6" or "8" means
- relative judgments tend to be more stable across annotators

RL from human preferences

Consider an episodic MDP $(\mathcal{S}, \mathcal{A}, P, H)$. The policy $\pi_\theta(a \mid s)$.

- A trajectory is $\tau = (s_1, a_1, \ldots, s_H, a_H)$.
- We do *not* assume access to the true reward $r(\tau)$.
- Instead, we may observe preferences over trajectories τ_1 and τ_2 from human:

$$\tau_1 \succ \tau_2, \text{ or } \tau_2 \succ \tau_1.$$

Core objective

Learn a policy that induces trajectories preferred by humans.

RL from human preferences

The Bandit View

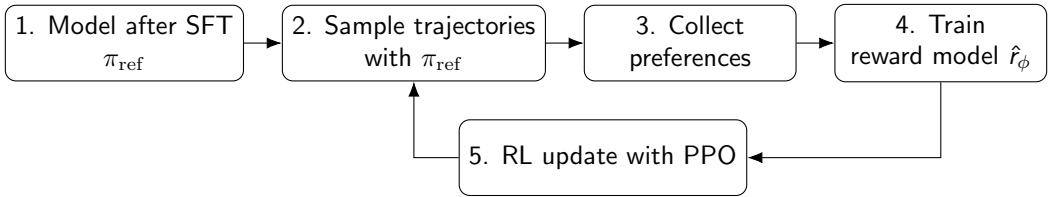
Given a prompt x, each response y is viewed as an arm. The LLM is policy

$$\pi_\theta(y|x).$$

The MDP View: Token-level

- State $s_t = (x, y_1, \cdot, y_t)$
- Action $a_t = y_{t+1}$
- Next state $s_{t+1} = (x, y_1, \cdots, y_{t+1})$
- Policy $\pi(y_{t+1}|x; y_{\leq t})$ (next token prediction).
- Reward $r(s_t, a_t)$???

The RLHF pipeline (GPT 3)



Interpretation

We first **learn the reward function** from preferences and use classic RL to **optimize around a reference policy**.

Reward model learning from preferences

Reward model

A model (neural network) $r_\phi(x; y)$ that gives a scalar reward for given a (prompt, response) pair.

Reward model training

- Reward model is trained with human feedback (preferences).
- Given (x, y_1) and (x, y_2) (two responses to the same prompt), a human annotator provides preference (y_1 is better or y_2 is better).

Link Function or Preference Model

Bradley-Terry Model (1952)

Given two trajectories (τ_1, τ_2), assume humans generate their preferences based on the true cumulative reward difference:

$$\mathbb{P}(\tau_1 \succ \tau_2) = \frac{e^{r(\tau_1)}}{e^{r(\tau_1)} + e^{r(\tau_2)}} = \frac{1}{1 + e^{-[r(\tau_1) - r(\tau_2)]}}.$$

Or

$$\mathbb{P}(\tau_1 \succ \tau_2) = \sigma(r(\tau_1) - r(\tau_2))$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function.

Human Feedback

- For each human annotator, we query the preference over (τ_1, τ_2).
- With probability $\mathbb{P}(\tau_1 \succ \tau_2)$, they will say τ_1 is better.
- With probability $1 - \mathbb{P}(\tau_1 \succ \tau_2)$, they will say τ_2 is better.

Reward model learning from preferences

Training loss

Assume we use a neural network $\hat{r}_\phi(x, y)$ with parameter ϕ to approximate the true reward function $r(x, y)$. Given a preference dataset $\mathcal{D}_P = \{(\tau_1^+, \tau_1^-), (\tau_2^+, \tau_2^-), \cdots\}$,

$$\min_\phi \mathcal{L}_{\mathrm{RM}}(\phi) = -\frac{1}{|\mathcal{D}_P|} \sum_{(\tau_k^+, \tau_k^-) \in \mathcal{D}_P} \log \sigma \left(\hat{r}_\phi(x_k, y_k^+) - \hat{r}_\phi(x_k, y_k^-) \right).$$

- supervised learning similar to BC: expected log likelihood risk.
- when humans prefer one trajectory over the other, what will the reward of each trajectory most likely be?

Reward model learning from preferences

How to use the reward model?

From the token-level $r(\tau) = r(x, y_1, \cdots, y_m) \approx \hat{r}_\phi(x, y)$, i.e. the terminal reward only. For $t < m - 1$, we choose

$$r(s_t, a_t) = r(x, y_{\leq t}; y_{t+1}) = -\beta \log \frac{\pi_\theta(y_{t+1}|x, y_{\leq t})}{\pi_{\text{ref}}(y_{t+1}|x, y_{\leq t})}$$

RL from human preferences

The MDP View: Token-level

- State $s_t = (x, y_1, \cdot, y_t)$
- Action $a_t = y_{t+1}$
- Next state $s_{t+1} = (x, y_1, \cdots, y_{t+1})$
- Policy $\pi(y_{t+1}|x; y_{\leq t})$ (next token prediction).
- A well defined Reward $r(s_t, a_t)$

RL from human preferences

The MDP View: Token-level

- State $s_t = (x, y_1, \cdot, y_t)$
- Action $a_t = y_{t+1}$
- Next state $s_{t+1} = (x, y_1, \cdots, y_{t+1})$
- Policy $\pi(y_{t+1}|x; y_{\leq t})$ (next token prediction).
- A well defined Reward $r(s_t, a_t)$

PPO-Clip

Consider a modified objective

$$\max_{\tilde{\theta}} \mathbb{E}_{\rho_\theta, a\sim\pi_\theta} \left[\min \left\{ \frac{\pi_{\tilde{\theta}}(a|s)}{\pi_\theta(a|s)} A_w(s,a), \left(\frac{\pi_{\tilde{\theta}}(a|s)}{\pi_\theta(a|s)} \right)_{1-\epsilon}^{1+\epsilon} A_w(s,a) \right\} \right]$$

RL with learned reward and KL control

- The terminal reward pushes the policy toward preferred behavior.
- The KL penalty prevents a large distribution shift away from the reference policy.
- Also train a critic to estimate the value of state s, $V_W(s)$, and to compute the advantage function

$$A_W(s_t, a_t) = R(s_t, a_t) - V_W(s),$$

where $R(s_t, a_t) = \hat{r}_\phi(x, y) - \sum_{\tau \geq t} \beta \log \frac{\pi_\theta(y_{t+1}|x, y_{\leq t})}{\pi_{\text{ref}}(y_{t+1}|x, y_{\leq t})}$

RL with learned reward and PPO

Potential Issues

- Two-stage pipeline, three enormous neural networks (reward, critic, actor).
- Reward model is hard to train, and often mis-specifies.

RL with learned reward and PPO

Potential Issues

- Two-stage pipeline, three enormous neural networks (reward, critic, actor).
- Reward model is hard to train, and often mis-specifies.

Can we optimize the preference objective *without* training a reward model?

From RLHF to DPO

Bandit View

Recall the bandit model $r(x, y)$, where x is the prompt and y is the response. Consider the following KL-regularized objective

$$\max_{\pi} \mathbb{E}_{x\sim d_0, y\sim\pi_\theta}\Big[r(x,y)\Big] - \beta\,\mathbb{E}_{x\sim d_0}\Big[\mathrm{KL}\big(\pi(\cdot\,|\,x)\,\|\,\pi_{\mathrm{ref}}(\cdot\,|\,x)\big)\Big]$$

Theorem

The optimal policy of the KL-regularized objective in bandits can be expressed as:

$$\pi^*(y|x) \propto \pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right).$$

Proof

Since this is a bandit problem, the optimization decouples into sub-optimization problem for each x (prompt). So for a fixed x, we only need to optimize over the distribution of the responses to the given x, $\pi(\cdot|x)$, subject to

$$\sum_y \pi(y|x) = 1, \qquad \pi(y|x) \geq 0.$$

For any state x, let

$$V^\pi(x) = \sum_y \pi(y|x) r(x,y) - \beta \sum_y \pi(y|x) \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)}.$$

We will solve this constrained optimization problem using a Lagrange multiplier.

Proof

For a fixed state x, introduce a Lagrange multiplier λ and define

$$\mathcal{L}(\pi, \lambda) = \sum_y \pi(y|x) r(x, y) - \beta \sum_y \pi(y|x) \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} + \lambda \left(\sum_y \pi(y|x) - 1 \right).$$

Take the partial derivative of $\mathcal{L}$ with respect to $\pi(y|x)$:

$$\begin{aligned} \frac{\partial}{\partial \pi(y|x)} \mathcal{L}(\pi, \lambda) =& r(x, y) - \beta \frac{\partial}{\partial \pi(y|x)} \left(\pi(y|x) \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} \right) + \lambda \\ =& r(x, y) - \beta \left(\log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} + 1 \right) + \lambda. \end{aligned}$$

Proof

The KKT condition for the optimal policy $\pi^*(y|x)$ satisfies $\frac{\partial \mathcal{L}}{\partial \pi(y|x)}|_{\pi=\pi^*} = 0$, or

$$0 = r(x,y) - \beta\left(\log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + 1\right) + \lambda.$$

Rearranging gives

$$\beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} = r(x,y) + \lambda - \beta.$$

Exponentiating both sides yields

$$\frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} = \exp\left(\frac{r(x,y)}{\beta}\right)\exp\left(\frac{\lambda-\beta}{\beta}\right).$$

Proof

$$\exp\left(\frac{\lambda-\beta}{\beta}\right)$$

does not depend on y, so it is a normalization constant. Hence

$$\pi^*(y|x) \propto \pi_{\mathrm{ref}}(y|x)\exp\left(\frac{r(x,y)}{\beta}\right).$$

Interpretation of the optimal policy

Optimal Policy

$$\pi^*(y|x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right).$$

Interpretation

- The exponential factor
$$\exp\left(\frac{1}{\beta} r(x,y)\right)$$
upweights high-reward actions.
- The parameter β controls how strongly we stay close to the reference policy:
  - small β: more reward-seeking, more concentrated on high-reward actions
  - large β: more conservative, closer to π_{ref}
- This is exactly a Gibbs/Boltzmann reweighting of the reference policy (soft policy iteration).

DPO reward parameterization

Optimal Policy

From the closed-form of the optimal policy,

$$r(x, y) = \beta \log \frac{\pi^*(x|y)}{\pi_{\text{ref}}(x|y)} + \beta \log Z(x),$$

where $Z(x)$ does not depend on y.

Bradley-Terry Model

Given two trajectories $(\tau_1 = (x, y_1), \tau_2 = (x, y_2))$,

$$\mathbb{P}(\tau_1 \succ \tau_2) = \sigma(r(x, y_1) - r(x, y_2))$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function.

DPO reward parameterization

Combining the optimal Policy and the BT model to remove the reward model:

$$\mathbb{P}(\tau_1 \succ \tau_2) = \sigma\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} - \beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)}\right).$$

The regression problem for finding π^*

Assume we have a set of preferences $\mathcal{D}_p = \{(\tau_{k,1}, \tau_{k,2}, o_k)\}_k$ where $\tau_{k,1} = (x_k, y_{k,1})$, $\tau_{k,2} = (x_k, y_{k,2})$, $y_{k,1}$ and $y_{k,2}$ are two different responses to x_k and $o_k \in \{1, 2\}$ is the human preference. Define a predictive model

$$f_\theta(\tau_{k,1}, \tau_{k,2}) = \sigma\left(\beta \log \frac{\pi_\theta(y_1|x)}{\pi_{\text{ref}}(y_1|x)} - \beta \log \frac{\pi_\theta(y_2|x)}{\pi_{\text{ref}}(y_2|x)}\right),$$

which predicts the preference. Learning the optimal policy is equivalent to a logistic regression to learn f_θ.

DPO loss

DPO Loss

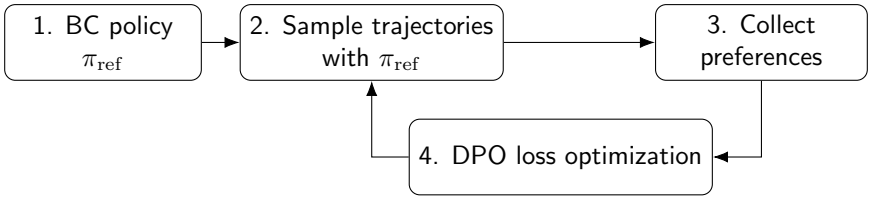
To train with cross-entropy loss, we have

$$\min_{\theta} -\frac{1}{|\mathcal{D}_p|} \sum_{k} o_k \log f_\theta(\tau_{k,1}, \tau_{k,2})$$

or

$$\min_{\theta} \mathcal{L}_{\mathrm{DPO}}(\theta) = -\frac{1}{|\mathcal{D}_P|} \sum_{(\tau_{k,1},\tau_{k,2})\in\mathcal{D}_p, o_k=1} \log \sigma\left(\beta \log \frac{\pi_\theta(y_{k,1}|x_k)}{\pi_{\mathrm{ref}}(y_{k,1}|x_k)} - \beta \log \frac{\pi_\theta(y_{k,2}|x_k)}{\pi_{\mathrm{ref}}(y_{k,2}|x_k)}\right).$$

DPO pipeline



Intuition: what is DPO really doing?

Classification view

Given two responses to the same prompt, DPO trains the policy so that the preferred response gets a larger **relative log-probability advantage** over the dispreferred one.

RL view

DPO is solving the same KL-regularized reward maximization problem, but the reward is *eliminated analytically* instead of learned explicitly and optimized by RL.

Practical consequence

No separate reward model, no critic, no PPO roll-outs inside the fine-tuning loop.

DPO versus reward-model + PPO

Aspect	RM + PPO	DPO
Optimization style	online RL	offline supervised loss
Reward representation	explicit $\hat{r}_\phi$	implicit via log-ratio
Sampling during update	usually yes	not required in-loop
Variance / tuning	often higher	often simpler
Expressiveness	very flexible	deterministic/bandits

But not a free lunch

DPO is simpler, but it inherits assumptions from the KL-regularized preference model. DPO is best seen as a **direct alignment objective** derived from a specific regularized RLHF model (bandits or deterministic transitions).

Reference

- R. Goffin, J. Olson. Is it all relative? Comparative judgments and the possible improvement of self-ratings and ratings of others. Perspect Psychol Sci. 2011.
- P. F. Christiano, *et al.* Deep reinforcement learning from human preferences. NeurIPS, 2017.
- L. Ouyang, *et al.* Training language models to follow instructions with human feedback. NeurIPS, 2022.
- R. Rafailov, *et al.* Direct preference optimization: your language model is secretly a reward model. NeurIPS, 2023.
- R. Rafailov, *et al.* From r to Q*: your language model is secretly a Q-function. COLM, 2024.