Natural Language Processing: RLHF

HSE Faculty of Computer Science

Machine Learning and Data-Intensive Systems

Murat Khazhgeriev

Table of Content

- Intro to RL
- RLHF pipeline
- PPO
- DPO

Table of Content

- **Intro to RL**
- RLHF pipeline
- PPO
- DPO

An Agent acts upon an Environment, changes the State and gets a Reward

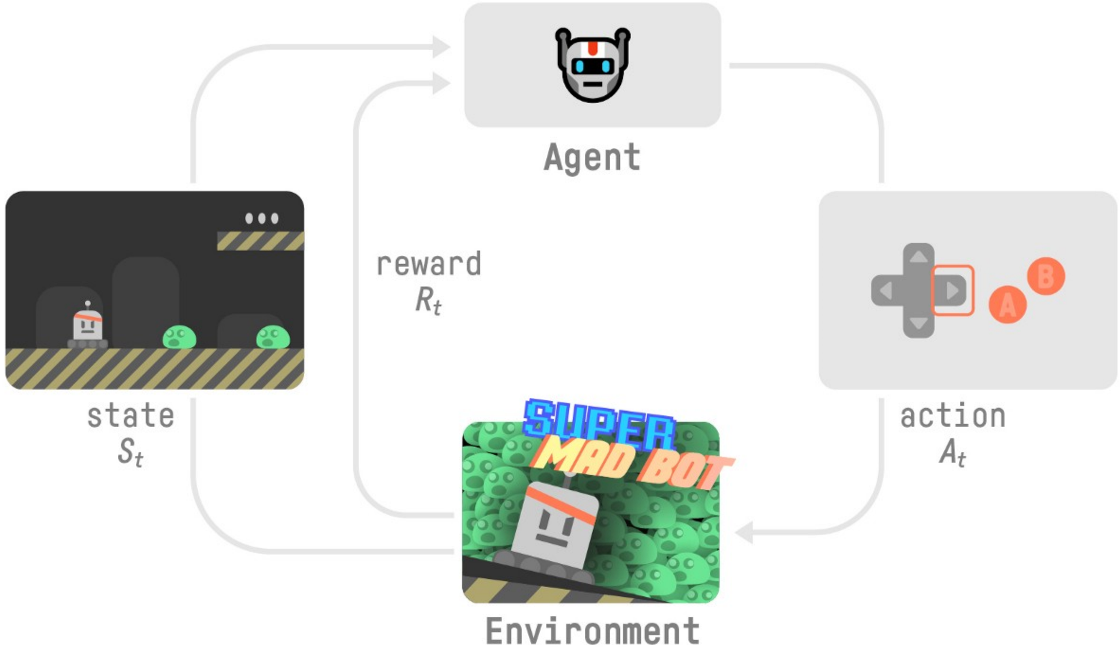
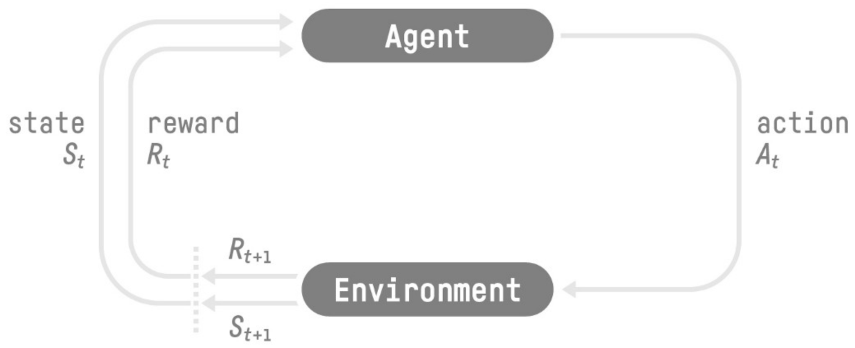


Image source: https://huggingface.co/learn/deep-rl-course/en/unit1/what-is-rl

An Agent acts upon an Environment, changes the State and gets a Reward



Where to learn more?

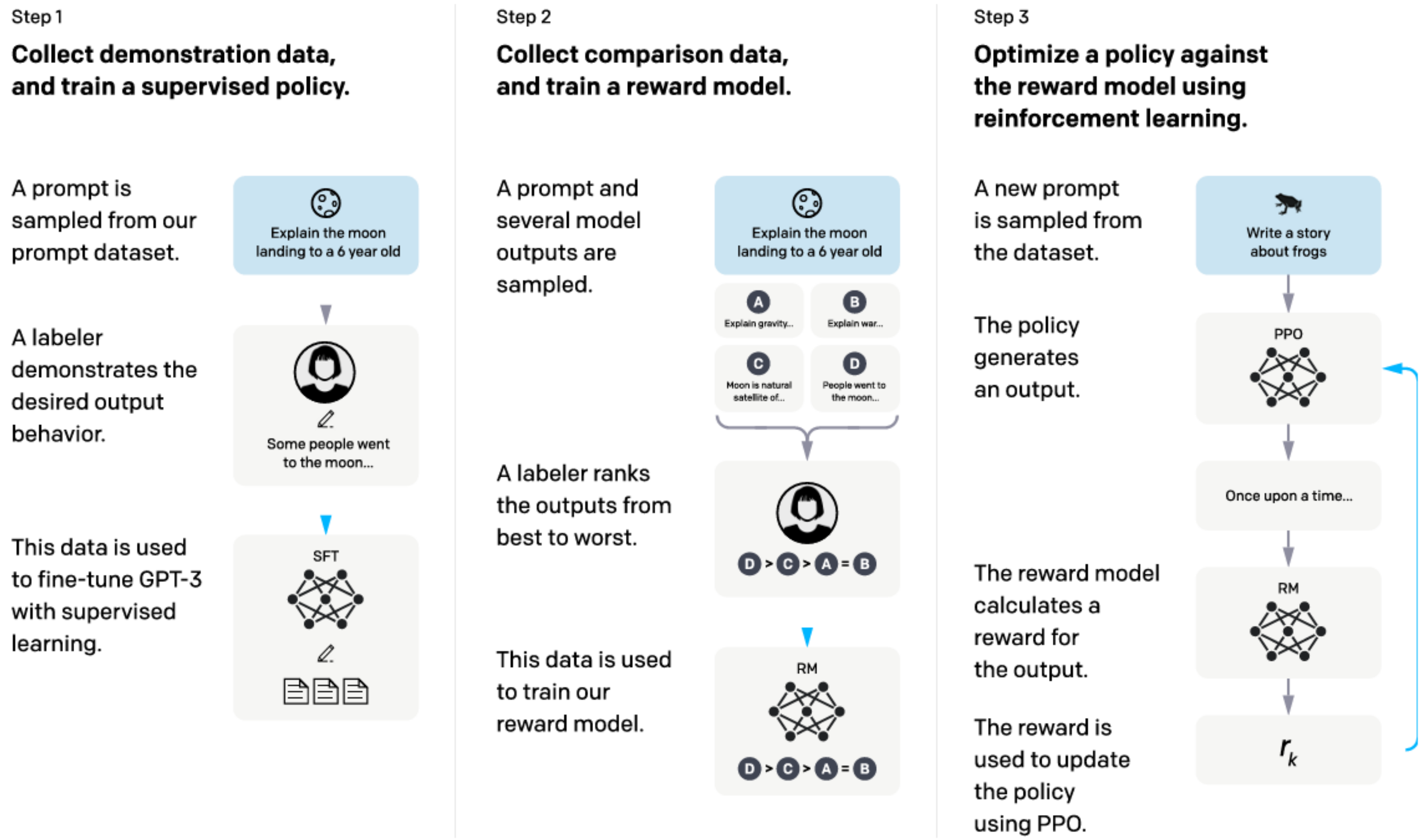
- [Deep RL Course by HF]
- [YSDA Practical RL]

Image source: https://huggingface.co/learn/deep-rl-course/en/unit1/what-is-rl

Table of Content

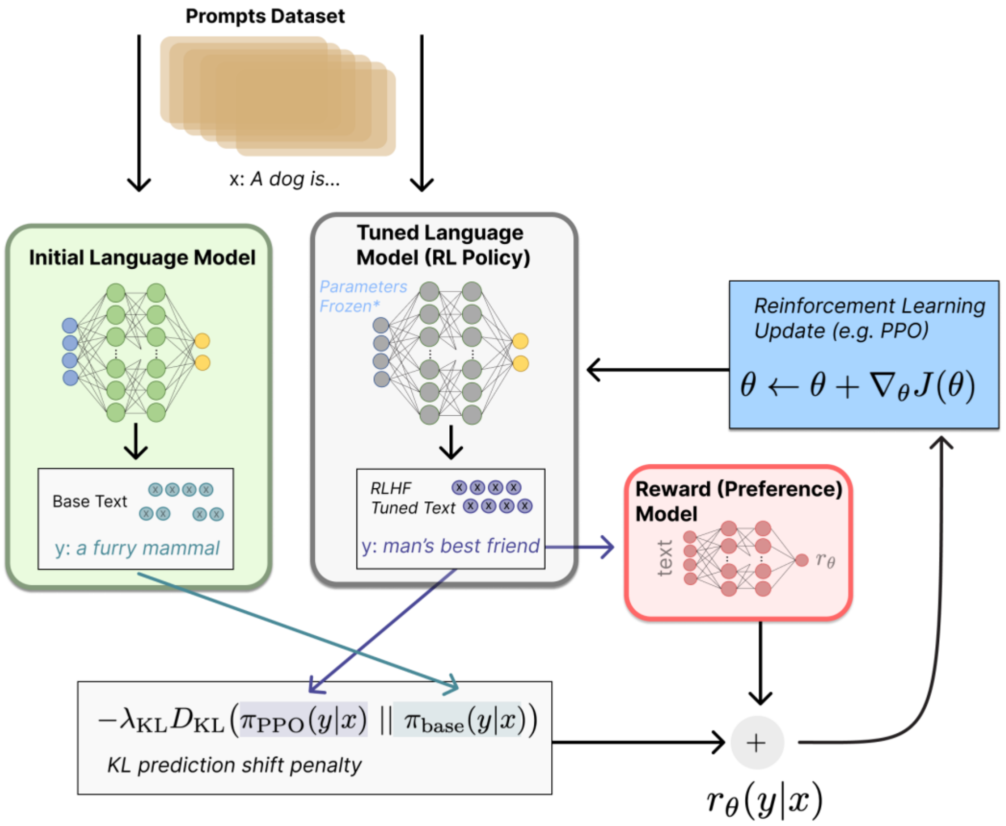
- Intro to RL
- **RLHF pipeline**
- PPO
- DPO

RLHF consists of 3 main steps: SFT, RM, RL



Source: https://arxiv.org/abs/2203.02155

RLHF consists of 3 main steps: SFT, RM, RL

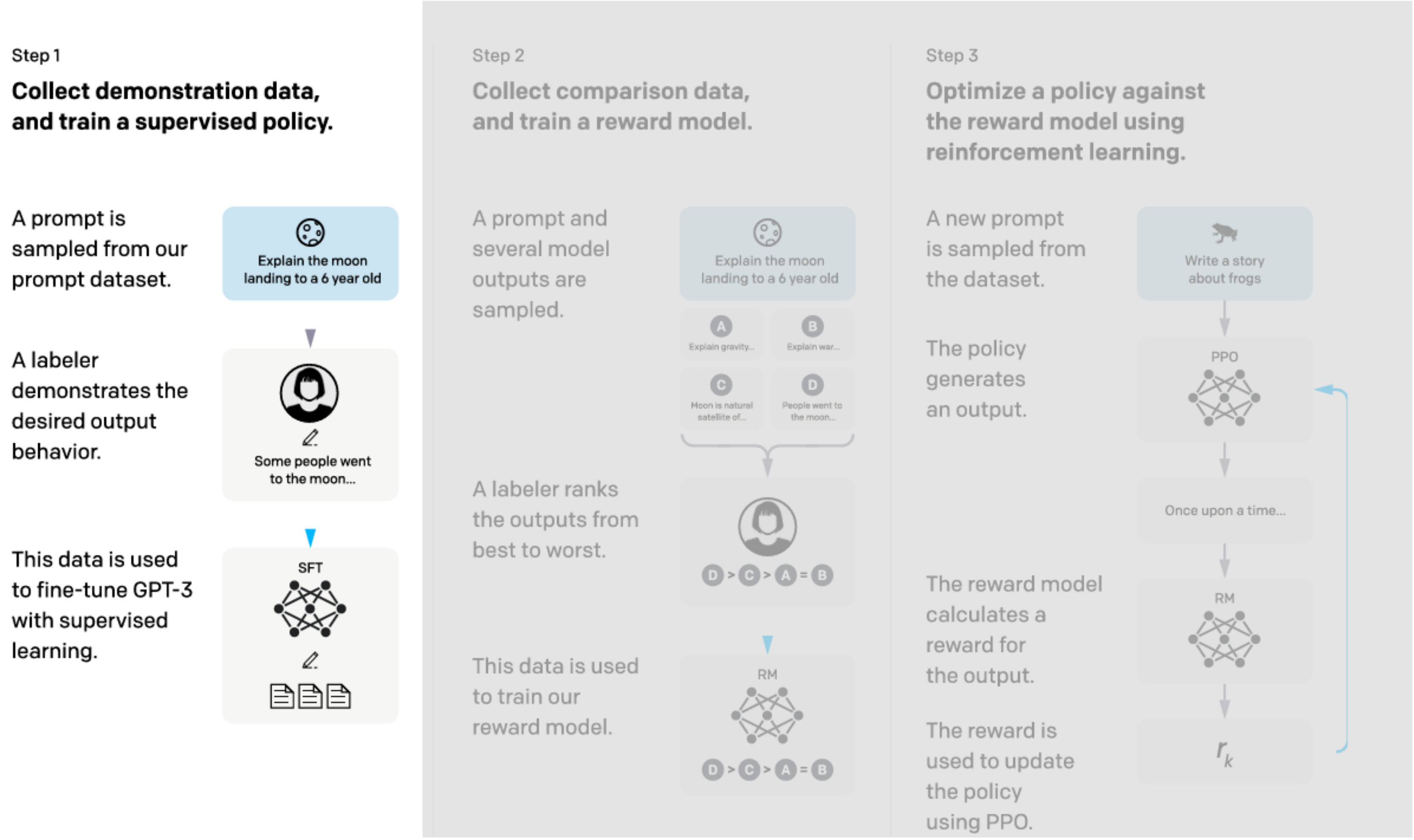


Source: https://huggingface.co/blog/rlhf

Table of Content

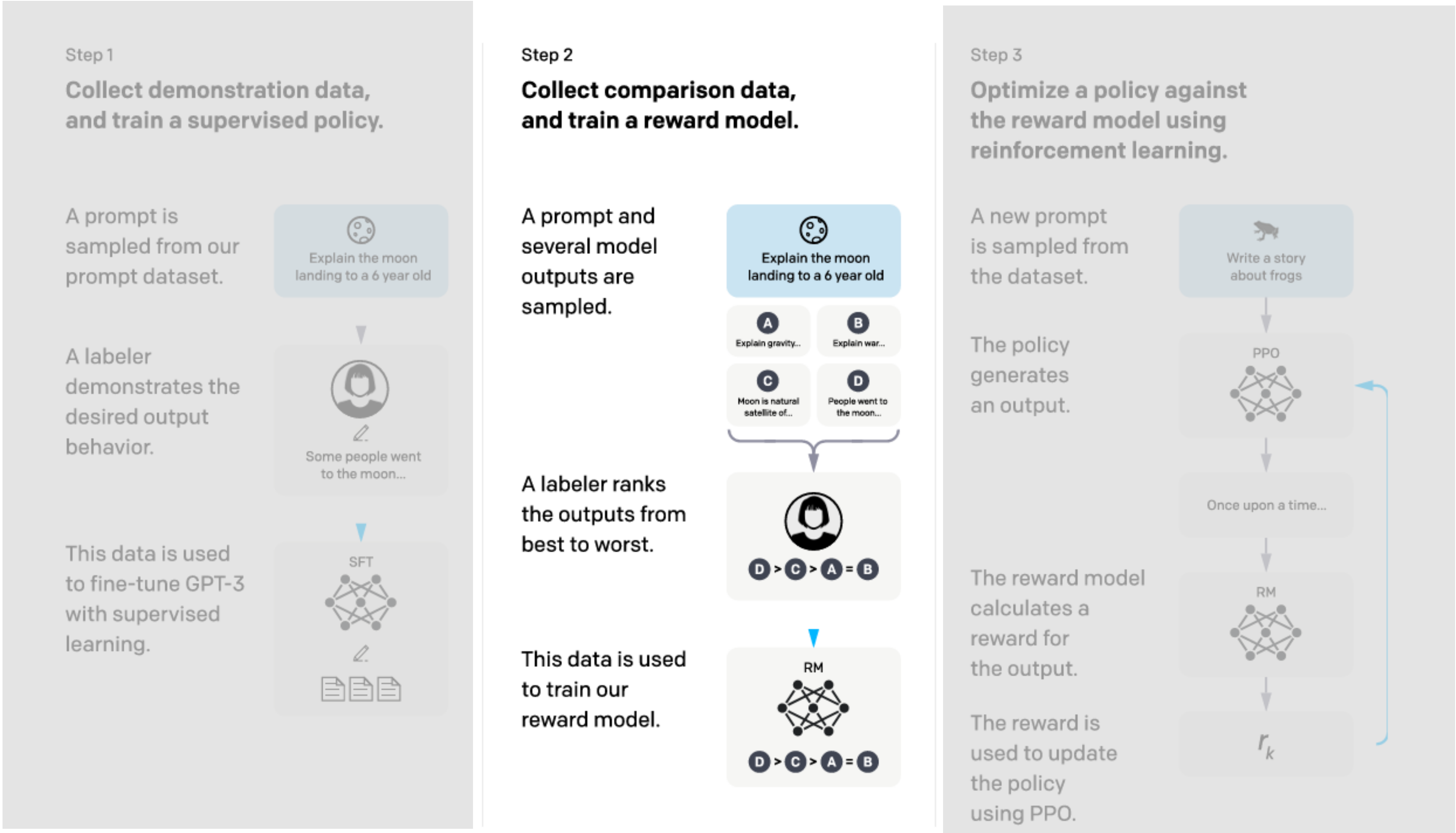
- Intro to RL
- RLHF pipeline
- **PPO**
- DPO

Step 1: Train an initial model with Supervised Finetuning



Source: https://arxiv.org/abs/2203.02155

Step 2: Train a Reward Model



Source: https://arxiv.org/abs/2203.02155

Reward Model learns to differentiate between winner's and loser's scores

$$\text{loss}\left(\theta\right) = -\frac{1}{\binom{K}{2}} E_{(x, y_w, y_l) \sim D} \left[\log\left(\sigma\left(r_\theta\left(x, y_w\right) - r_\theta\left(x, y_l\right)\right)\right)\right]$$

Source: https://arxiv.org/abs/2203.02155

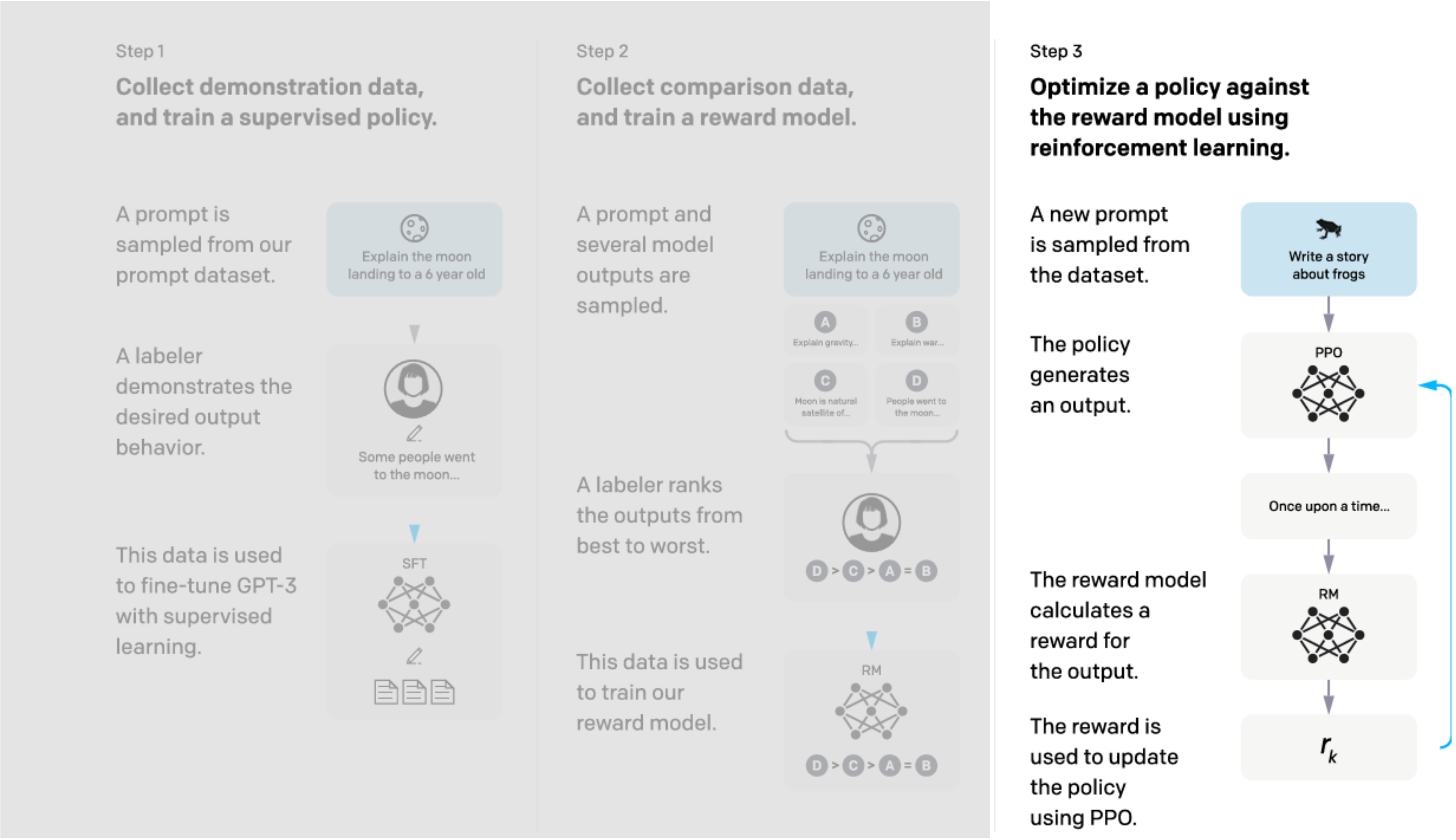
Reward Model learns to differentiate between winner's and loser's scores

$$\text{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\theta\left(x,y_w\right)-r_\theta\left(x,y_l\right)\right)\right)\right]$$

…see the whiteboard for more!

Source: https://arxiv.org/abs/2203.02155

Step 3: Introduce Reinforcement Learning to the problem



Source: https://arxiv.org/abs/2203.02155

Leverage a vanilla Proximal Policy Optimization setup

$$\text{objective}\,(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_\theta(x,y) - \beta \log\left(\pi_\phi^{\mathrm{RL}}(y \mid x)/\pi^{\mathrm{SFT}}(y \mid x)\right)\right]$$

Source: https://arxiv.org/abs/2203.02155

Add regularization via pretrain tasks to prevent deterioration

$$\text{objective}(\phi) = E_{(x,y)\sim D_{\pi_\phi^{\mathrm{RL}}}}\left[r_\theta(x,y) - \beta \log\left(\pi_\phi^{\mathrm{RL}}(y \mid x)/\pi^{\mathrm{SFT}}(y \mid x)\right)\right] + \gamma E_{x\sim D_{\text{pretrain}}}\left[\log(\pi_\phi^{\mathrm{RL}}(x))\right]$$

…see the whiteboard for more!

Source: https://arxiv.org/abs/2203.02155

Table of Content

- Intro to RL
- RLHF pipeline
- PPO
- **DPO**

Your Language Model is Secretly a Reward Model

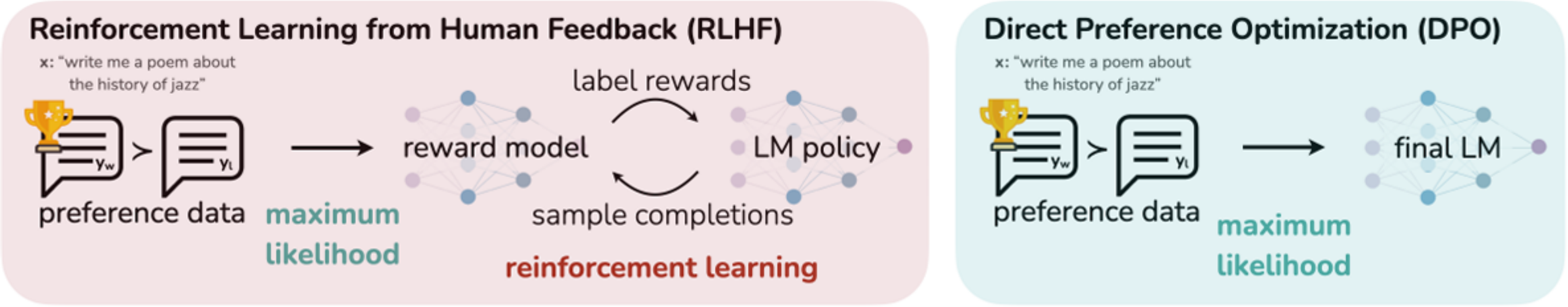


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, fitting an *implicit* reward model whose corresponding optimal policy can be extracted in closed form.

Source: https://arxiv.org/abs/2305.18290

Your Language Model is Secretly a Reward Model

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]$$

…see the whiteboard for more!

Source: https://arxiv.org/abs/2305.18290

Your Language Model is Secretly a Reward Model

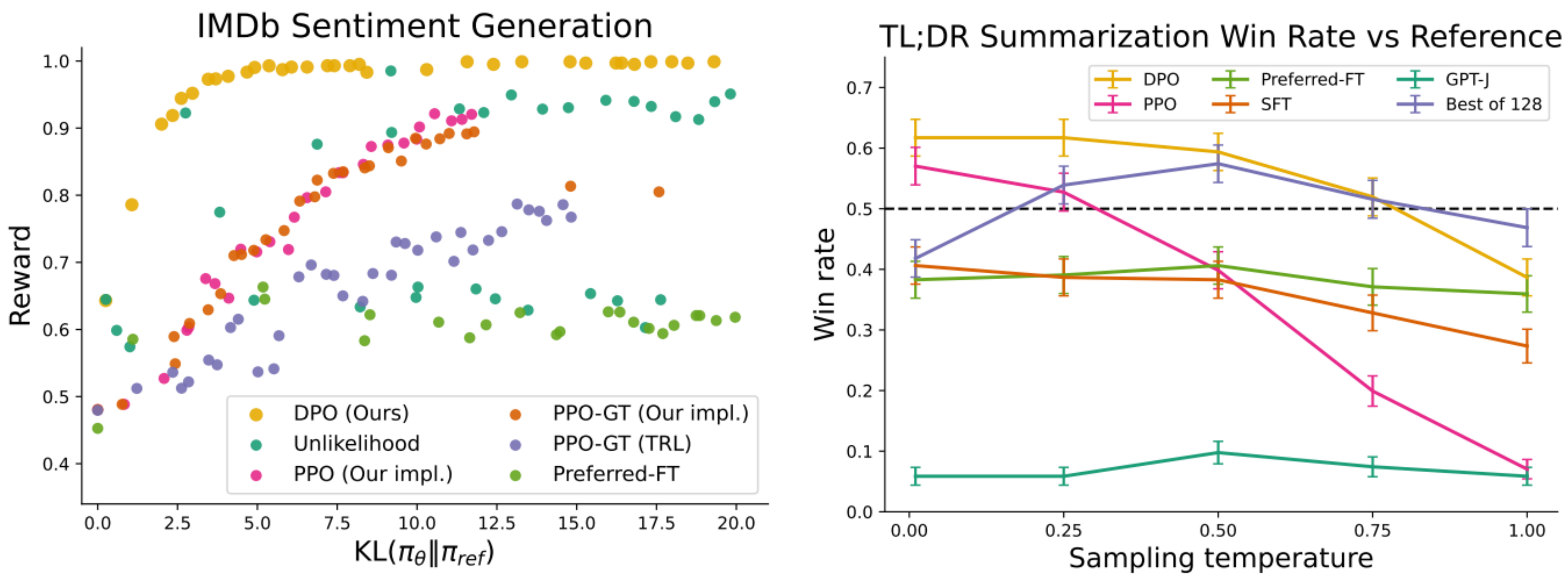


Figure 2: **Left.** The frontier of expected reward vs KL to the reference policy. DPO provides the highest expected reward for all KL values, demonstrating the quality of the optimization. **Right.** TL;DR summarization win rates vs. human-written summaries, using GPT-4 as evaluator. DPO exceeds PPO's best-case performance on summarization, while being more robust to changes in the sampling temperature.

Source: https://arxiv.org/abs/2305.18290

HSE UNIVERSITY • EST 1992 •