Aligning with Human Preferences: Methods and Challenges

Thanh H. Nguyen

Assistant Professor, University of Oregon

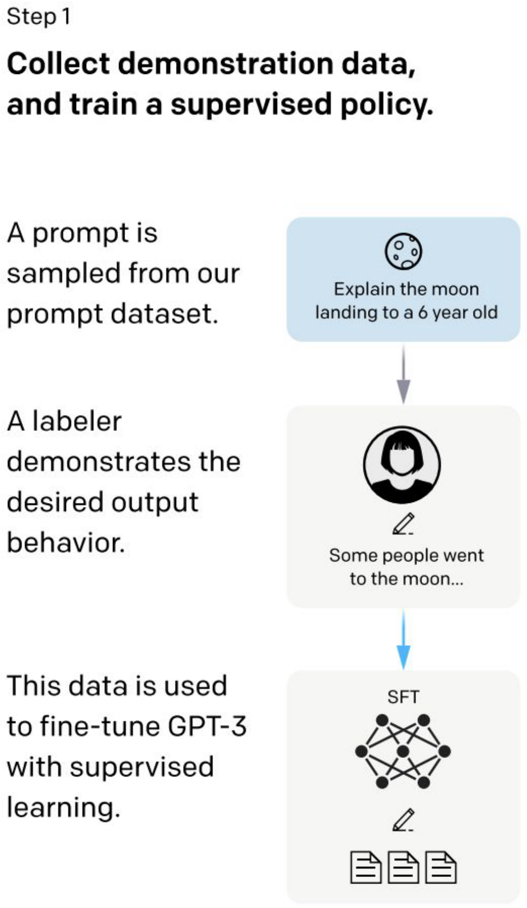
Overview

- Large language models (e.g., GPT):
  - Pre-trained on a vast textual corpus to predict subsequent tokens.
  - Equip LLMs with world knowledge
  - Facilitate the generation of coherent and influent text in response to various input
- Limitations
  - Not always adept at interpreting a wide range of instructions
  - Can produce biased/toxic content or invent facts
- Recent research:
  - Empowering LLM to understand instructions and align with human expectations

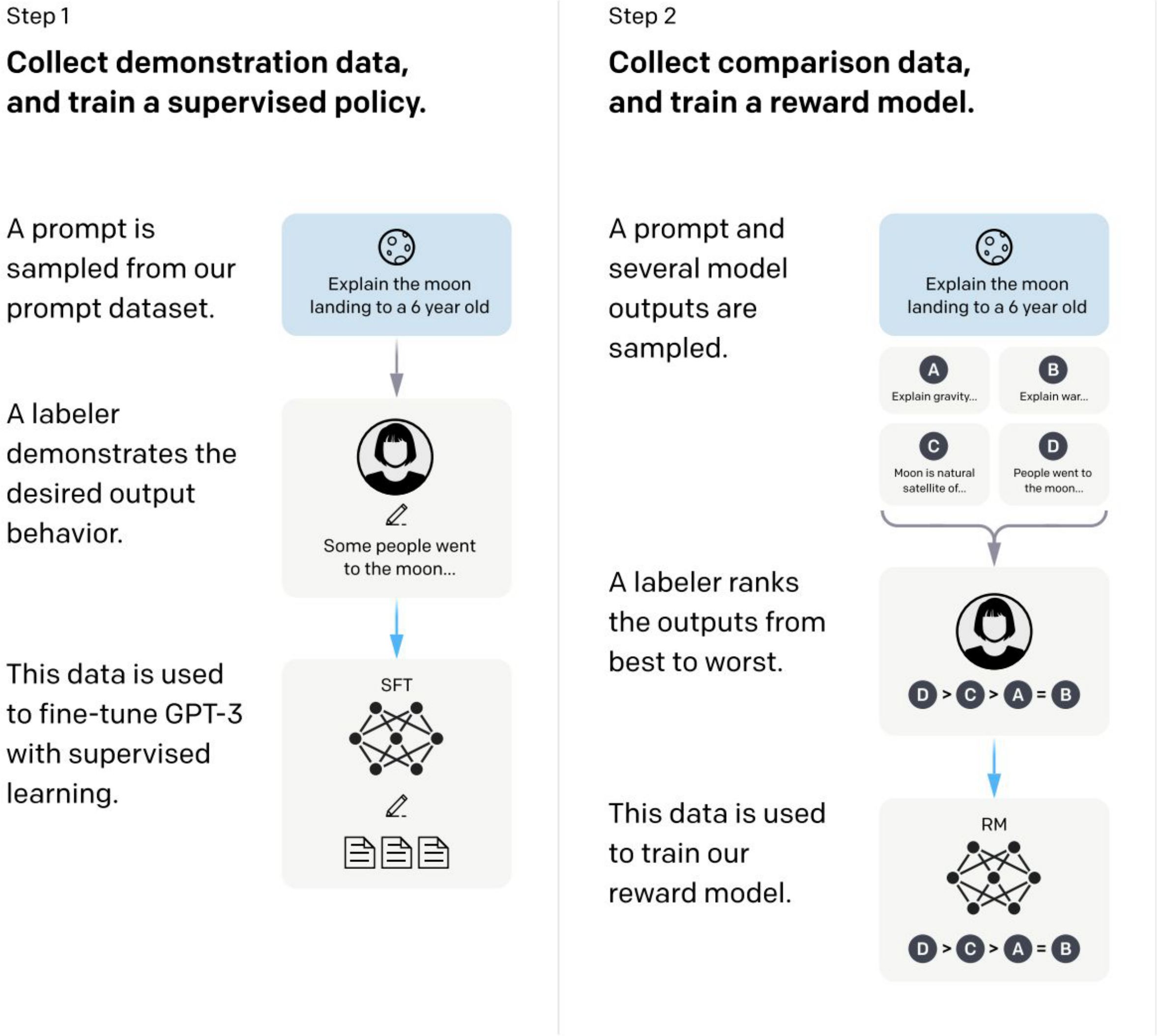
Reinforcement Learning From Human Feedback

Ouyang et al. "Training language models to follow instructions with human feedback." *NeurIPS 2022.*

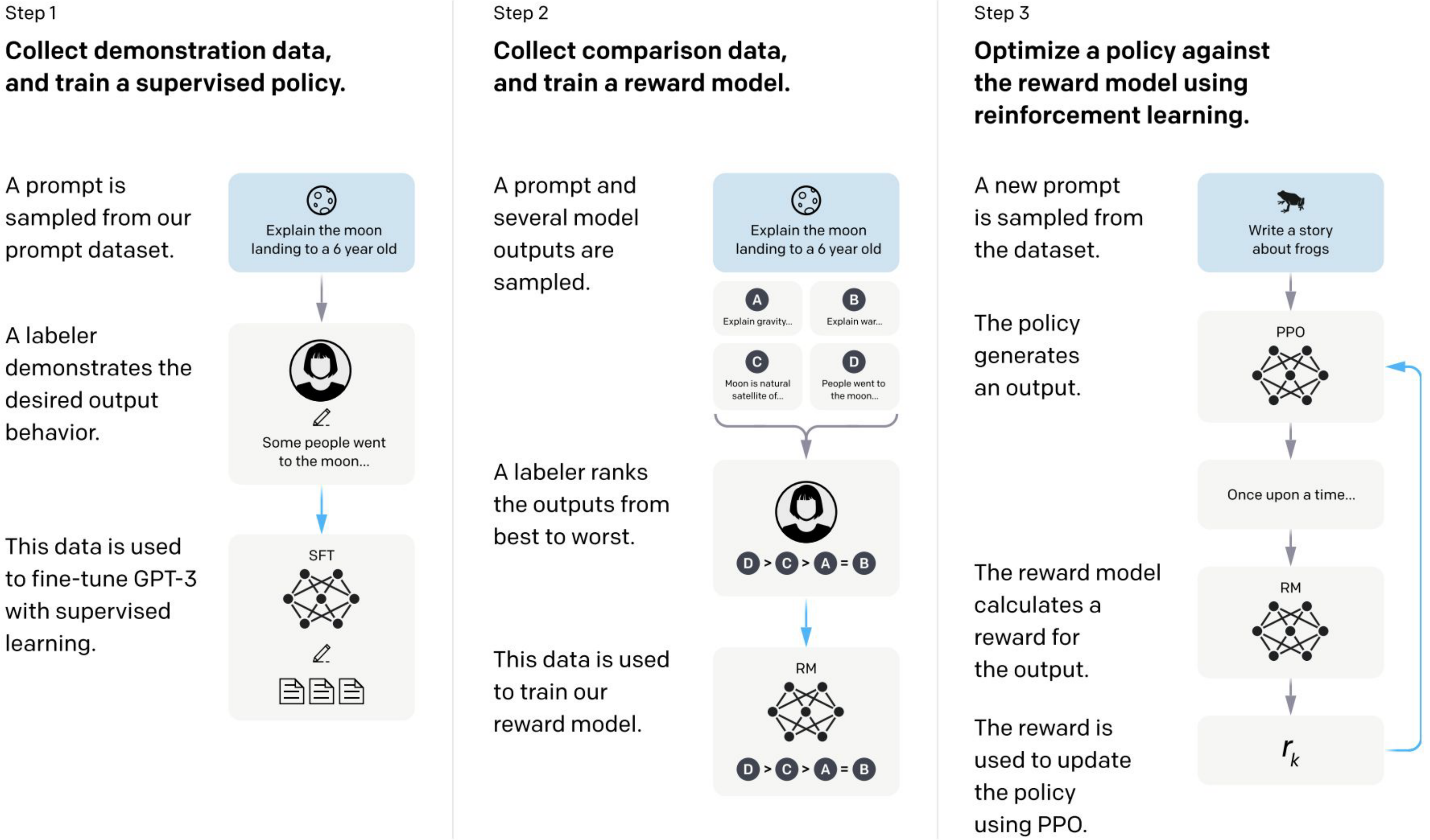
RLHF: Overview



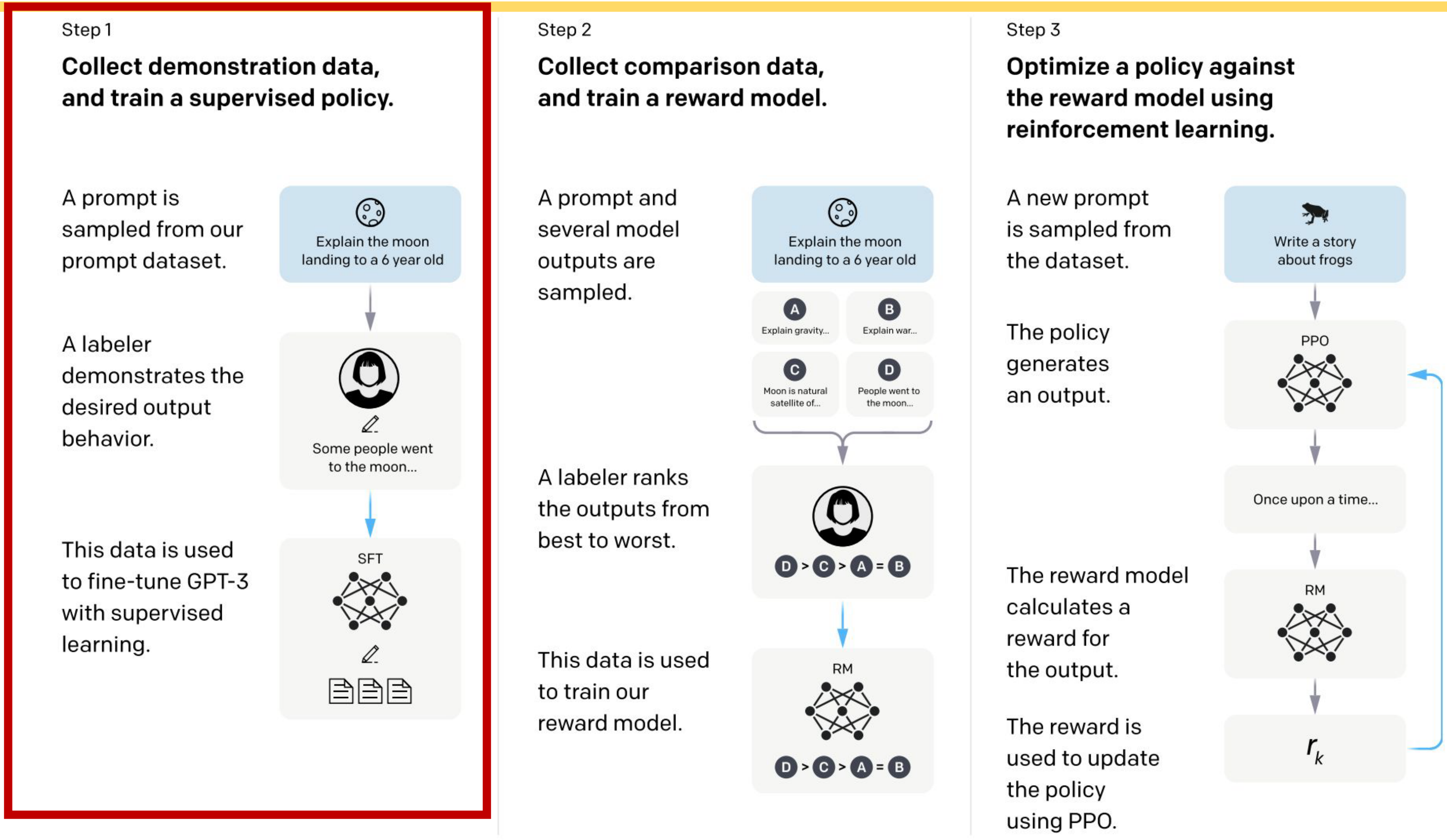
RLHF: Overview



RLHF: Overview



RLHF: Train a Supervised Policy from Demonstration Data



RLHF: Train a Supervised Policy from Demonstration Data

- Firstly, hiring a team of 40 contractors to label data, based on their performance on a screening test.
- Then collecting a dataset of human-written demonstrations of the desired output behavior on (mostly English) prompts submitted to the OpenAI API and some labeler-written prompts,
- Use this dataset to train their supervised learning baselines.

Supervised Fine-tuning

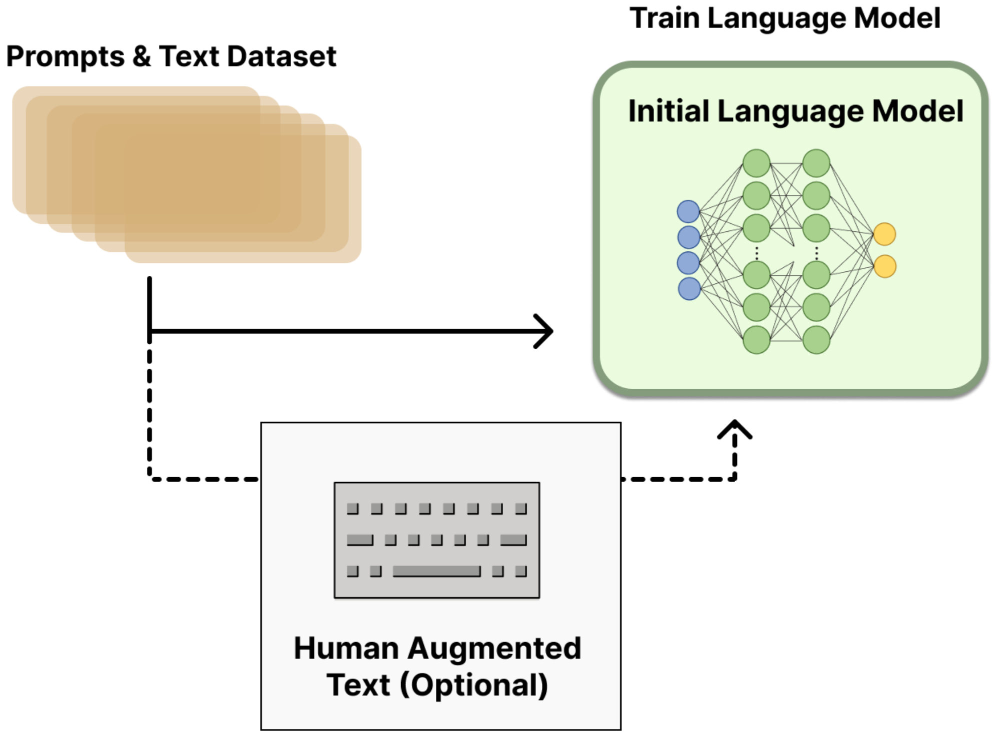
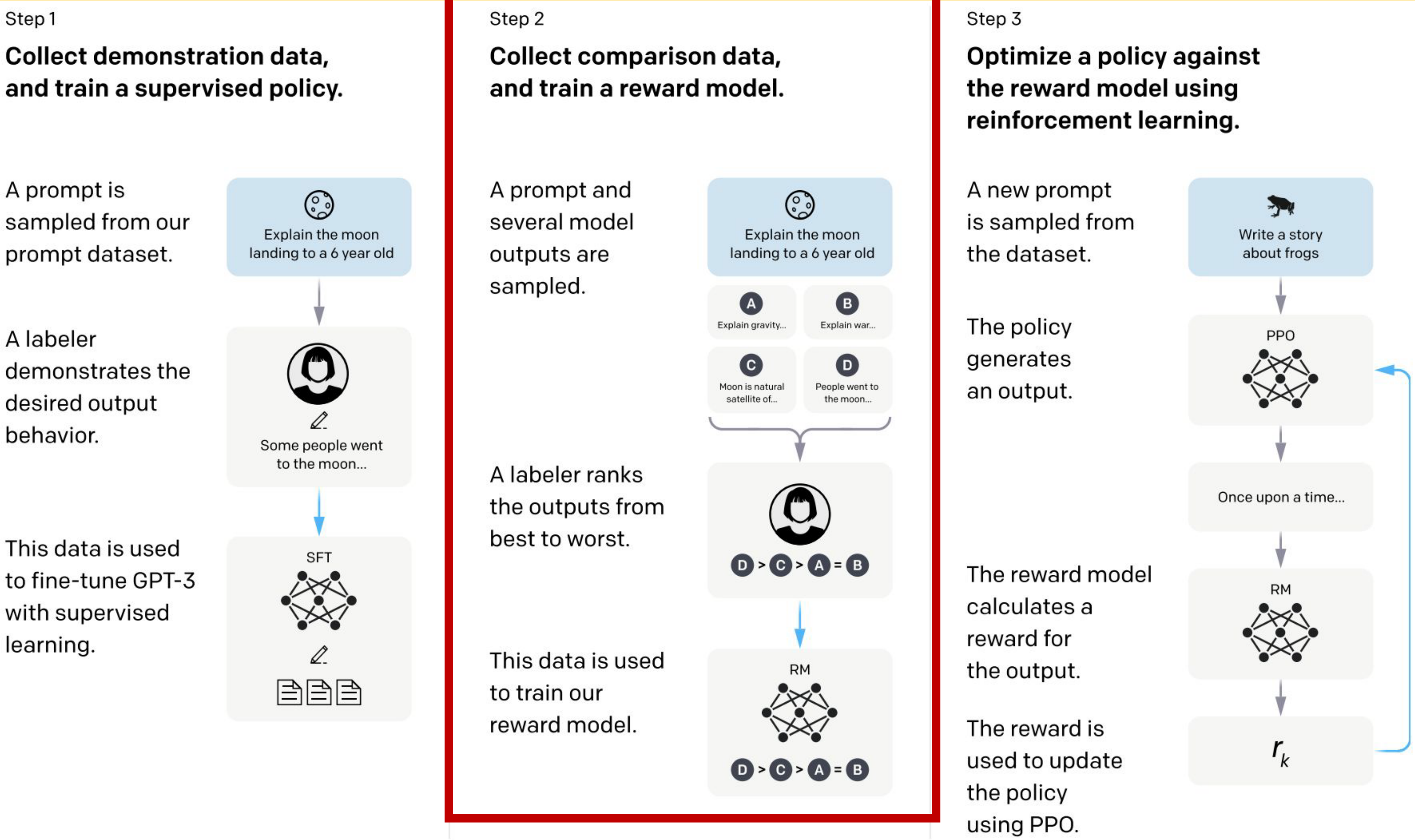


Image source: HuggingFace

RLHF: Learning a Reward Model from Human Feedback



RLHF: Learning a Reward Model from Human Feedback

- Collect a dataset of human-labeled comparisons between outputs from OpenAI's models on a larger set of API prompts.

- And then train a reward model (RM) on this dataset to predict which model output their labelers would prefer.

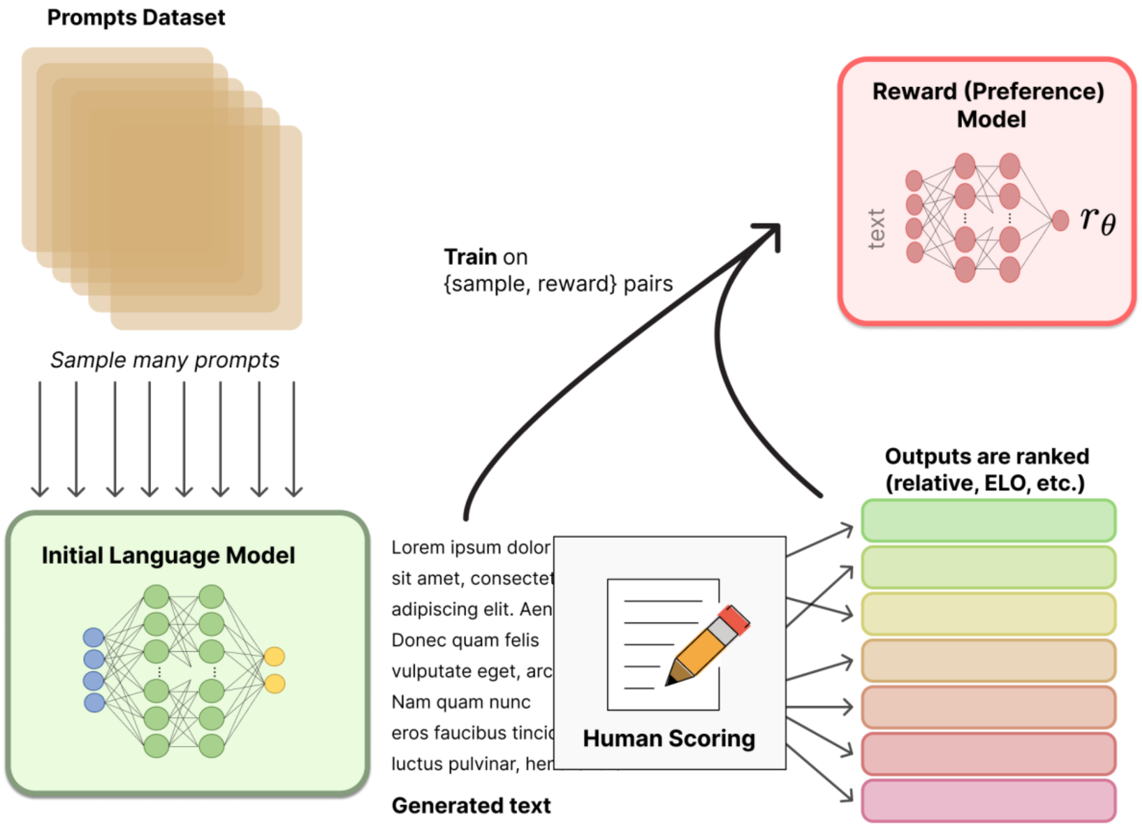


Image source: HuggingFace

RLHF: Learning a Reward Model from Human Feedback

- Feedback comes as preferences over model samples:

$$\mathcal{D} = \{\boxed{x^i}, \boxed{y_w^i}, \boxed{y_l^i}\}$$

Prompt

Preferred response

Dis-preferred response

- Bradley-Terry model connects rewards to preferences

Sigmoid function

$$p(y_w \succ y_l) = \sigma\big(r(x, y_w) - r(x, y_l)\big) = \frac{\exp[r(x, y_w)]}{\exp[r(x, y_w)] + \exp[r(x, y_l)]}$$

Rewards assigned to preferred and dis-preferred responses

RLHF: Learning a Reward Model from Human Feedback

- Bradley-Terry Model connects rewards to preferences

Sigmoid function

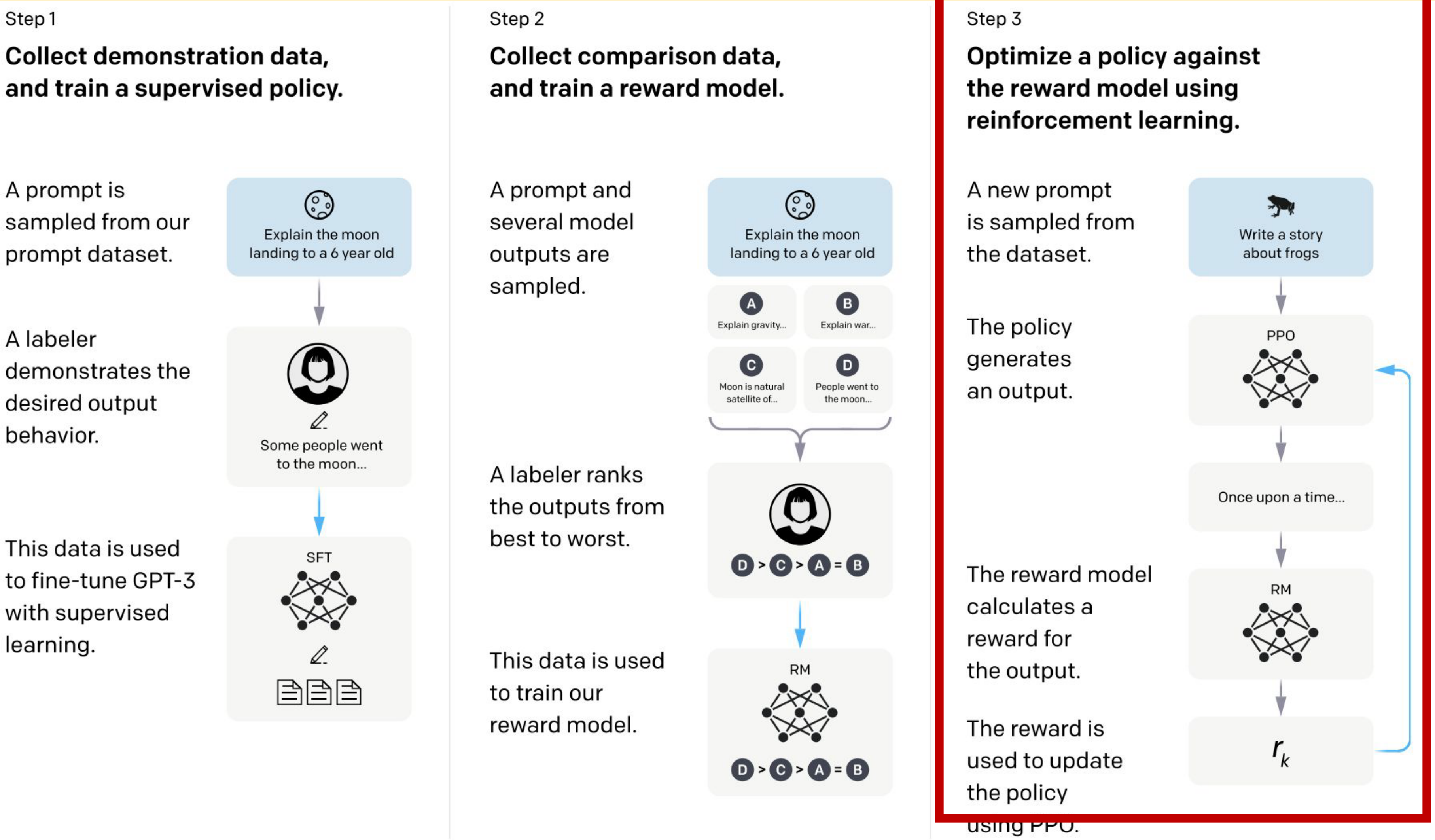
$$p(y_w \succ y_l) = \sigma\big(r(x, y_w) - r(x, y_l)\big) = \frac{\exp[r(x, y_w)]}{\exp[r(x, y_w)] + \exp[r(x, y_l)]}$$

Rewards assigned to preferred and dis-preferred responses

- Train the reward model by minimizing negative log likelihood

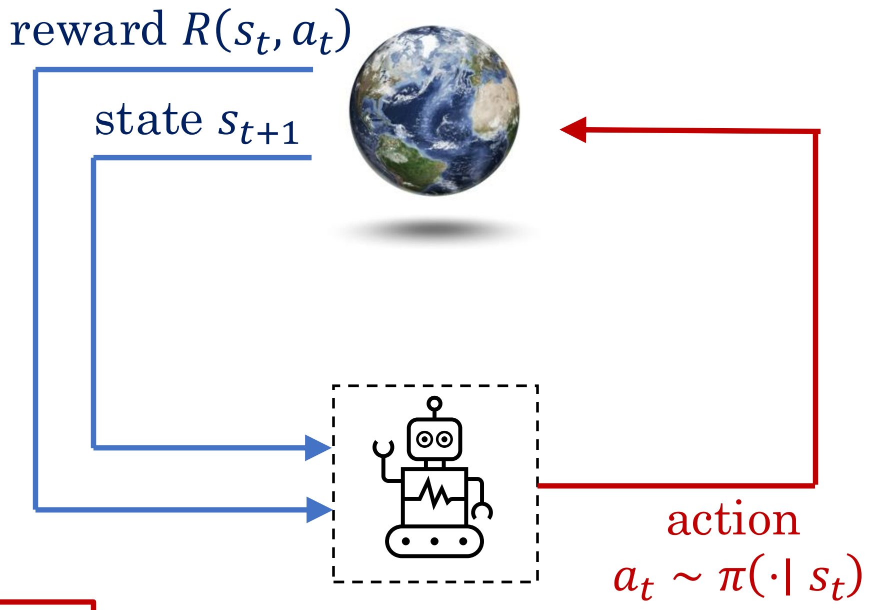
$$\boxed{\mathcal{L}_R(\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[\log \sigma\left(r_\phi(x, y_w) - r_\phi(x, y_l)\right)\right]}$$

RLHF: Learning a Policy that Optimize the Reward



Background on Reinforcement Learning

- MDP setup
  - States: S
  - Actions: A
  - Transitions: $P(s' \mid s, a)$ (unknown)
  - Reward function: $R(s, a)$ (unknown)



- Goal: find an optimal policy $\pi_\theta(\cdot \mid s), \forall s$

$$\boxed{\max_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]}$$

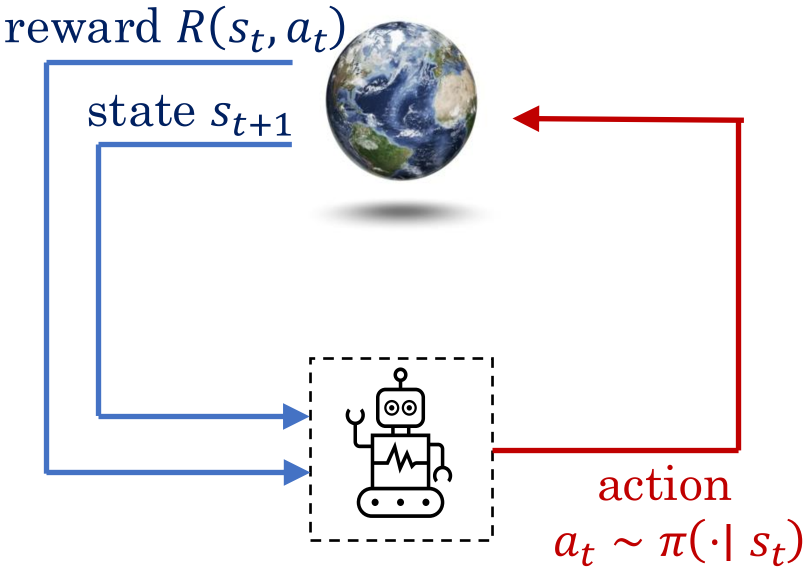
  - Where $\tau = (s_0, a_0, s_1, a_1, \cdots)$

Background on Reinforcement Learning

- Goal: find an optimal policy $\pi_\theta(\cdot \mid s), \forall s$

$$\boxed{\max_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right]}$$

  - Where $\tau = (s_0, a_0, s_1, a_1, \cdots)$



- Some important notions:
  - State-action value function $Q_\pi(s,a) = R(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V_\pi(s')$
  - Value function $V_\pi(s) = \sum_a \pi(a \mid s)\, Q_\pi(s,a)$
  - Advantage function: $A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s)$

Background on Reinforcement Learning

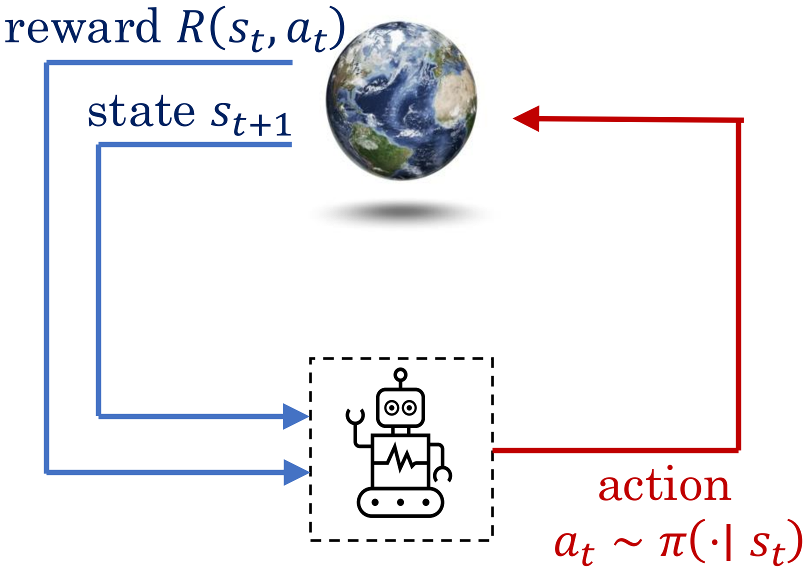
- Goal: find an optimal policy $\pi_\theta(\cdot \mid s), \forall s$

$$\boxed{\max_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]}$$

  - Where $\tau = (s_0, a_0, s_1, a_1, \cdots)$
- Policy gradient theorem

$$\boxed{\nabla J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla \ln \pi_\theta(a \mid s)\, Q_{\pi_\theta}(s, a)\right]}$$

- Policy optimization: gradient ascent
  - Challenge: unstable training



TRPO: Trust Region Policy Optimization

- Goal: improve training stability
- Policy optimization's objective:

$$J(\theta) = \mathbb{E}_{s\sim\rho^{\pi_{\theta_{old}}}, a\sim\pi_{\theta_{old}}}\Bigg[\underbrace{\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)}}_{\text{Important sampling}}\ \underbrace{\hat{A}_{old}(s,a)}_{\text{Estimated advantage}}\Bigg]$$

$$\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{x\sim q}\left[f(x)\frac{p(x)}{q(x)}\right]$$

Schulman et al. "Trust region policy optimization." In *ICML*, 2015.

TRPO:
Trust Region Policy Optimization

- Goal: improve training stability
- Policy optimization's objective:

$$J(\theta) = \mathbb{E}_{s\sim\rho^{\pi_{\theta_{old}}},a\sim\pi_{\theta_{old}}}\left[\underbrace{\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)}}_{\text{Important sampling}}\ \underbrace{\hat{A}_{old}(s,a)}_{\text{Estimated advantage}}\right]$$

  - Subject to KL divergence constraint:

$$\mathbb{E}_{s\sim\rho^{\pi_{\theta_{old}}}}\left[\mathbb{D}_{KL}\left(\pi_{\theta_{old}}(\cdot \mid s)\|\pi_\theta(\cdot \mid s)\right)\right] \le \delta$$

PPO: Proximal Policy Optimization

- Goal: simplifying TRPO
- Two primary variants
  - PPO-Penalty: penalty-based approach instead of KL constraints

$$J(\theta) = \mathbb{E}\left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)} \hat{A}_{old}(s,a) - \beta KL\left(\pi_{\theta_{old}}(\cdot \mid s), \pi_\theta(\cdot \mid s)\right)\right]$$

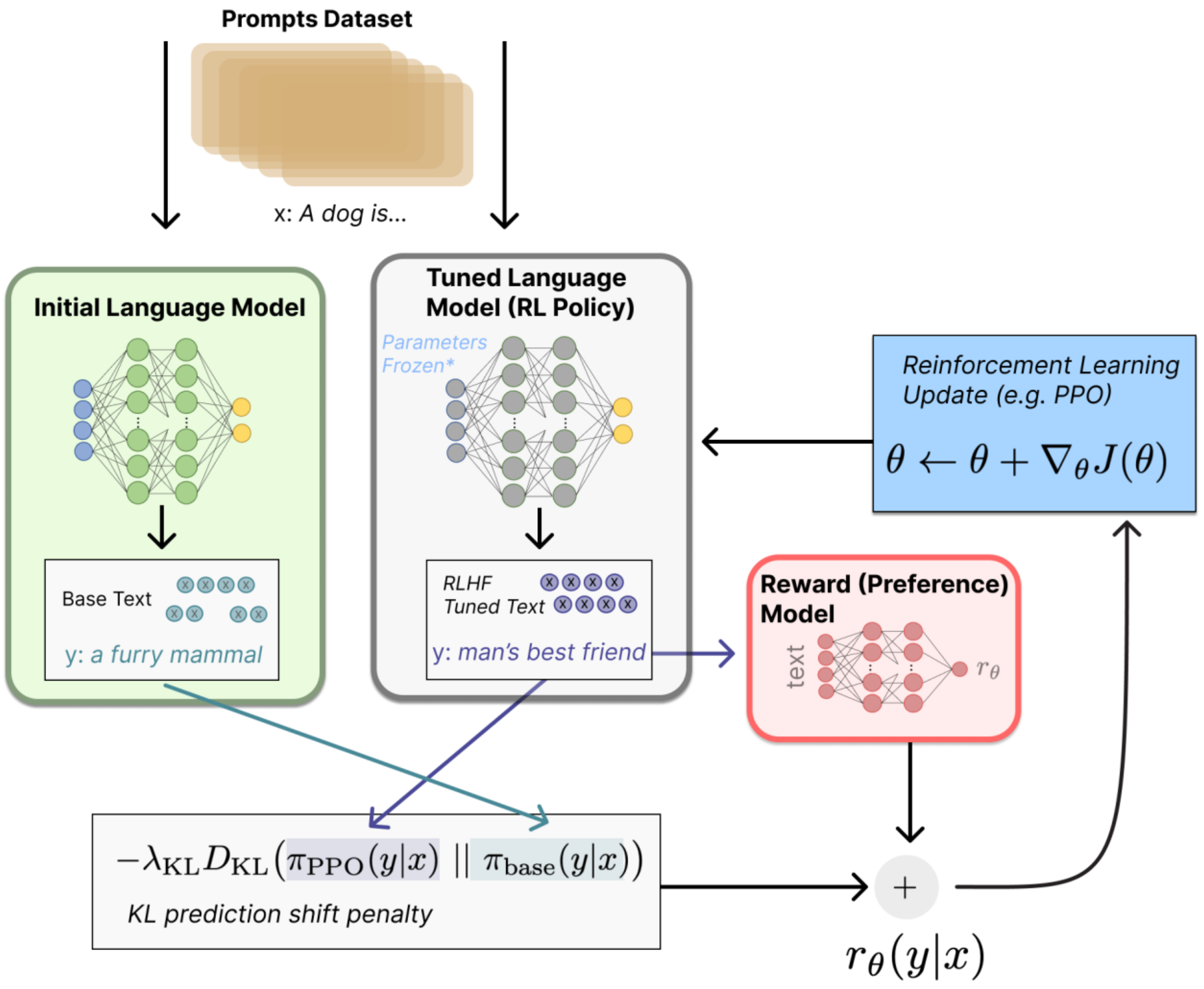
  - PPO-Clipped: simplify objective using clipping function

$$J(\theta) = \mathbb{E}\left[\min\left(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)} \hat{A}_{old}(s,a), clip\left(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_{old}}(a \mid s)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{old}(s,a)\right)\right]$$

Schulman et al. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347* (2017).

RLHF: Learning a Policy that Optimize the Reward

- Use this RM as a reward function and fine-tune supervised learning baseline to maximize this reward using the PPO algorithm.



RLHF: Learning a Policy that Optimize the Reward

- Now we have a reward model r_ϕ that represents goodness according to humans
- Next, learn a policy π_θ achieving a high reward

- Objective

$$\boxed{\max_\theta \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta}\left[r_\phi(x,y)\right]}$$

Sample from policy

Want high rewards

RLHF: Learning a Policy that Optimize the Reward

- Now we have a reward model r_ϕ that represents goodness according to humans
- Next, learn a policy π_θ achieving a high reward while staying close to original model π_{ref}
- Objective

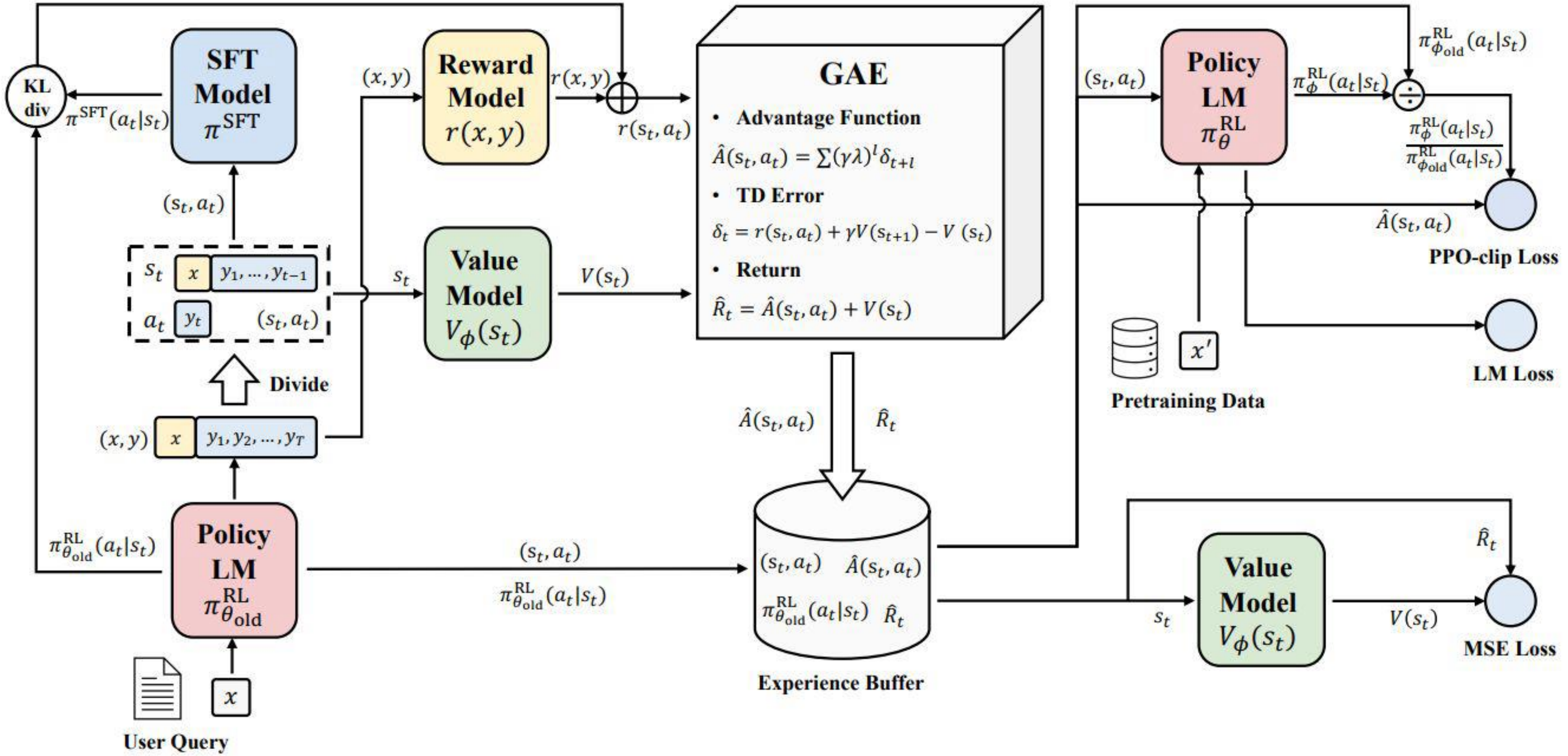
$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[r_\phi(x, y)\right] - \beta\, \mathbb{D}_{KL}\left[\pi_\theta(y \mid x) \| \pi_{ref}(y \mid x)\right]$$

Sample from policy

Want high rewards

But keep KL to original model small

RLHF: Learning a Policy that Optimize the Reward



Zheng et al. "[Secrets of RLHF in large language models part i: Ppo](#)." *arXiv preprint arXiv:2307.04964*(2023).

Evaluation

- API distribution
  - Main metric is human preference ratings on a held out set of prompts from the same source as their training distribution.

- Public NLP datasets
  - They evaluate on two types of public datasets, which are FLAN and TO, both consist of a variety of NLP tasks. Also, conduct human evaluations of toxicity on the RealToxicityPrompts dataset

Table 3: Labeler-collected metadata on the API distribution.

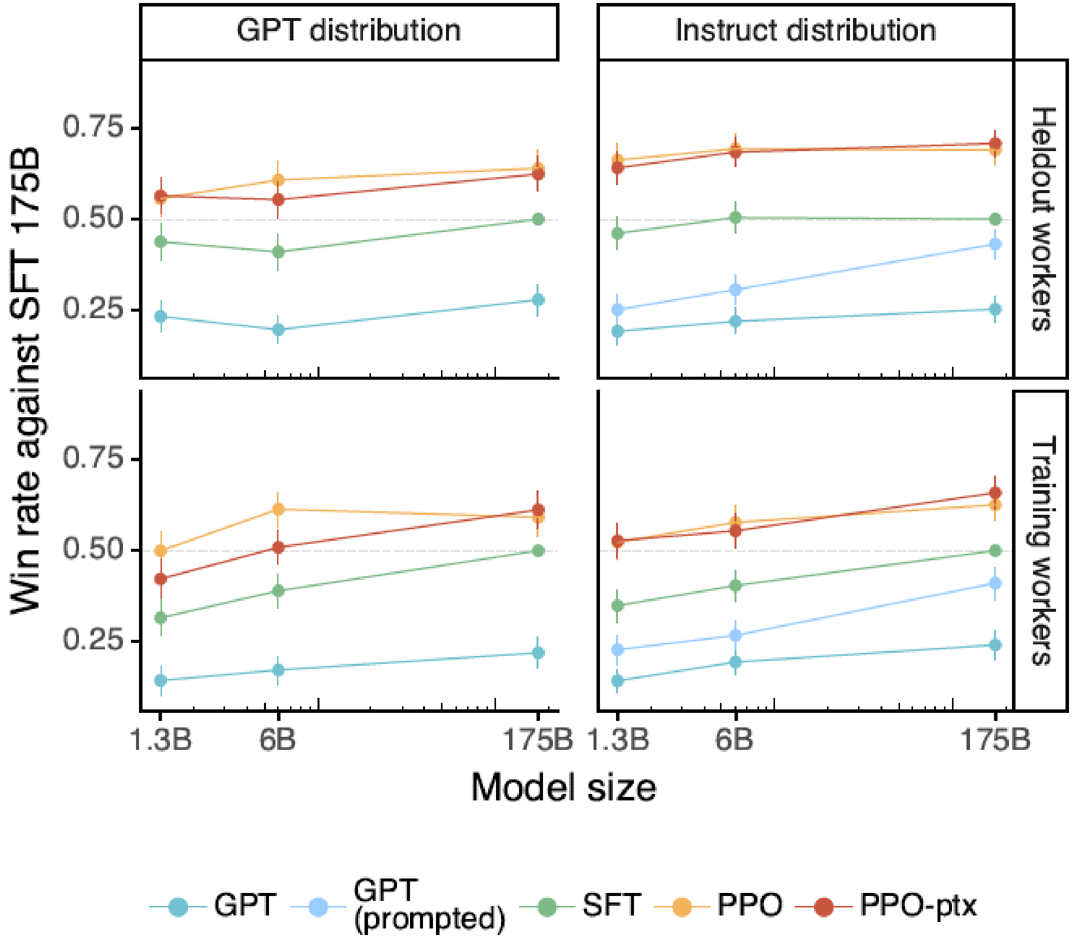
Metadata	Scale
Overall quality	Likert scale; 1-7
Fails to follow the correct instruction / task	Binary
Inappropriate for customer assistant	Binary
Hallucination	Binary
Satisifies constraint provided in the instruction	Binary
Contains sexual content	Binary
Contains violent content	Binary
Encourages or fails to discourage violence/abuse/terrorism/self-harm	Binary
Denigrates a protected class	Binary
Gives harmful advice	Binary
Expresses opinion	Binary
Expresses moral judgment	Binary

Results

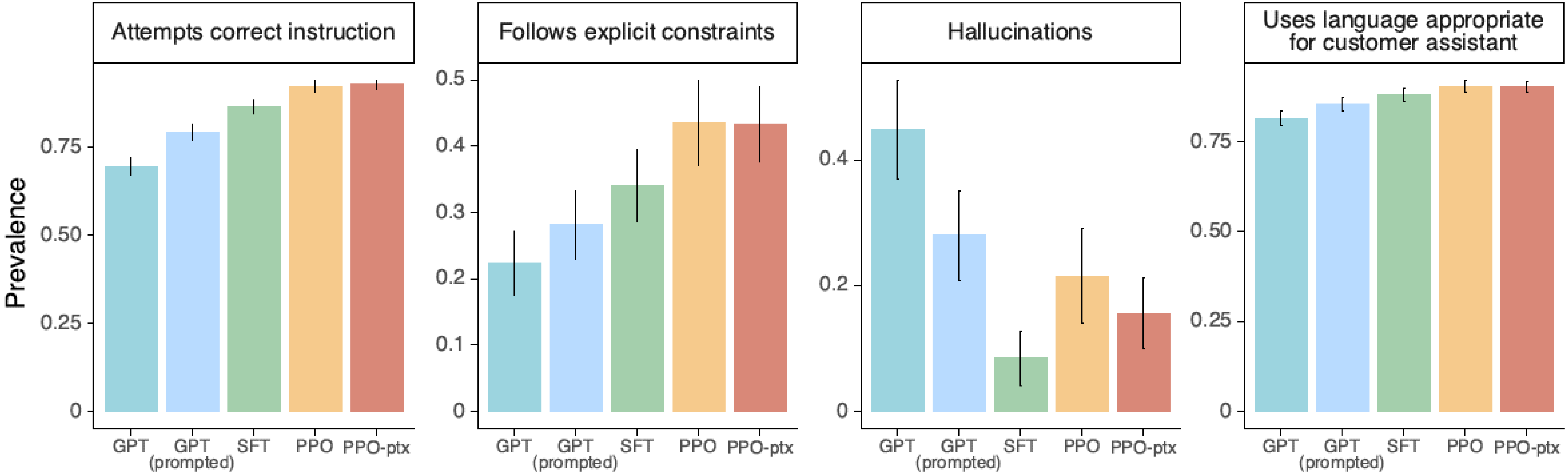
- API distribution
  - Labelers significantly prefer InstructGPT outputs over outputs from GPT-3
  - Generalizing to the preferences of "held-out" labelers
  - Public NLP datasets are not reflective of how their language models are used

- Public NLP datasets
  - Showing improvements in truthfulness over GPT-3
  - Showing small improvements in toxicity over GPT-3, but not bias
  - Minimizing performance regressions on public NLP datasets by modifying their RLHF fine-tuning procedure

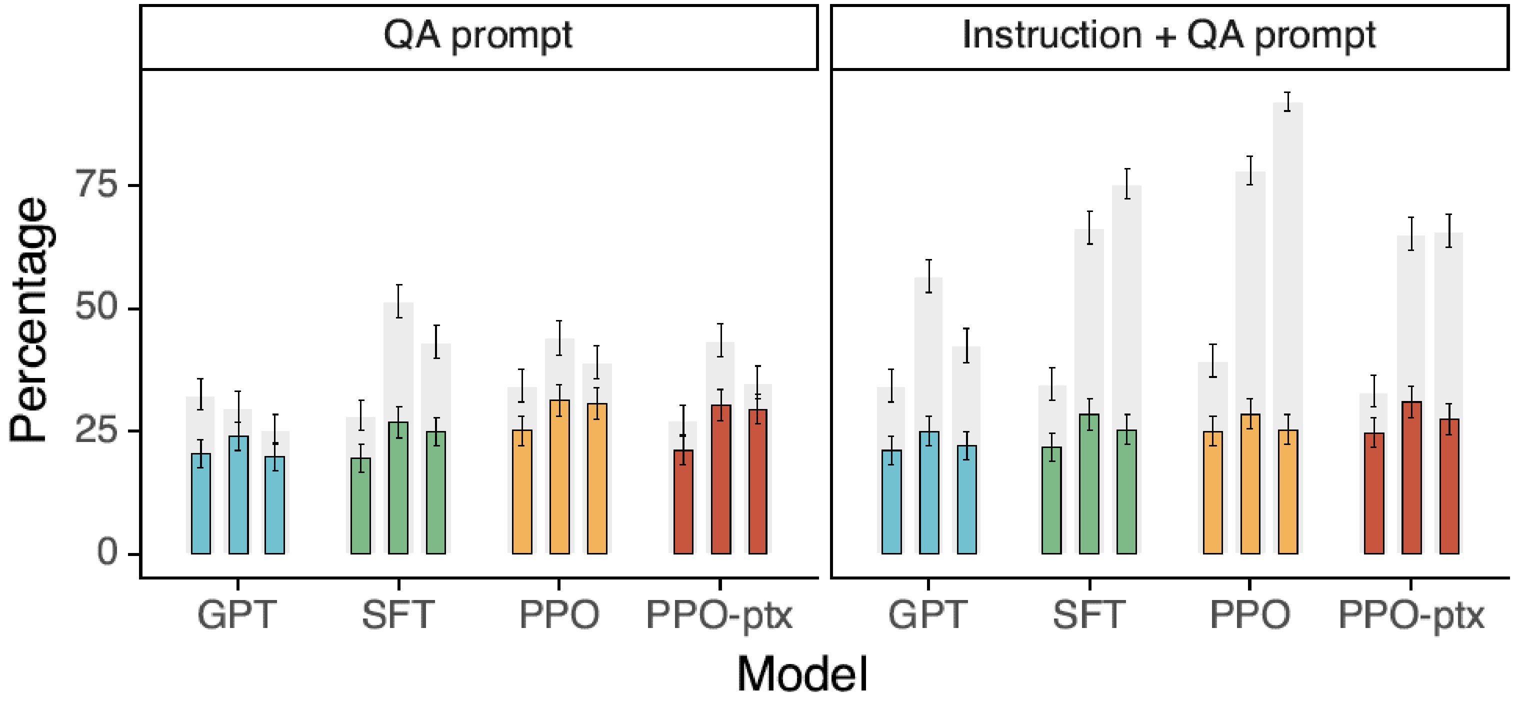
Preference Results



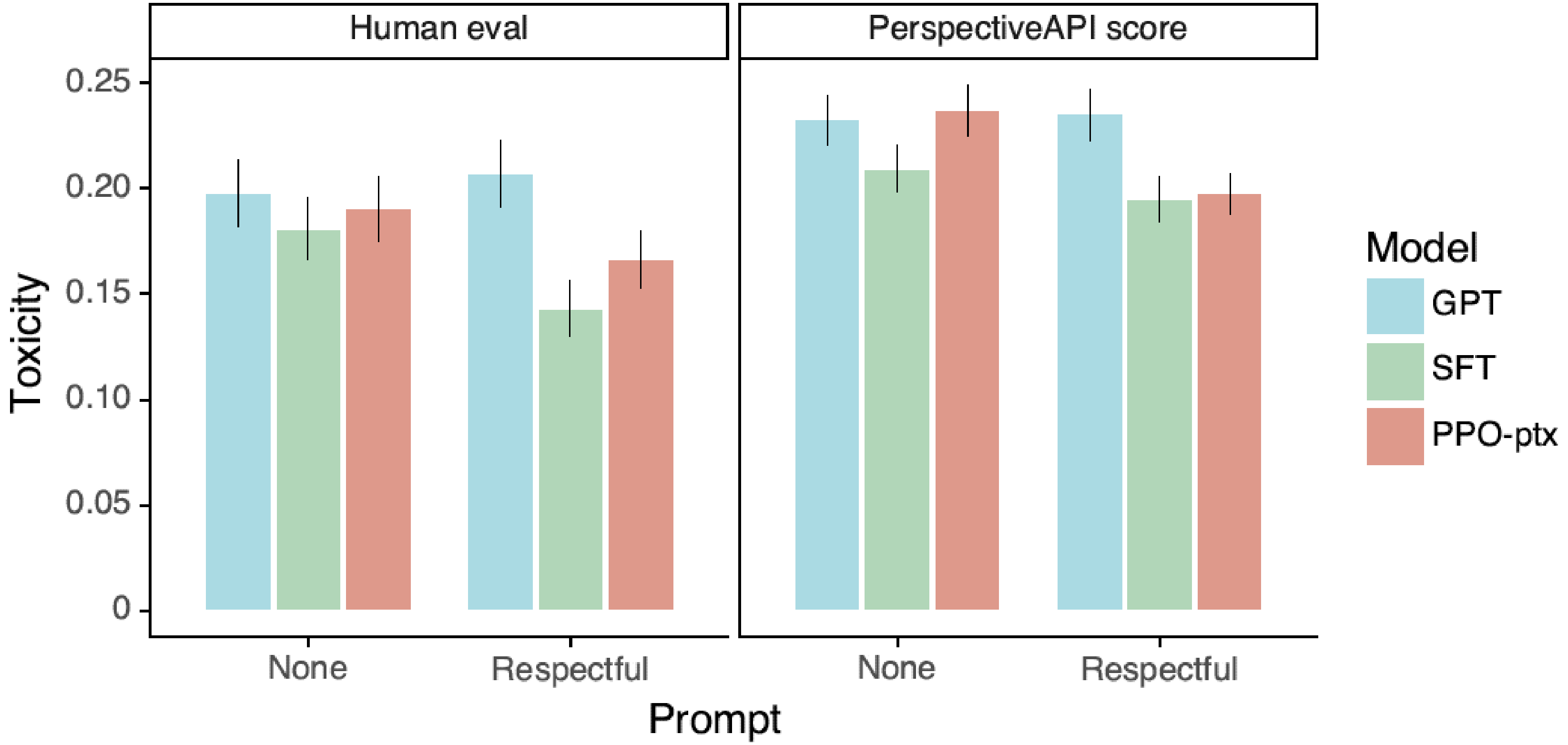
Metadata results on the API Distribution



Results on Truthful Dataset

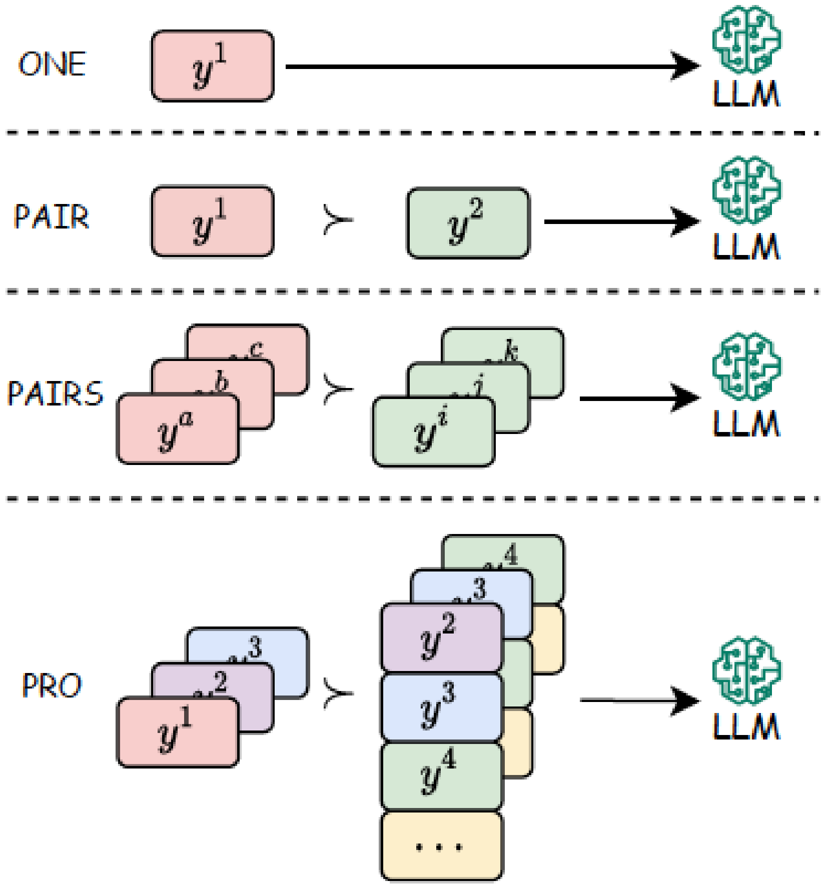


Results on RealToxicityPrompts Dataset



Some Extension of RLHF

Human Alignment: Preference Ranking Optimization



Different Supervised Finetuning Paradigms

Song et al. "Preference ranking optimization for human alignment." In *AAAI*. 2024.

Human Alignment: Preference Ranking Optimization (PRO)

- From RLHF to PRO
  - Convert listwise ranking to pairwise ranking

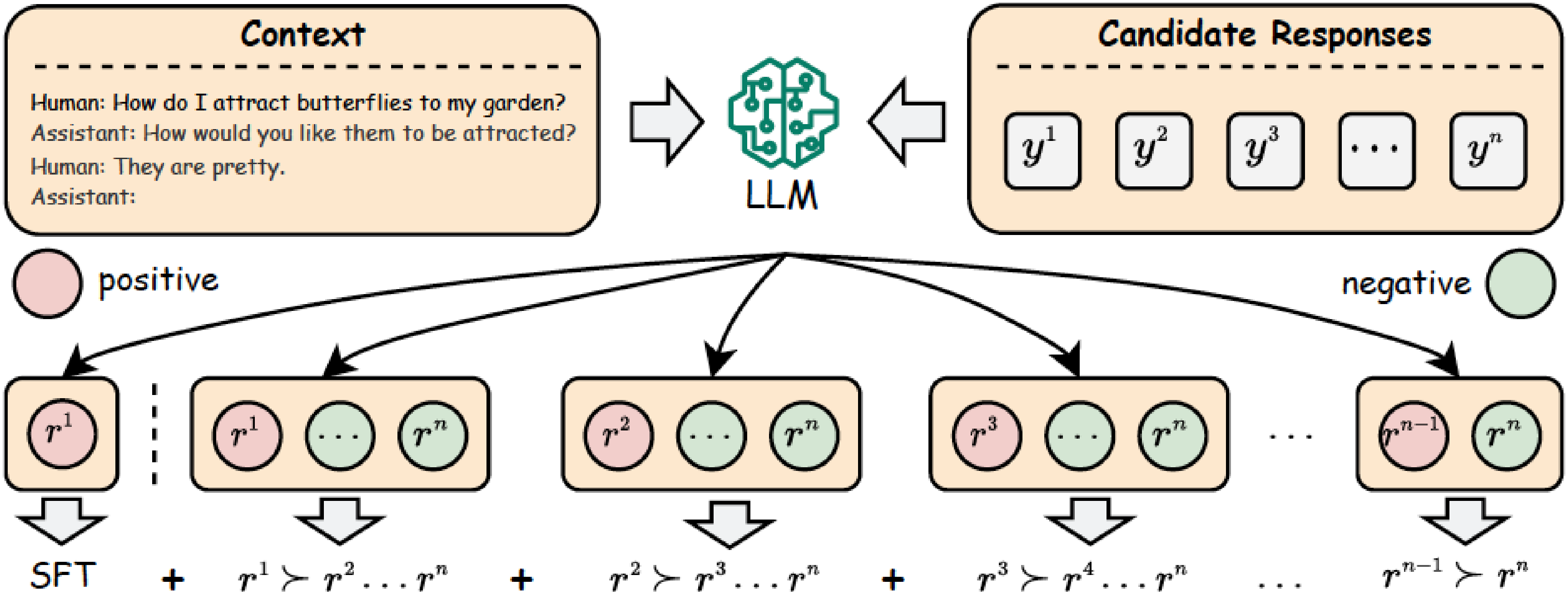
$$\boxed{y_1 \succ y_2 \succ \cdots \succ y_n} \longrightarrow \boxed{y_1 \succ \{y_2, \cdots, y_n\}}$$

$$\boxed{\mathcal{L}(y_1 \succ \{y_2, \cdots, y_n\}) = -\log \frac{\exp\left(r_\pi(x, y^1)\right)}{\sum_{i=1}^{n} \exp\left(r_\pi(x, y^i)\right)}}$$

InfoNCE loss

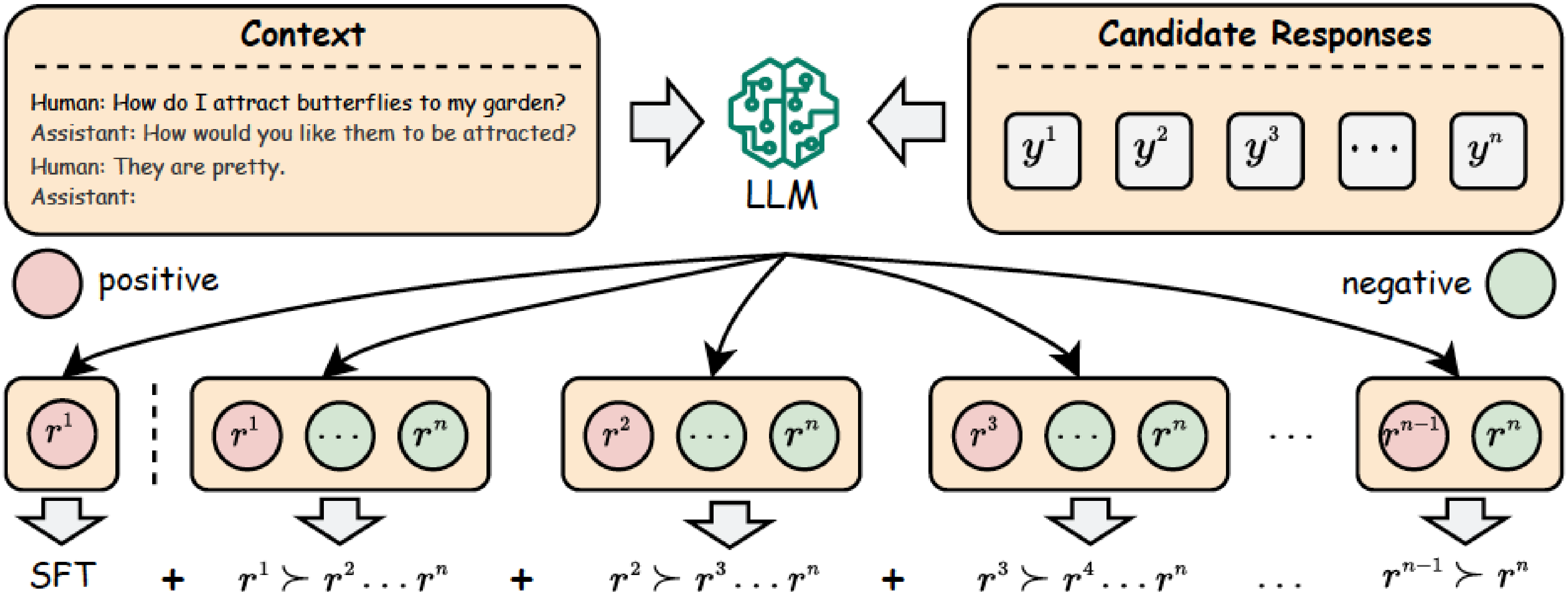
  - Issue: Does not fully leverage the ranking

Human Alignment: Preference Ranking Optimization (PRO)



The pipeline of PRO for Human Feedback Alignment learning

Human Alignment: Preference Ranking Optimization (PRO)



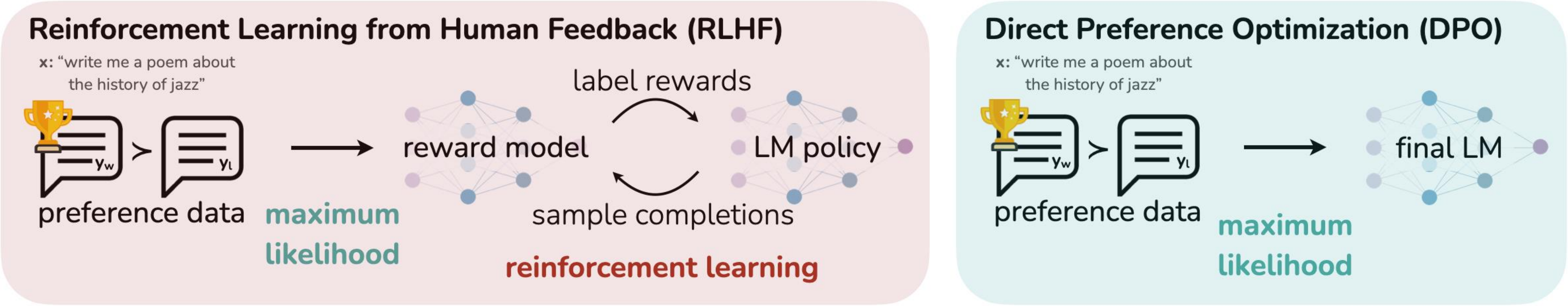
The pipeline of PRO for Human Feedback Alignment learning

$$\mathcal{L}(y_1 \succ y_2 \succ \cdots \succ y_n) = -\log \prod_{k=1}^{n-1} \frac{\exp\left(r(x, y_k)\right)}{\sum_{i=k}^{n} \exp\left(r(x, y_i)\right)}$$

Direct Preference Optimization

Rafailov et al. "Direct preference optimization: Your language model is secretly a reward model." *NeurIPS 2024*).

Direct Preference Optimization



Direct Preference Optimization

- RLHF Objective:

get high reward, stay close to reference model

Any reward functions

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}[r(x,y)] - \beta\mathbb{D}_{KL}\left[\pi(y \mid x) \| \pi_{ref}(y \mid x)\right]$$

- There is a closed-form solution of the above optimization

Direct Preference Optimization

- Deriving Closed-Form Optimal Policy:

$$\mathbb{D}_{KL}(p\|q) = \mathbb{E}_{u\sim p}\left[\log \frac{p(u)}{q(u)}\right]$$

$$Z(x) = \sum_y \pi_{ref}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

$$\max_\pi \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}[r(x,y)] - \beta\mathbb{D}_{KL}\left[\pi(y \mid x) \| \pi_{ref}(y \mid x)\right]$$

$$= \max_\pi \mathbb{E}_{x\sim\mathcal{D}} \mathbb{E}_{y\sim\pi(y|x)}\left[r(x,y) - \beta \log \frac{\pi(y \mid x)}{\pi_{ref}(y \mid x)}\right]$$

$$= -\beta \min_\pi \mathbb{E}_{x\sim\mathcal{D}} \mathbb{E}_{y\sim\pi(y|x)}\left[\log \frac{\pi(y \mid x)}{\pi_{ref}(y \mid x)} - \frac{1}{\beta} r(x,y)\right]$$

$$= -\beta \min_\pi \mathbb{E}_{x\sim\mathcal{D}} \mathbb{E}_{y\sim\pi(y|x)}\left[\log \frac{\pi(y \mid x)}{\frac{1}{Z(x)} \pi_{ref}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)} - \log Z(x)\right]$$

$$= -\beta \min_\pi \mathbb{E}_{x\sim\mathcal{D}} \mathbb{E}_{y\sim\pi(y|x)}\left[\mathbb{D}_{KL}\left[\pi(y \mid x) \middle\| \frac{1}{Z(x)} \pi_{ref}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)\right] - \log Z(x)\right]$$

Direct Preference Optimization

- RLHF Objective:

get high reward, stay close to reference model

Any reward functions

$$\boxed{\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}[r(x,y)] - \beta \mathbb{D}_{KL}\left[\pi(y \mid x) \| \pi_{ref}(y \mid x)\right]}$$

- Closed-form Optimal Policy:

write optimal policy as function of reward function

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

$$with\ Z(x) = \sum_{y} \pi_{ref}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

intractable sum over possible response

Direct Preference Optimization

- Closed-form Optimal Policy:

write optimal policy as
function of reward function

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

$$with\ Z(x) = \sum_y \pi_{ref}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$$

intractable sum over possible response

- Rearrange:

ratio is positive if policy likes response more
than reference model; negative if otherwise.

$$\boxed{r(x, y) = \underbrace{\beta \log \frac{\pi^*(y \mid x)}{\pi_{ref}(y \mid x)} + \beta \log Z(x)}_{\text{Some parameterization of a reward function}}}$$

Direct Preference Optimization: Putting It Together

derived from the Bradley-Terry model of human preferences

A loss function on reward functions

$$\boxed{\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[\log \sigma\left(r(x, y_w) - r(x, y_l)\right)\right]}$$

+

A transformation between reward functions and policy

$$r(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{ref}(y \mid x)} + \beta \log Z(x)$$

When substituting, the log Z term cancels, because the loss only care about difference in rewards

Reward of preferred response

Reward of dis-preferred response

=

A loss function on policy

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

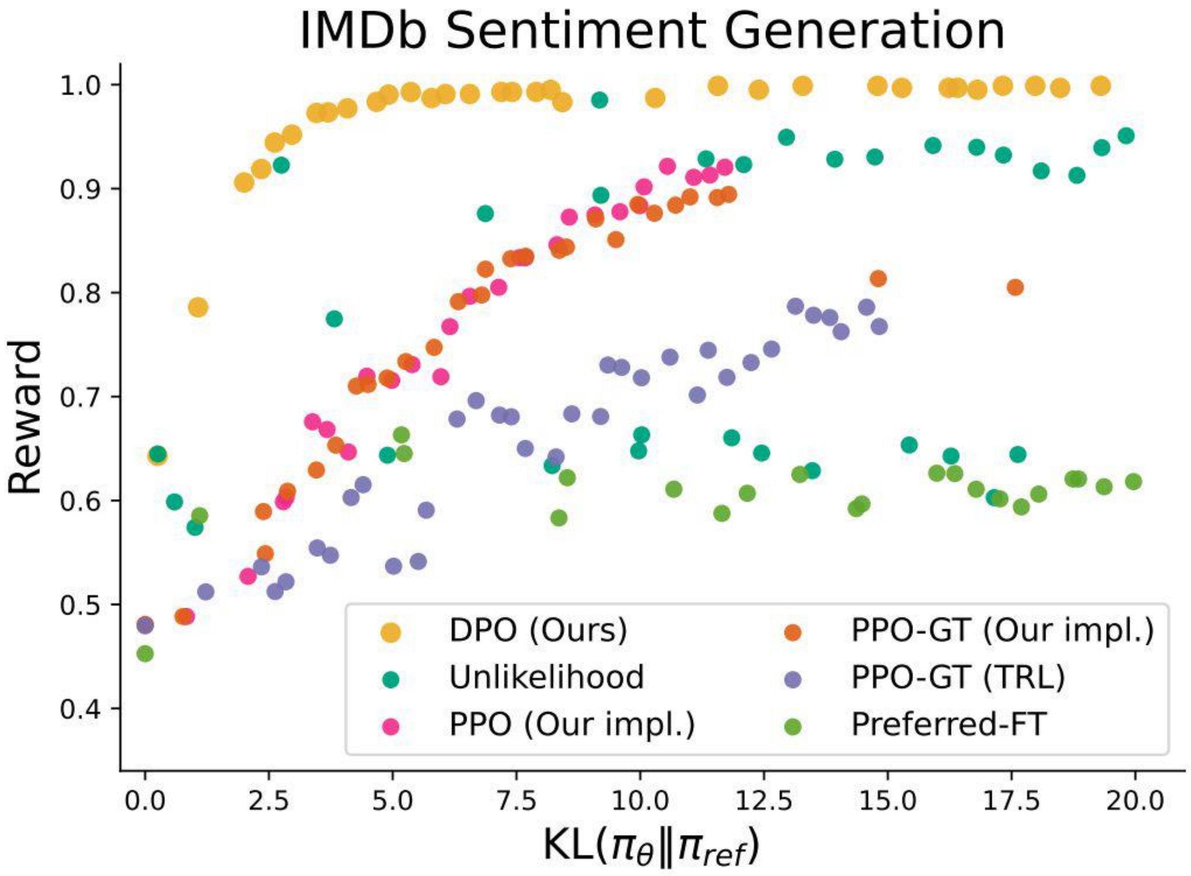
Results

- Three tasks
  - Controlled sentiment generation (IMDb dataset)
  - Summarization (Reddit dataset)
  - Single-turn dialogue (Anthropic Helpful and Harmless dialogue dataset)

- Evaluation
  - Controlled sentiment generation: pre-trained sentiment classifier (rewards)
  - Win rates against a baseline policy
    - Use GPT-4 as a proxy for human evaluation of summary quality and response helpfulness
    - Summarization: reference summaries in the test set as baseline
    - Dialog: preferred response in test dataset as baseline

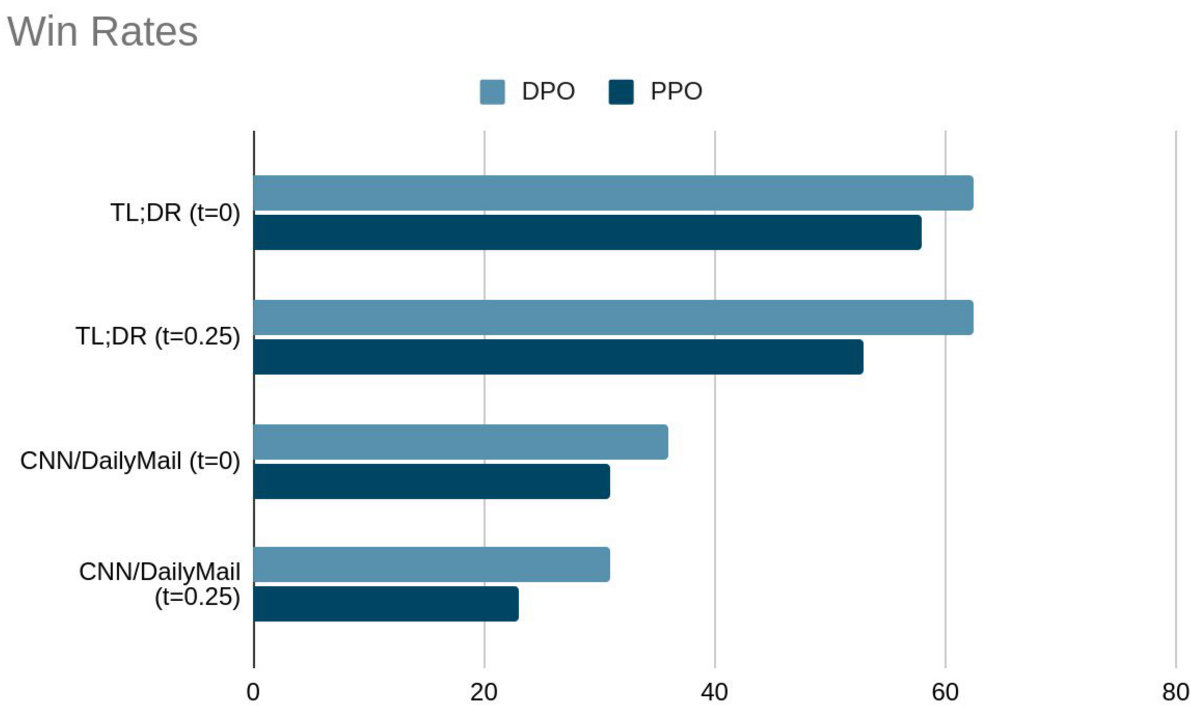
How Efficiently does DPO Trade off Reward & KL?

1. Generate positive IMDB reviews from GPT2-XL
2. Use pre-trained sentiment classifier as Gold RM
3. Create preferences based on Gold RM
4. Optimize with PPO and DPO



DPO vs PPO: Empirics

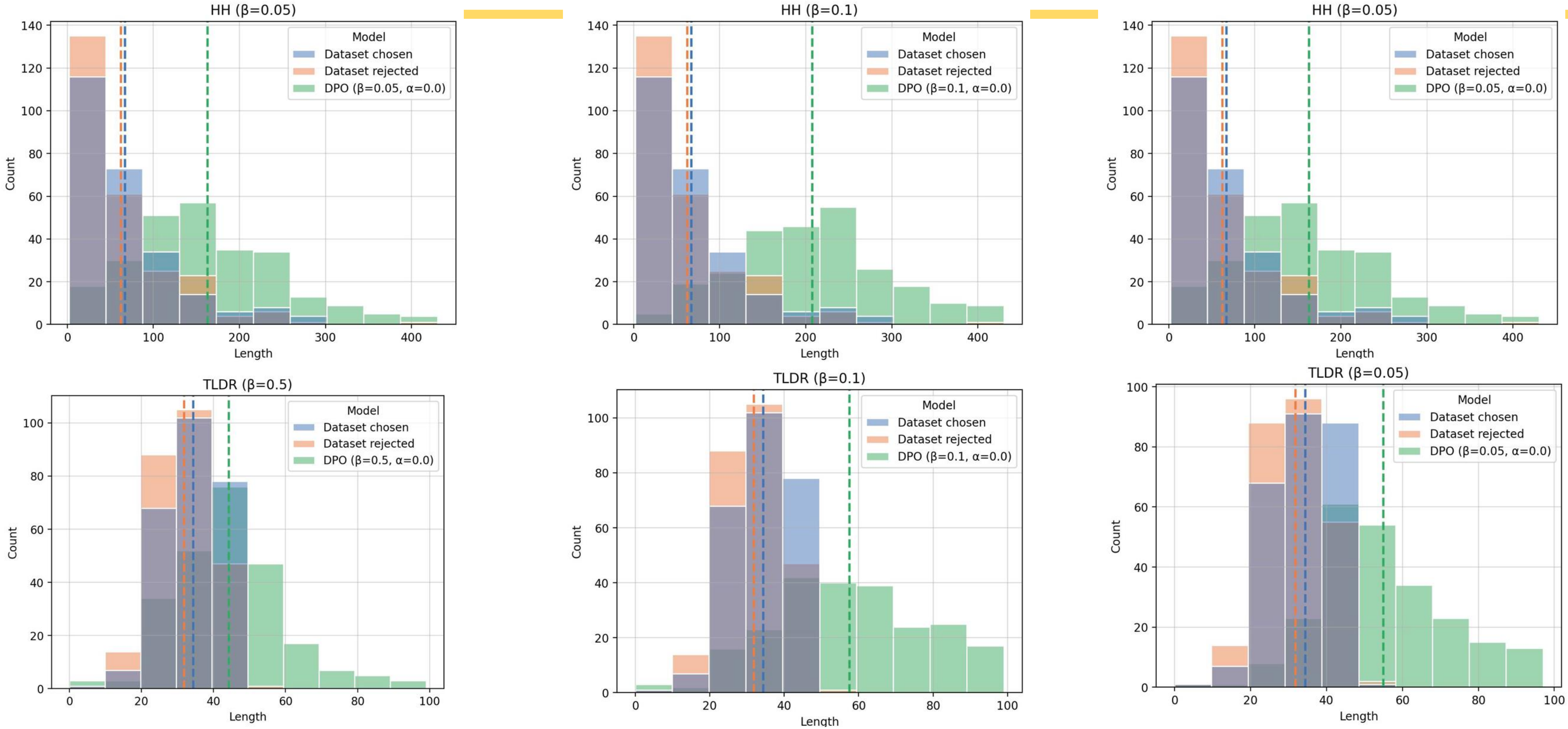
1. DPO is trained only on the Reddit TL;DR feedback data.
2. PPO uses a trained reward function and additional prompts for RL training.
3. We evaluate the trained policies on OOD CNN/DailyMail news summarization task.



DPO: Summary

- DPO optimizes the same classical RLHF objective
- Is simple and computationally cheap
- Like classical RLHF it is prone to hacking

DPO: Reward Hacking Issue



Park et al. "[Disentangling length from quality in direct preference optimization](#)." *arXiv preprint arXiv:2403.19159* (2024).

Length Regularization in DPO

- RLHF objective:

get high reward, stay close to reference model

Any reward functions

$$\max_{\pi} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta} [r(x, y)] - \beta \mathbb{D}_{KL} \left[\pi(y \mid x) \| \pi_{ref}(y \mid x) \right]$$

- RLHF objective with length regularization:

$$\max_{\pi} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta} [r(x, y)] - \alpha |y| - \beta \mathbb{D}_{KL} \left[\pi(y \mid x) \| \pi_{ref}(y \mid x) \right]$$

Length regularization

- Closed-form optimal policy

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

$$with\ Z(x) = \sum_y \pi_{ref}(y \mid x) \exp\left(\frac{1}{\beta} (r(x, y) - \alpha |y|) \right)$$

Park et al. "Disentangling length from quality in direct preference optimization." *arXiv preprint arXiv:2403.19159* (2024).

Length Regularization in DPO

- Closed-form Optimal Policy:

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

$$with\ Z(x) = \sum_y \pi_{ref}(y \mid x) \exp\left(\frac{1}{\beta}\left(r(x,y) - \alpha|y|\right)\right)$$

- Rearrange:

ratio is positive if policy likes response more than reference model; negative if otherwise.

$$\boxed{r(x,y) = \underbrace{\beta \log \frac{\pi^*(y \mid x)}{\pi_{ref}(y \mid x)} + \beta \log Z(x) - \alpha|y|}_{\text{Some parameterization of a reward function}}}$$

DPO with Regularization: Putting It Together

derived from the Bradley-Terry model of human preferences

A loss function on reward functions

$$\boxed{\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w\, y_l) \sim \mathcal{D}}\left[\log \sigma\left(r(x, y_w) - r(x, y_l)\right)\right]}$$

+

A transformation between reward functions and policy

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

When substituting, the log Z term cancels, because the loss only care about difference in rewards

Reward of preferred response

Reward of dis-preferred response

=

A loss function on policy

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

Summary

- LLMs with human feedback
  - Goal: align LLM responses with human preferences
  - RLHF: two stages of reward modeling and policy learning
  - DPO: end-to-end policy learning
  - Variants:
    - Listwise rankings versus pairwise rankings
    - Length regularization
- Future works
  - Generalizability of LLM alignment
  - LLM alignment for non-English languages
  - Human-in-the-loop factors