Reinforcement Learning with Human Feedback

Pu Yang

2023.3.22

Content

Introduction

What properties do we want a LM to satisfy?

	Importance	ChatGPT / GPT4
Helpful	😄	😄
Honest	😟	🤦
Harmless	😄	😟

Deep Reinforcement Learning from Human Preferences

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in neural information processing systems, 30.

Preliminaries

- At each time t
  - Observation $o_t \in O$
  - Action $a_t \in A$
  - Reward $r_t \in R$ (×)

- Trajectory segment
  - $\sigma = (o_0, a_0, o_1, a_1, \cdots, o_{k-1}, a_{k-1})$
  - $\sigma^1 > \sigma^2$: The human preferred trajectory segment σ^1 than trajectory segment σ^2

Preliminaries – Evaluate in two ways

- Quantitative: Preferences “>” are generated by a reward function $r: O \times A \rightarrow R$ if

$$\left(o_0^1, a_0^1, \cdots, o_{k-1}^1, a_{k-1}^1\right) > \left(o_0^2, a_0^2, \cdots, o_{k-1}^2, a_{k-1}^2\right)$$

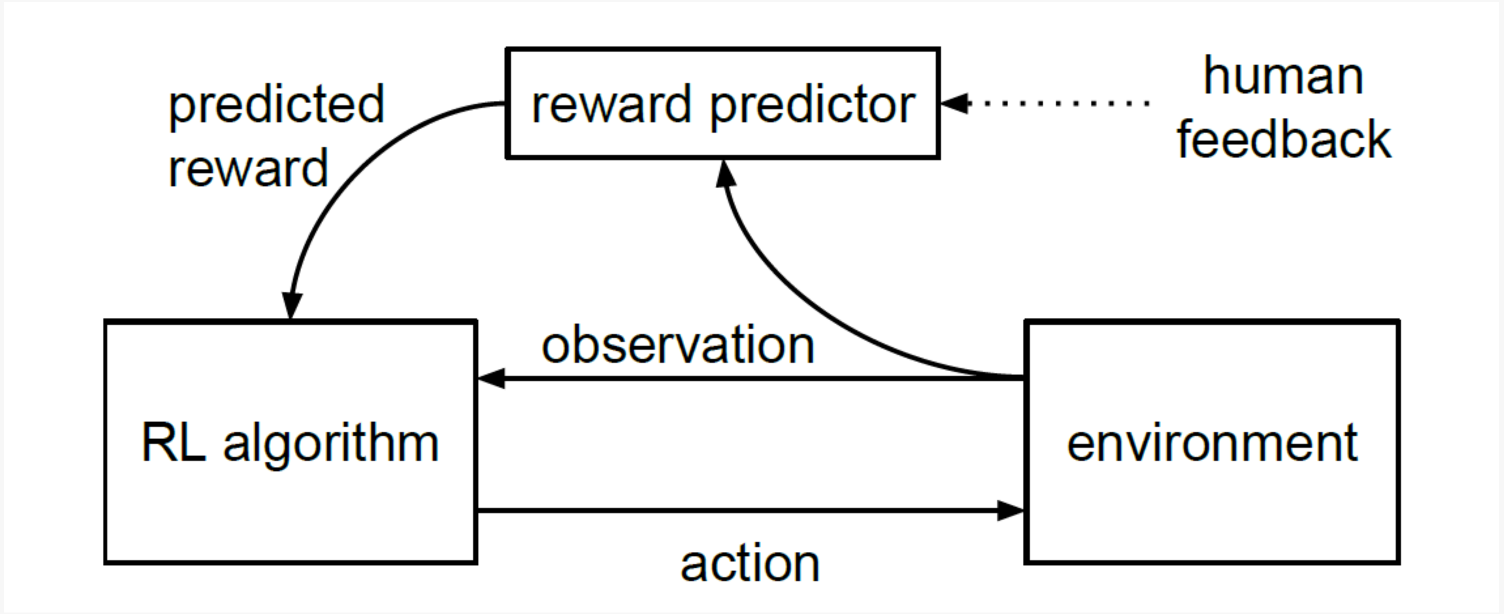
where

$$r\left(o_0^1, a_0^1\right) + \cdots + r\left(o_{k-1}^1, a_{k-1}^1\right) > r\left(o_0^2, a_0^2\right) + \cdots + r(o_{k-1}^2, a_{k-1}^2)$$

- Qualitative: Sometimes we have no reward function by which we can quantitatively evaluate behavior. In these cases, all we can do is qualitatively evaluate how well the agent satisfies to the human’s preferences.

Offline RL

- Inaccessible to the complete environment
- Model-based method:



Method: Fitting a Reward Function

- We can interpret a reward function estimate $\hat{r}$ as a preference-predictor if we view $\hat{r}$ as a latent factor explaining the human's judgments, and utilize logistic function

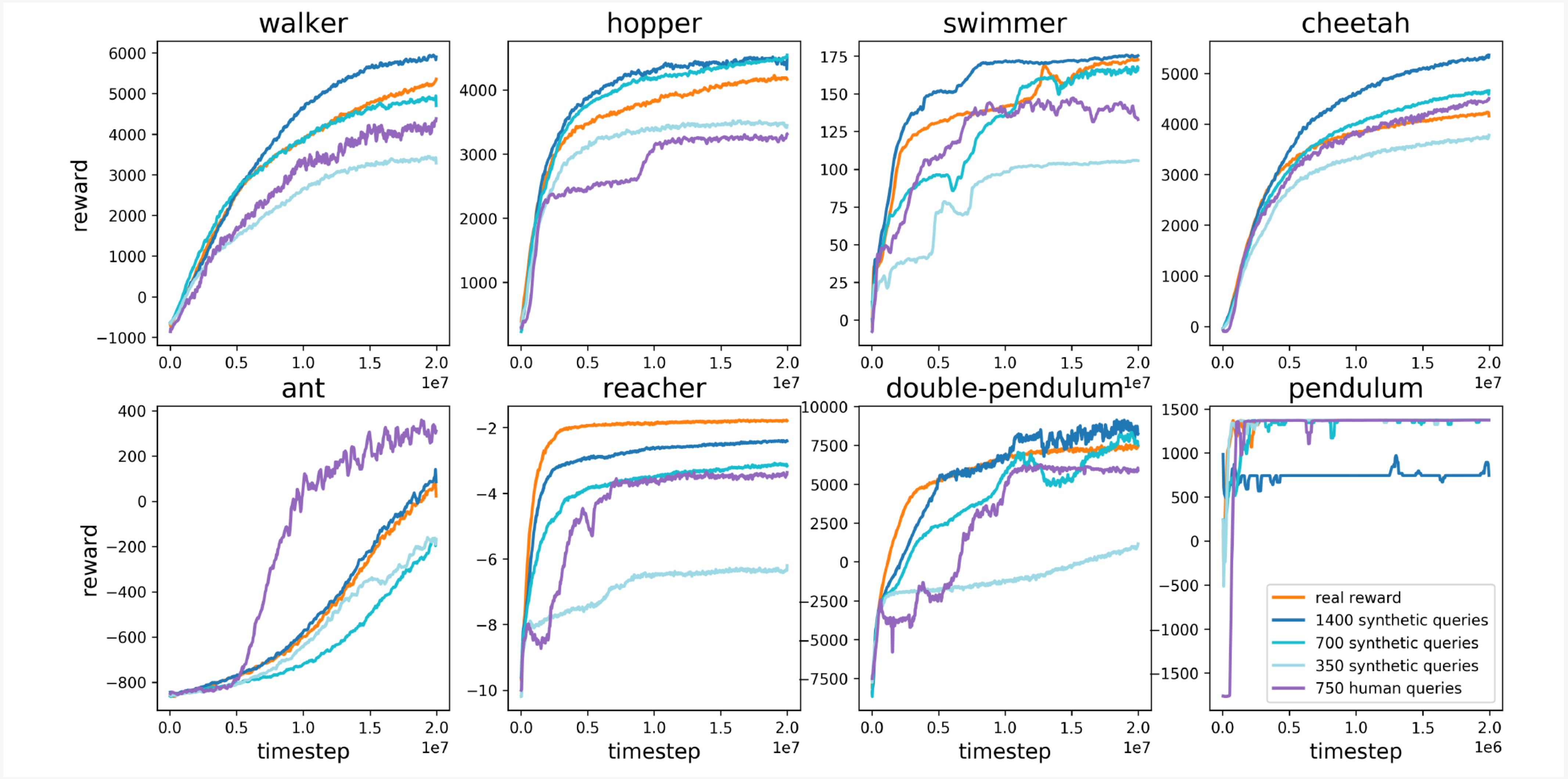
$$\hat{P}\left[\sigma^1 \succ \sigma^2\right] = \frac{\exp \sum \hat{r}\left(o_t^1, a_t^1\right)}{\exp \sum \hat{r}(o_t^1, a_t^1) + \exp \sum \hat{r}(o_t^2, a_t^2)}.$$

- Cross-Entropy loss (pairwise ranking loss):

$$\operatorname{loss}(\hat{r}) = -\sum_{(\sigma^1, \sigma^2, \mu) \in \mathcal{D}} \mu(1) \log \hat{P}\left[\sigma^1 \succ \sigma^2\right] + \mu(2) \log \hat{P}\left[\sigma^2 \succ \sigma^1\right].$$

where μ is an one-hot vector.

Experimental Results on MuJoCo

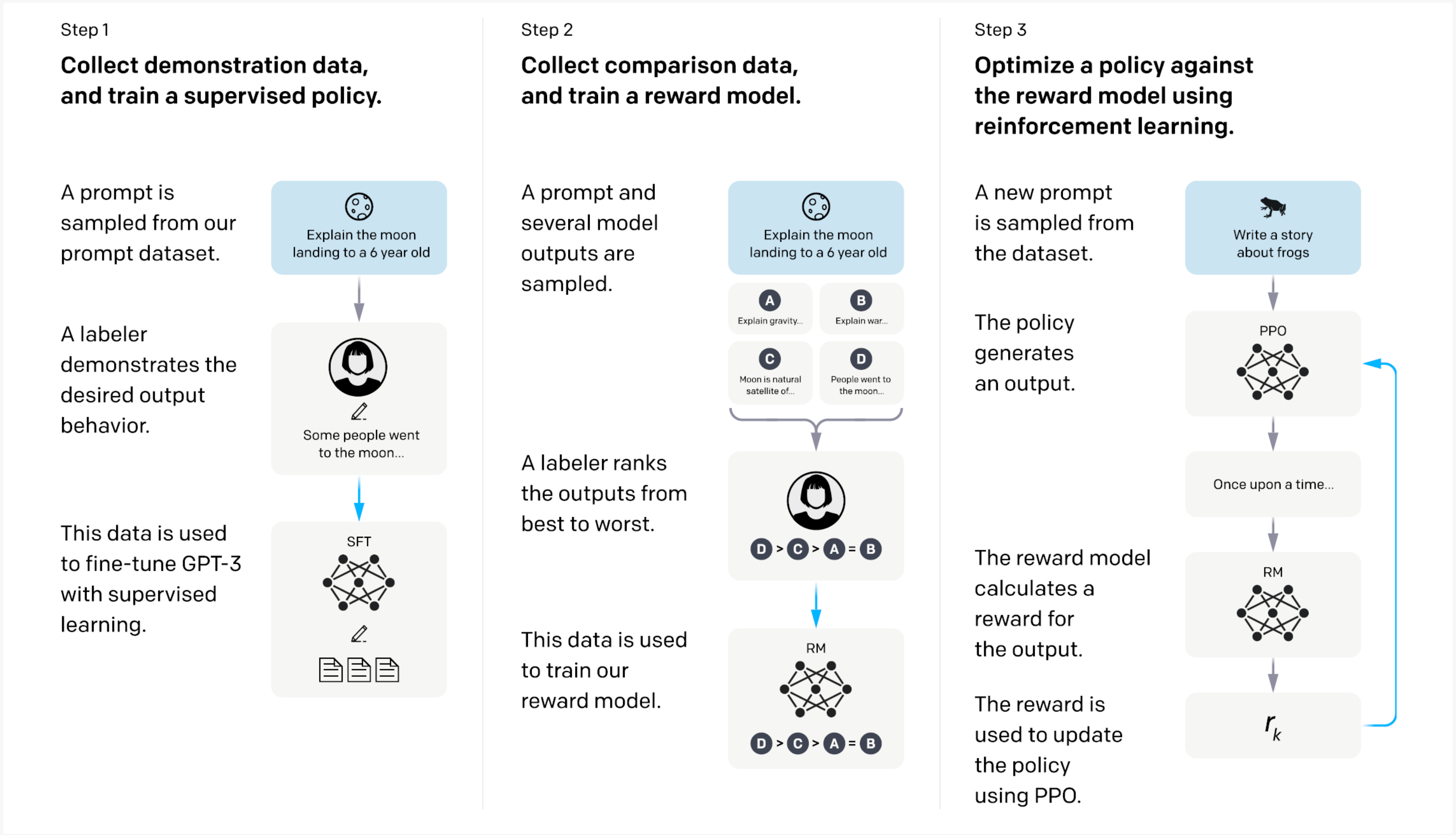


Instruct GPT

Ziegler D M, Stiennon N, Wu J, et al. Fine-tuning language models from human preferences[J]. arXiv preprint arXiv:1909.08593, 2019.

Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback[J]. arXiv preprint arXiv:2203.02155, 2022.

Instruct GPT



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

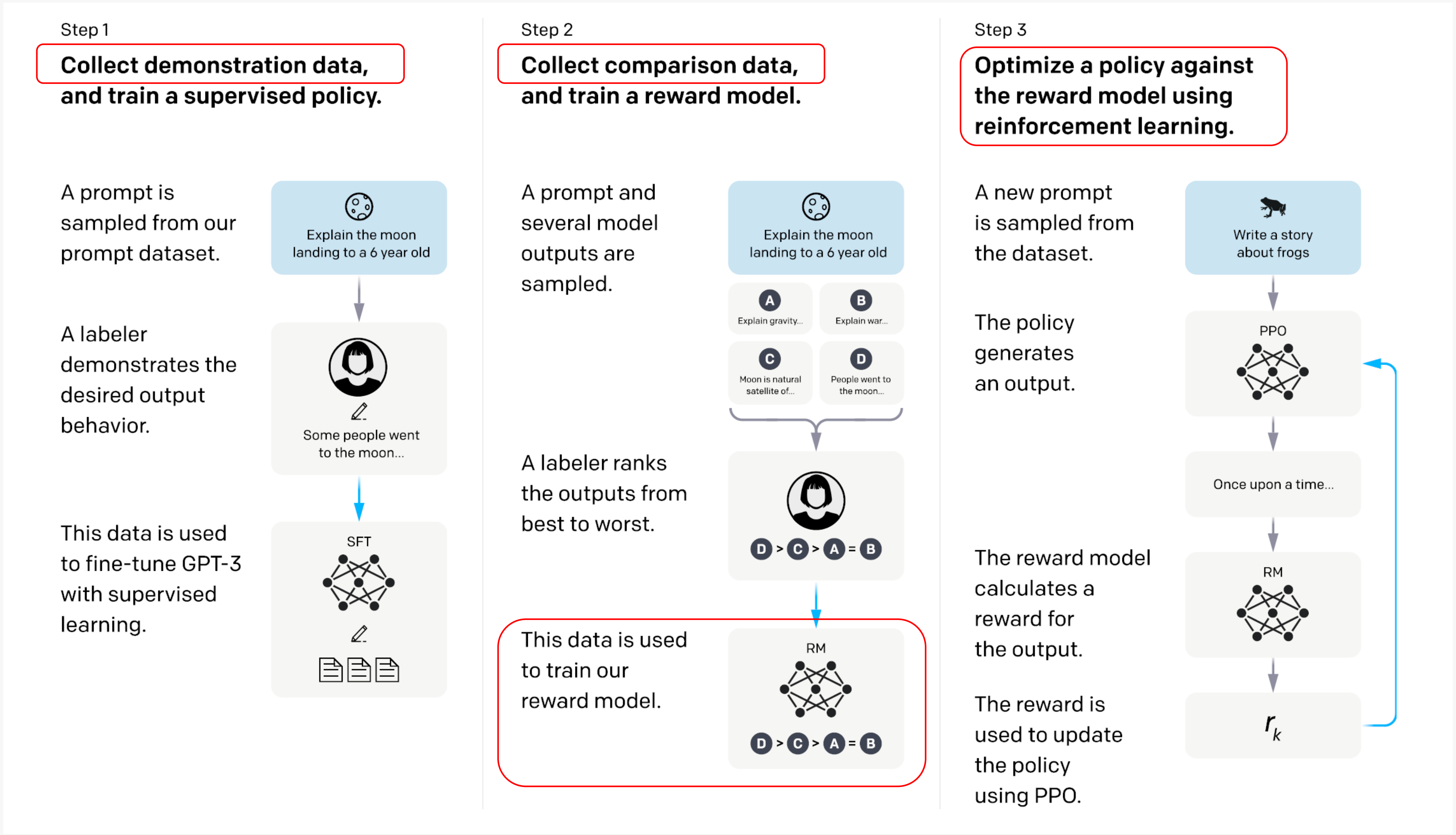
The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A Explain gravity...
B Explain war...
C Moon is natural satellite of...
D People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

Instruct GPT



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A Explain gravity...
B Explain war...
C Moon is natural satellite of...
D People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

Collect Data

- Collect Prompts
  - Ask labelers to write prompts themselves to train the very first InstructGPT models
  - Let customers to use it and collect 200 prompts per user ID
- Produce 3 datasets
  - SFT datasets (with labeler demonstrations), about 13k training prompts
  - RM datasets (with labeler rankings of model outputs), about 33k training prompts
  - PPO datasets, about 31k training prompts

Data

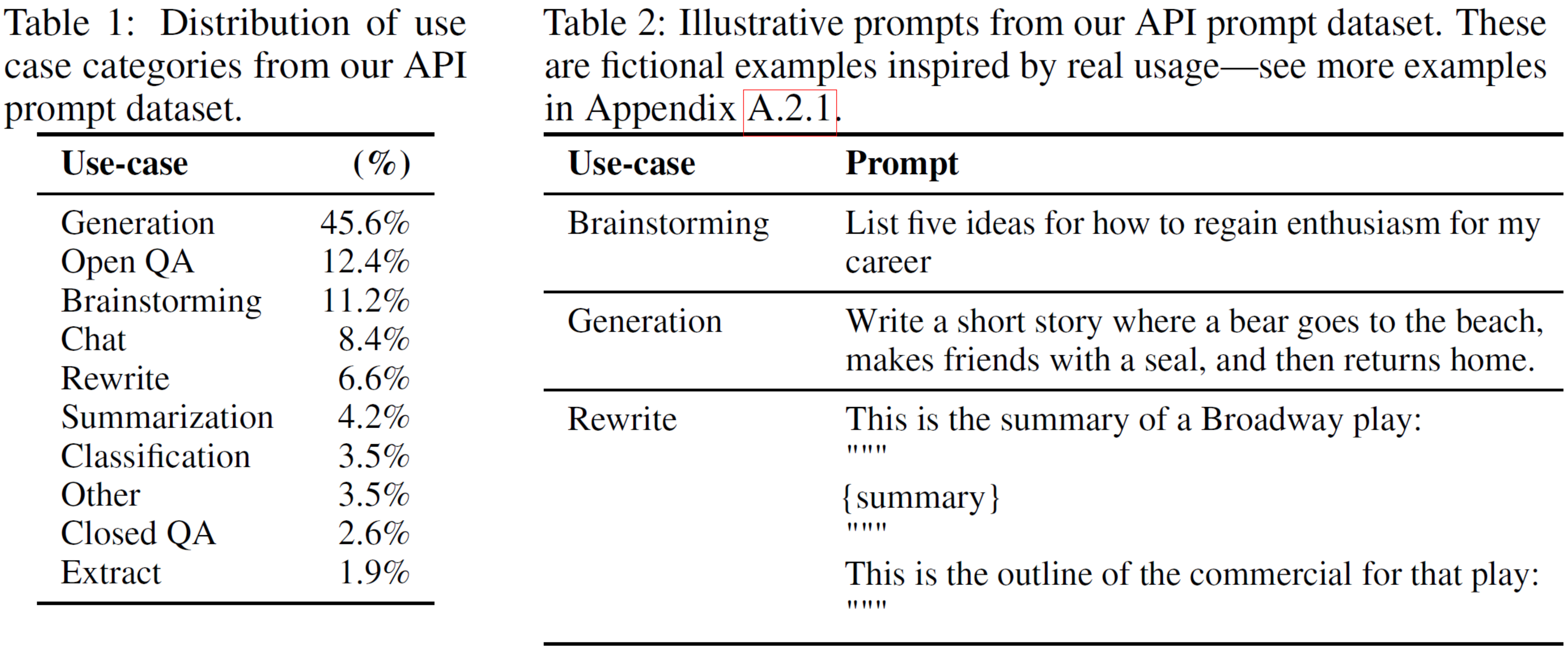


Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

Reward Modeling $r_\theta(prompt, output)$

- Model
  - Start from the SFT model
  - Remove the final unembedding layer (softmax layer)
  - Add a FC layer to output a scaler
- Data
  - Present labelers with anywhere between K=9 responses to rank
- Pairwise ranking loss

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

where $(x, y_w) > (x, y_l)$ and σ is the sigmoid function.

Use RM to Learn a "Policy"

- Optimization Problem:

$$\max_{\phi} E_{(x,y)\sim D_{\pi_\phi}} \; r_\theta(x,y)$$

where x is prompt, y is output, π_ϕ is the LM

- Notice that y is depend on ϕ, so we can not use normal supervised learning.
- Reparameterization v.s. Reinforcement Learning

Reinforcement Learning

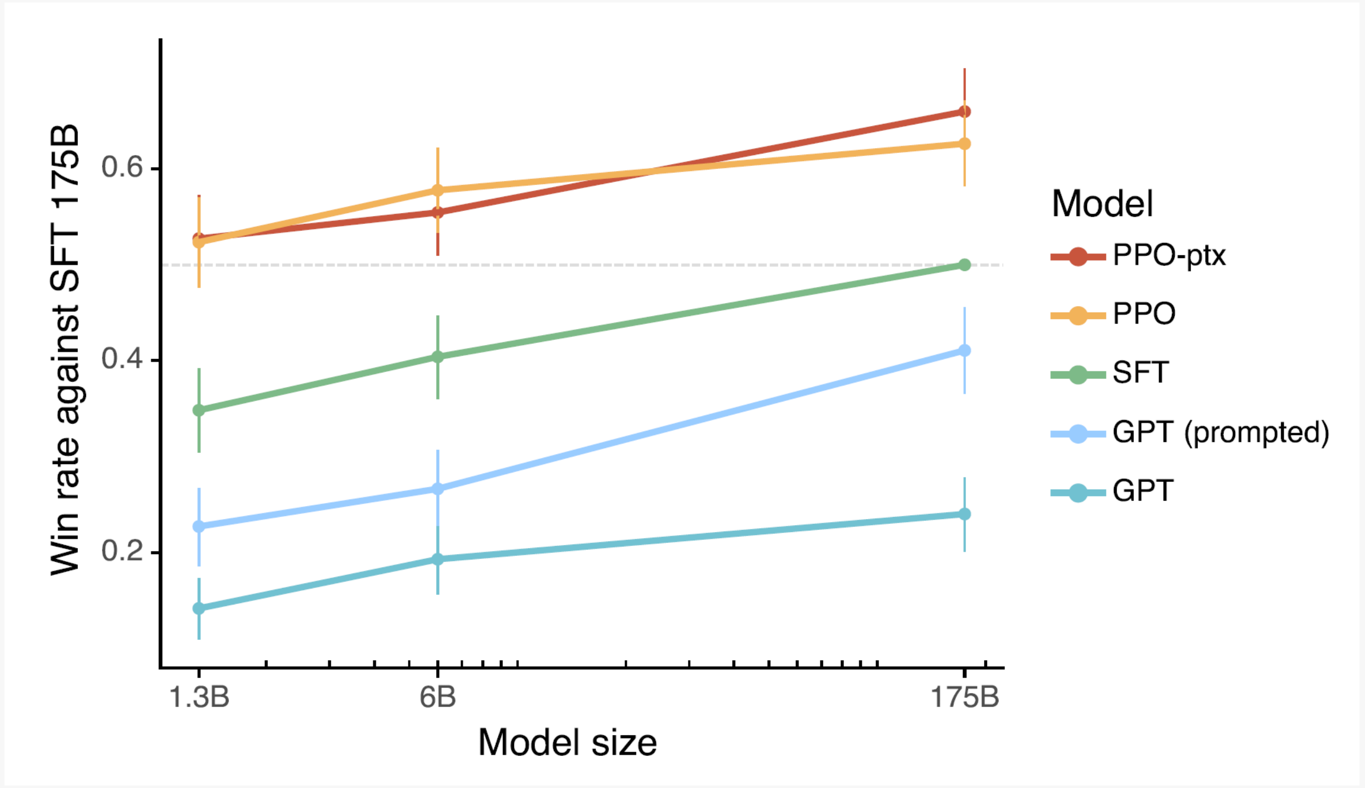
- MDP (Incomplete: No transition)
  - State: prompt x
  - Action: output y
  - Reward: RM $r_\theta(x, y)$
- PPO-ptx Loss (First term + Second term = PPO loss)

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

  - First term: reward from RM
  - Second term: KL, Prevent distributional mismatch in testing (action)
  - Third term: Prevent distributional mismatch in training (state)

Experimental Results

- Human evaluations of various models



Anthropic LLM

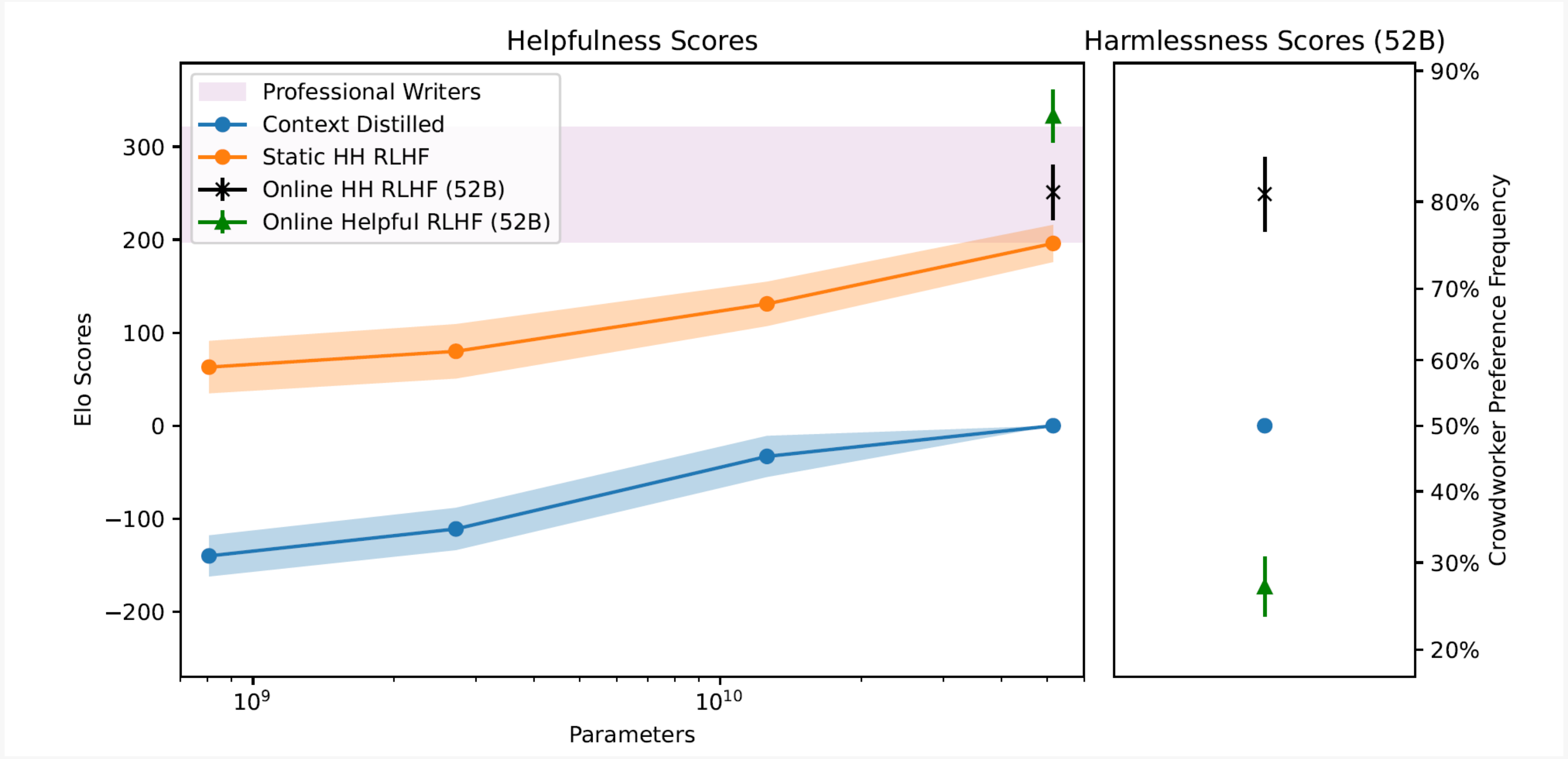
Bai Y, Jones A, Ndousse K, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback[J]. arXiv preprint arXiv:2204.05862, 2022.

Anthropic

- 官网：https://www.anthropic.com/
- Google Bard: https://blog.google/technology/ai/bard-google-ai-search-updates/

Two Important Improvements

- Trade-off between helpful and harmless
- Iterated Online Mode Training



Data Collection for Prompts

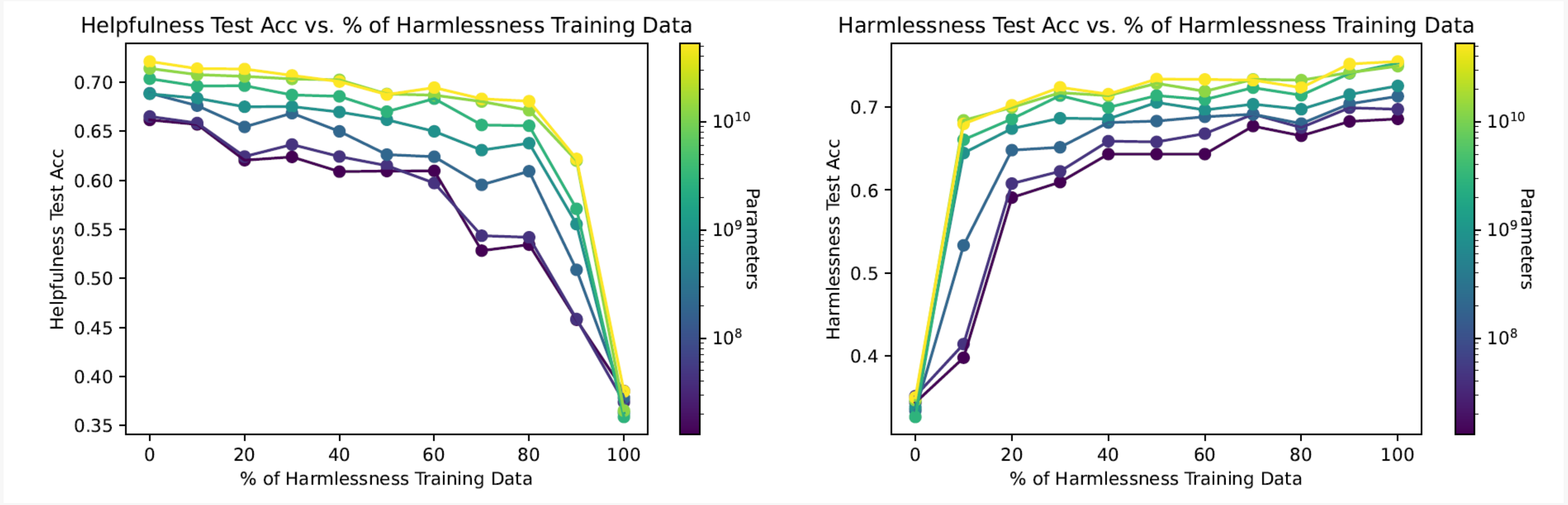
- Helpful: we ask crowdworkers to solicit our models to assist with any purely text-based tasks
  - answer questions
  - write or edit documents
  - discuss plans and decisions
- Harmless: we invite crowdworkers to adversarially probe or ‘red-team’ our language models in order to provoke harmful responses
  - help them with harmful goals
  - cause the AI to use toxic language

Datasets

- STF: 44k helpful, 42k harmless
- RM: 52k helpful, 2k harmless
- RL: 22k helpful, 0 harmless

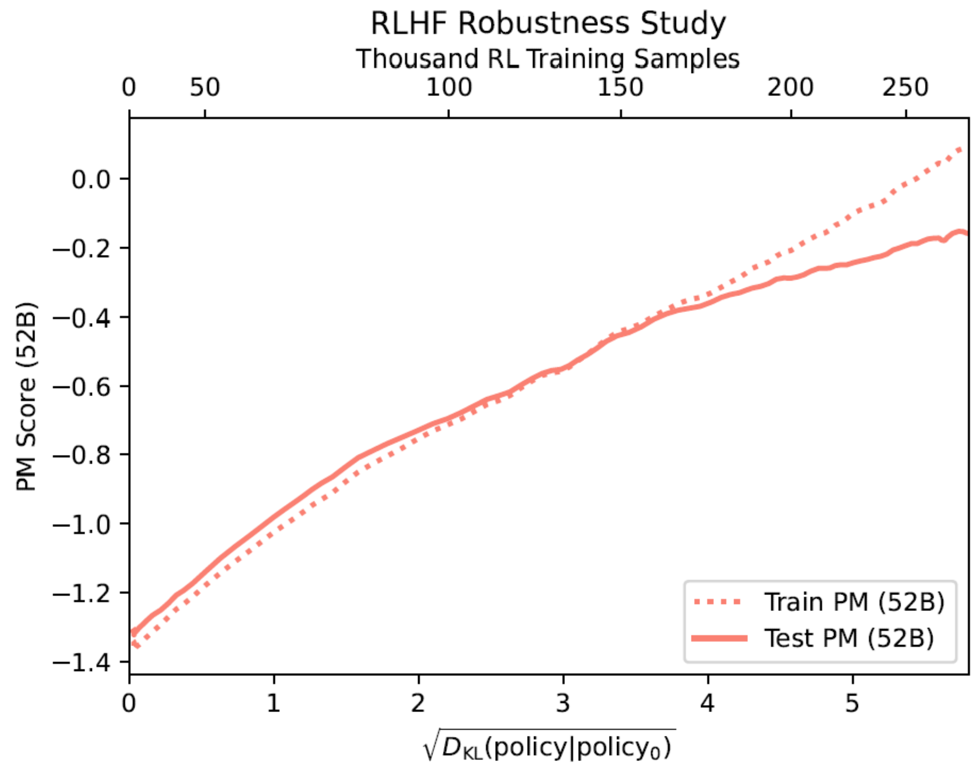
Varying Helpful vs Harmless Data Fraction

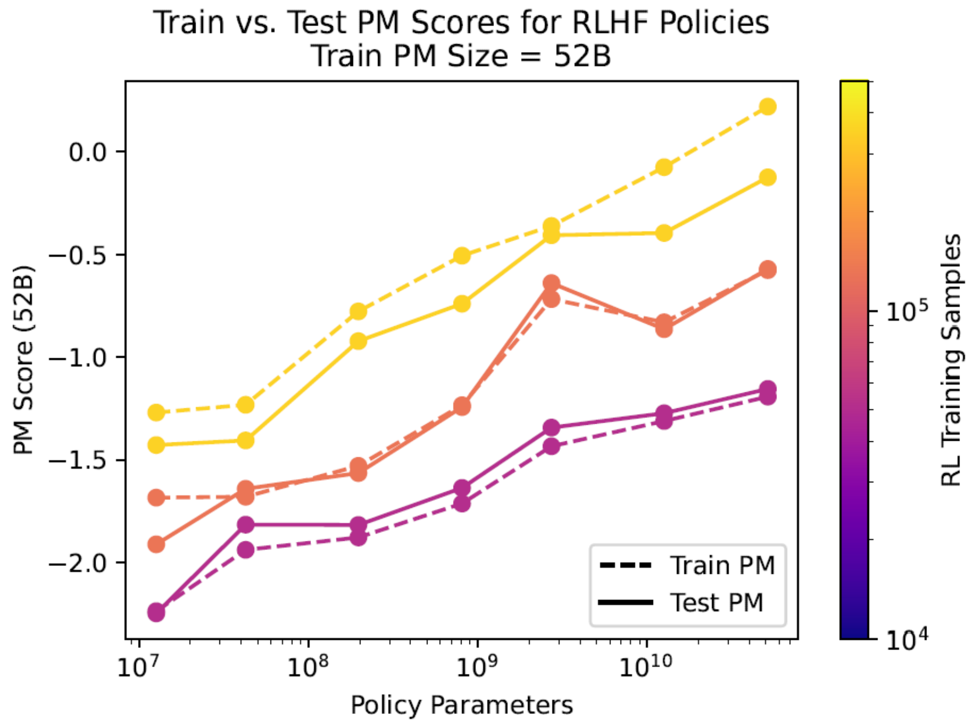
- STF Model



Robustness for Reward Model

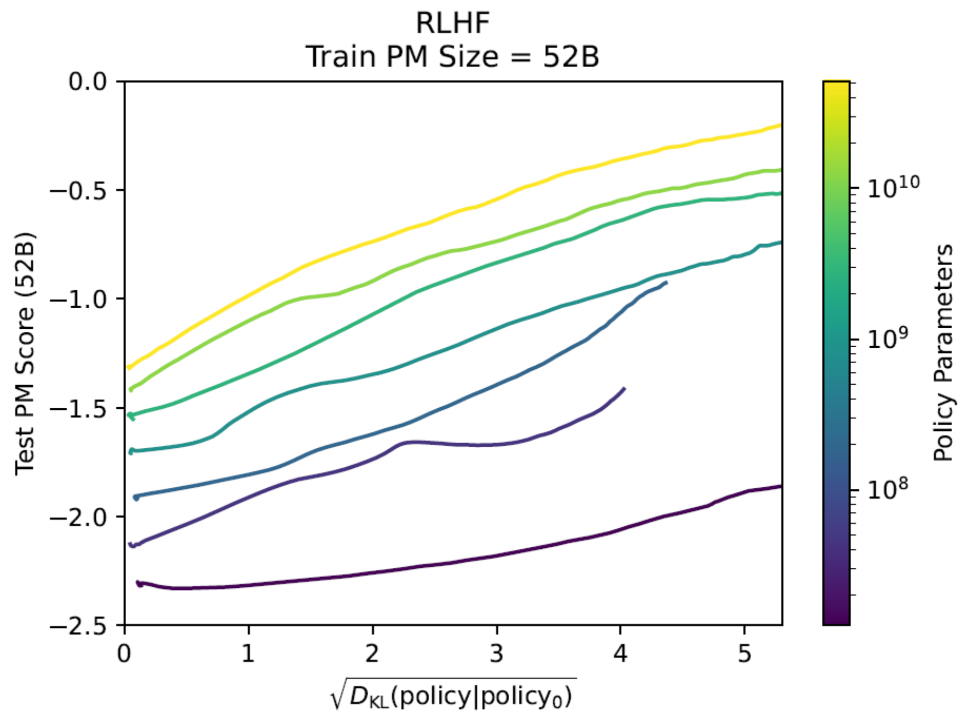
- Learn a Train RM and a Test RM

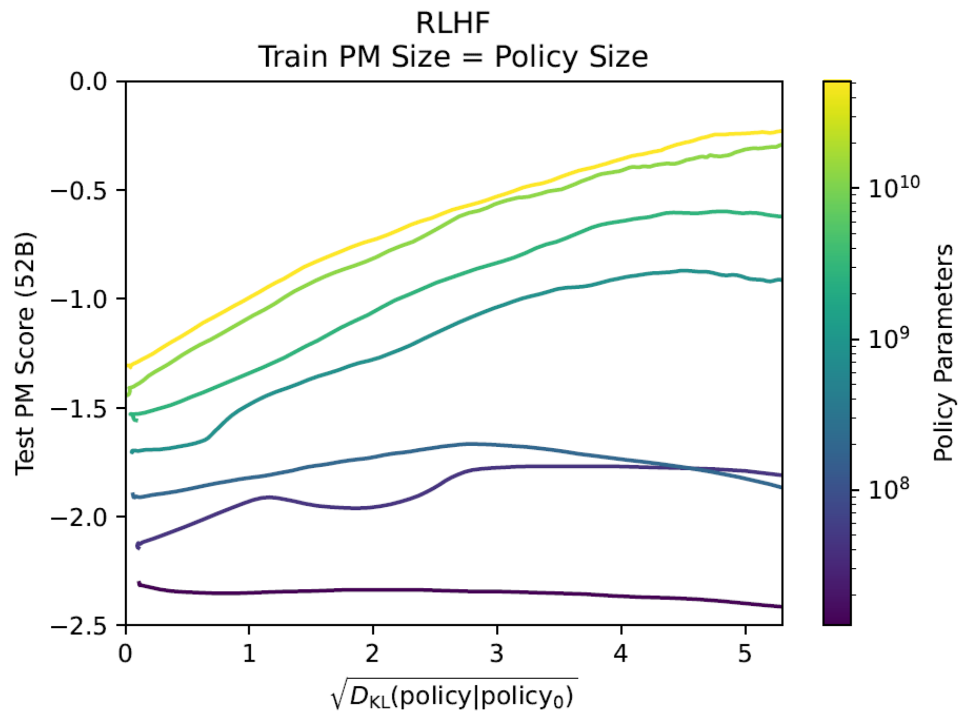




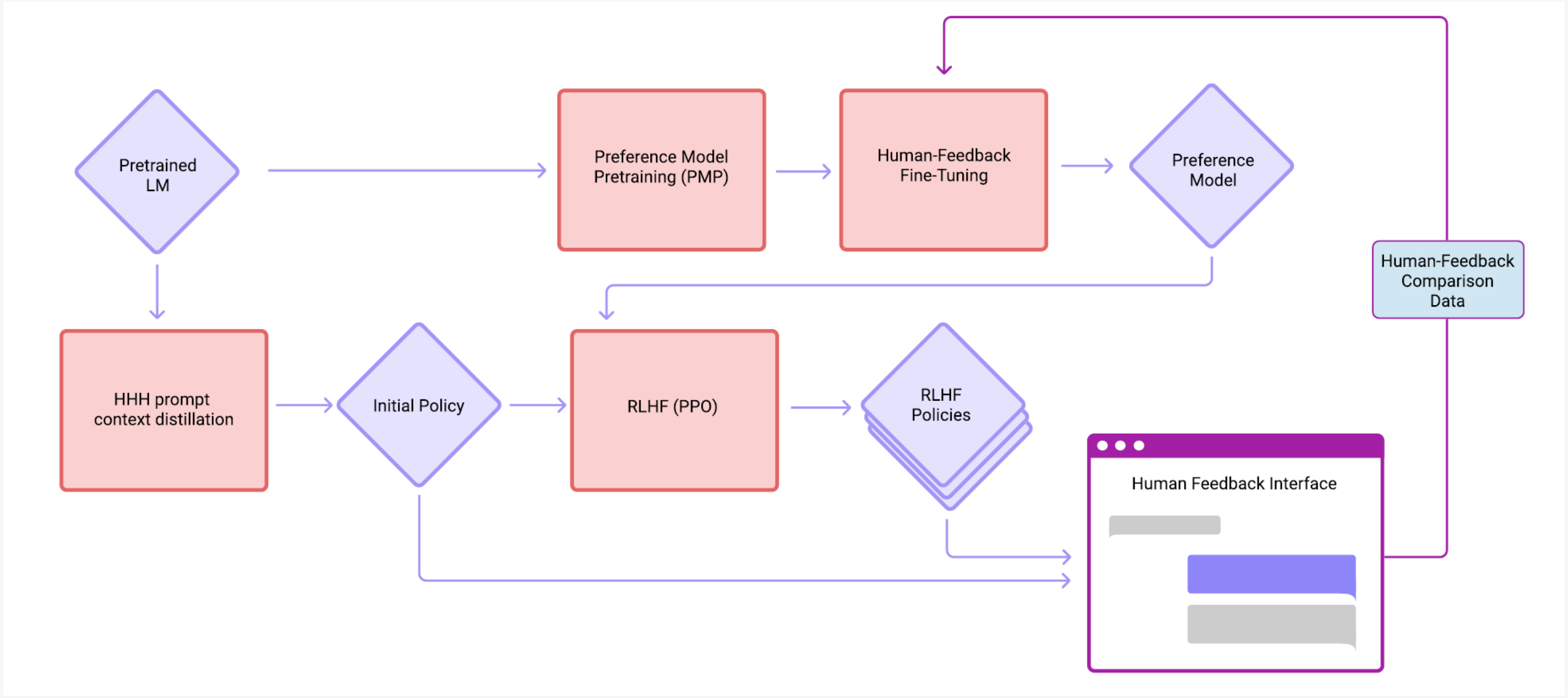
Robustness for Reward Model

- Train PM Size = 52B v.s. Train PM Size = Policy Size

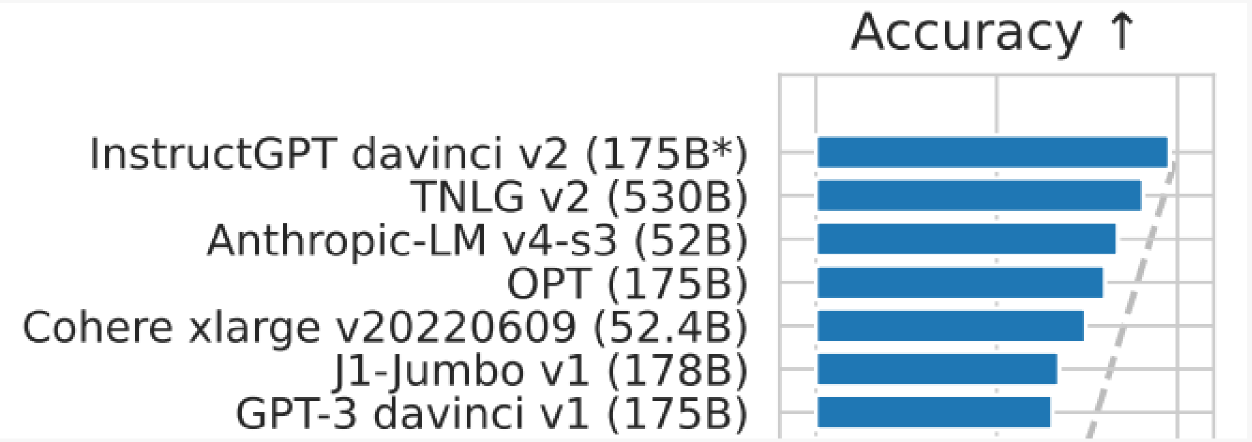


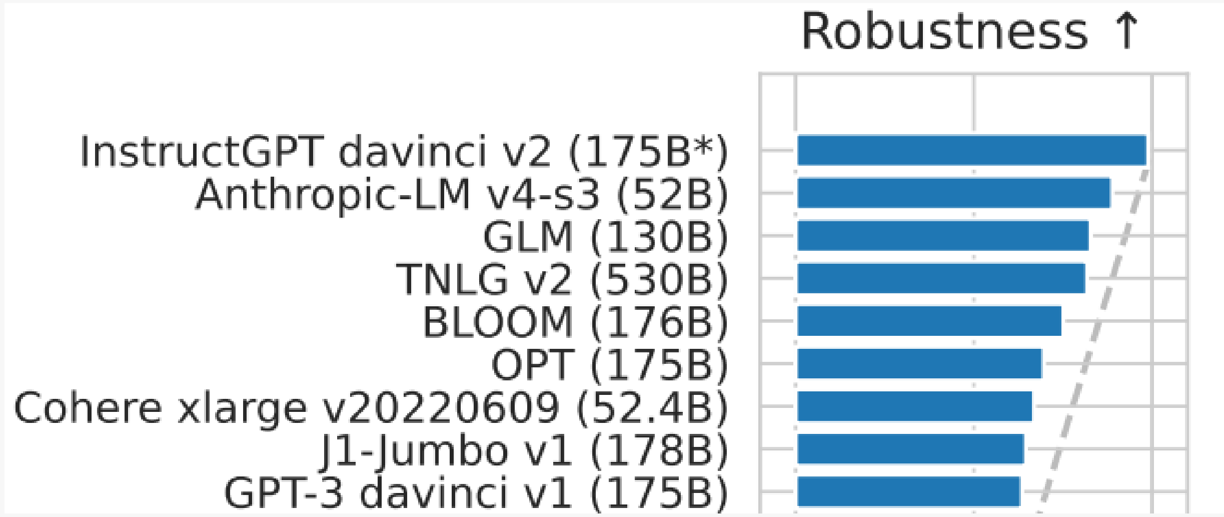


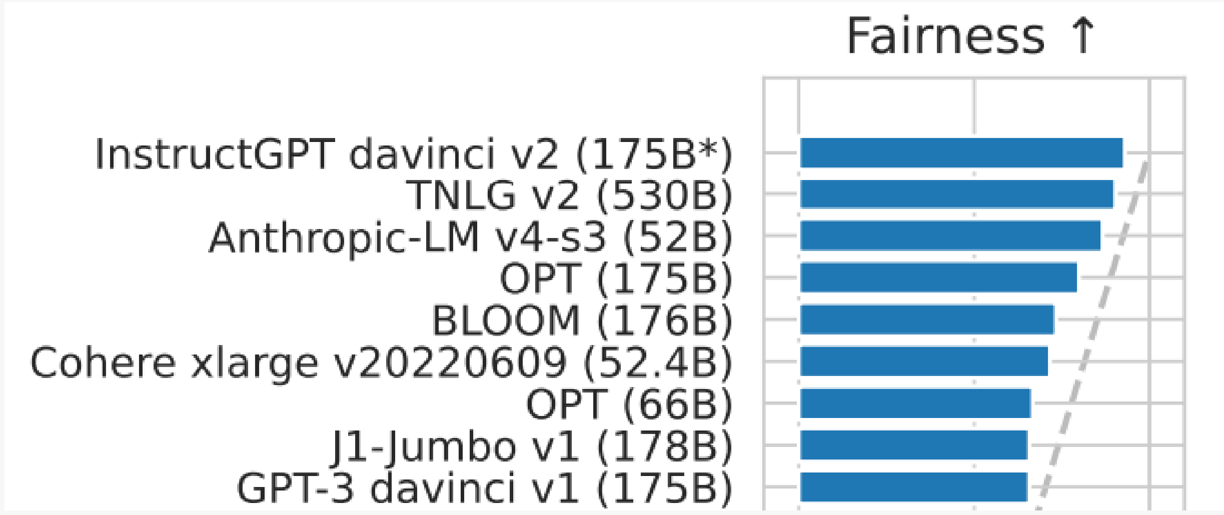
Online RLHF


Pretrained LM
Preference Model Pretraining (PMP)
Human-Feedback Fine-Tuning
Preference Model
Human-Feedback Comparison Data
HHH prompt context distillation
Initial Policy
RLHF (PPO)
RLHF Policies
Human Feedback Interface

Holistic Evaluation of Language Models (HELM)



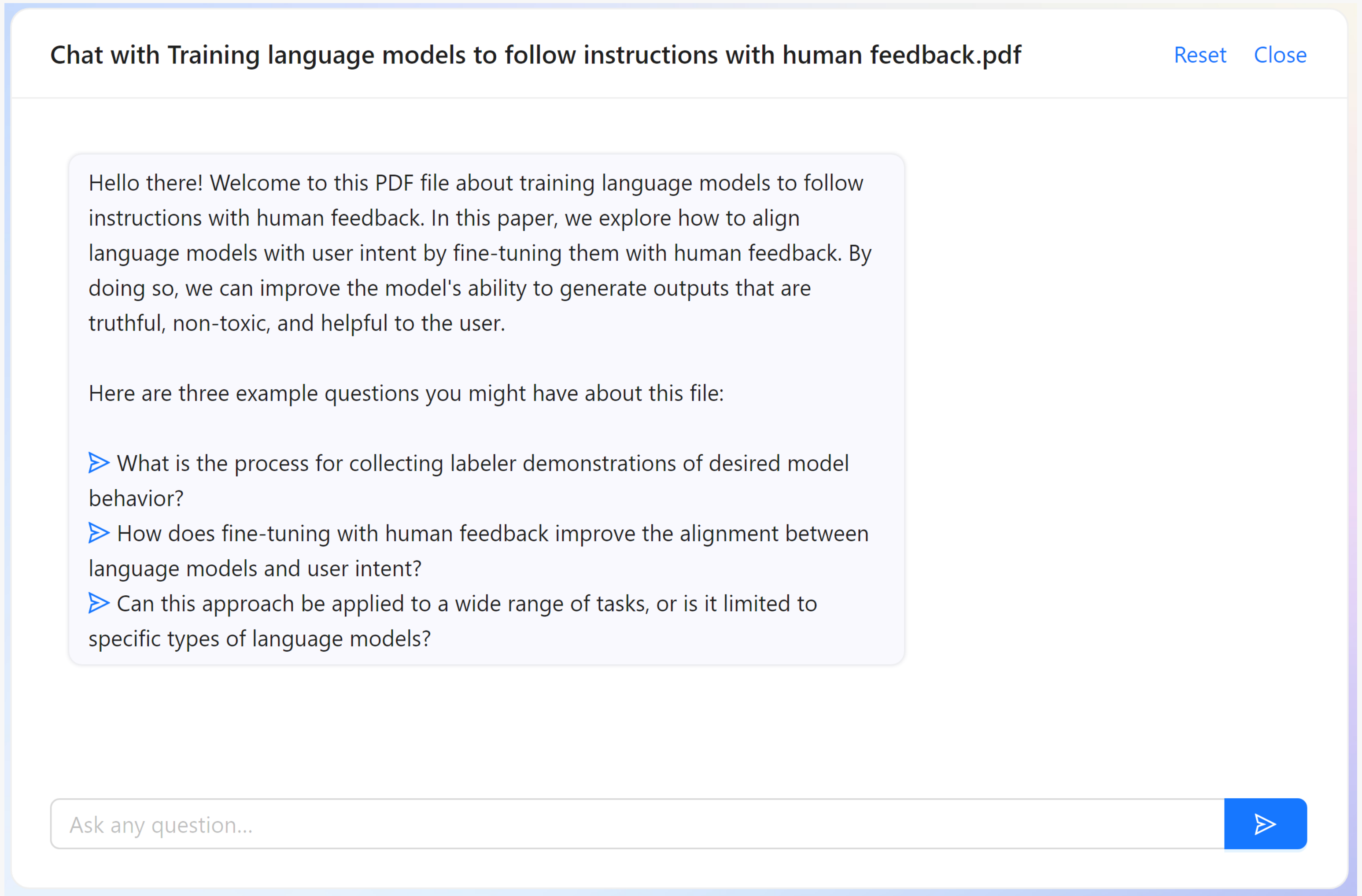




Liang P, Bommasani R, Lee T, et al. Holistic evaluation of language models[J]. arXiv preprint arXiv:2211.09110, 2022.

ChatPDF

- https://www.chatpdf.com/



Chat with Training language models to follow instructions with human feedback.pdf

Hello there! Welcome to this PDF file about training language models to follow instructions with human feedback. In this paper, we explore how to align language models with user intent by fine-tuning them with human feedback. By doing so, we can improve the model's ability to generate outputs that are truthful, non-toxic, and helpful to the user.

Here are three example questions you might have about this file:

▷ What is the process for collecting labeler demonstrations of desired model behavior?
▷ How does fine-tuning with human feedback improve the alignment between language models and user intent?
▷ Can this approach be applied to a wide range of tasks, or is it limited to specific types of language models?