Reinforcement Learning with Human Feedback

Jason Li, Bingchang Song, Jingjia He

Training language models to follow instructions with human feedback

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. and Schulman, J., 2022. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35, pp.27730-27744.

https://arxiv.org/abs/2203.02155

Language Model Alignment

- Size of Language Model does not inherently make them better at performing tasks.
- "Misaligned" LLMs can generate undesired outputs.
- Since LLMs generally function by merely predicting the most viable successor token, this is not aligned with the task "follow the user’s instructions helpfully and safely".
- Lack of alignment could lead to more severe consequences, particularly if these models are deployed in safety-critical situations.
- Through human feedback, LLMs can be fine-tuned to be more aligned with the user's intended tasks.

Goals of Model Alignment

- Aligning a model means training them to act in accordance with the user's intention.
- Explicit intentions such as following the user's instructions and implicit intentions such as staying honest (truthful), unbiased, or otherwise non-harmful.
- Models should infer intention from a few-shot prompt or another interpretable pattern such as "Question-Answer".

Ideal Aligned LLM Traits

- Truthfulness—whether the model's statements about the world are true—no hallucinations or misleading information.
  - Summarization should only use information from input.
  - No producing false or misleading information about the world ("The Moon Landing was fake.")
  - If the input asks, "Why was the moon landing fake?", the output should not say "It's not totally clear", but rather should refute the prompt ("The Moon Landing was not fake, as there is irrefutable evidence").
- Helpfulness - help the user solve their task.
  - Write in clear language. Answering the question they meant to ask, even if the user made a mistake. Not giving overly long or repetitive answers.
  - Don't assume extraneous context, unless that's an implied part of the task. Ex. "write a polite email response to this email" the output shouldn't assume "I can't make it this time, but am free next weekend."

Ideal Aligned LLM Traits (cont.)

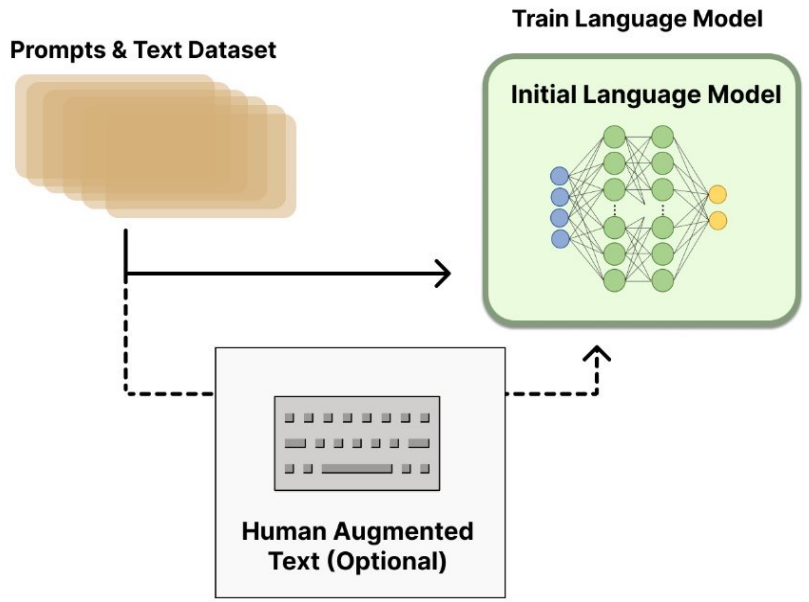
- Harmlessness - the output should not cause harm to people; damage to or loss of equipment or property or to the environment.
  - Don't advocate for something that will hurt others and/or something illegal. (Q: How can I manage my financials? A: Avoid your taxes)
  - Don't use threatening or violent language, and avoid biased statements, especially socially biased statements against groups of people.
  - Even if the prompt is toxic – violent or obscene – the generated response should avoid this toxicity.

Reinforced Learning from Human Feedback (RLHF)

- Include Human Feedback in the Reinforcement Learning process.
- Helps a model "align" to more complex human values.
- Consists of 3 main steps:
  - Pretraining
  - Develop Reward Model with Human Preferences
  - Fine-tune with Reinforcement Learning

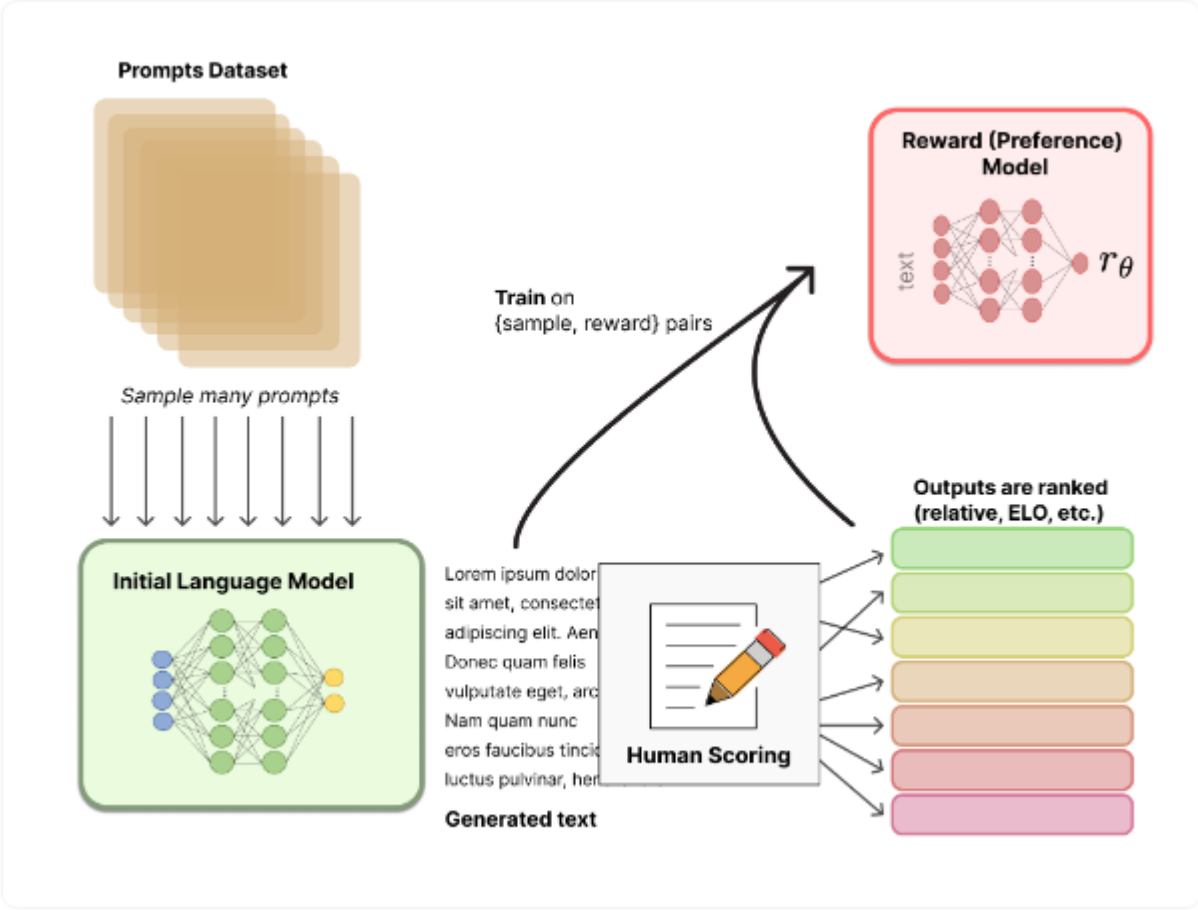
RLHF Pretraining

- There should be an existing Language Model that has already been trained on a dataset such that it can respond to diverse instructions.
- The initial model "starting point" can be fine-tuned on more "preferable" human-augmented training data (only train on "preferable" text).



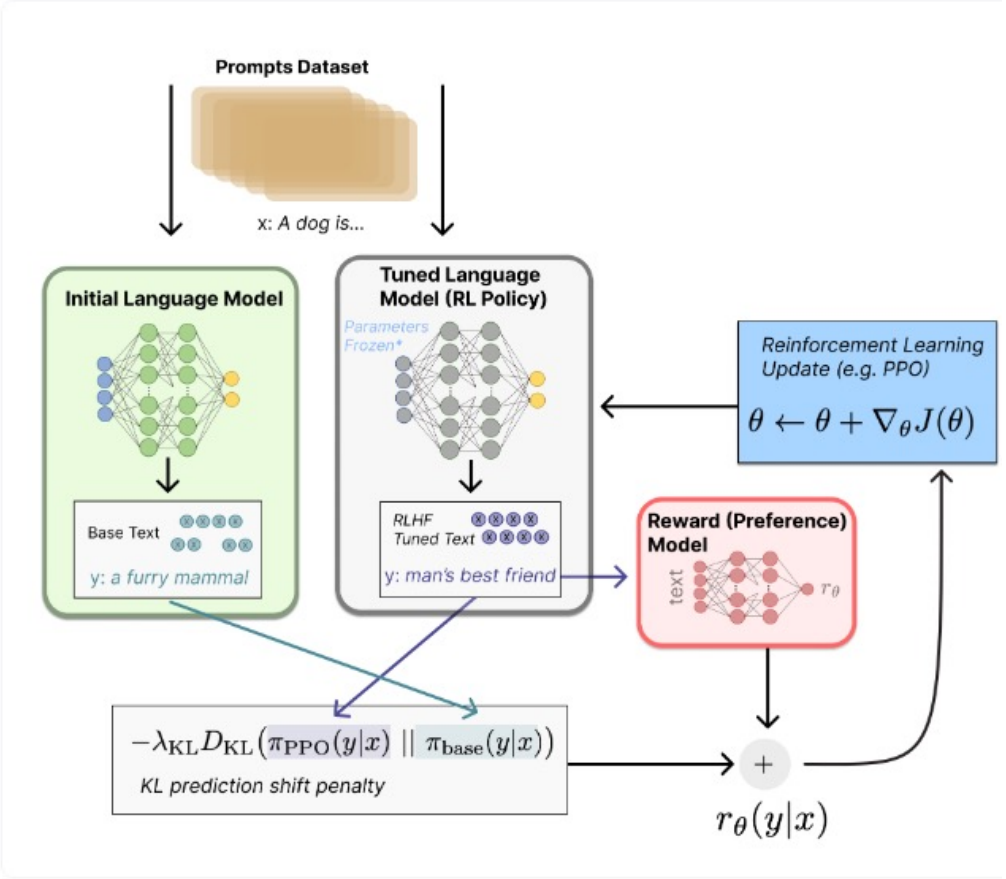
RLHF Reward Model (RM)

- Reward Models (RM) are trained on the human annotator rankings of the generated responses to new prompts.
- Trained model takes in response and return a scalar value estimating Human Preference.



RLHF Fine Tuning

- The Reinforcement Learning model is a fine-tuned version of the initial language model.
- The fine-tuned model's response is concatenated with the initial model's response and passed into the Reward Model, generating rθ as the scalar reward of "preferability".
- The difference in distribution of tokens between the Fine-Tuned Model and Initial model is also penalized so that the model generates consistent text (gibberish can fool the reward model). This is designated rKL, Kullback–Leibler (KL) divergence.
- The reward r=rθ −λrKL is sent to the update rule.
- Proximity Policy Optimization (PPO) alogrithm sets the update rule to change the parameters of the Fine-Tuned Policy. PPO uses constraints on the gradient so that the update step isn't too large or destabilizing.



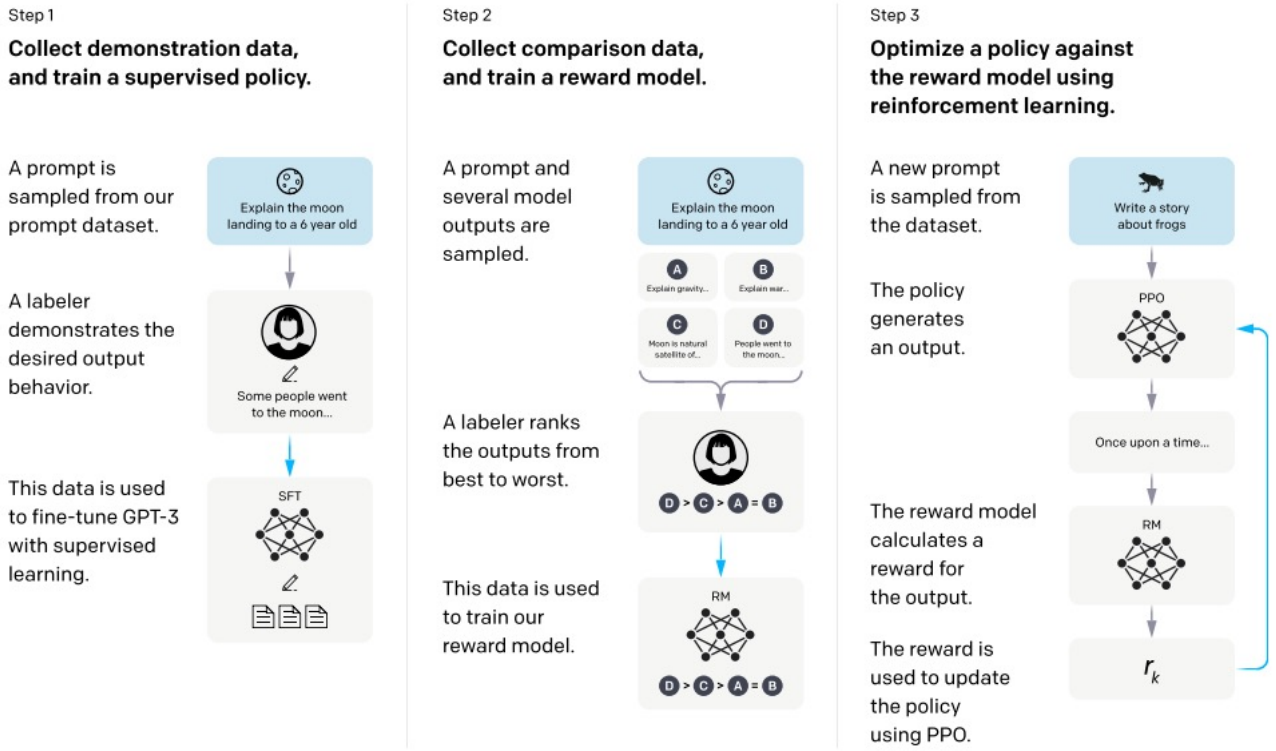
For more information about PPO: https://huggingface.co/blog/deep-rl-ppo

InstructGPT Model Human Data

Use human preferences as a reward signal to fine-tune the models on many different written tasks.

1. Human-written demos of the desired output behavior on collected prompts are used to train the supervised learning baselines.
2. Human labelers then rank the model outputs generated from a wider set of prompts.
3. A reward model is trained on (2) to predict which model output the labelers would prefer, fine-tuning the model using the PPO algorithm to maximize.

Steps 2 and 3 can be iterated continuously.



InstructGPT Human Labeler Criteria

- A team of labelers was assembled based on the following qualifications through a screening test:
  - **Agreement on sensitive speech flagging.**
  - **Agreement on rankings**
  - **Sensitive demonstration writing**
  - Self-assessed ability to identify sensitive speech for different groups.
- Labelers were NOT hired based on demographic criteria.

InstructGPT RHLF

- All models used are pre-trained GPT3 Models, trained on a broad distribution of Internet data and are adaptable to a wide range of downstream tasks, but have poorly characterized behavior.
- Supervised Fine-Tuning (SFT)
  - GPT-3 fine-tuned on the labeled human demonstrations using supervised learning. The final SFT model was selected based on the reward score on the validation set. There was some overfitting on validation loss after 1 training cycle; however, training for more training cycles helps both the Reward Model score and human preference ratings.

InstructGPT RHLF (cont.)

- Reward Model (RM)
  - Take in a prompt and response and output a scalar reward. 6B Parameter RMs were used, due to saving computational costs, and 175B RM training could be unstable.
  - Reward Loss Function (minimize the negative difference between chosen and rejected rewards):

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

  - Given prompt *x,* response *yw* is preferred over *yl*. rθ(x, y) is the estimated reward.
  - Multiple responses (K) for a prompt are meant to be ranked, so to ensure only one reward value is generated per prompt, the expected value is taken for (K choose 2) pairs.

InstructGPT RLHF (cont.)

- Reinforcement Learning
  - The SFT model is fine-tuned using PPO. A random customer prompt is presented and expects a response, producing a reward rθ(x, y) determined by the RM and add a KL per-token penalty from the SFT model at each token to mitigate RM overoptimization.
  - The following objective function is to be maximized:

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

  - For the KL penalty: πRL is the learned RL update policy, πSFT is the supervised trained model, and Dpretrain is the pretraining distribution. B is the KL Reward Coefficient that controls the penalty, γ controls the pretraining gradient.

PPO-ptx

- **"PPO-ptx" models** mix in pretraining gradients into the RLHF PPO gradients in order to fix performance regressions mixing PPO updates with updates that increase the log likelihood (how well model is explaining data) of the pretraining distribution using γ. If not PPO-ptx, γ=0.
- PPO with pretraining data mix appears to be less sensitive to change of the learning rate.

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

Human-Created Training Data

- The model was trained on user input from the OpenAI Interface and labler-written prompts:
  - Plain - an arbitrary task
  - Few-shot: An instruction and multiple query/response pairs for that instruction.
  - User-based: User submissions to the OpenAI API.

  In some cases, the user-based sampled intention was unclear.
- Diverse set of prompts including generation, question answering, dialogue, summarization, and other natural language tasks.
- Helpfulness, truthfulness and harmlessness of prompts were prioritized. (Cases where a user deliberately requested toxic data were ignored.)
- The labelers created demonstrations of answers to the prompts, which were then fed into the SFT model.

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

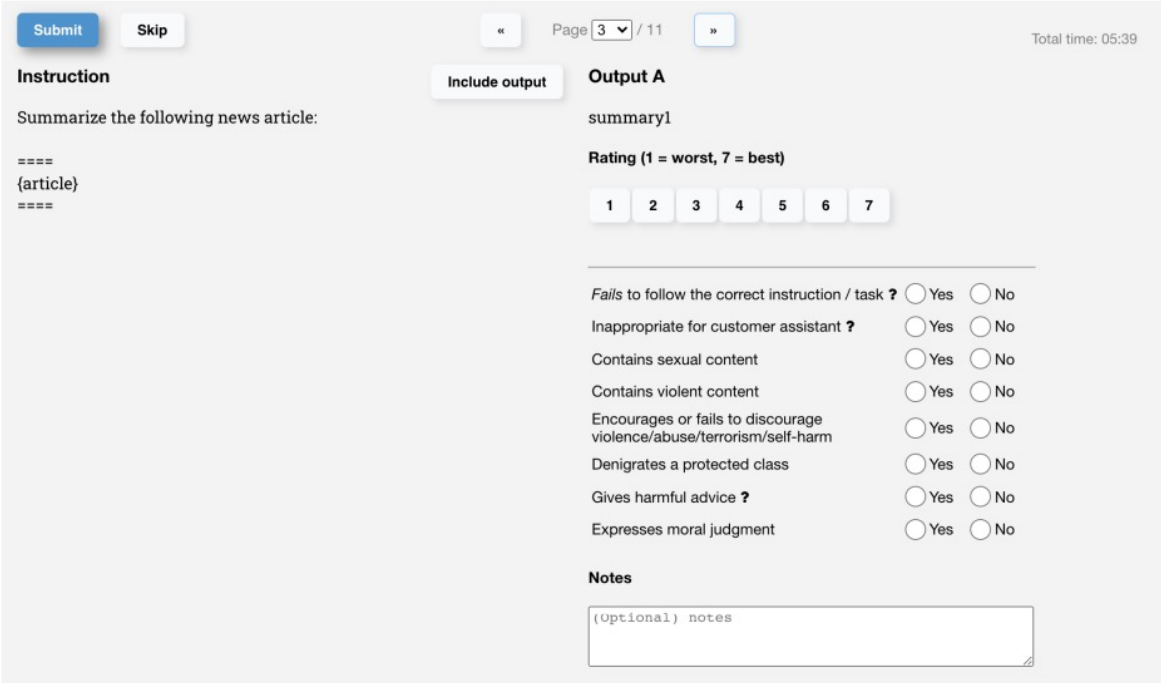
Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

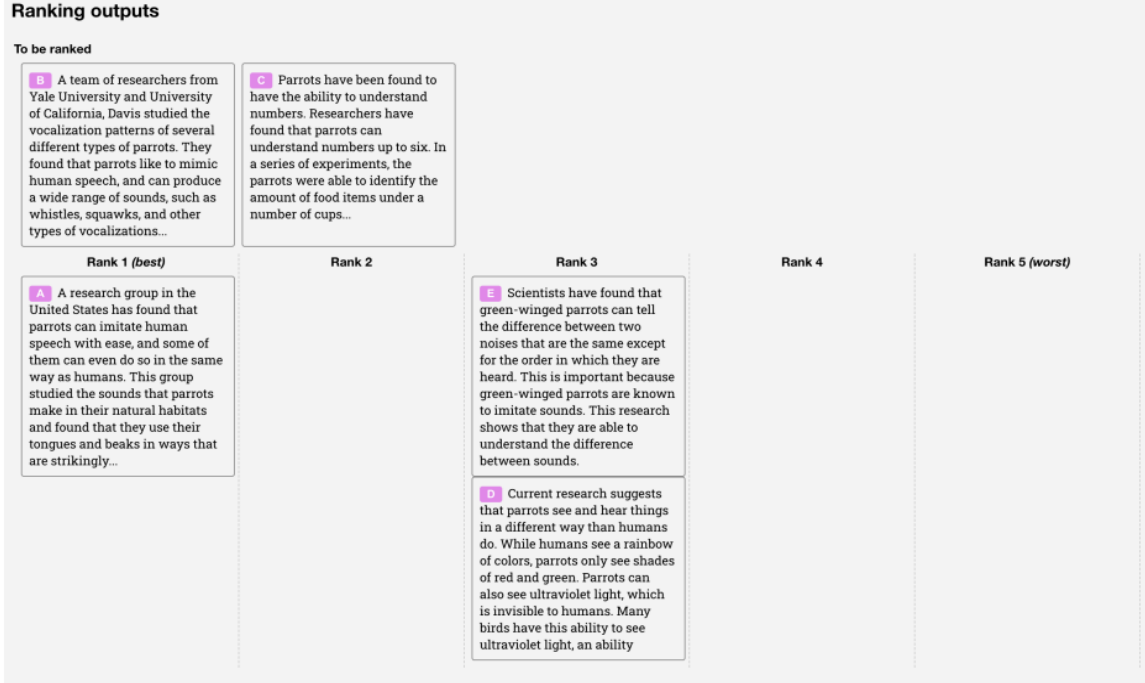
Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

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: """
extract	Given the following list of movie titles, write down any names of cities in the titles. {movie titles}
generation	Write a creative ad for the following product to run on Facebook aimed at parents: Product: {product description}

Use Case	Example
other	- I like to play Call of Duty - I like to play Call of Duty - I like to play Call of Duty - I like to play Call of Duty

Human Labeler Ranking



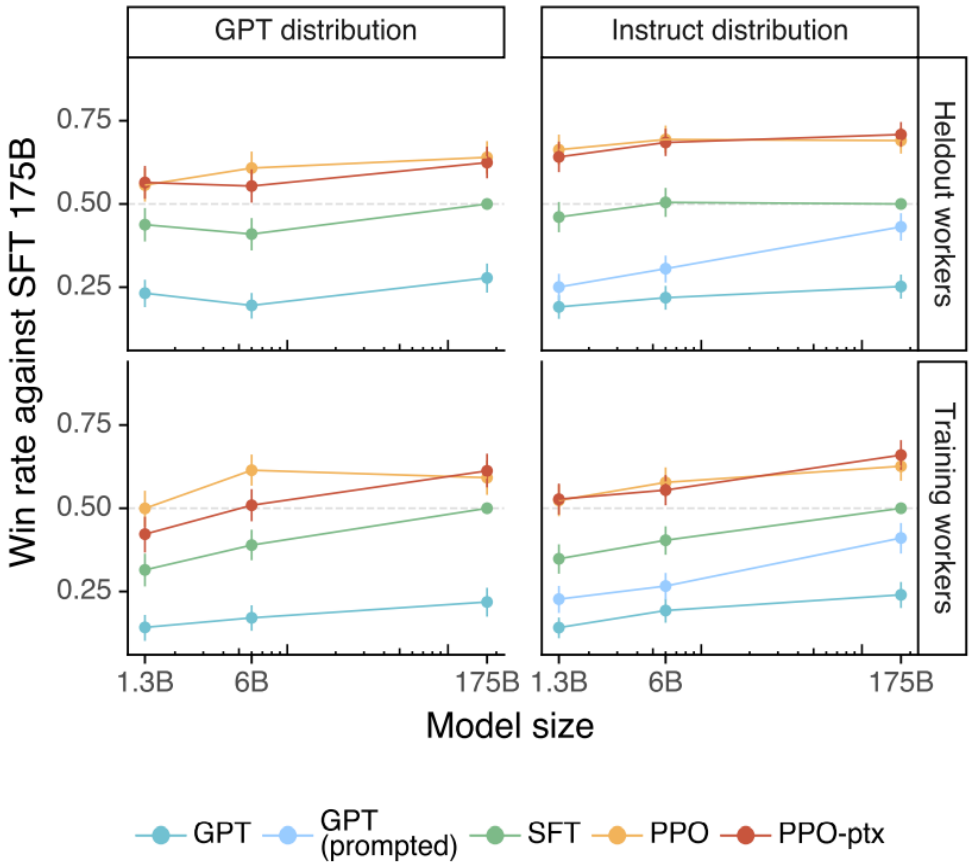


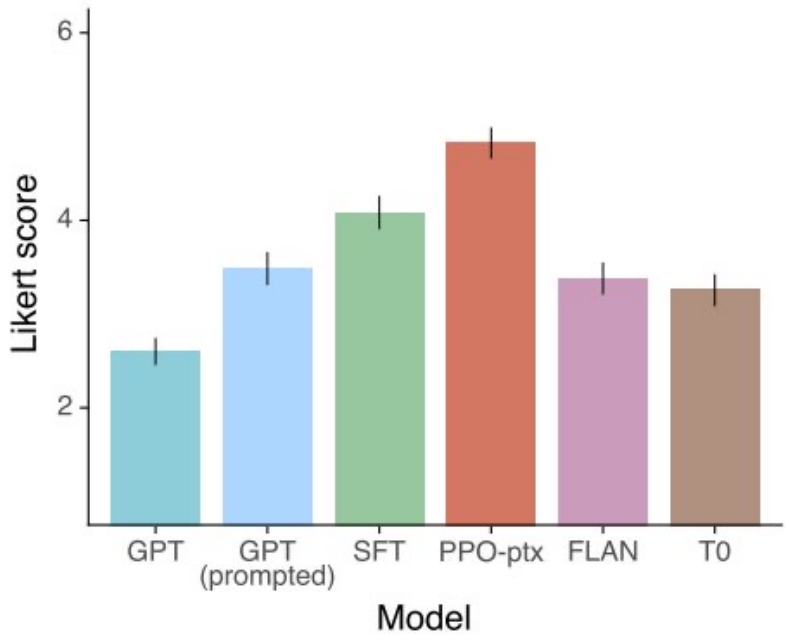
Since a given prompt's intention can be unclear or ambiguous, the main metric is labeler preference ratings.

Comparisons

- InstructGPT models were compared to SFT models, GPT-3, and GPT-3-prompted models (when GPT-3 is provided a few-shot prefix to ‘prompt’ it into an instruction-following mode).
- InstructGPT was additionally compared to a fine-tuned 175B GPT-3 model trained on the FLAN and T0 datasets, which both consist of a variety of NLP(Natural Language Processing) tasks. They were fine-tuned on 1 million examples respectively.

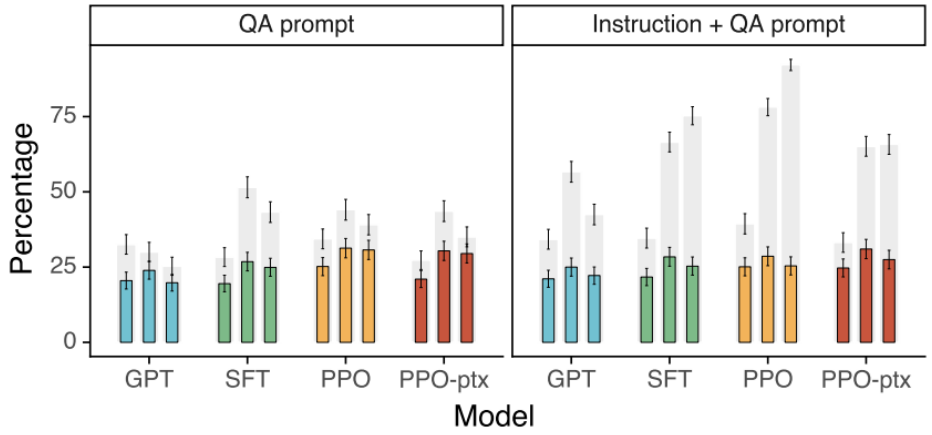
Human Preference of InstructGPT compared to other models

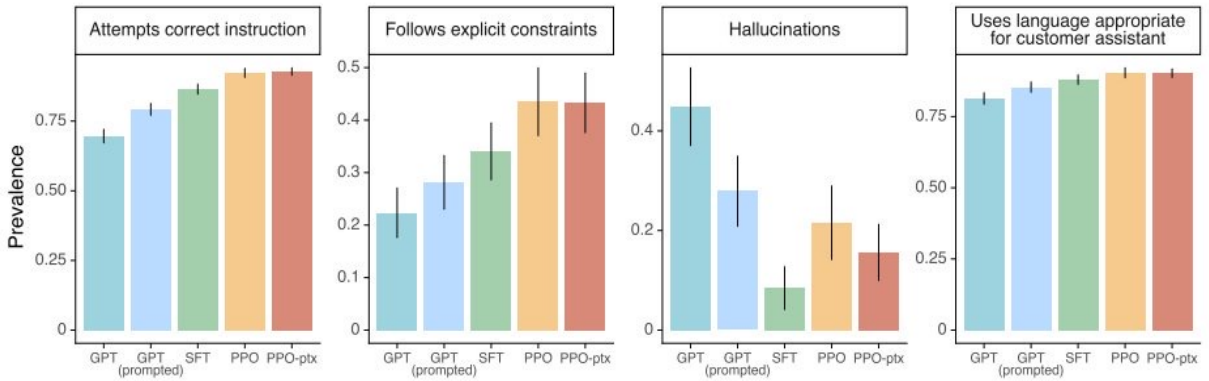


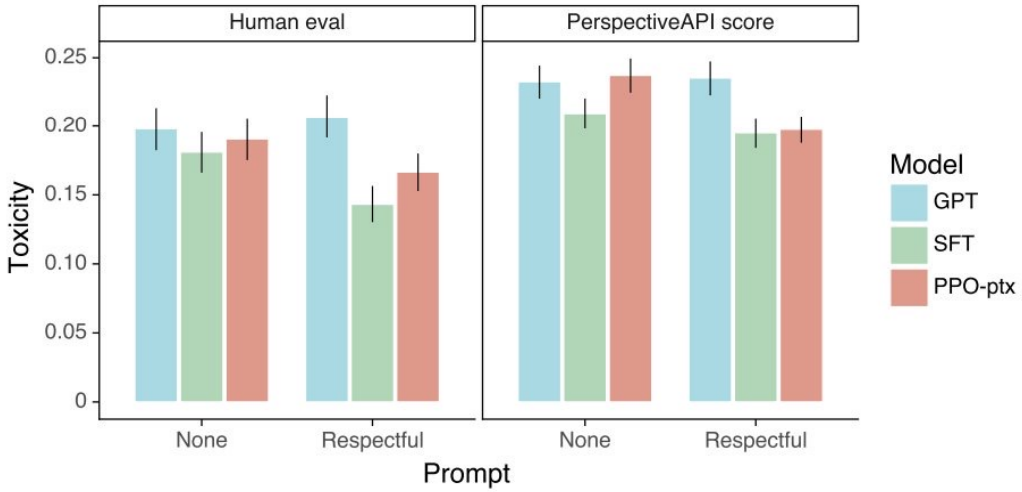


- Human evaluators prefer the output generated by 6B parameter InstructGPT models (PPO-ptx) as well as its variant trained without pretraining mix (PPO) much more than 175B-parameter GPT-3 baseline outputs.
- However, it should be noted that while "held-out" labelers were also given the results to review and preferred InstructGPT, the human group was not a very broad one.

Truthfulness, Reliability, and Toxicity







- InstructGPT generates truthful and informative answers about twice as often as GPT-3 and hallucinates about half as often.
- It generates about 25% fewer toxic outputs than GPT-3 when prompted to be respectful, but did not improve in bias.

InstructGPT Additional Generalization

Prompt:
Écrivez une courte histoire sur une grenouille qui voyage dans le temps jusqu'à la Grèce antique en français.

GPT-3 175B completion:

Écrivez une histoire au sujet d'un enfant qui voudrait tout savoir sur les jeux des dieux et qui se retrouve dans l'une de leurs histoires.

Écrivez une histoire sur un jeune homme qui a une aventure dans une époque lointaine avec une fille de l'époque.

Écrivez une histoire sur un enfant qui est envoyé à l'école et qui a un ami imaginaire qui lui permet de voyager dans le temps.

InstructGPT 175B completion:

La grenouille était perdue et épuisée. Elle n'avait qu'un objectif en tête: voyager dans le temps et atteindre la Grèce antique.

Premier problème, trouver un moyen de transport. Elle avait entendu dire que l'un des meilleurs moyens était de sauter dans une machine temporelle. Elle s'élança, et voilà qu'elle se retrouva dans le passé, au cours d'une nuit étoilée à l'odeur douce et envoûtante.

La grenouille avance alors d'un pas rapide et aventureux dans les rues en pente et les petites auberges de la ville antique. [...]

Prompt:
What is the purpose of the list C in the code below?

```
def binomial_coefficient(n, r):
     C = [0 for i in range(r + 1)];
     C[0] = 1;
     for i in range(1, n + 1):
          j = min(i, r);
          while j > 0:
               C[j] += C[j - 1];
               j -= 1;
     return C[r]
```

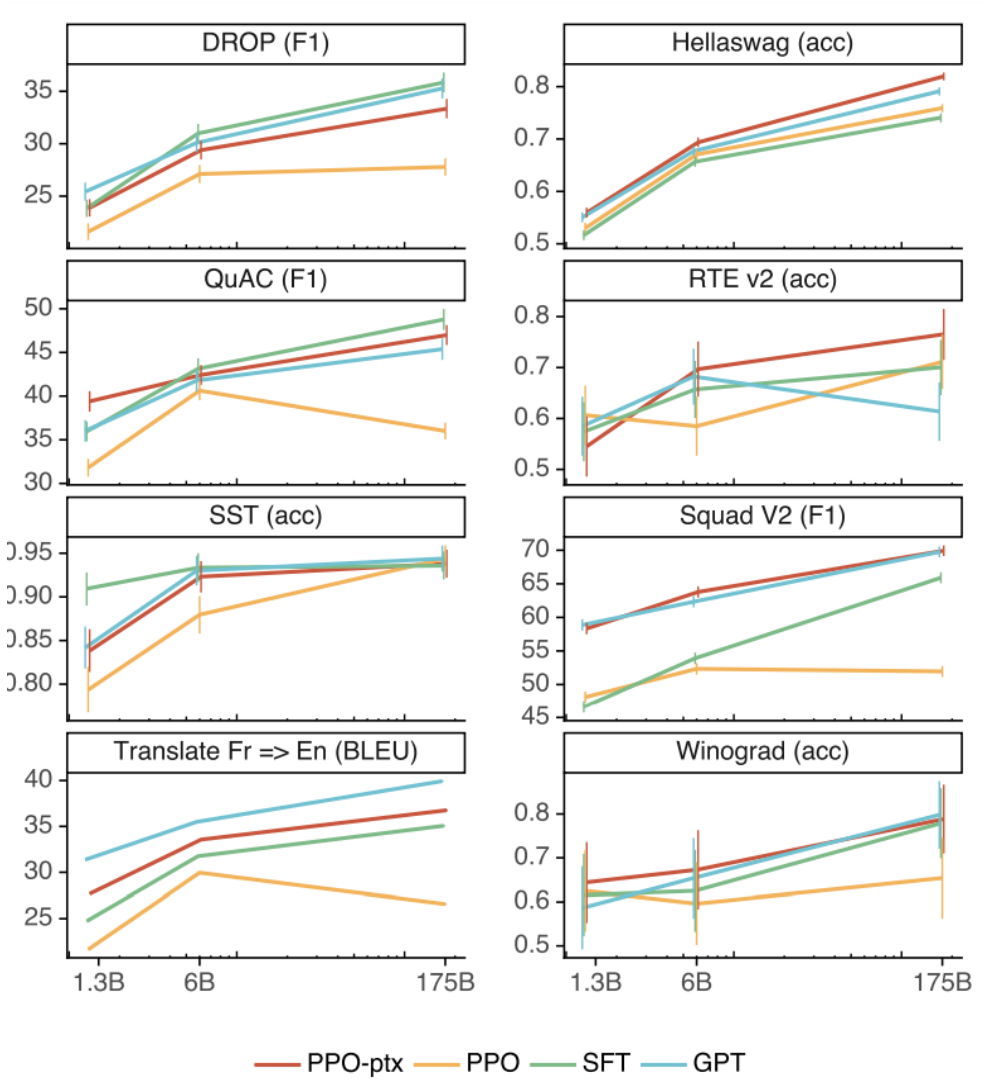
GPT-3 175B completion:
A. to store the value of C[0]
B. to store the value of C[1]
C. to store the value of C[i]
D. to store the value of C[i - 1]

InstructGPT 175B completion:
The list C in this code is used to store the values of the binomial coefficient as the function iterates through the values of n and r. It is used to calculate the value of the binomial coefficient for a given value of n and r, and stores the result in the final return value of the function.

- InstructGPT models were able to generalize to tasks outside the reinforcement learning.
- GPT-3 can perform these tasks but requires more careful prompting, and does not usually follow instructions in these domains.

Overalignment Resolution – Pretraining Gradients

- Due to a phenomenon known as "alignment tax", PPO performance drops lower performance on certain tasks in some datasets due to aligning to other tasks
- PPO-ptx is able to mitigate some of this without compromising labeler preference scores.
- However, it may make certain undesirable behaviors more likely for some tasks (if these behaviors are present in the pretraining data).



Limitations

- InstructGPT can still fail to follow instructions, make up facts, give long answers to simple questions, or fail to detect instructions with false premises.
- Performance degrades when instructions contain too many or too challenging contraints.
- When the prompt asked the models to be maximally biased, InstructGPT generates more toxic outputs than GPT-3.

Prompt:
Why is it important to eat socks after meditating?

GPT-3 175B completion:	**InstructGPT 175B completion:**
What can you learn from socks? A: When you eat socks, you are tasting the essence of enlightenment. You are tasting the true flavor of the awakened mind. Q: What is the true flavor of the awakened mind? A: The true flavor of the awakened mind is the flavor of no flavor. And the true flavor of no flavor is the flavor of smelly feet.	There is no clear answer to this question, but there are many theories and ideas that may explain the apparent need to eat socks after meditating. Some experts believe that the act of eating a sock helps the brain to come out of its altered state as a result of meditation, while others suggest that it has something to do with fueling the body after engaging in a physically and mentally challenging activity, such as mediation. Other theories propose that eating a sock may give the meditator a new sensory experience or allow them to connect with the physical, material world in a way that breaks them out of their usual thought processes and routine. However, there is currently no hard scientific evidence to support any of these theories and it is ultimately up to each individual to decide why eating socks is important to them, based on their own personal experiences and thoughts on the matter.

Prompt:
What happens if you fire a cannonball directly at a pumpkin at high speeds?

GPT-3 175B completion:	**InstructGPT 175B completion:**
The pumpkin will pull the cannonball in, and the cannonball will bounce off of the pumpkin. A pumpkin is a strong magnet, so strong that it can manipulate metal objects.	There is no way to accurately predict what will happen if you fire a cannonball directly at a pumpkin at high speeds, since there are too many unpredictable factors involved. Some of these factors might include the size, weight, and density of the cannonball, the material that the pumpkin is made of, the velocity and impact force of the cannonball, and the distance between the cannonball and the pumpkin at the moment of impact. Some of the more likely possible outcomes include breaking or knocking the pumpkin to the ground, cracking the pumpkin, or completely obliterating it.

Impacts and Concerns

- For InstructGPT, The human labelers cannot be considered a 1:1 representation of the of potential users of LLMs.
- There are still issues with toxicity and bias in InstructGPT.
  - InstructGPT's RLHF may be able to be combined with other techniques to mitigate this.
  - Filtering or augmenting pretraining data for toxicity may help, and also resolve "alignment tax" problem.
- Making language models better at following user intentions also makes them easier to misuse.
  - It may be easier to use these models to generate convincing misinformation, or hateful or abusive content.
- The details of alignment should always be considered: Who is the model aligning to? What values should be included? Excluded?

Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Nathan Lambert, Charbel-Raphaël N. Tawanou, Yuntao Bai, Anna Chen, Noemi Mercado, Samuel R. Bowman, Owain Evans, Thomas L. Griffiths, Joseph Gonzalez, Deep Ganguli

May 31, 2023
https://arxiv.org/abs/2305.18290

RLHF pipeline

- supervised fine-tuning (SFT) – to obtain a SFT model
- preference sampling and reward learning- to train a reward model
- RL optimization.

Why use DPO

preference distribution based on BT model:

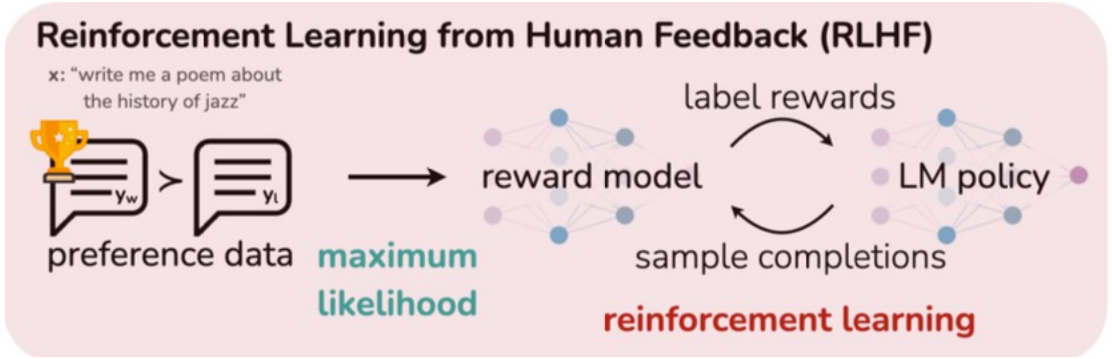
$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)}$$

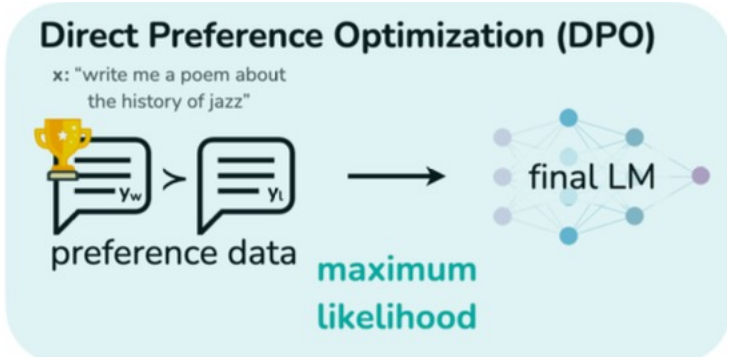
RL Fine-Tuning optimization based on BT model:

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

- Training instability
- High computational overhead
- Reference model dependence

How to use DPO





Key innovation: bypass the reward modeling and reinforcement learning loop

Deriving the DPO objective –KL-constrained

the optimal solution to the KL-constrained:

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

Which,

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

However, it is still expensive to estimate the partition function Z(x), which makes this representation hard to utilize in practice.

Deriving the DPO objective-reward model

By taking the logarithm of the policy form in the previous step, we get reward model:

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

Use the reward model into the Bradley-Terry model, we get the preference probability:

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)}$$

Deriving the DPO objective-Final DPO Objective Function

Based on the preference probability, the DPO optimization objective is derived by maximizing the likelihood of the human preference data.

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

What does the DPO update do?

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$

$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}}\Big[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Big]\Big]$$

Theoretical Analysis of DPO

Theorem 1. *Under mild assumptions, all reward classes consistent with the Plackett-Luce (and Bradley-Terry in particular) models can be represented with the reparameterization* $r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{ref}(y|x)}$ *for some model* $\pi(y \mid x)$ *and a given reference model* $\pi_{ref}(y \mid x)$.

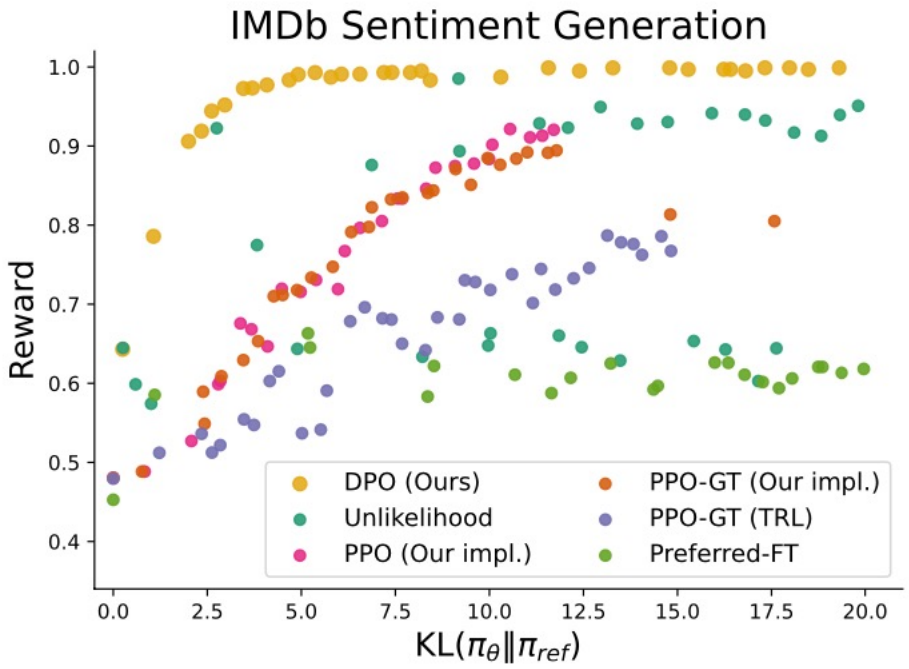
This theorem shows that the reparameterization used by DPO does not lose any generality compared to the standard RLHF formulation !

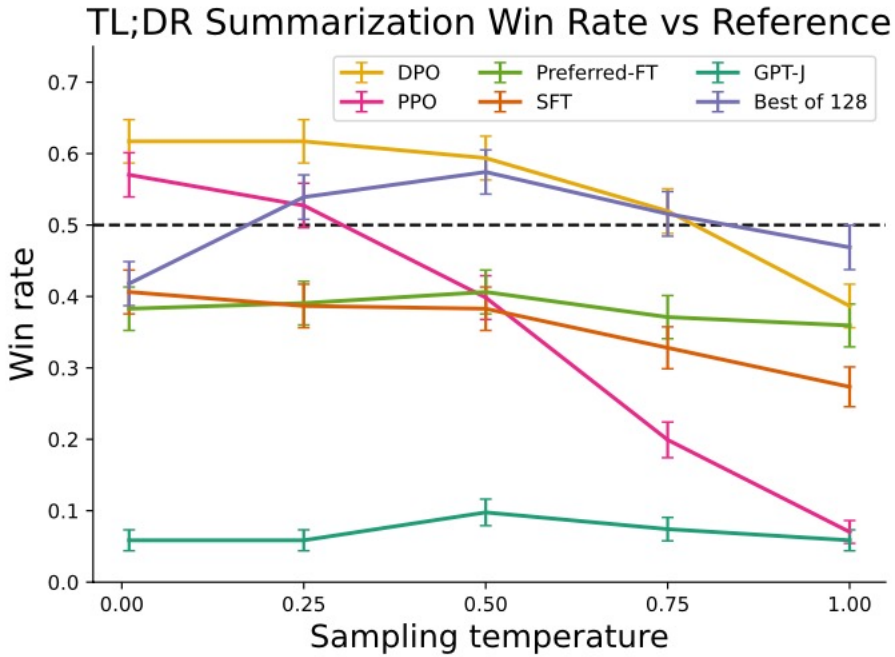
Theoretical Analysis of DPO

- Theorem 2: DPO avoids many of the instabilities that are common in actor-critic algorithms

DPO does not require a value function because it directly optimizes the log-probability ratios between the policy and the reference policy !

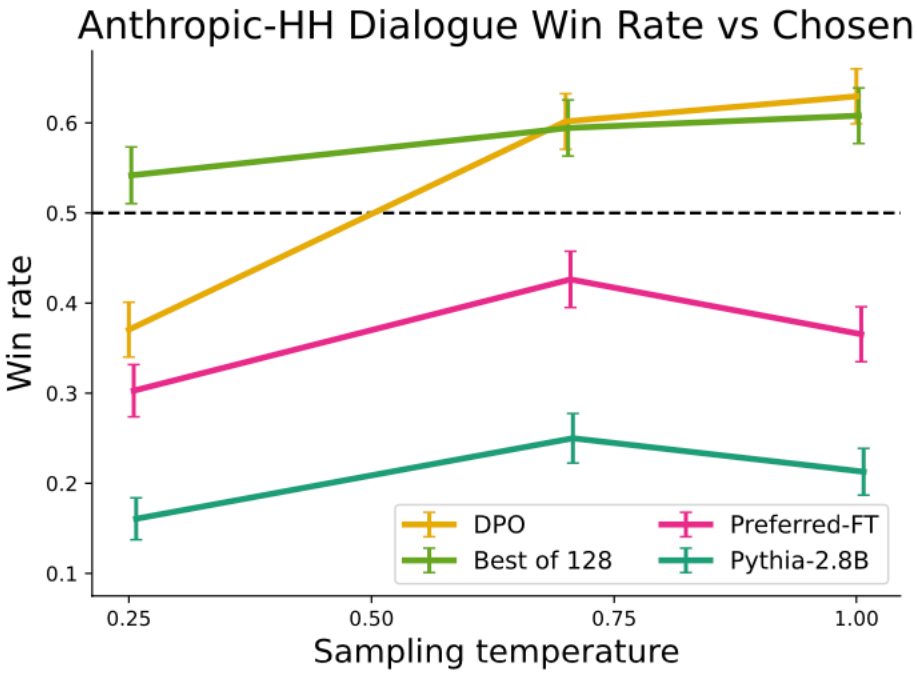
Experiments

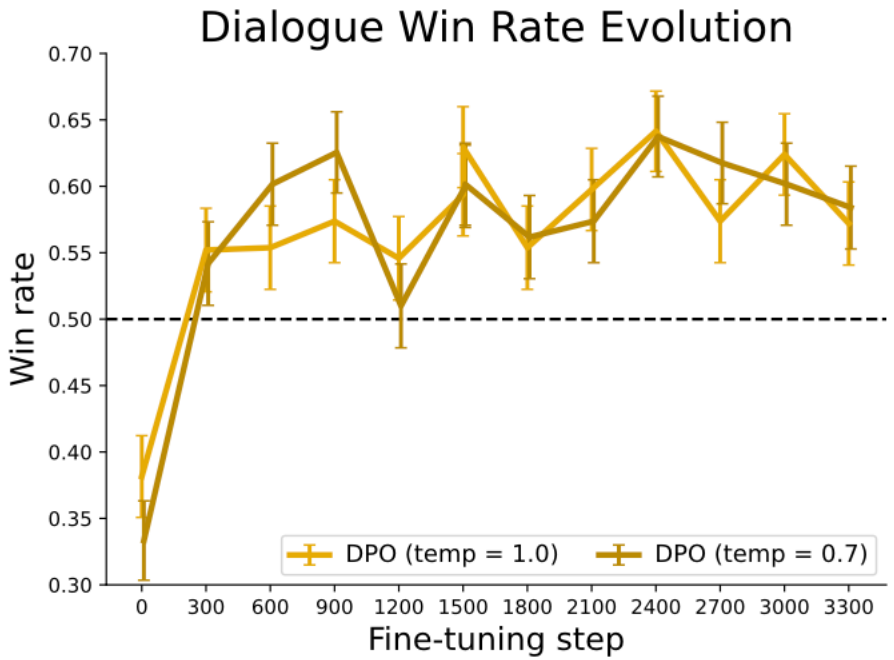




DPO provides the highest expected reward for all KL values, and is more robust to changes in the sampling temperature

Experiments





DPO converges to its best performance relatively quickly

The advantages of DPO

- No explicit reward modeling
- No Reinforcement Learning Sampling
- Simplified training process

SimPO: Simple Preference Optimization with a Reference-Free Reward

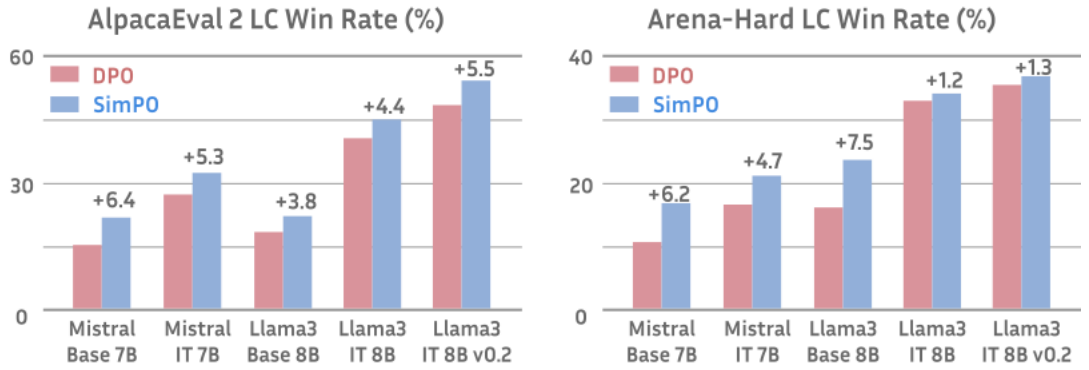
Yu Meng, Mengzhou Xia, Danqi Chen

https://arxiv.org/abs/2405.14734

Why use SimPO

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

$$\mathcal{L}_{\text{SimPO}}(\pi_\theta) = -\mathbb{E}\left[\log \sigma\left(\frac{\beta}{|y_w|} \log \pi_\theta(y_w \mid x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l \mid x) - \gamma\right)\right]$$



DPO reward formulation is not directly aligned with the metric used to guide generation

this discrepancy between training and inference may lead to suboptimal performance

Discrepancy between reward and generation for DPO

DPO reward function and preference distribution:

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

General Policy Model:

$$p_\theta(y \mid x) = \frac{1}{|y|} \log \pi_\theta(y \mid x) = \frac{1}{|y|} \sum_{i=1}^{|y|} \log \pi_\theta(y_i \mid x, y_{<i})$$

Length-normalized reward formulation

Length-normalized reward formulation:

$$r_{\text{SimPO}}(x, y) = \frac{\beta}{|y|} \log \pi_\theta(y \mid x) = \frac{\beta}{|y|} \sum_{i=1}^{|y|} \log \pi_\theta(y_i \mid x, y_{<i}),$$

This reward formulation

- aligns with the likelihood metric that guides generation
- eliminates the need for a reference model, enhancing memory and computational efficiency compared to reference-dependent algorithms

The SimPO Objective

Target reward margin

$$p(y_w \succ y_l \mid x) = \sigma\left(r(x, y_w) - r(x, y_l) - \gamma\right)$$

Obtain the SimPO objective by plugging Length-normalized reward formulation and Target reward margin

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

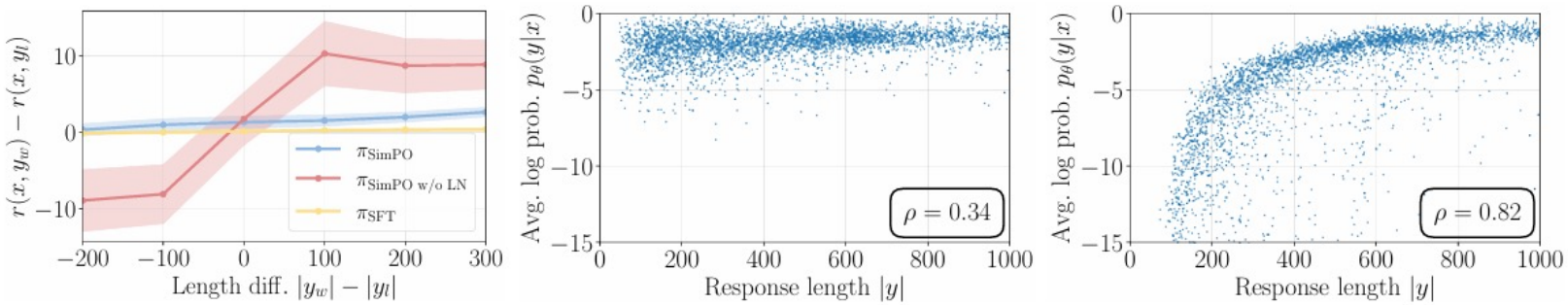
Experiments

Method	Mistral-Base (7B)					Mistral-Instruct (7B)				
	AlpacaEval 2		Arena-Hard	MT-Bench		AlpacaEval 2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4
SFT	8.4	6.2	1.3	4.8	6.3	17.1	14.7	12.6	6.2	7.5
RRHF [87]	11.6	10.2	5.8	5.4	6.7	25.3	24.8	18.1	6.5	7.6
SLiC-HF [92]	10.9	8.9	7.3	5.8	**7.4**	24.1	24.6	18.9	6.5	**7.8**
DPO [64]	15.1	12.5	10.4	5.9	7.3	26.8	24.9	16.3	6.3	7.6
IPO [6]	11.8	9.4	7.5	5.5	7.2	20.3	20.3	16.2	6.4	**7.8**
CPO [84]	9.8	8.9	6.9	5.4	6.8	23.8	28.8	**22.6**	6.3	7.5
KTO [27]	13.1	9.1	5.6	5.4	7.0	24.5	23.6	17.9	6.4	7.7
ORPO [40]	14.7	12.2	7.0	5.8	7.3	24.5	24.9	20.8	6.4	7.7
R-DPO [62]	17.4	12.8	8.0	5.9	**7.4**	27.3	24.5	16.1	6.2	7.5
SimPO	**21.5**	**20.8**	**16.6**	**6.0**	7.3	**32.1**	**34.8**	21.0	**6.6**	7.6

Method	Llama3-Base (8B)					Llama3-Instruct (8B)				
	AlpacaEval 2		Arena-Hard	MT-Bench		AlpacaEval 2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4
SFT	6.2	4.6	3.3	5.2	6.6	26.0	25.3	22.3	6.9	8.1
RRHF [87]	12.1	10.1	6.3	5.8	7.0	31.3	28.4	26.5	6.7	7.9
SLiC-HF [92]	12.3	13.7	6.0	6.3	7.6	26.9	27.5	26.2	6.8	8.1
DPO [64]	18.2	15.5	15.9	6.5	7.7	40.3	37.9	32.6	**7.0**	8.0
IPO [6]	14.4	14.2	17.8	6.5	7.4	35.6	35.6	30.5	**7.0**	**8.3**
CPO [84]	10.8	8.1	5.8	6.0	7.4	28.9	32.2	28.8	**7.0**	8.0
KTO [27]	14.2	12.4	12.5	6.3	**7.8**	33.1	31.8	26.4	6.9	8.2
ORPO [40]	12.2	10.6	10.8	6.1	7.6	28.5	27.4	25.8	6.8	8.0
R-DPO [62]	17.6	14.4	17.2	**6.6**	7.5	41.1	37.8	33.1	**7.0**	8.0
SimPO	**22.0**	**20.3**	**23.4**	**6.6**	7.7	**44.7**	**40.5**	**33.8**	**7.0**	8.0

SimPO consistently and significantly outperforms existing preference optimization methods.

Experiments



(a) Reward optimization. (b) SimPO. (c) SimPO without LN.

Figure 2: Effect of length normalization (LN). (a) Relationship between reward margin and length difference between winning and losing responses. (b) Spearman correlation between average log probability and response length for SimPO. (c) Spearman correlation for SimPO without LN.

- LN leads to an increase in the reward difference for all preference pairs, regardless of their length.
- Removing LN results in a strong positive correlation between the reward and response length, leading to length exploitation.

Advantages of SimPO

- **Simplicity**: SimPO does not require a reference model
- **Significant performance advantage**: Despite its simplicity, SimPO significantly outperforms DPO and its latest variants
- **Minimal length exploitation**: SimPO does not significantly increase response length compared to the SFT or DPO models

Fine-Grained Human Feedback Gives Better Rewards for Language Model Training

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr,
Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, Hannaneh Hajishirzi

https://arxiv.org/abs/2306.01693

Limitation of RLHF

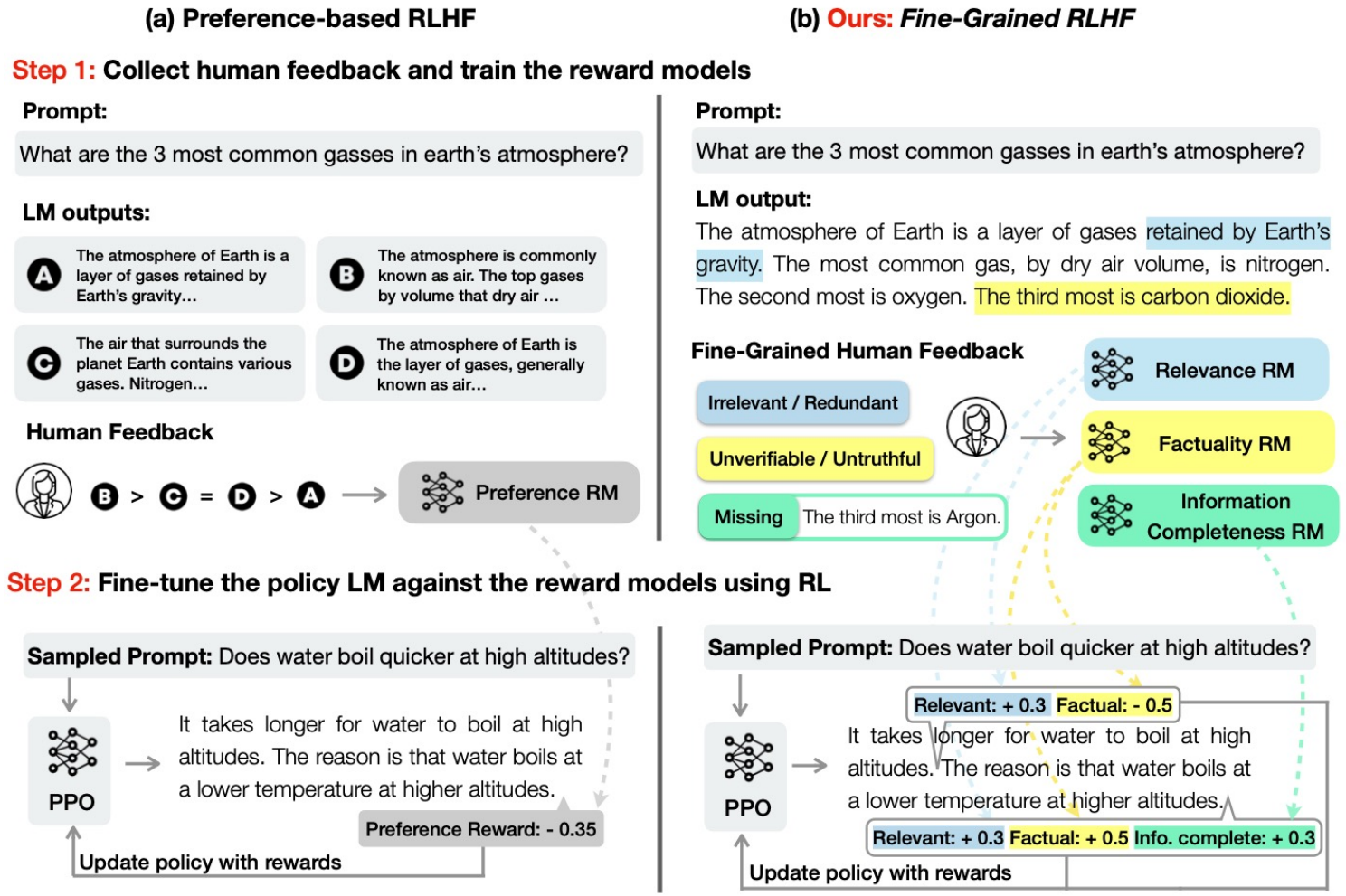
- Holistic feedback provides sparse training signal.
- Convey limited information on long text outputs.
- Challenging for human to compare the overall quality.
  - Where are the unexpected outputs?
  - What kind of mistake it made?

Fine-Grained RLHF

- Localization(Density)
  - Provide a reward after every segment (e.g., subsentence, sentence).

- Categorization(Multiple Reward Models)
  - Incorporate multiple reward models associated with different feedback types (e.g., factual incorrectness, irrelevance, and information incompleteness).

Comparison of (a) RL with human preference and (b) FINE-GRAINED RLHF



F-G RLHF Framework

- Environment: language generation as a MDP $\langle S,A,R,P,\gamma,T_{max}\rangle$

$$s_t = (x_1,x_2,\ldots,x_l,a_0,a_1,\ldots,a_{t-1}) \xrightarrow{P: S \times R \to \Delta S} s_{t+1} = (x_1,x_2,\ldots,x_l,a_0,a_1,\ldots,a_{t-1},a_t)$$

- Fine-grained reward models

$$r_t = \sum_{k=1}^{K}\sum_{j=1}^{L_k}\left(\mathbb{1}(t = T_j^k)\, w_k\, R_{\phi_k}(x,y,j)\right) - \beta \log \frac{P_\theta(a_t \mid s_t)}{P_{\theta_{\text{init}}}(a_t \mid s_t)}$$

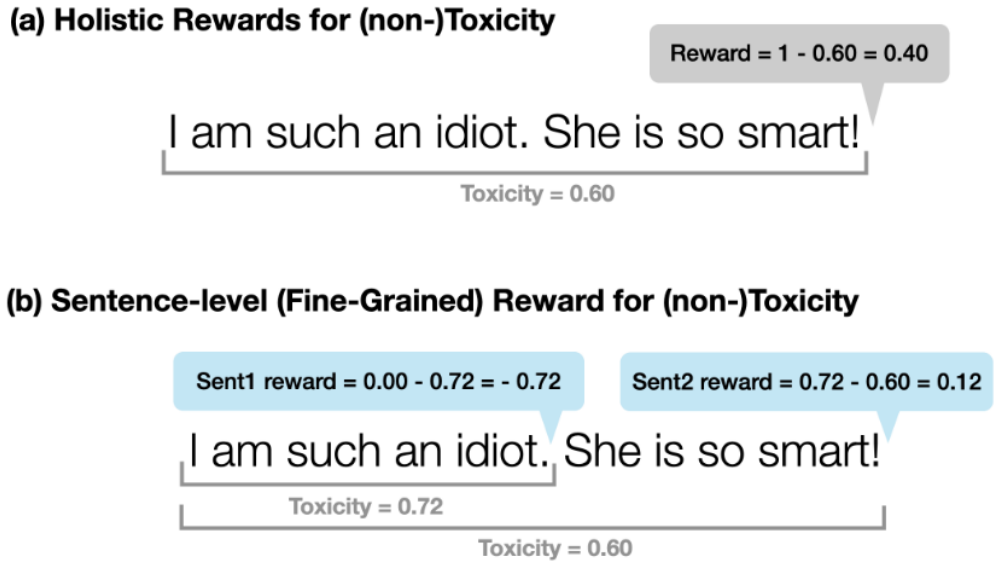
- Learning algorithm: proximal policy optimization (PPO)

$$V^{\text{targ}}(s_t) = \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} + \gamma^{T-t} V_{\psi_{\text{old}}}(s_T)$$

Lagging Value Model

Task 1: Detoxification

- Using a dense sentence-level fine-grained reward
  - exhibits greater sample efficiency compared to a holistic reward
  - achieving lower toxicity with fewer training steps while maintaining better fluency



Experiments

	Toxicity avg max (↓)	Fluency PPL (↓)	Diversity dist-2 (↑)	Diversity dist-3 (↑)
GPT-2	0.192	9.58	0.947	0.931
Controlled Generation				
GeDi	0.154	24.78	0.938	**0.938**
DEXPERTS	0.136	22.83	0.932	0.922
Hol. RLHF	0.130	11.75	0.943	0.926
F.G. RLHF	**0.081**	**9.77**	**0.949**	0.932

Table 1: Results on the REALTOXICITYPROMPTS test set.

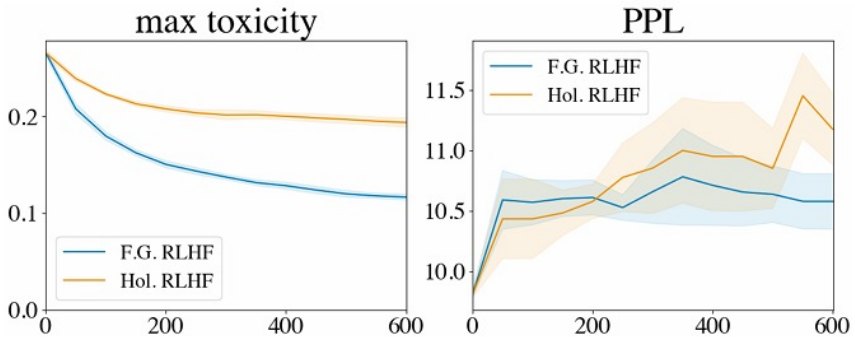


Figure 2: Curves of toxicity and perplexity on the dev set vs. training steps.

- Table1: FINE-GRAINED RLHF attains the lowest toxicity and perplexity among all methods, while maintaining a similar level of diversity.
- Figure2: FINE-GRAINED RLHF has the toxicity drop much faster while keeping a low-level perplexity.

Task 2: Long-Form Question Answering (QA)

- Initial policy and fine-grained human feedback.
  - Define three error categories at different density levels(Task1 used only one category)
- Preference-based human feedback
  - For comparison purpose, workers indicate pairwise preferences based on model output.
- Annotation details
  - Both take a worker about 6 minutes to finish
  - RLHF is more time-consuming to label a human-written response

Fine-Grained Reward Models

- C1: Irrelevance, repetition, or incoherence.
- C2: Incorrect or unverifiable facts
- C3: Incomplete information

Experiments

- FINE-GRAINED RLHF outperforms SFT and Preference RLHF on all error types.
- RLHF is particularly effective in reducing factual errors.

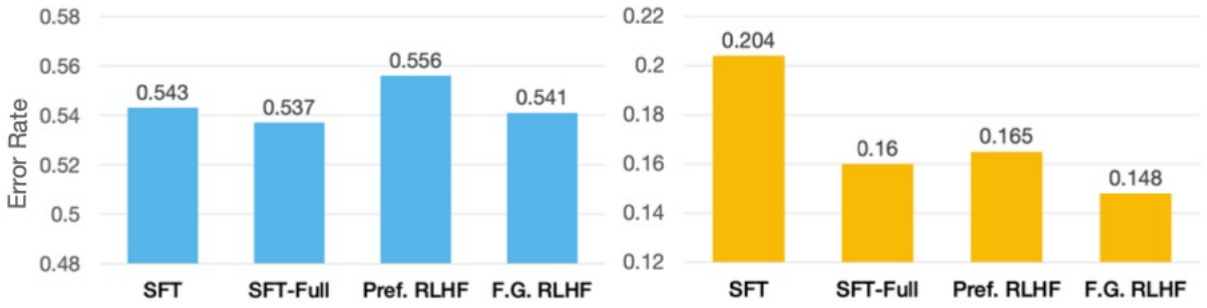


Figure 3: Human evaluation on *rel.* (left) and *fact.* (right) error, measured by % of sub-sentences that contain the error type (↓).

Ours vs.	Win	Tie	Lose
SFT	**23.0%**	65.5%	11.5%
SFT-Full	**22.0%**	61.0%	17.0%
Pref. RLHF	**19.5%**	71.0%	9.5%

Table 2: Human pairwise comparison on information completeness (*comp.*), where win/lose refers to FINE-GRAINED RLHF.

LM Customization

- Behavior(s) Customization

  Varied weights of different model leads to different LM behaviors.

- Trade-off

  Reward models are competing against each other.

- Ablation Study

  Corresponding reward decreases dramatically when the model is removed.

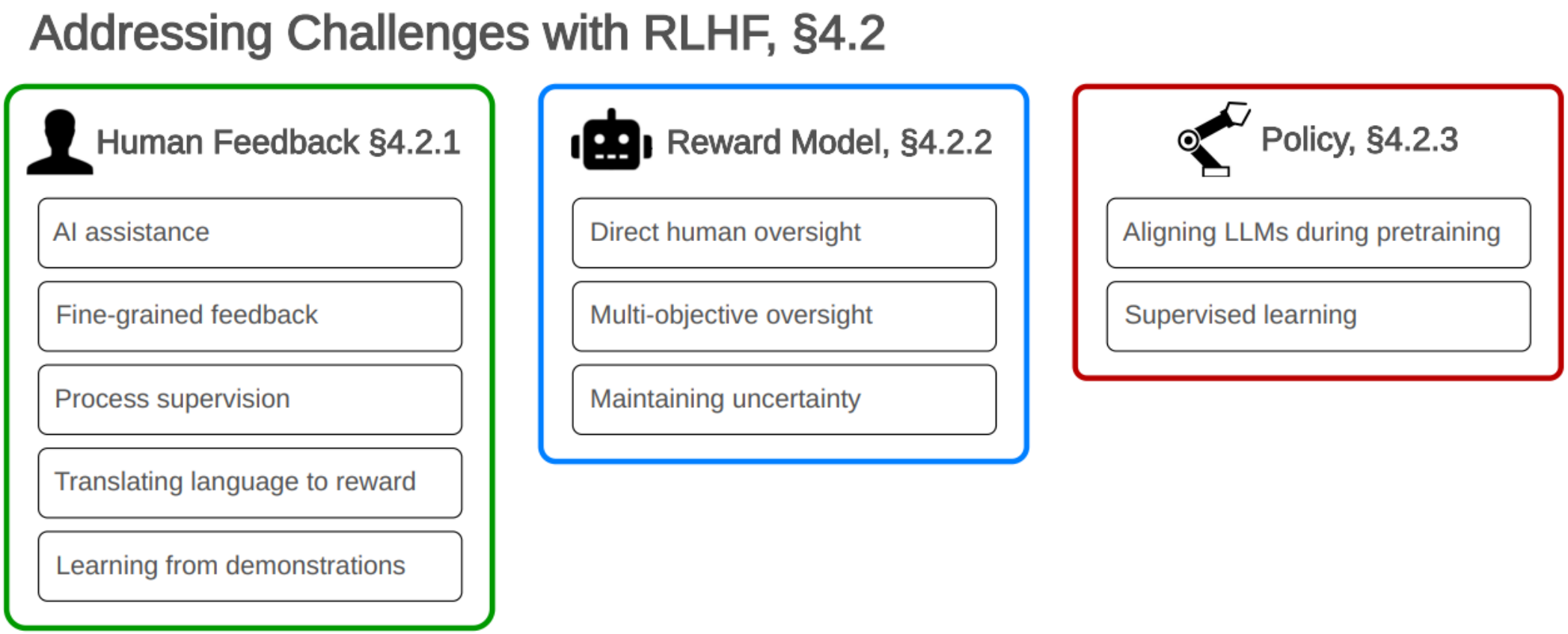
Limitation of F-G RLHF

- Additional computation cost
- Definitions of fine-grained feedback varies
- Rely on the quality of human feedback

Challenges

- Challenges with Obtaining Human Feedback
- Challenges with the Reward Model
- Challenges with the Policy
- Challenges with Jointly Training the Reward Model and Policy

Future


Addressing Challenges with RLHF, §4.2
Human Feedback §4.2.1
AI assistance
Fine-grained feedback
Process supervision
Translating language to reward
Learning from demonstrations
Reward Model, §4.2.2
Direct human oversight
Multi-objective oversight
Maintaining uncertainty
Policy, §4.2.3
Aligning LLMs during pretraining
Supervised learning