

Presentation

Jinyuan Li, Sizhe Wang, Shengxin Hong
09 / 18 / 2025

Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*

Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell[†]

Peter Welinder

Paul Christiano^{*†}

Jan Leike*

Ryan Lowe*

OpenAI

What kind of large language models do we want?

- The Prompt (User Request): "Explain gravity to a 6-year-old."

Output from a Base Language Model (Pre-trained):

"Gravity is a fundamental force of nature by which all things with mass or energy—including planets, stars, galaxies, and even light—are attracted to one another. In general relativity, gravity is described as a consequence of the curvature of spacetime caused by the uneven distribution of mass..."

Factually correct.

BUT... it ignores the instruction "to a 6-year-old." ❌

Output from an Instruct Model (After RLHF):

"Imagine the Earth is a giant trampoline! Because it's so big, it pulls everything toward its middle. This pulling power is called gravity. It's what keeps your feet on the floor so you don't float away, and it's what makes an apple fall down from a tree instead of up!"

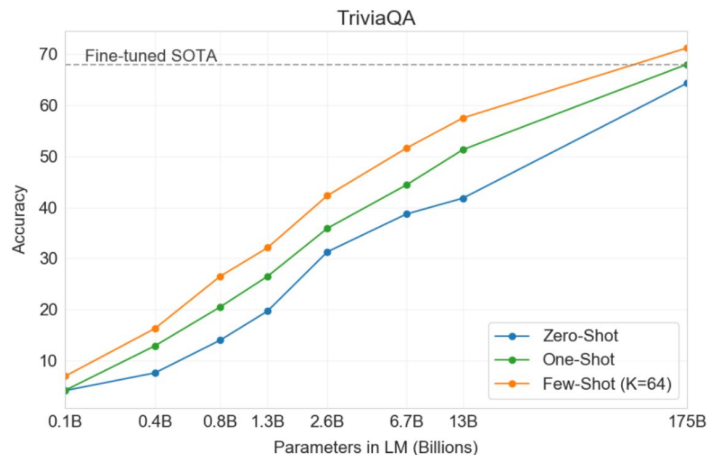
Directly follows the instruction.

Aligned with what the user actually wanted. ✅

What kind of large language models do we want?

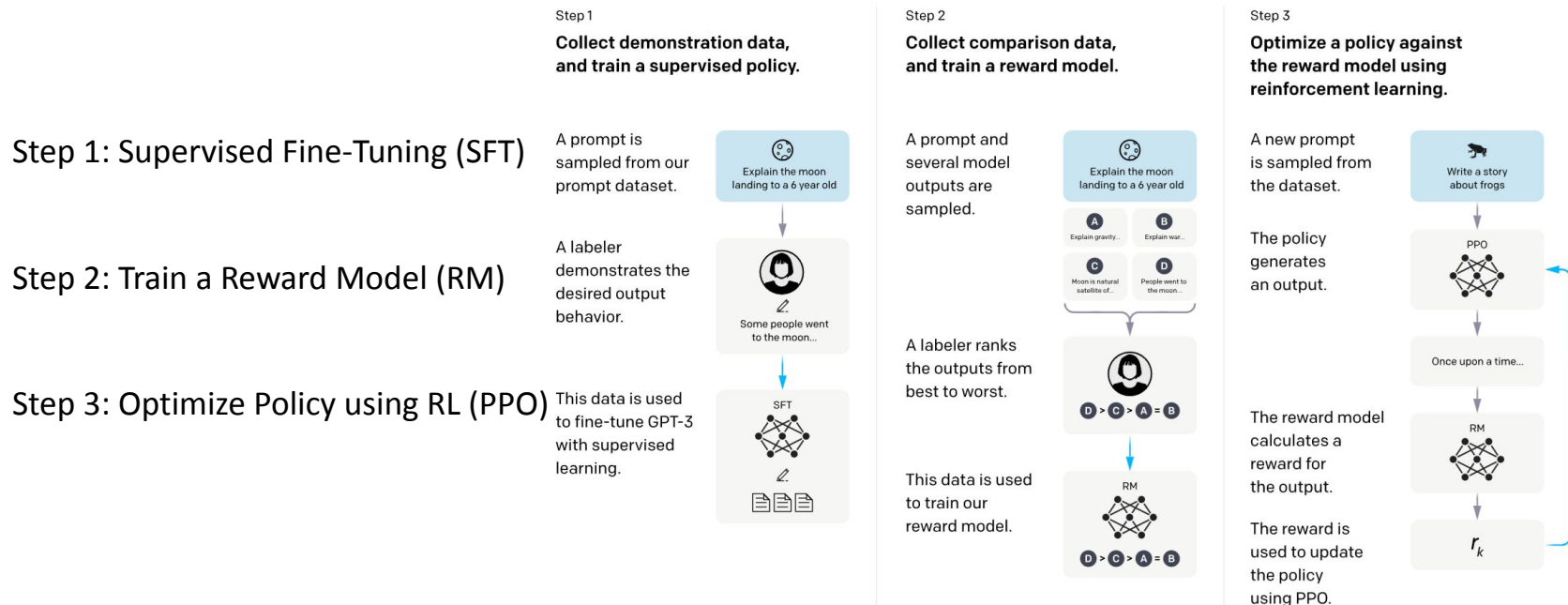
- This scale-up cannot automatically align with human preferences.

Model	Release Date	Parameters	Pre-training Data Size
GPT	June 2018	117 Million	~5GB
GPT-2	February 2019	1.5 Billion	40GB
GPT-3	May 2020	175 Billion	45TB



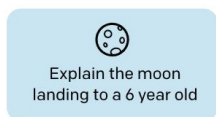
The Solution: A 3-Step Alignment Process

- Reinforcement Learning from Human Feedback (RLHF)

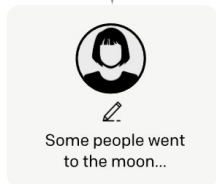


Step 1: Collect Demonstrations & Train SFT Policy

- Goal: Teach the model the desired style of answering instructions.

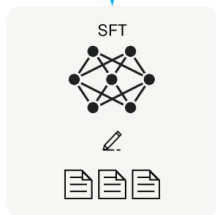


Input: Sampled prompts from: (1) Prompts written by 40 human labelers and (2) Prompts submitted via the OpenAI API.



Action: A human labeler demonstrates the desired, high-quality output behavior for that prompt.

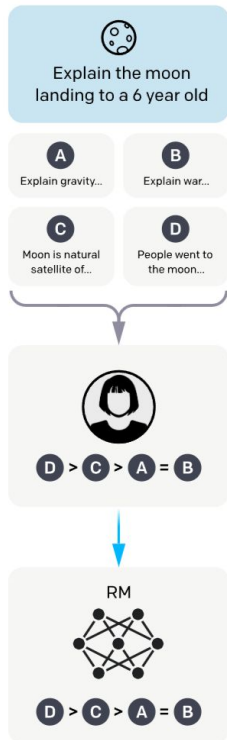
Output: This creates a dataset of human demonstrations (approx. 13k prompts).



Training: This curated dataset is used to fine-tune the pre-trained GPT-3 model using standard supervised learning.

Step 2: Collect Comparisons & Train Reward Model

- Goal: Train a "judge" (the RM) to learn what humans prefer.



Input: Sample a prompt and use the SFT model to generate multiple (K=4 to K=9) different outputs for that single prompt.

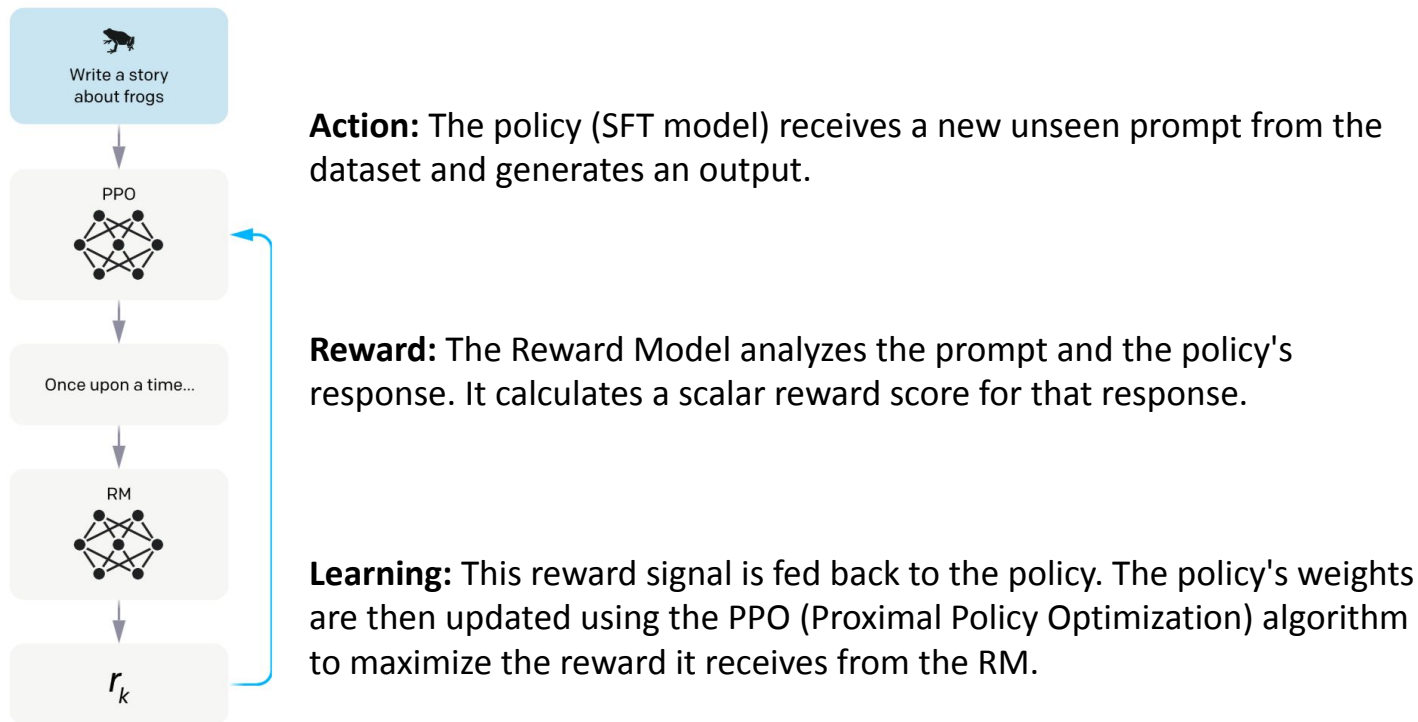
Human Action: A labeler is shown all K outputs and ranks them from best to worst.

Output: A new dataset of human preference data (33k training prompts). This data is structured as comparisons (e.g., for Prompt X, Output A is preferred > Output C).

Training: Train a separate Reward Model (RM). The RM takes any output and returns a "reward" predicting how much a human would prefer that output.

Step 3: Optimize Policy via Reinforcement Learning (PPO)

- Goal: Use the "judge" (RM) to teach the SFT model how to generate better answers.



Step 3.5: Proximal Policy Optimization (PPO)

- The Role of PPO:

Constrained Optimization & Preventing Reward Hacking

- How PPO Solves This (The KL Constraint)?:

Maximize the score from the Reward Model, BUT do not become too different from the original SFT model.

- The Final Objective:

Final Score = (Reward_from_RM) - (KL_Penalty_Score)

Be helpful

Stay sane and coherent

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

Step 4: Final Model -- PPO-ptx (*InstructGPT*)

- "Alignment Tax":

The model lost capability in order to "buy" alignment. It was forgetting some of its general knowledge.

- "PPO-ptx" Solution:

"mixed in" the gradients from the original pre-training dataset (the general internet text GPT-3 was first trained on).

- The Final Objective:

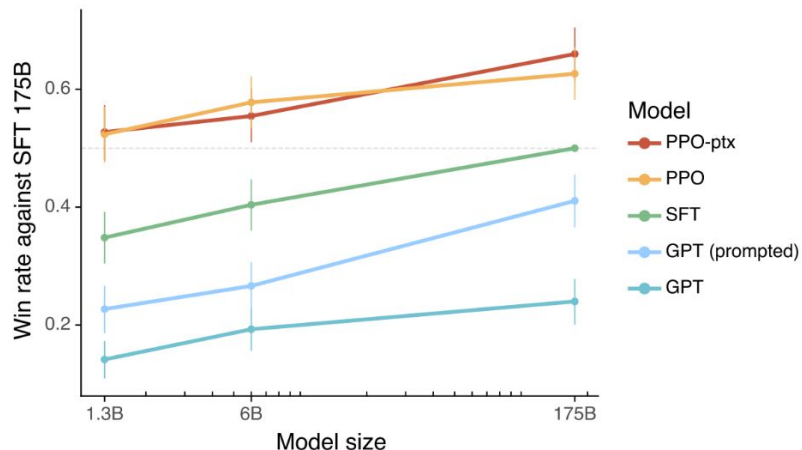
Final Score = (Reward_from_RM) - (KL_Penalty_Score) + (Pre-training Objective Bonus)

Be helpful Stay sane and coherent Don't forget general knowledge

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

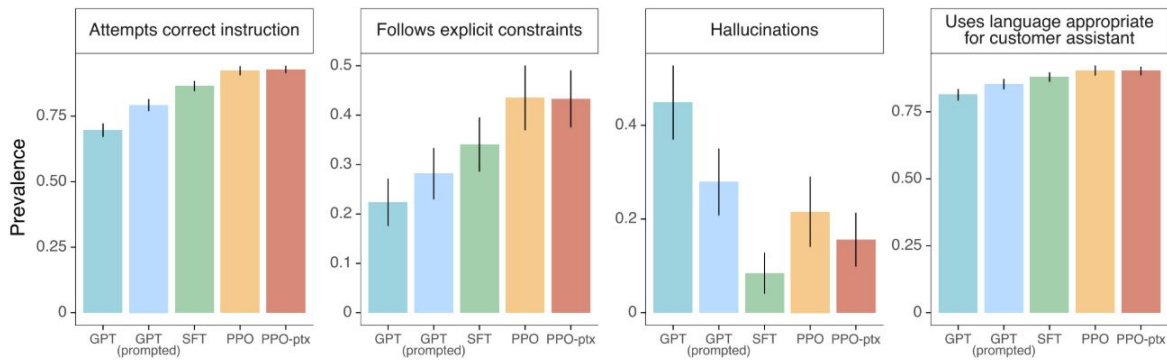
$\gamma E_{x \sim D_{\text{pretrain}}} [\log(\pi_{\phi}^{\text{RL}}(x))]$

Main Result 1: Humans Vastly Prefer InstructGPT



- Key Result: Outputs from the 1.3B InstructGPT model were preferred by labelers over outputs from the 175B GPT-3 model.
- Alignment training is significantly more effective and parameter-efficient than simply scaling the model size.

Main Result 2: Improvements in Helpfulness & Honesty



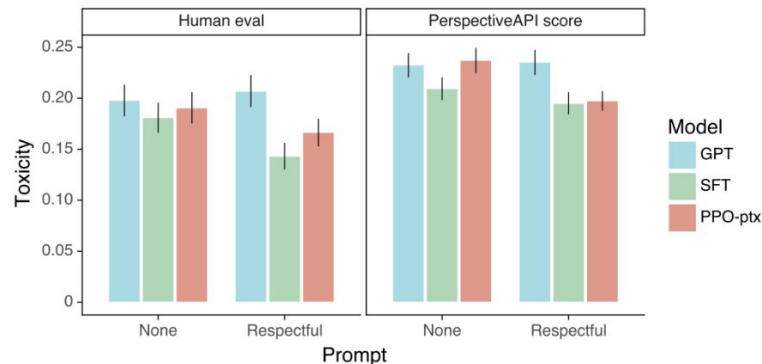
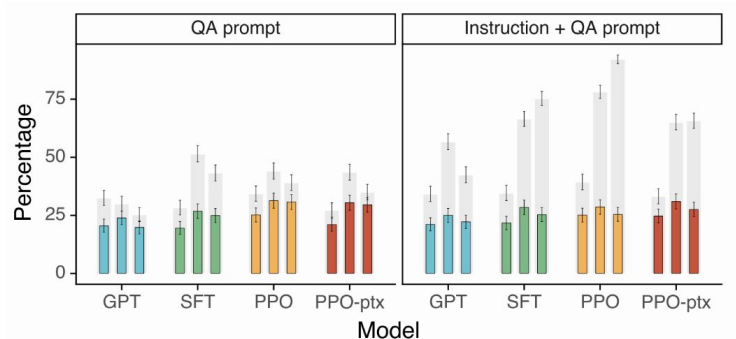
More Helpful (Following Instructions):

- More likely to attempt the correct instruction.
- Better at following explicit constraints given in the prompt (e.g., "Write this in two paragraphs").
- More likely to use language appropriate for a customer assistant.

More Honest (Fewer Hallucinations):

- On closed-domain tasks (like summarization), the base GPT-3 model "hallucinates" over 40% of the time.
- The alignment process cut this hallucination rate.

Main Result 3: Improvements in Truthfulness & Harmlessness



Honesty (Truthfulness):

- On the TruthfulQA benchmark, which tests a model's tendency to mimic human falsehoods, the InstructGPT were significantly more truthful than GPT-3.

Harmlessness (Toxicity):

- On the RealToxicityPrompts dataset, models were tested for toxic output generation.
- When instructed to be "respectful," InstructGPT generated about 25% fewer toxic outputs than the base GPT-3 model.

Q & A

- This seems hard to scale because it requires so much human labeling. Can we just train an "AI feedback model" and learn from that?

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

Rafael Rafailov^{*†}

Archit Sharma^{*†}

Eric Mitchell^{*†}

Stefano Ermon^{†‡}

Christopher D. Manning[†]

Chelsea Finn[†]

[†]Stanford University [‡]CZ Biohub
{rafailov,architsh,eric.mitchell}@cs.stanford.edu

Overview of Framework

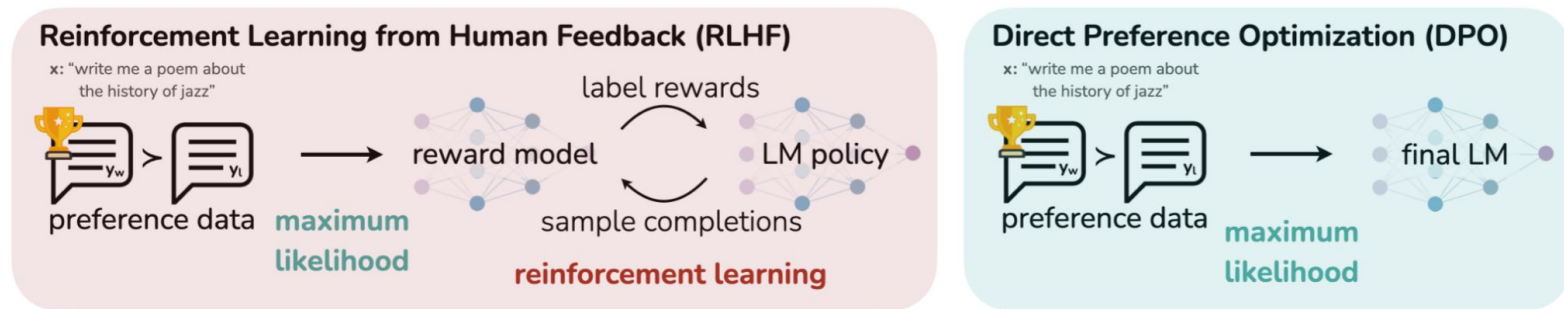


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes 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.

RLHF Preliminaries: Reward Modeling

- Reward Modelling Phase in RLHF:

$$(y_1, y_2) \sim \pi^{\text{SFT}}(\mathbf{y} \mid x)$$

- Bradley-Terry (BT) model

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp(r^*(x, y_1))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))}. \quad (1)$$

RLHF Preliminaries: Policy Optimization

- PPO objective function

$$\max_{\pi_{\theta}} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta}(y|x)} [r_{\phi}(x, y)] - \beta \mathbb{D}_{\text{KL}} [\pi_{\theta}(y | x) || \pi_{\text{ref}}(y | x)], \quad (3)$$

Deriving the DPO Objective

- Following prior work, the optimal solution to Eq. 3 is:

$$\pi_r(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp \left(\frac{1}{\beta} r(x, y) \right), \quad (4)$$

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

- Reparameterizing the reward function in terms of the policy:

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x). \quad (5)$$

Deriving the DPO Objective

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x). \quad p^*(y_1 \succ y_2 \mid x) = \frac{\exp(r^*(x, y_1))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))}.$$

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} - \beta \log \frac{\pi^*(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)}\right)} \quad (6)$$

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]. \quad (7)$$

Experimental Setup: Tasks & Metrics

- Dataset
 - IMDb Sentiment Generation
 - TL;DR Summarization
 - Anthropic Helpful & Harmless (HH)
- Evaluation
 - IMDb: Reward-KL Frontier.
 - TL;DR summarization: win rate vs. human-written summaries, using GPT-4 as the evaluator.
 - HH dialogue: win rate vs. the chosen response baseline, evaluated by GPT-4.

Experimental Setup: Base Models

- Base model:
 - IMDb: GPT-2-large.
 - TL;DR: an SFT model fine-tuned on human-written forum post summaries.
 - HH: Pythia-2.8B

Results: Sentiment & Summarization

- DPO is a more efficient optimizer than PPO, achieving a better reward-KL tradeoff on the sentiment task.
- On summarization, DPO not only achieves a higher peak win rate but is also significantly more robust to changes in sampling temperature.

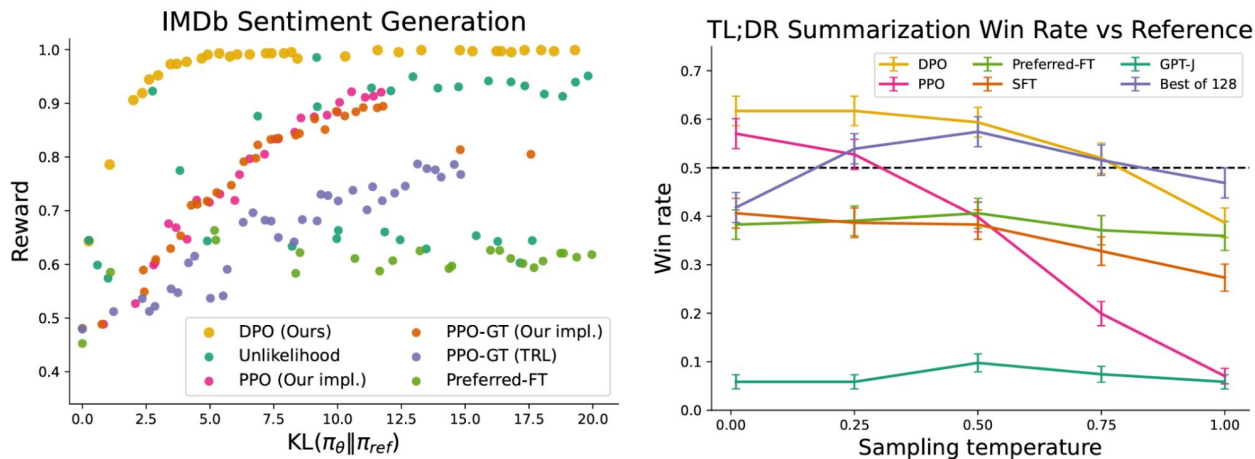


Figure 2: **Left.** The frontier of expected reward vs KL to the reference policy. DPO provides the highest expected reward for all KL values, demonstrating the quality of the optimization. **Right.** TL;DR summarization win rates vs. human-written summaries, using GPT-4 as evaluator. DPO exceeds PPO’s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

Results: Dialogue Task

- DPO is the only efficient method that improves upon the dataset's preferred responses, matching the performance of the computationally expensive "Best of 128" baseline.
- The training process is highly stable, with DPO converging quickly to its peak performance and maintaining it throughout training.

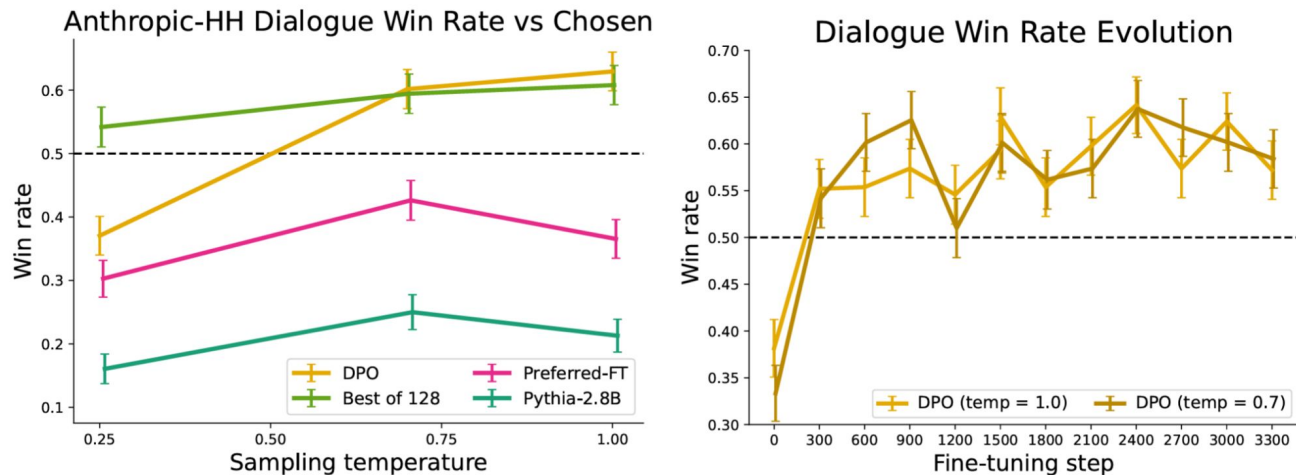


Figure 3: **Left.** Win rates computed by GPT-4 for Anthropic-HH one-step dialogue; DPO is the only method that improves over chosen summaries in the Anthropic-HH test set. **Right.** Win rates for different sampling temperatures over the course of training. DPO's improvement over the dataset labels is fairly stable over the course of training for different sampling temperatures.

Results: Generalization & Evaluator Agreement

- DPO demonstrates better generalization than PPO, maintaining a higher win rate on an out-of-distribution dataset.
- The use of GPT-4 as an evaluator is validated, as its judgments show a high level of agreement with human preferences.

Alg.	Win rate vs. ground truth	
	Temp 0	Temp 0.25
DPO	0.36	0.31
PPO	0.26	0.23

Table 1: GPT-4 win rates vs. ground truth summaries for out-of-distribution CNN/DailyMail input articles.

	DPO	SFT	PPO-1
N respondents	272	122	199
GPT-4 (S) win %	47	27	13
GPT-4 (C) win %	54	32	12
Human win %	58	43	17
GPT-4 (S)-H agree	70	77	86
GPT-4 (C)-H agree	67	79	85
H-H agree	65	-	87

Table 2: Comparing human and GPT-4 win rates and per-judgment agreement on TL;DR summarization samples. **Humans agree with GPT-4 about as much as they agree with each other.** Each experiment compares a summary from the stated method with a summary from PPO with temperature 0.

Q & A

- Q: Is there a downside to DPO's simplicity? Since it learns directly from human feedback, is it more likely to copy errors in that data.
 - Yes, its simplicity creates trade-offs. DPO's direct fitting approach makes it sensitive to noise and biases in the preference data, as it can directly propagate these errors into the policy. While its gradient weighting offers some mitigation, it doesn't solve the issue. Performance also critically depends on the quality of the reference policy and the tuning of β .
- Q: How sensitive is DPO to violations of the Bradley–Terry assumption and to the choice of reference/ β
 - The DPO loss is derived from the Bradley-Terry model, so significant violations of this assumption can misspecify the objective and bias the results.
 - Performance is very sensitive to the choice of π_{ref} , making a high-quality, distribution-matched reference policy essential.
 - The β is critical as it controls the KL-regularization strength: a large β results in a conservative policy close to the reference, while a small β allows more aggressive fitting to preferences, risking drift and overfitting.

SimPO: Simple Preference Optimization with a Reference-Free Reward

Yu Meng^{1*} Mengzhou Xia^{2*} Danqi Chen²

¹Computer Science Department, University of Virginia

²Princeton Language and Intelligence (PLI), Princeton University

yumeng5@virginia.edu

{mengzhou, danqic}@cs.princeton.edu

Motivation: Overcoming the Limitations of DPO

- Motivations
 - DPO's reliance on a reference model creates computational overhead and a mismatch between the training objective and the inference-time generation metric.
 - There is a need to align the reward formulation directly with the model's generation metric (average log-likelihood) to improve performance and efficiency.
- Contributions
 - SimPO: A simple, reference-free preference optimization algorithm.
 - Length-Normalized Reward: A novel reward formulation to prevent length bias.
 - Target Reward Margin: A mechanism to enhance the model's ability to distinguish between winning and losing responses.

SimPO's Core Idea: Length-Normalized Reward

- The reward is directly aligned with the generation metric: average log-likelihood.

$$p_{\theta}(y \mid x) = \frac{1}{|y|} \log \pi_{\theta}(y \mid x) = \frac{1}{|y|} \sum_{i=1}^{|y|} \log \pi_{\theta}(y_i \mid x, y_{<i}). \quad (3)$$

$$r_{\text{SimPO}}(x, y) = \frac{\beta}{|y|} \log \pi_{\theta}(y \mid x) = \frac{\beta}{|y|} \sum_{i=1}^{|y|} \log \pi_{\theta}(y_i \mid x, y_{<i}), \quad (4)$$

The SimPO Objective: Enforcing a Reward Margin

- A target reward margin, $\gamma > 0$, is added to the Bradley-Terry objective.
- This encourages the winning response's reward to exceed the loser's by at least γ , which improves generalization.

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l) - \gamma). \quad (5)$$

$$\mathcal{L}_{\text{SimPO}}(\pi_\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\frac{\beta}{|y_w|} \log \pi_\theta(y_w | x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l | x) - \gamma \right) \right]. \quad (6)$$

DPO vs. SimPO

- Across both AlpacaEval 2 (LC) and Arena-Hard benchmarks, SimPO consistently and significantly outperforms DPO across all tested models and setups.

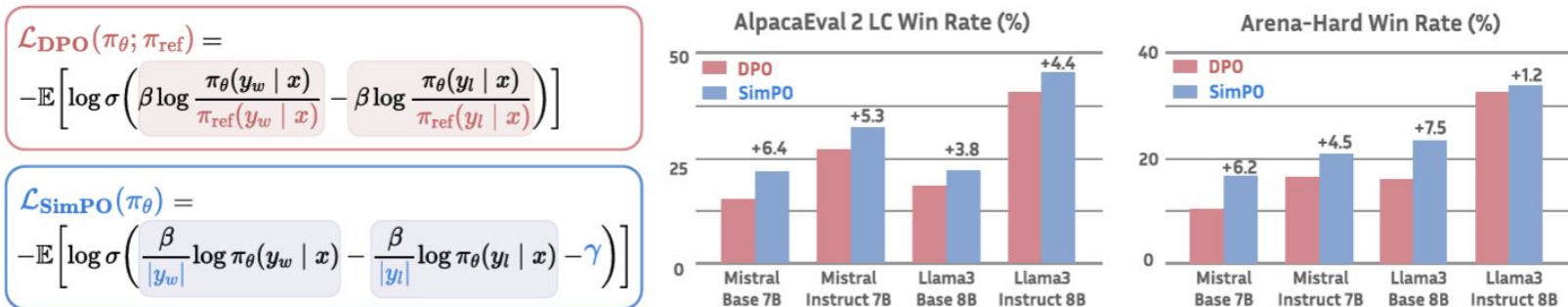


Figure 1: SimPO and DPO mainly differ in their reward formulation, as indicated in the shaded box. SimPO outperforms DPO significantly across a range of settings on AlpacaEval 2 and Arena-Hard.

Experiment Setup: The Base Configuration

- Goal: To test performance starting from a standard pre-trained model.
- Models:
 - Llama-3-8B
 - Mistral-7B
- Datasets & Pipeline:
 - Step 1 : Fine-tune the base model on UltraChat-200K to create a supervised fine-tuned (SFT) model.
 - Step 2 : Perform preference optimization (e.g., SimPO, DPO) on the SFT model using the UltraFeedback dataset.

Experiment Setup: The Instruct Configuration

- Goal: To test performance starting from a powerful, instruction-tuned model.
- Models:
 - Llama-3-8B-Instruct
 - Mistral-7B-Instruct-v0.2
- Dataset & On-Policy Generation:
 - To mitigate distribution shift, a new preference dataset is generated "on-policy".
 - Process:
 - Use the Instruct models to generate 5 responses for each prompt from UltraFeedback.
 - Use an external reward model (PairRM) to score the 5 responses.
 - The highest-scoring response becomes the 'winner' (y_w) and the lowest-scoring becomes the 'loser' (y_l).

Evaluation: Benchmarks & Metrics

- Models are assessed on three benchmarks.

Table 2: Evaluation details for AlpacaEval 2 [55], Arena-Hard [54], and MT-Bench [99]. The baseline model refers to the model compared against. GPT-4 Turbo corresponds to GPT-4-Preview-1106.

	# Exs.	Baseline Model	Judge Model	Scoring Type	Metric
AlpacaEval 2	805	GPT-4 Turbo	GPT-4 Turbo	Pairwise comparison	LC & raw win rate
Arena-Hard	500	GPT-4-0314	GPT-4 Turbo	Pairwise comparison	Win rate
MT-Bench	80	-	GPT-4/GPT-4 Turbo	Single-answer grading	Rating of 1-10

Main Results: SimPO Consistently Outperforms Baselines

- In all four settings, SimPO achieves the highest performance on the benchmarks of AlpacaEval 2 and Arena-Hard, significantly surpassing DPO and other preference optimization methods.

Table 4: AlpacaEval 2 [55], Arena-Hard [54], and MT-Bench [99] results under the four settings. LC and WR denote length-controlled and raw win rate, respectively. We train SFT models for Base settings on the UltraChat dataset. For Instruct settings, we use off-the-shelf models as the SFT model.

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 [91]	11.6	10.2	5.8	5.4	6.7	25.3	24.8	18.1	6.5	7.6
SLiC-HF [96]	10.9	8.9	7.3	5.8	7.4	24.1	24.6	18.9	6.5	7.8
DPO [66]	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 [88]	9.8	8.9	6.9	5.4	6.8	23.8	28.8	22.6	6.3	7.5
KTO [29]	13.1	9.1	5.6	5.4	7.0	24.5	23.6	17.9	6.4	7.7
ORPO [42]	14.7	12.2	7.0	5.8	7.3	24.5	24.9	20.8	6.4	7.7
R-DPO [64]	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	Llama-3-Base (8B)					Llama-3-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 [91]	12.1	10.1	6.3	5.8	7.0	31.3	28.4	26.5	6.7	7.9
SLiC-HF [96]	12.3	13.7	6.0	6.3	7.6	26.9	27.5	26.2	6.8	8.1
DPO [66]	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 [88]	10.8	8.1	5.8	6.0	7.4	28.9	32.2	28.8	7.0	8.0
KTO [29]	14.2	12.4	12.5	6.3	7.8	33.1	31.8	26.4	6.9	8.2
ORPO [42]	12.2	10.6	10.8	6.1	7.6	28.5	27.4	25.8	6.8	8.0
R-DPO [64]	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

Ablation Study

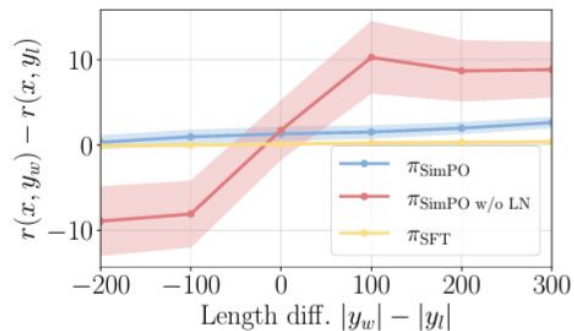
- Confirm that both of SimPO's key design choices are crucial. Removing length normalization (w/o LN) causes the most significant performance drop, while setting the reward margin γ to 0 also degrades performance.

Table 5: Ablation studies under Mistral-Base and Mistral-Instruct settings. We ablate each key design of SimPO: (1) removing length normalization in Eq. (4) (*i.e.*, w/o LN); (2) setting target reward margin γ to be 0 in Eq. (6) (*i.e.*, $\gamma = 0$).

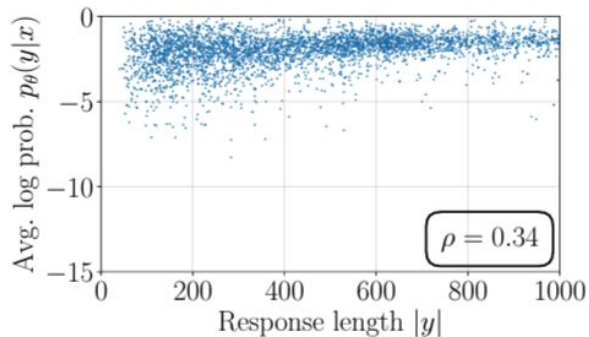
Method	Mistral-Base (7B) Setting					Mistral-Instruct (7B) Setting				
	AlpacaEval 2		Arena-Hard	MT-Bench		AlpacaEval 2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4
DPO	15.1	12.5	10.4	5.9	7.3	26.8	24.9	16.3	6.3	7.6
SimPO	21.5	20.8	16.6	6.0	7.3	32.1	34.8	21.0	6.6	7.6
w/o LN	11.9	13.2	9.4	5.5	7.3	19.1	19.7	16.3	6.4	7.6
$\gamma = 0$	16.8	14.3	11.7	5.6	6.9	30.9	34.2	20.5	6.6	7.7

Analysis of Length Normalization

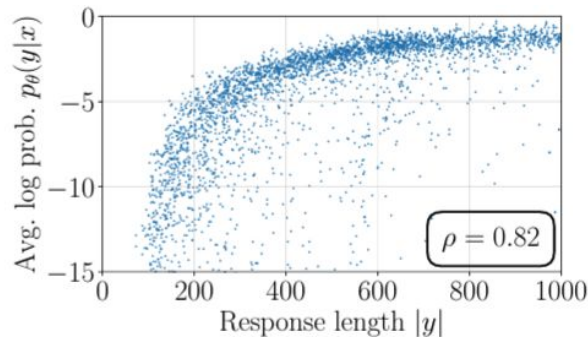
- Length normalization (LN) successfully prevents the model from learning a spurious correlation between response length and reward.



(a) Reward optimization.



(b) SimPO.



(c) SimPO without LN.

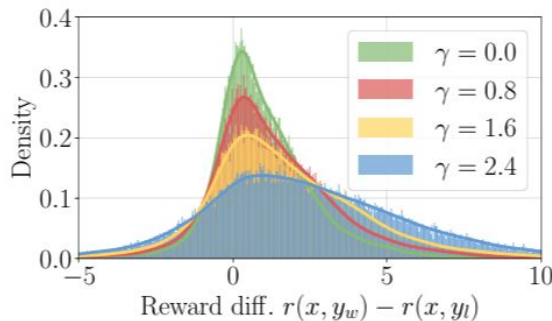
Figure 2: Effect of length normalization (LN). (a) Relationship between reward margin and length difference between winning and losing responses. (b) Spearman correlation between average log probability and response length for SimPO. (c) Spearman correlation for SimPO without LN.

The Impact of Target Reward Margin (γ)

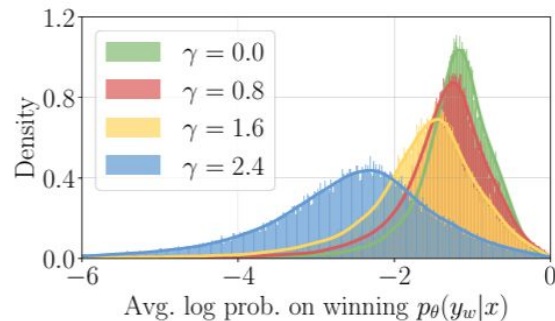
- Increasing γ improves the model's ability to correctly rank responses.
- However, an excessively large margin can degrade overall generation quality, indicating a trade-off between reward separation and maintaining a well-calibrated model.



(a) Performance w/ different γ .



(b) Reward diff. distribution.

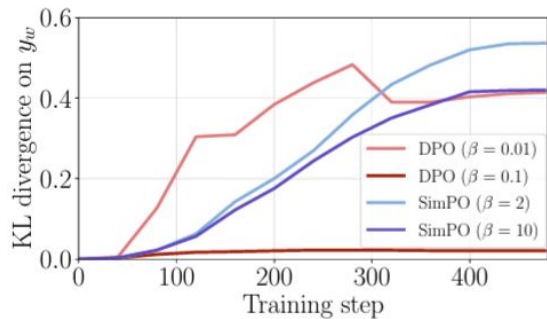


(c) Log prob. distribution.

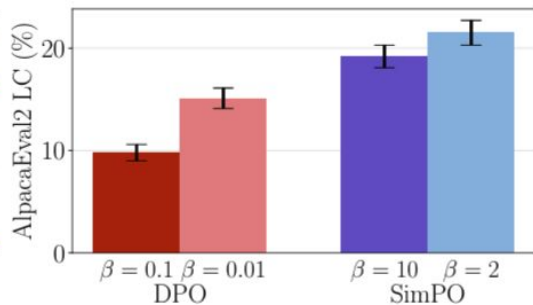
Figure 3: Study of the margin γ . (a) Reward accuracy and AlpacaEval2 LC win rate under different γ values. (b) Reward difference distribution under different γ values. (c) Log likelihood distribution on chosen responses under different γ values.

SimPO vs. DPO on Efficiency & Stability

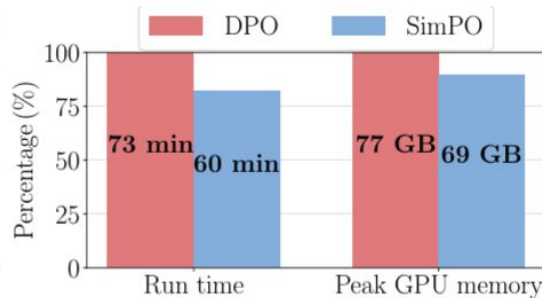
- SimPO is significantly more compute and memory efficient than DPO due to its reference-free design.
- Although without KL regularization term, SimPO's policy remains stable and does not diverge excessively from the initial model when using a small learning rate



(a) KL divergence w/ different β .



(b) Performance w/ different β .



(c) Efficiency of DPO vs. SimPO.

Figure 5: Comparison between SimPO and DPO (continued). (a) With different β in DPO and SimPO, KL divergence from the policy model to the reference model on y_w . (b) AlpacaEval2 LC win rate of DPO and SimPO with different β . (c) Runtime and memory usage for DPO and SimPO.

Q & A

- Q: Why does SimPO requires a much larger beta than DPO?
 - SimPO uses a **length-normalized average log-likelihood** as reward and removes the reference model, so reward gaps and per-example gradients are much smaller in scale; a larger β is therefore needed to bring the Bradley–Terry logits into a useful range. In contrast, DPO multiplies β by a log-ratio to a reference policy (and β also controls the KL strength), so practical β values are much smaller.

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

Zequi Wu^{1*} Yushi Hu^{1*} Weijia Shi¹ Nouha Dziri² Alane Suhr³
Prithviraj Ammanabrolu⁴⁵ Noah A. Smith¹² Mari Ostendorf¹ Hannaneh Hajishirzi¹²

¹University of Washington

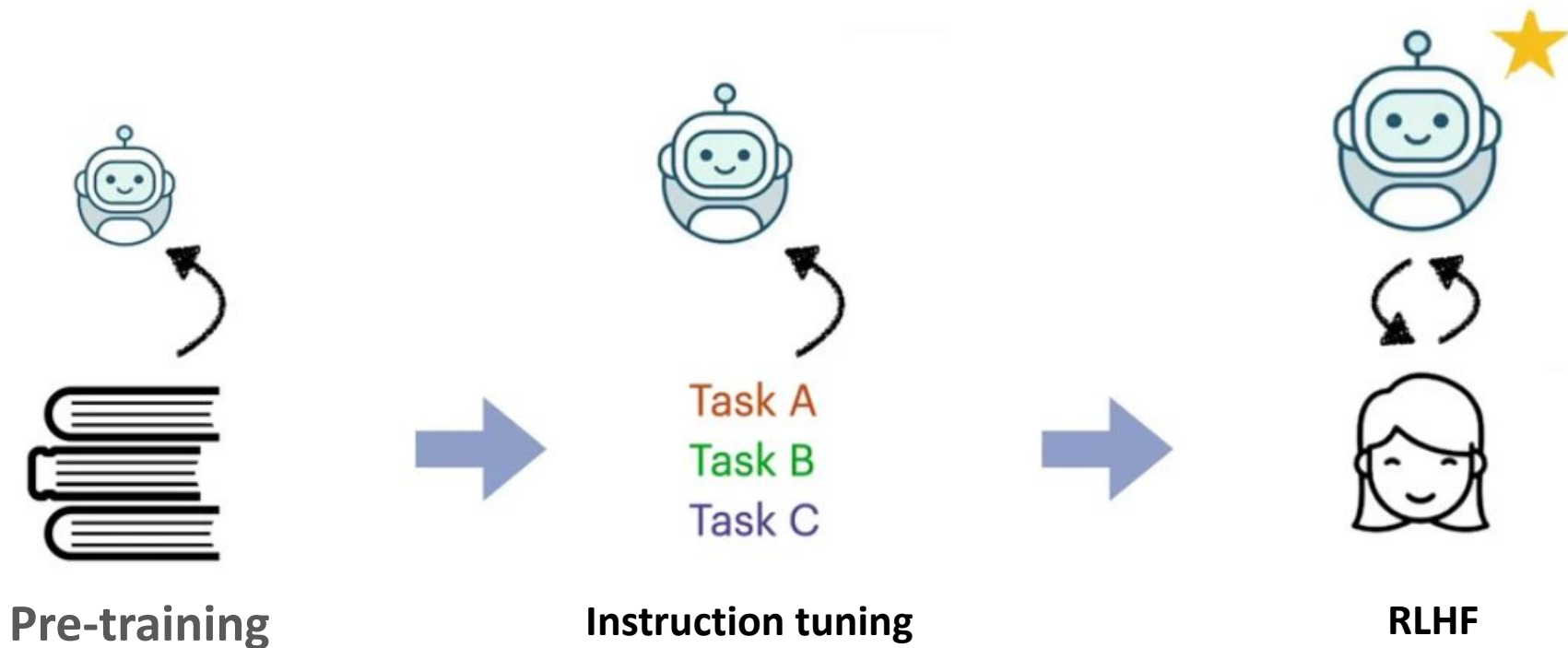
²Allen Institute for Artificial Intelligence

³University of California, Berkeley

⁴University of California, San Diego

⁵MosaicML

State-of-the-art AI is built on...



RLHF

Step 1: Collect preference feedback and train a reward model

Prompt:

What are the 3 most common gasses in earth's atmosphere?

LM outputs:

A

The atmosphere of Earth is a layer of gases retained by Earth's gravity...

B

The atmosphere is commonly known as air. The top gases by volume that dry air ...

C

The air that surrounds the planet Earth contains various gases. Nitrogen...

D

The atmosphere of Earth is the layer of gases, generally known as air...

Human Feedback



B >

C =

D >

A



Preference RM

Step 2: Fine-tune the policy LM against the reward models using RL

Sampled Prompt: Does water boil quicker at high altitudes?



PPO

It takes longer for water to boil at high altitudes. The reason is that water boils at a lower temperature at higher altitudes.

Preference Reward: - 0.35

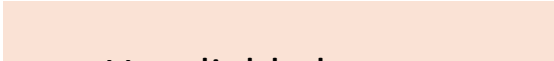
Update policy with rewards

Problem 1: Challenging overall quality comparison

Hard to compare LM outputs with a **mixture of diverse undesired behaviors**

Output A:

- Sentence 1 - Factual **[good]** but informative **[bad]**
- Sentence 2 - ...

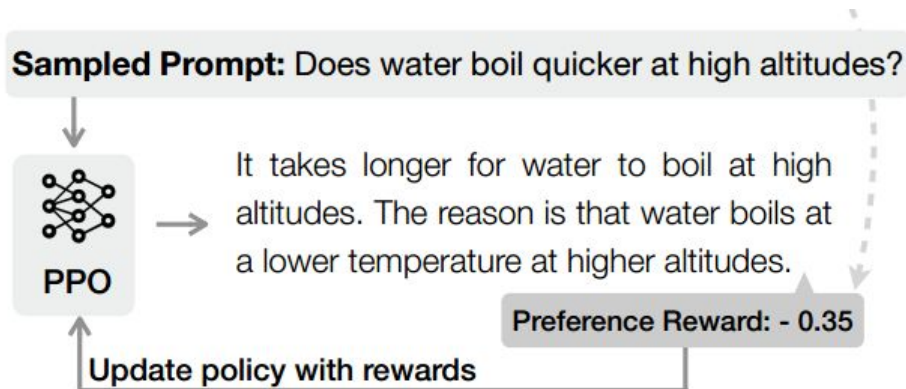


Unreliable human
feedback

Output B:

- Sentence 1 - Informative **[good]** but verifiable **[bad]**
- Sentence 2 - ...

Problem 2: Sparse reward for training



Single holistic reward for the full output



Unreliable RL training

Fine-grained feedback is more explicit and reliable

Prompt:

What are the 3 most common gasses in earth's atmosphere?

LM output:

The atmosphere of Earth is a layer of gases retained by Earth's gravity. The most common gas, by dry air volume, is nitrogen. The second most is oxygen. The third most is carbon dioxide.



Localizing
feedback/reward

Fine-Grained Human Feedback

Irrelevant / Redundant

Unverifiable / Untruthful

Missing

The third most is Argon.



Relevance RM



Factuality RM



Information
Completeness RM



Categorizing
feedback/reward

Fine-grained feedback is more explicit and reliable

Step 1: Collect fine-grained feedback and train reward model

Prompt:

What are the 3 most common gasses in earth's atmosphere?

LM output:

The atmosphere of Earth is a layer of gases retained by Earth's gravity. The most common gas, by dry air volume, is nitrogen. The second most is oxygen. The third most is carbon dioxide.

Fine-Grained Human Feedback

Irrelevant / Redundant

Unverifiable / Untruthful

Missing The third most is Argon.



Relevance RM



Factuality RM



Information
Completeness RM

Step 2: Refine the policy LM against the reward model using RL

Sampled Prompt: Does water boil quicker at high altitudes?



PPO

Relevant: + 0.3 Factual: - 0.5

It takes longer for water to boil at high altitudes. The reason is that water boils at a lower temperature at higher altitudes.

Relevant: + 0.3 Factual: + 0.5 Info. complete: + 0.3

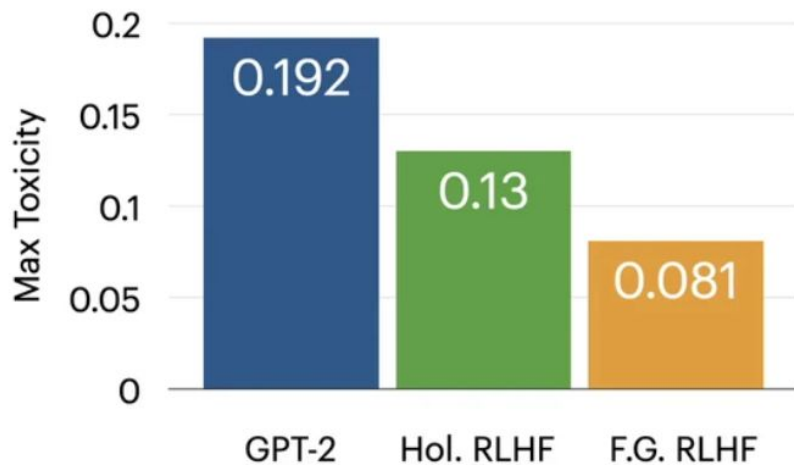
Update policy with rewards

Task 1: Detoxification

- Explore **learning with dense (sentence-level)** reward compared to holistic reward from a single reward model that measures toxicity (0-1).
- Use perspective API as reward model.
- **Data:** RealToxicityPrompts (prompts known to easily elicit problematic LM generations)
- **Initial policy model:** GPT2-large

Result

- The toxicity score is reported as the max score among 4 sampled model outputs, averaged over all test input prompts.

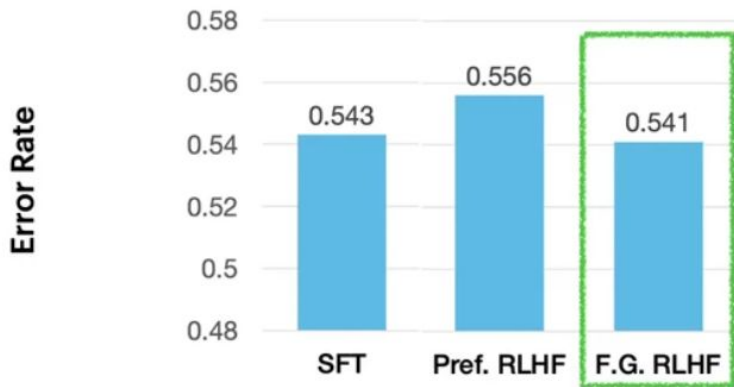


Task 2: Long-form QA

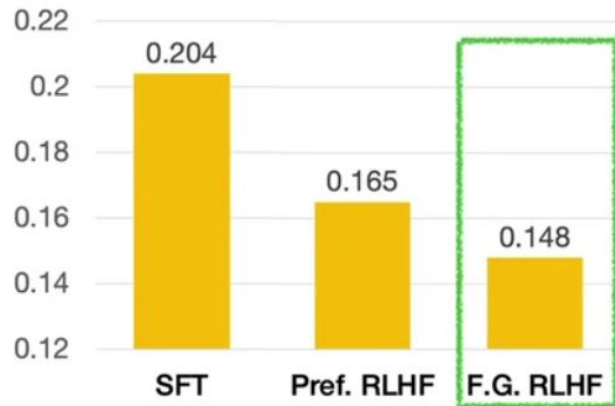
- **Task input:** A question and a set of knowledge passages
- **Data:** QA-FEEDBACK (based on ASQA)
- **Initial policy model:** T5-large supervised fine-tuned with 1K examples (SFT)

Human evaluation

C1: irrelevance, repetition, and incoherence error



C2: incorrect or unverifiable facts



C3: information completeness

FG RLHF v.s.	Win	Tie	Lose
SFT	23.0%	65.5%	11.5%
Pref. RLHF	19.5%	71.0%	9.5%

Fine-grained RLHF outperforms SFT and preference RLHF on all error type

Summary

- Fine-grained RLHF enables LM training and learning from **dense rewards** associated with **different feedback types**.
- Learning with fine-grained reward functions leads to **improved performance** in long-form generation and allows **LM behavior customization**.

Question

- If segment-level rewards and multi-aspect heads are trained on human-annotated spans, how robust is the method to label sparsity and annotator disagreement, and can the model learn to generate its own proxy spans (via uncertainty/attribution) that improve the reward models without additional human labels?
- How do conflicts between fine-grained reward models (e.g., relevance vs. completeness) affect the stability and generalization of the trained LM ?