Mainly from **LLM Training: RLHF and Its Alternatives**

Reinforcement learning

Reinforcement learning is a machine learning training method based on rewarding desired behaviors and punishing undesired ones. In general, a reinforcement learning agent -- the entity being trained -- is able to perceive and interpret its environment, take actions and learn through trial and error.

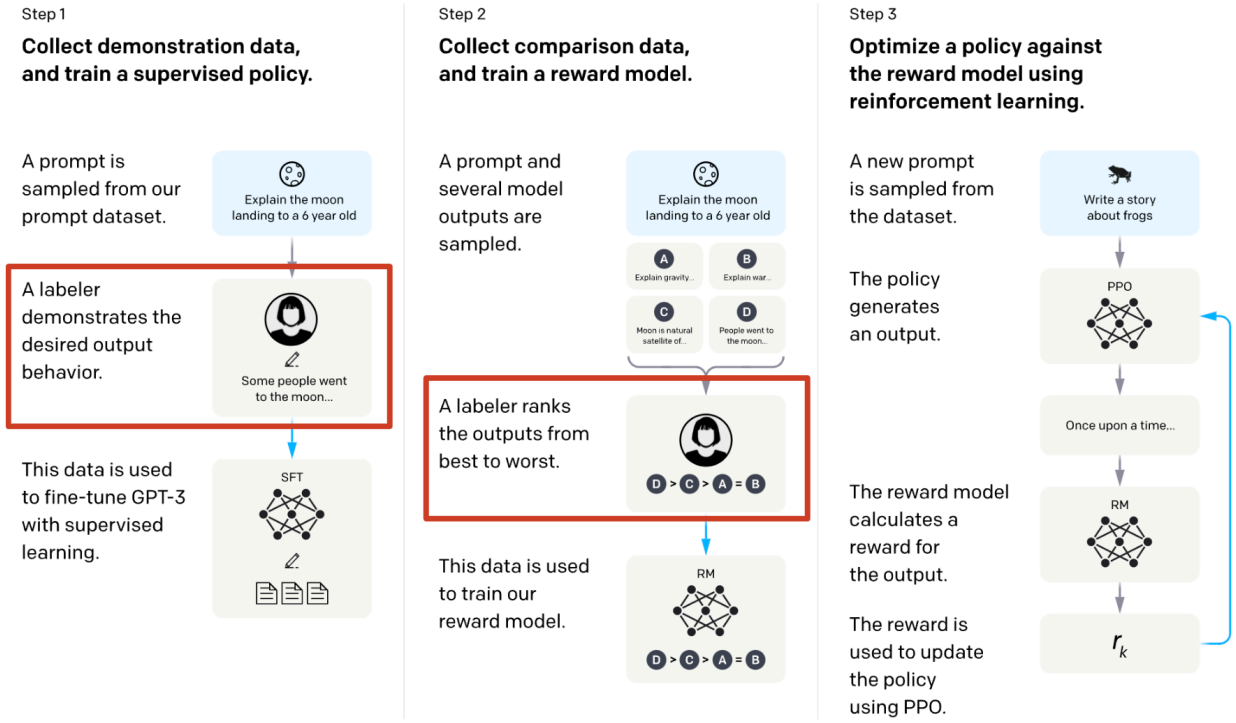
Supervised fine tuning cons:

models are incentivized to place probability mass on all human demonstrations, including those that are low-quality; and distributional shift during sampling can degrade performance

RLHF goal

The essential goal here is to make a conventional large language model (GPT-3 in our case) align with human principles or preferences. This makes our LLMs less toxic, more truthful, and less biased.

Steps:



1. Pretraining a language model (LM),
    1. fine tune with preference data set or supervised learning
    2. supervised fine tuning uses a smaller dataset
2. gathering data and training a reward model, and
    1. These LMs for reward modeling can be both another fine-tuned LM or a LM trained from scratch on the preference data.
        1. fine-tuned LM's output layer (the next-token classification layer) is substituted with a regression layer, which features a single output node.
    2. sample a set of prompts from a predefined dataset
    3. pass through the initial language model to generate new text.
    4. for each prompt, we generate four to nine responses from the finetuned LLM created in the prior step. An individual then ranks these responses based on their preference.
    5. Human annotators are used to rank the generated text outputs from the LM
        1. There are multiple methods for ranking the text. One method that has been successful is to have users compare generated text from two language models conditioned on the same prompt. By comparing model outputs in head-to-head matchups, an **Elo** system can be used to generate a ranking of the models and outputs relative to each-other. These different methods of ranking are normalized into a scalar reward signal for training.

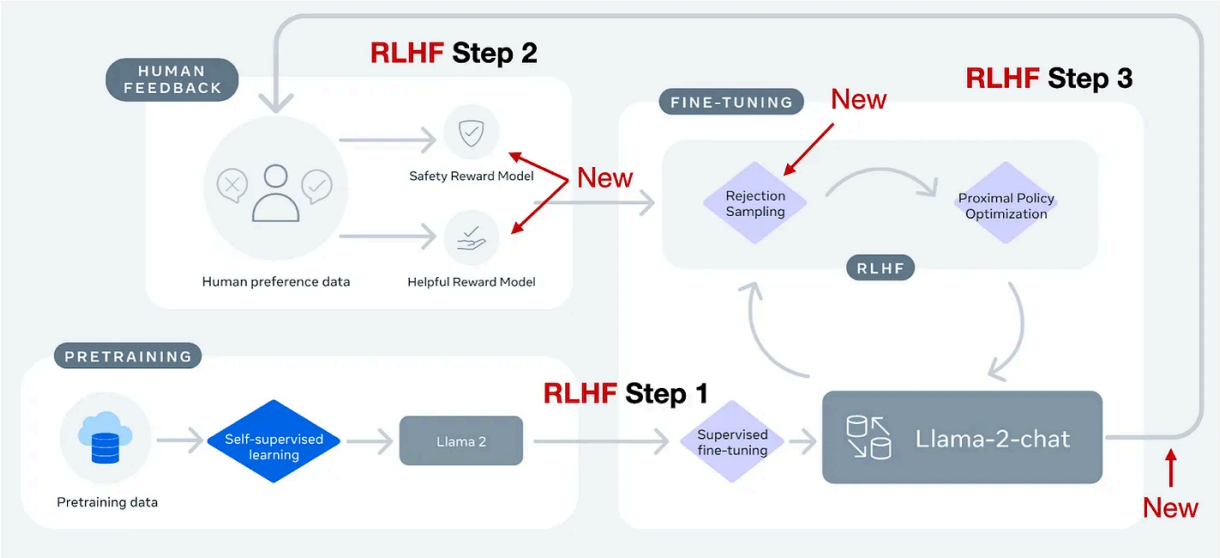
6. An interesting artifact of this process is that the successful RLHF systems to date have used reward language models with varying sizes relative to the text generation (e.g. OpenAI 175B LM, 6B reward model, Anthropic used LM and reward models from 10B to 52B, DeepMind uses 70B Chinchilla models for both LM and reward).
7. An intuition would be that these preference models need to have similar capacity to understand the text given to them as a model would need in order to generate said text.

3. fine-tuning the LM with reinforcement learning.
    1. fine-tuning some or all of the parameters of a **copy of the initial LM** with a policy-gradient RL algorithm, Proximal Policy Optimization (PPO)
    2. Some parameters of the LM are frozen

InstructGPT stats

- Pretrain: 100B - 5T tokens
- Supervised Finetuning: 1k - 50k instruction response pairs
- RLHF > 50k examples

RLHF in Llama 2



- 2 reward model
    - Helpfulness
    - Harmlessness

- Different human ranking method
    - InstructGPT ask human labelers to rank 4 responses at a time
    - LLama 2 only presents 2 responses for ranking but an additional "margin" label (ranging from "significantly better" to "negligibly better") is gathered
- Different ranking loss function to train reward model
    - InstructGPT loss $\mathcal{L}_{\text{ranking}} = -\log\left(\sigma\left(r_\theta\left(x, y_c\right) - r_\theta\left(x, y_r\right)\right)\right)$
    - Llama2 loss $\mathcal{L}_{\text{ranking}} = -\log\left(\sigma\left(r_\theta\left(x, y_c\right) - r_\theta\left(x, y_r\right) - m(r)\right)\right)$
    - Llama 2 added the the margin “m(r)” as a discrete function of the preference rating as follows:
    - returning a higher margin via “m(r)” will make the difference between the reward of the preferred and rejected responses smaller, resulting in a larger loss, which in turn results in larger gradients, and consequently model changes, during the policy gradient update.
- 2 RLHF stages
    - Rejection sampling
        - K outputs are drawn, and the one with the highest reward is chosen for the gradient update during the optimization step
    - PPO stage

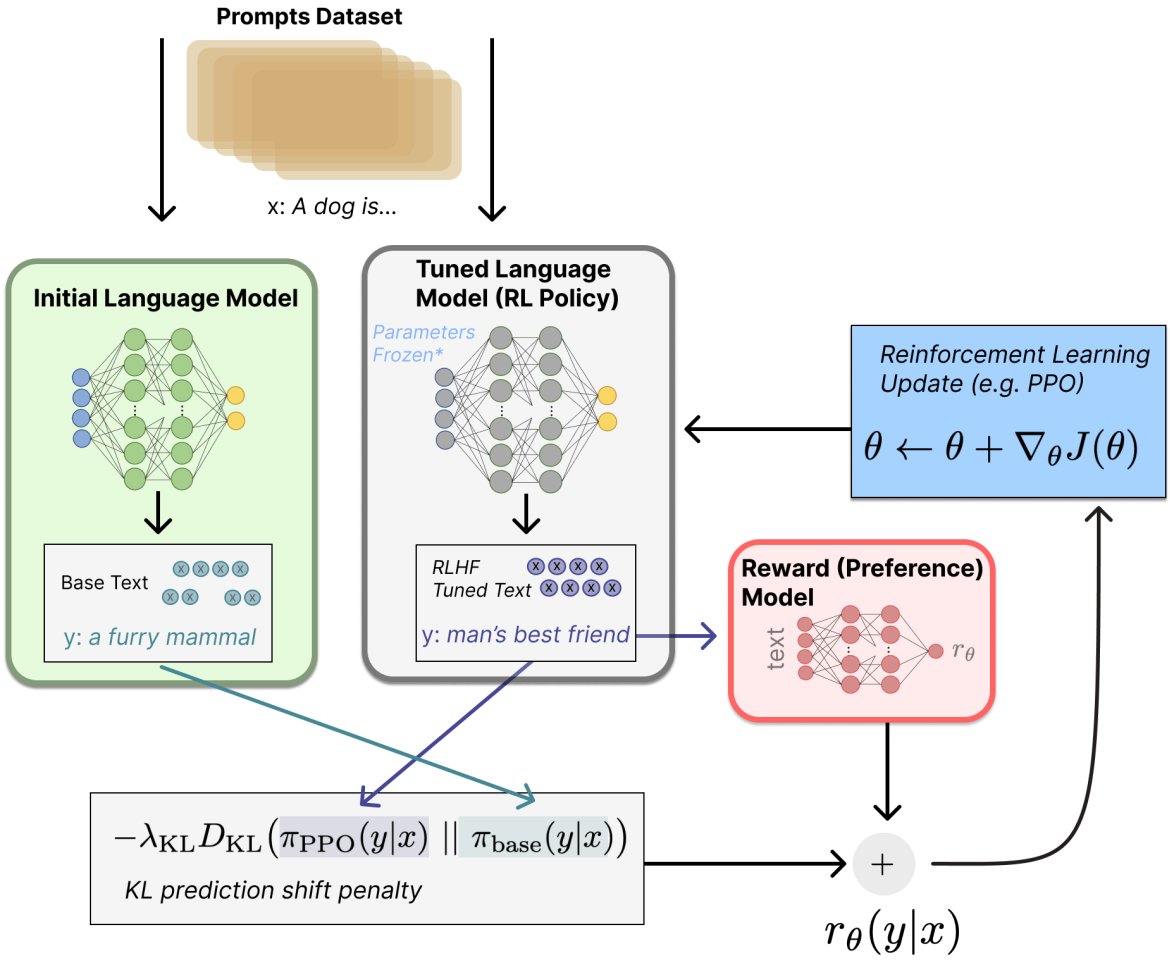
Proximal Policy Optimization Algorithms

Human feedback policies. We want to use the reward model trained above to train a policy that generates higher-quality outputs as judged by humans. We primarily do this using reinforcement learning, by treating the output of the reward model as a reward for the entire summary that we maximize with the PPO algorithm [58], where each time step is a BPE token.[8] We initialize our policy to be the model fine-tuned on Reddit TL;DR. Importantly, we include a term in the reward that penalizes the KL divergence between the learned RL policy π_ϕ^{RL} with parameters ϕ and this original supervised model π^{SFT}, as previously done in [25]. The full reward R can be written as:

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

This KL term serves two purposes. First, it acts as an entropy bonus, encouraging the policy to explore and deterring it from collapsing to a single mode. Second, it ensures the policy doesn't learn to produce outputs that are too different from those that the reward model has seen during training.

For the PPO value function, we use a Transformer with completely separate parameters from the policy. This prevents updates to the value function from partially destroying the pretrained policy early in training (see ablation in Appendix G.1). We initialize the value function to the parameters of the reward model. In our experiments, the reward model, policy, and value function are the same size.



1. Given a prompt, *x*, from the dataset, the text *y* is generated by the current iteration of the fine-tuned policy.
2. Concatenated with the original prompt, that text is passed to the preference model, which returns a scalar notion of "preferability", *rθ*.
3. In addition, per-token probability distributions from the RL policy are compared to the ones from the initial model to compute a penalty on the difference between them.
    1. In multiple papers from OpenAI, Anthropic, and DeepMind, this penalty has been designed as a scaled version of the Kullback–Leibler **(KL) divergence** between these sequences of distributions over tokens, Rkl.
    2. The KL divergence term penalizes the RL policy from moving substantially away from the initial pretrained model with each training batch, which can be useful to make sure the model outputs reasonably coherent text snippets.
    3. Without this penalty the optimization can start to generate text that is gibberish but fools the reward model to give a high reward.

    4. In practice, the KL divergence is approximated via sampling from both distributions (explained by John Schulman **here**). The final reward sent to the RL update rule is $r = r\theta - \lambda \text{Rkl}$.
        1. This KL term serves two purposes. First, it acts as an entropy bonus, encouraging the policy to explore and deterring it from collapsing to a single mode.
        2. Second, it ensures the policy doesn't learn to produce outputs that are too different from those that the reward model has seen during training.
4. the **update rule** is the parameter update from PPO that maximizes the reward metrics in the current batch of data (PPO is on-policy, which means the parameters are only updated with the current batch of prompt-generation pairs). PPO is a trust region optimization algorithm that uses constraints on the gradient to ensure the update step does not destabilize the learning process.

Results for RLHF with PPO:

- better than supervised learning in summarizing reddit posts and news articles
- over optimize the reward model hurt the true preference on llm output
- doubling the training data amount leads to a ~1.1% increase in the reward model validation set accuracy, whereas doubling the model size leads to a ~1.8% increase
- our reward models are sensitive to small but semantically important details in the summary.
- our learned reward models consistently outperform other metrics such as ROUGE, summary length, amount of copying from the post, and log probability under our baseline supervised models.

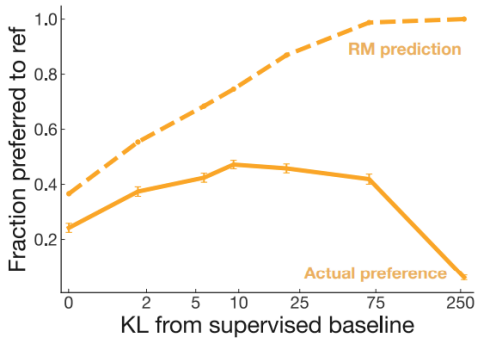


Figure 5: Preference scores versus degree of reward model optimization. Optimizing against the reward model initially improves summaries, but eventually overfits, giving worse summaries. This figure uses an earlier version of our reward model (see rm3 in Appendix C.6). See Appendix H.2 for samples from the KL 250 model.

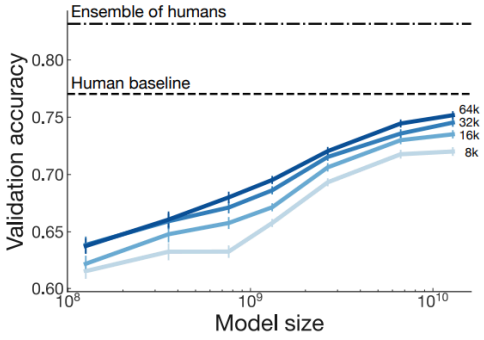


Figure 6: Reward model performance versus data size and model size. Doubling amount of training data leads to a ~1.1% increase in reward model validation accuracy, whereas doubling the model size leads to a ~1.8% increase. The 6.7B model trained on all data begins approaching the accuracy of a single human.

Direct Preference Optimization:

RLHF cons

- More complex
- High computational cost

In this paper, we show how to directly optimize a language model to adhere to human preferences, without explicit reward modeling or reinforcement learning.

Given a dataset of human preferences over model responses, DPO can therefore optimize a policy using a simple binary cross entropy objective, producing the optimal policy to an implicit reward function fit to the preference data.

our key insight is to leverage an analytical mapping from reward functions to optimal policies, which enables us to transform a loss function over reward functions into a loss function over policies. This change-of-variables approach avoids fitting an explicit, standalone reward model, while still optimizing under existing models of human preferences, such as the Bradley-Terry model. In essence, the policy network represents both the language model and the (implicit)

reward.

RL Fine-Tuning Phase: During the RL phase, we use the learned reward function to provide feedback to the language model. In particular, we formulate the following optimization problem

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

Deriving the DPO objective. We start with the same RL objective as prior work, Eq. 3, under a general reward function r. Following prior work [29, 28, 17, 15], it is straightforward to show that the optimal solution to the KL-constrained reward maximization objective in Eq. 3 takes the form:

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

where $Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$ is the partition function. See Appendix A.1 for a

Now that we have the probability of human preference data in terms of the optimal policy rather than the reward model, we can formulate a maximum likelihood objective for a parametrized policy π_θ. Analogous to the reward modeling approach (i.e. Eq. 2), our policy objective becomes:

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

This way, we fit an implicit reward using an alternative parameterization, whose optimal policy is simply π_θ. Moreover, since our procedure is equivalent to fitting a reparametrized Bradley-Terry

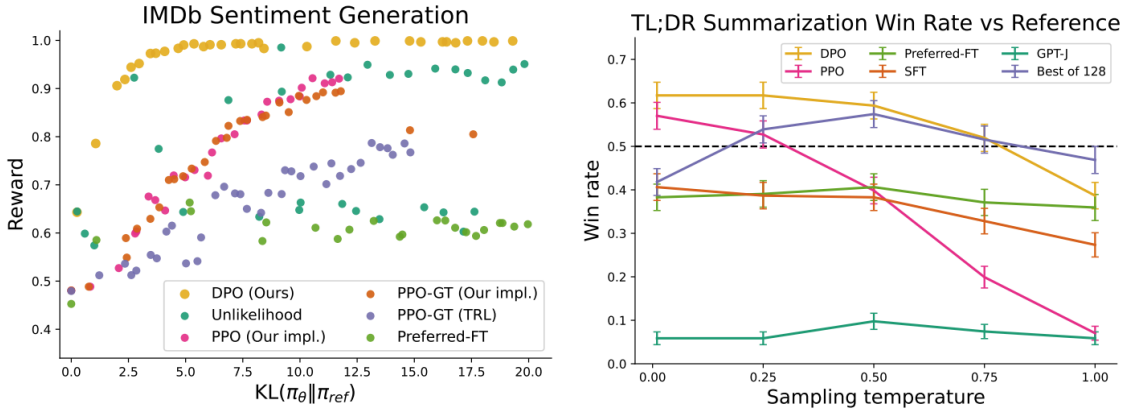


Figure 2: **Left.** The frontier of expected reward vs KL to the reference policy. DPO provides the highest expected reward for all KL values, demonstrating the quality of the optimization. **Right.** TL;DR summarization win rates vs. human-written summaries, using GPT-4 as evaluator. DPO exceeds PPO's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

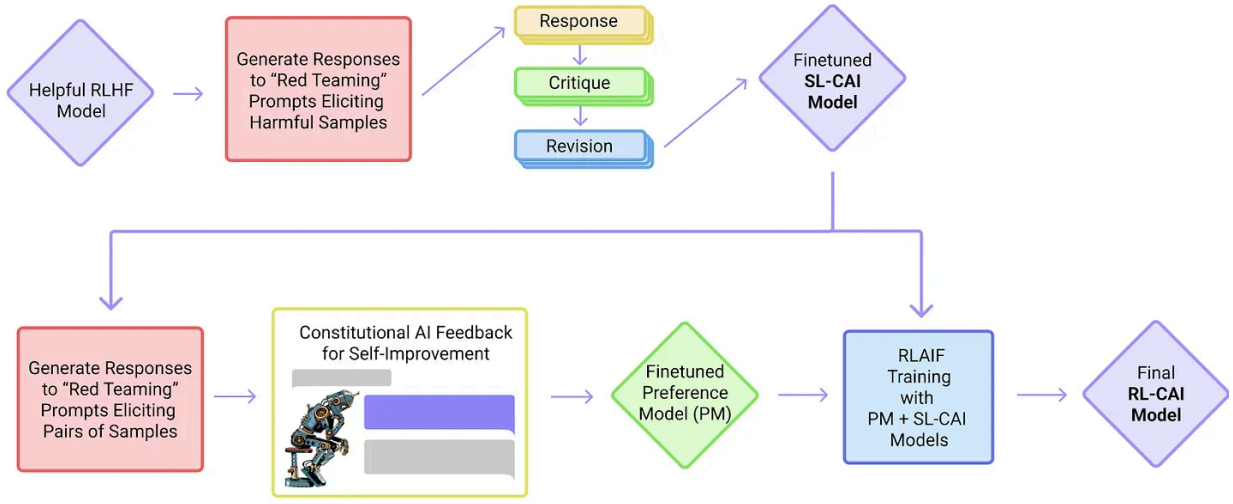
Experiments and Results

- DPO converges to its best performance relatively quickly.
- DPO policies can generalize similarly well to PPO policies, even though DPO does not use the additional unlabeled Reddit TL;DR prompts that PPO uses.
- These experiments are judged by GPT-4 e.g. GPT-4 decides if a completion is better than human written summaries

- And another experiment, We find that with both prompts, GPT-4 tends to agree with humans about as often as humans agree with each other, suggesting that GPT-4 is a reasonable proxy for human evaluations

RLAIF

Constitutional AI: Harmlessness from AI Feedback (Dec 2022, https://arxiv.org/abs/2212.08073)



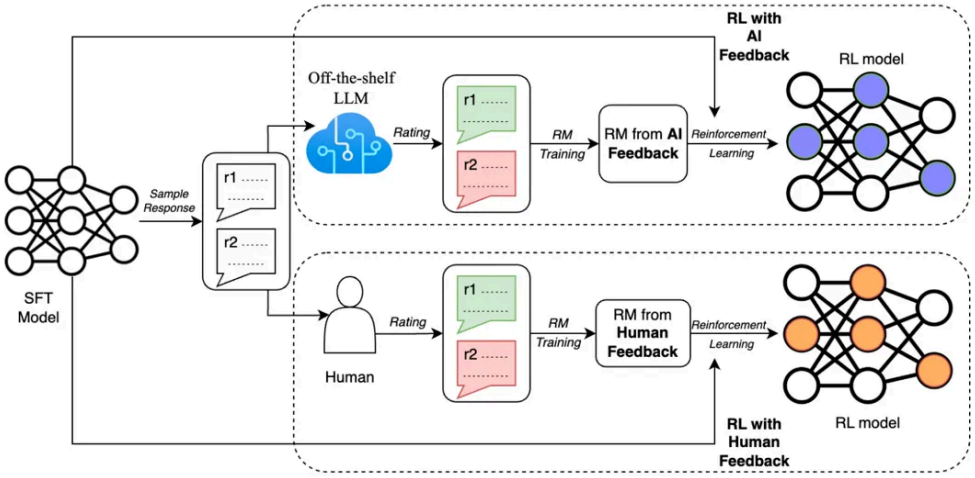
(Supervised Stage) Critique → Revision → Supervised Learning In the first stage of the process, we first generate responses to harmfulness prompts using a helpful-only AI assistant. These initial responses will typically be quite harmful and toxic. We then ask the model to critique its response according to a principle in the constitution, and then revise the original response in light of the critique. We revise responses repeatedly in a sequence, where we randomly draw principles from the constitution at each step. Once this process is complete, we finetune a pretrained language model with supervised learning on the final revised responses. The main purpose of this phase is to easily and flexibly alter the distribution of the model's responses, to reduce the need for exploration and the total length of training during the second RL phase.

(RL Stage) AI Comparison Evaluations → Preference Model → Reinforcement Learning This stage mimics RLHF, except that we replace human preferences for harmlessness with 'AI feedback' (i.e. we perform 'RLAIF'), where the AI evaluates responses according to a set of constitutional principles. Just as RLHF distills human preferences into a single preference model (PM), in this stage we distill LM interpretations of a set of principles back into a hybrid[5] human/AI PM (as we use human labels for helpfulness, but only AI labels for harmlessness). We begin by taking the AI assistant trained via supervised learning (SL) from the first stage, and use it to generate a pair of responses to each prompt in a dataset of harmful prompts (e.g. from [Ganguli et al., 2022]). We then formulate each prompt and pair into a multiple choice question, where we ask which response is best according to a constitutional principle. This produces an AI-generated preference dataset for harmlessness, which we mix with our human feedback helpfulness dataset. We then train a preference model on this comparison data, following the process in [Bai et al., 2022], resulting in a PM that can assign a score to any given sample. Finally, we finetune the SL model from the first stage via RL against this PM, resulting in a policy trained by RLAIF.

RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback (Sep 2023, https://arxiv.org/abs/2309.00267)

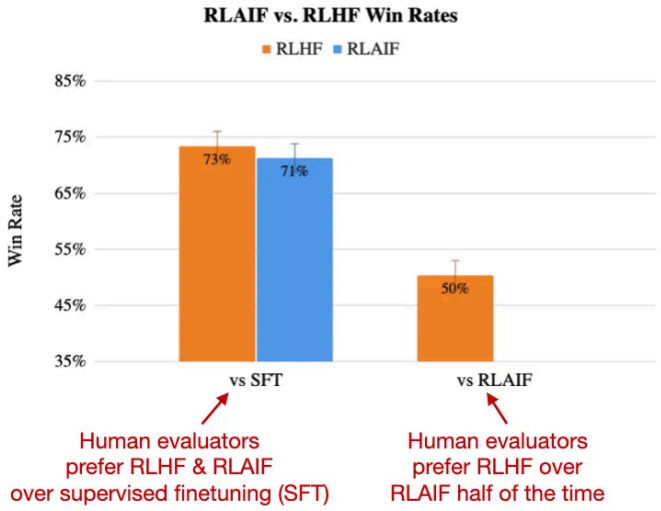
The main contributions of this work are as follows:

1. We demonstrate that RLAIF achieves comparable or superior performance to RLHF on the tasks of summarization, helpful dialogue generation, and harmless dialogue generation.
2. We show that RLAIF can improve upon a SFT policy even when the LLM labeler is the same size as the policy.
3. We find that directly prompting the LLM for reward scores during RL can outperform the canonical setup where a reward model is trained on LLM preferences.
4. We compare various techniques for generating AI labels and identify optimal settings for RLAIF practitioners.
    1. use chain of thoughts prompting and few shot prompting



RLHF where the ranking (rating) comes from an LLM

RLHF as in Llama 2



Human evaluators prefer RLHF & RLAIF over supervised finetuning (SFT)

Human evaluators prefer RLHF over RLAIF half of the time

Preamble	A good summary is a shorter piece of text that has the essence of the original. ... Given a piece of text and two of its possible summaries, explain which summary best adheres to coherence, accuracy, coverage, and overall quality as defined above.
Exemplar	»»»» Example «««« Text - We were best friends over 4 years ... Summary 1 - Broke up with best friend, should I wish her a happy birthday... And what do you think of no contact? Summary 2 - should I wish my ex happy birthday, I broke no contact, I'm trying to be more patient, I'm too needy, and I don't want her to think I'll keep being that guy. Thoughts on Summary 1 - Coherence - 7. Rationale: The summary is generally understandable, though it could be written with better grammar. Accuracy - 9. Rationale: The summary doesn't say things that aren't in the original text, and isn't misleading. Coverage - 6. Rationale: The summary covers most of the important information in the post and conveys the gist of the original text. However, it places more emphasis on "no contact" and could have mentioned the smothering/neediness to be more complete. Overall Quality - 7. Rationale: The summary represents the post fairly well with only minor areas where it could be improved. Thoughts on Summary 2 - Coherence - 3. Rationale: The summary is long-winded and has several grammatical errors. Accuracy - 4. Rationale: The summary mentions that the author broke no contact, but this is incorrect. Otherwise, it is accurate. Coverage - 8. Rationale: The summary covers the key points in the original text. Overall Quality - 4. Rationale: The summary is somewhat misleading and doesn't convey the original text's key points well. Preferred Summary=1 »»»» Follow the instructions and the example(s) above ««««
Sample to Annotate	Text - {text} Summary 1 - {summary1} Summary 2 - {summary2}
Ending	Thoughts on Summary 1 -

Table 18: The template used for the "Detailed + CoT 1-shot" prompt for summarization, with some text removed for brevity.

2 approaches:

1. Distilled RLAIF: produces soft labels (e.g. [0.6, 0.4]), and train a reward model on it

2. Direct RLAIF: ask LLM model to rate from 1 - 10

Evaluation:

- AI Labeler Alignment measures the accuracy of AI-labeled preferences with respect to human preferences.
- Win Rate evaluates the end-to-end quality of two policies by measuring how often one policy is preferred by human annotators over another.
- Harmless Rate measures the percentage of responses that are considered harmless by human evaluators

Results

- RLAIF achieves performance gains on par with or better than RLHF on all three tasks
- One natural question that arises is whether there is value in combining human and AI feedback. We experimented with combining both types of feedback but did not see an improvement beyond using human feedback alone.
- RLAIF can yield improvements even when the AI labeler model is the same size (in terms number of params) as the policy LLM.
    - We note that the AI labeler and initial policy are not the exact same model.
- Direct RLAIF performs better than Distilled RLAIF
    - One hypothesis for the improved quality is that bypassing the distillation from AI preferences into a RM enables information to flow directly from the off-the-shelf LLM to the policy.

Win Rate			Harmless Rate	
Comparison	**Summa -rization**	**Helpful dialogue**	**Model**	**Harmless dialogue**
RLAIF **vs** SFT	71%	63%	SFT	64%
RLHF **vs** SFT	73%	64%	RLHF	76%
RLAIF **vs** RLHF	50%	52%	RLAIF	88%
Same-size RLAIF **vs** SFT	68%			
Direct RLAIF **vs** SFT	74%			
Direct RLAIF **vs** Same-size RLAIF	60%			

Table 1: **Left side:** Win rates when comparing generations from two different models for the summarization and the helpful dialogue tasks, judged by human evaluators. **Right side:** Harmless rates across policies for the harmless dialogue task, judged by human evaluators.

- We observe that eliciting chain-of-thought reasoning generally improves AI labeler alignment, while the impacts of preamble specificity and in-context learning vary across tasks

- We also conduct experiments with selfconsistency (Wang et al., 2022b), where multiple chain-of-thought rationales are sampled with temperature T > 0. The preference distributions generated by the LLM are averaged together to arrive at the final preference label. We find that **selfconsistency** strictly degrades AI labeler alignment

	AI Labeler Alignment		
Prompt	Summary	H1	H2
Base 0-shot	76.1%	67.8%	69.4%
Base 1-shot	76.0%	67.1%	71.7%
Base 2-shot	75.7%	66.8%	**72.1%**
Base + CoT 0-shot	77.5%	**69.1%**	70.6%
Detailed 0-shot	77.4%	67.6%	70.1%
Detailed 1-shot	76.2%	67.6%	71.5%
Detailed 2-shot	76.3%	67.3%	71.6%
Detailed 8-shot	69.8%	–	–
Detailed + CoT 0-shot	**78.0%**	67.8%	70.1%
Detailed + CoT 1-shot	77.4%	67.4%	69.9%
Detailed + CoT 2-shot	76.8%	67.4%	69.2%

Table 2: We observe that eliciting chain-of-thought reasoning tends to improve AI labeler alignment, while few-shot prompting and detailed preambles have mixed effects across tasks. H1 refers to helpfulness, H2 to harmlessness.

- Results show that the policy trained with more aligned AI labels achieves a significantly higher win rate.
- larger ai labeler model size leads to better ai labeler alignment and produce even higher quality preference labels.
    - Since the AI labeler is only used to generate preference examples once and is not called during RL, using an even larger AI labeler is not necessarily prohibitively expensive.

Other Optimization options

The Wisdom of Hindsight Makes Language Models Better Instruction Followers

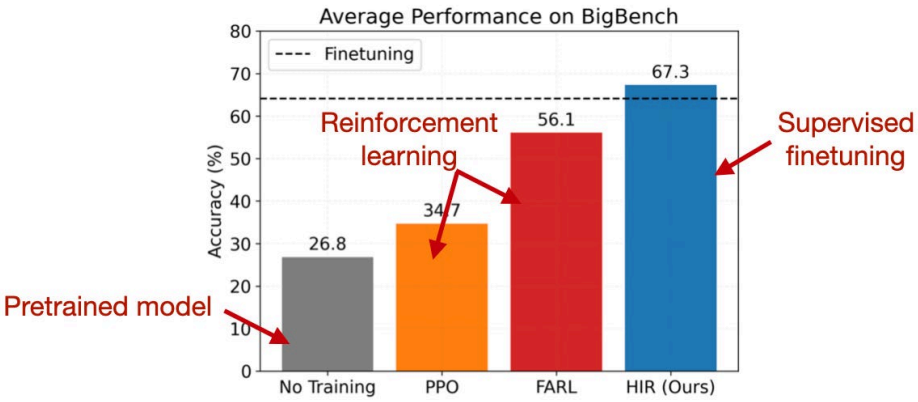
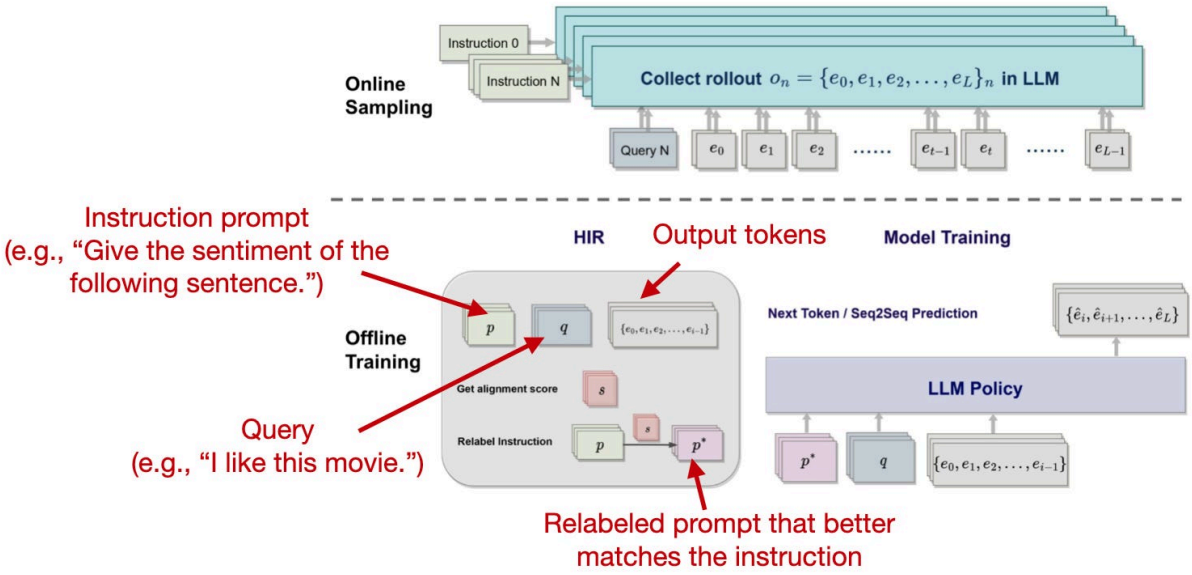


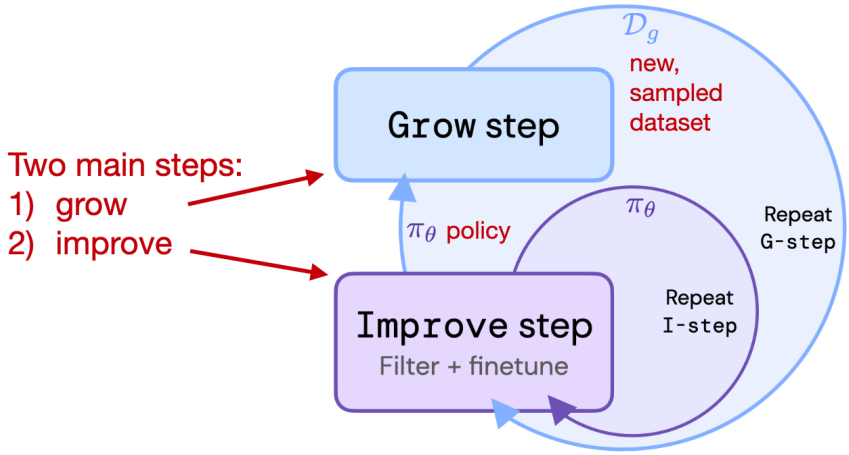
Figure 1. **Average Performance on BigBench.** HIR demonstrates a significant average performance gain over 12 tasks on BigBench compared to all baselines using FLAN-T5-Large.



Contrastive Preference Learning: Learning from Human Feedback without RL (Oct 2023, https://arxiv.org/abs/2310.13639)

Similar to DPO but used in robotics environment

(5) Reinforced Self-Training (ReST) for Language Modeling (Aug 2023, https://arxiv.org/abs/2308.08998)



References

- **An Introduction to Training LLMs Using Reinforcement Learning From Human Feedback (RLHF)**
- **Illustrating Reinforcement Learning from Human Feedback (RLHF)**
- **LLM Training: RLHF and Its Alternatives**
- **Learning to summarize from human feedback** https://arxiv.org/pdf/2009.01325.pdf
- Proximal Policy Optimization Algorithms https://arxiv.org/pdf/1707.06347.pdf
- RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback
- Direct Preference Optimization: Your Language Model is Secretly a Reward Model