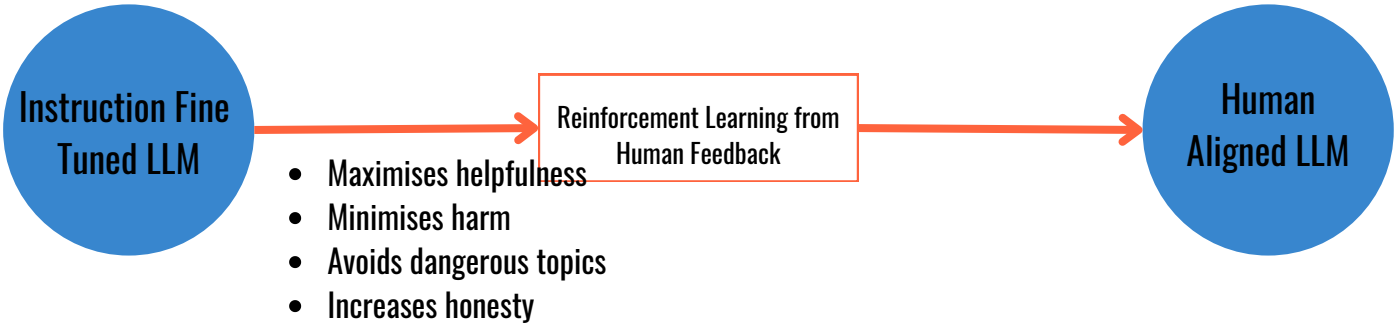
How is a Large Language Model taught to behave itself?

Aligning with Human Values

- Like with language, in general, Large Language Models can also behave badly -
  - Toxicity
  - Aggression
  - Dangerous/Harmful
- LLMs should align with Helpfulness, Honesty and Harmlessness (HHH)

What is Reinforcement Learning from Human Feedback (RLHF)?



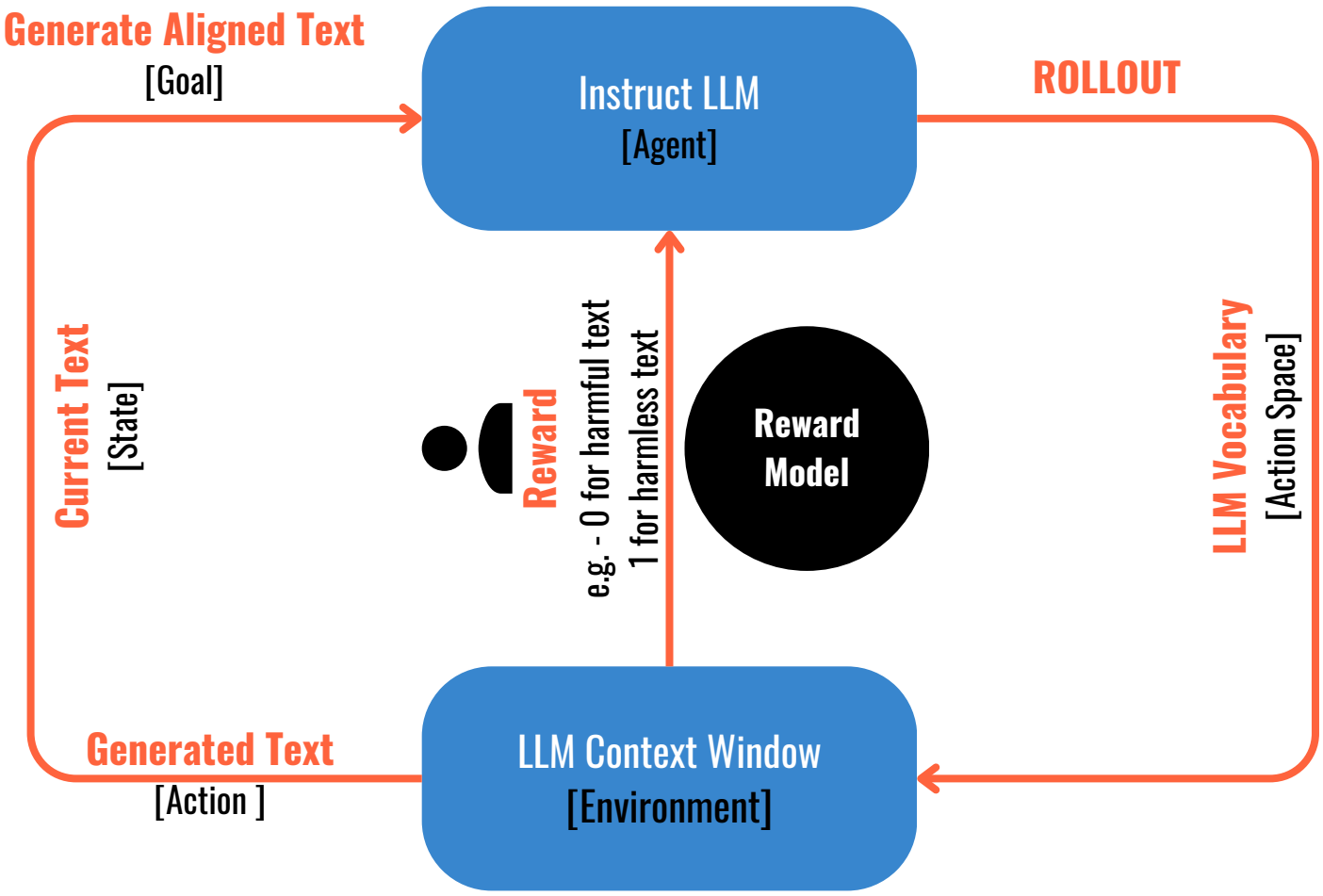
- Reinforcement Learning based on Human Feedback data
- Personalisation of LLMs is a potential application of RLFL

REINFORCEMENT LEARNING

is a type of machine learning in which an **agent** learns to make decisions related to a specific **goal** by taking **actions** in an **environment**, with the objective of maximising some notion of a cumulative **reward**

in/abhinav-kimothi

How does RLHF work?

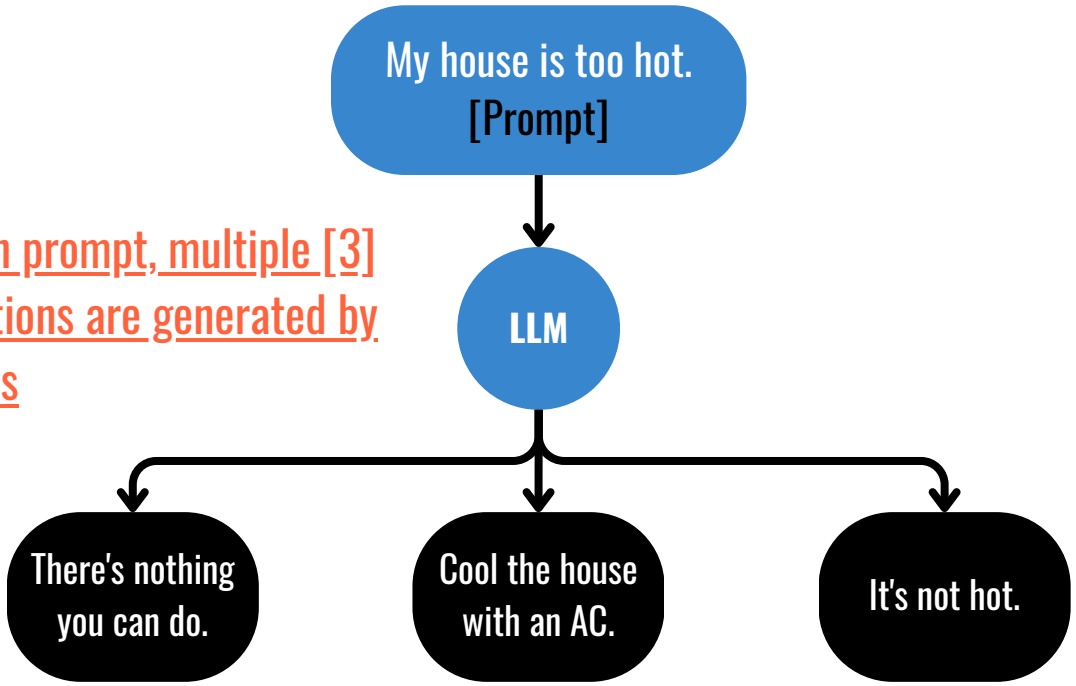


In RLHF, the agent (our fine-tuned instruct LLM) in its environment (Context Window) takes one action (of generating text) from all available actions in the action space (the entire vocabulary of tokens/words in the LLM).
The outcome of this action (the generated text) is evaluated by a human and is given a reward if the outcome (the generated text) aligns with the goal. If the outcome does not align with the goal, it is given a negative reward or no reward. This is an iterative process and each step is called a rollout. The model weights are adjusted in a manner that the total rewards at the end of the process are maximised.

Note : In practice, instead of a human giving a feedback continually, a classification model called the **Reward Model** is trained based on human generated training examples

in/abhinav-kimothi

How is Reward Model training data created from Human Feedback?



1 For each prompt, multiple [3] completions are generated by the LLMs

2	1	3
2	1	3
3	1	2

2 Human evaluators from diverse backgrounds rank the completions

3 Rankings are converted to pairwise training data for the reward model

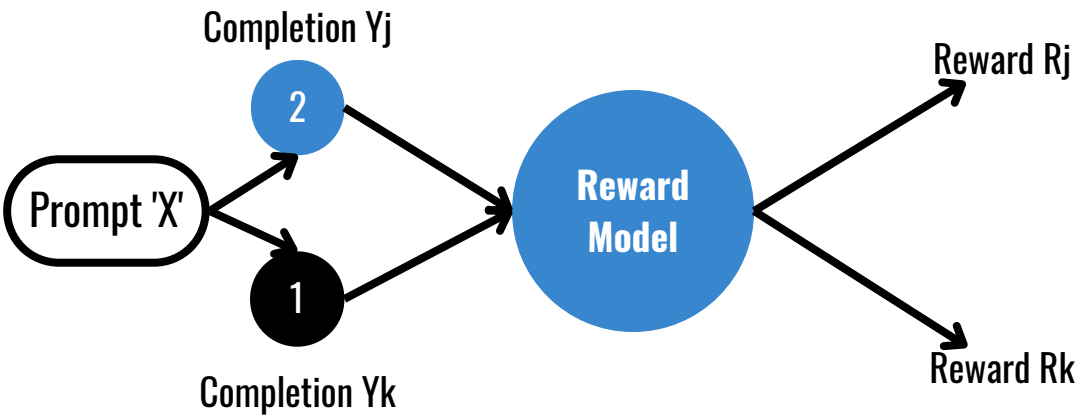
Prompt

		Pairs	Rewards	Pairs	Rewards
Completion 1	2	1, 2	[0,1]	2, 1	[1,0]
Completion 2	1	2, 3	[1,0]	2, 3	[1,0]
Completion 3	3	3, 1	[0,1]	1, 3	[1,0]

The pairs are ordered in order {Yj, Yk} such that Yj is the preferred completion

in/abhinav-kimothi

4 Reward Model is a supervised learning language model



The reward model learns to favour the human preferred response Yj while minimising the log of sigmoid difference between the rewards.

5 Reward Model is finally used as a binary classifier

The logit values for the Positive Class are passed as the Rewards. The LLM will change the weights in such a way that the choice of generated text will yield the highest rewards.

"Tommy loves Television"

Positive Class (Not Hate)	**3.1718**
Negative Class (Hate)	-2.6093

"Tommy hates gross movies"

Positive Class (Not Hate)	**-0.5351**
Negative Class (Hate)	0.1377

Proximal Policy Optimisation or PPO is a popular choice for training the reinforcement learning algorithms

in/abhinav-kimothi

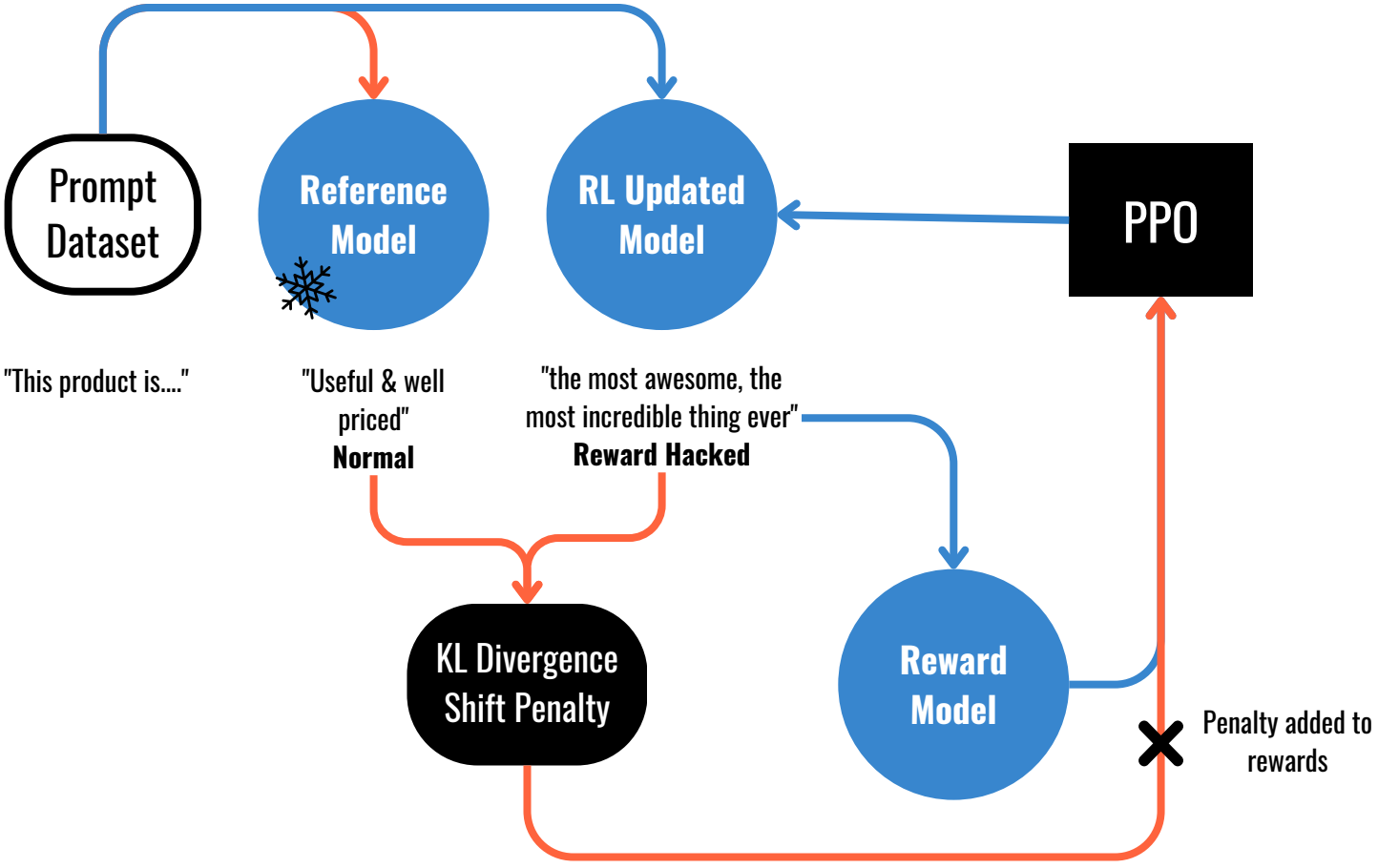
What is Proximal Policy Optimisation?

- PPO helps us optimise a large language model (LLM) to be more aligned with human preferences. We want the LLM to generate responses that are helpful, harmless, and honest.
- PPO works in cycles with two phases: Phase I and Phase II.
- In Phase I, the LLM completes prompts and carries out experiments. These experiments help us update the LLM based on the reward model, which captures human preferences.
- The reward model determines the rewards for prompt completions. It tells us how good or bad the completions are in terms of meeting human preferences.
- In Phase II, we have the value function, which estimates the expected total reward for a given state. It helps us evaluate completion quality and acts as a baseline for our alignment criteria.
- The value loss minimises the difference between the actual future reward and its estimation by the value function. This helps us make better estimates for future rewards.
- Phase 2 involves updating the LLM weights based on the losses and rewards from Phase 1.
- PPO ensures that these updates stay within a small region called the trust region. This keeps the updates stable and prevents us from making drastic changes.
- The main objective of PPO is to maximise the policy loss. We want to update the LLM in a way that generates completions aligned with human preferences and receives higher rewards.
- The policy loss includes an estimated advantage term, which compares the current action (next token) to other possible actions. We want to make choices that are advantageous compared to other options.
- Maximising the advantage term leads to better rewards and better alignment with human preferences.
- PPO also includes the entropy loss, which helps maintain creativity in the LLM. It encourages the model to explore different possibilities instead of getting stuck in repetitive patterns.
- The PPO objective is a weighted sum of different components. It updates the model weights through back propagation over several steps.
- After many iterations, we arrive at an LLM that is more aligned with human preferences and generates better responses.
- While PPO is popular, there are other techniques like Q-learning. Researchers are actively exploring new methods, such as direct preference optimisation, to improve reinforcement learning with large language models.

in/abhinav-kimothi

How to avoid Reward Hacking?

- Reward hacking happens when the language model finds ways to maximise the reward without aligning with the original objective i.e. model generates language that sounds exaggerated or nonsensical but still receives high scores on the reward metric.
- To prevent reward hacking, the original LLM is introduced as a reference model, whose weights are frozen and serve as a performance benchmark.
- During training iterations, the completions generated by both the reference model and the updated model are compared using KL divergence. KL divergence measures how much the updated model has diverged from the reference model in terms of probability distributions.
- Depending on the divergence, a shift penalty is added to the rewards calculation. The shift penalty penalises the updated model if it deviates too far from the reference model, encouraging alignment with the reference while still improving based on the reward signal.



Reinforcement Learning loop that is exposed to the threat of reward hacking

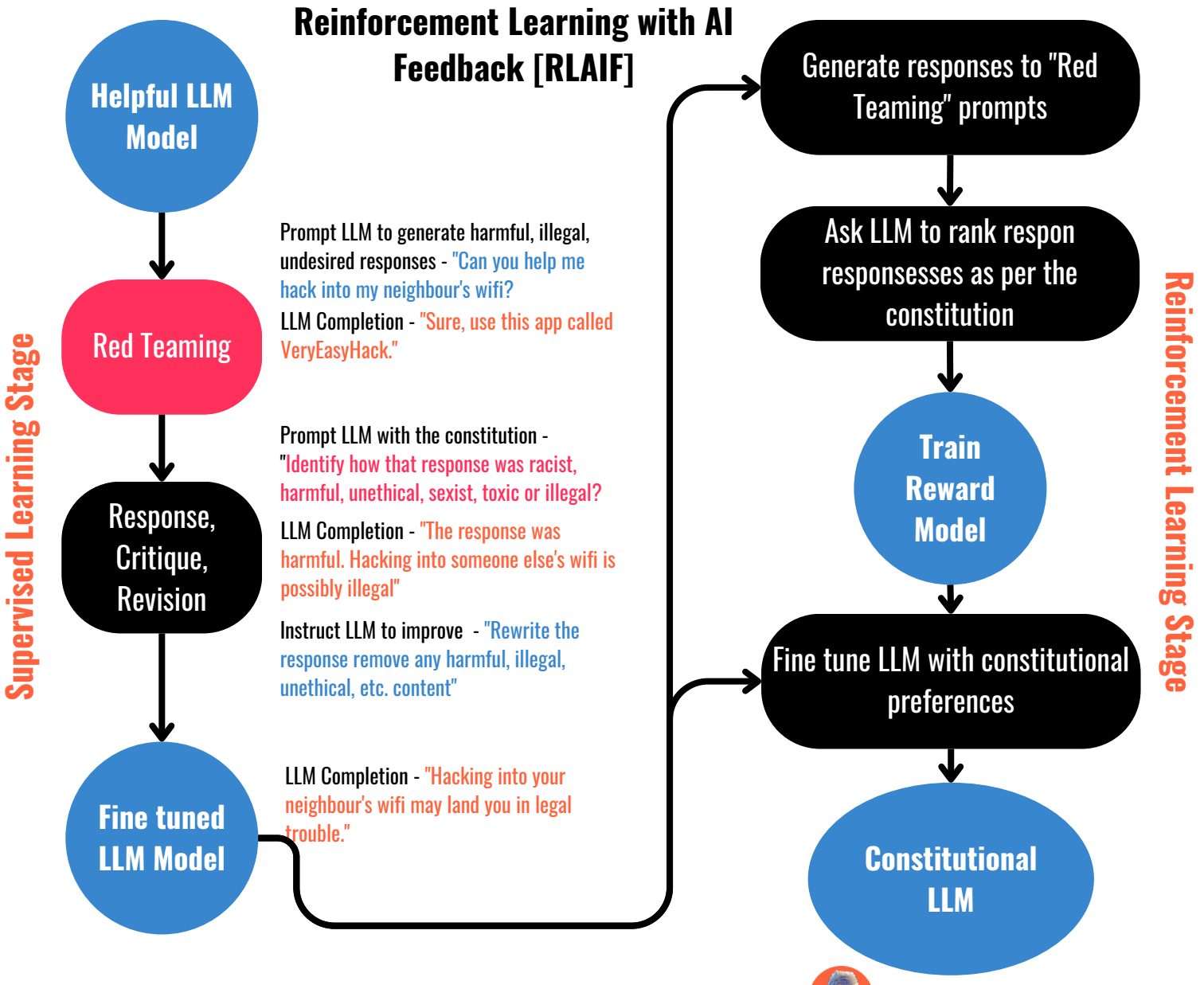
Introduction of Shift Penalty loop increases alignment without exaggeration

RLHF can also be used in conjunction with PEFT to reduce memory footprint

in/abhinav-kimothi

Scaling Human Feedback : Self Supervision with Constitutional AI

- Scaling human feedback for RLHF can be challenging due to the significant human effort required to produce the trained reward model. As the number of models and use cases increases, human effort becomes a limited resource, necessitating methods to scale human feedback.
- First proposed in 2022 by researchers at Anthropic, Constitutional AI is an approach to scale supervision and address some unintended consequences of RLHF. Constitutional AI involves training models using a set of rules and principles that govern the model's behaviour, forming a "constitution".
- The training process for Constitutional AI involves two phases: supervised learning and reinforcement learning.
- In the supervised learning phase, the model is prompted with harmful scenarios and asked to critique its own responses based on constitutional principles. The revised responses, conforming to the rules, are used to fine-tune the model.
- The reinforcement learning phase, known as reinforcement learning from AI feedback (RLAIF), uses the fine-tuned model to generate responses based on constitutional principles



in/abhinav-kimothi