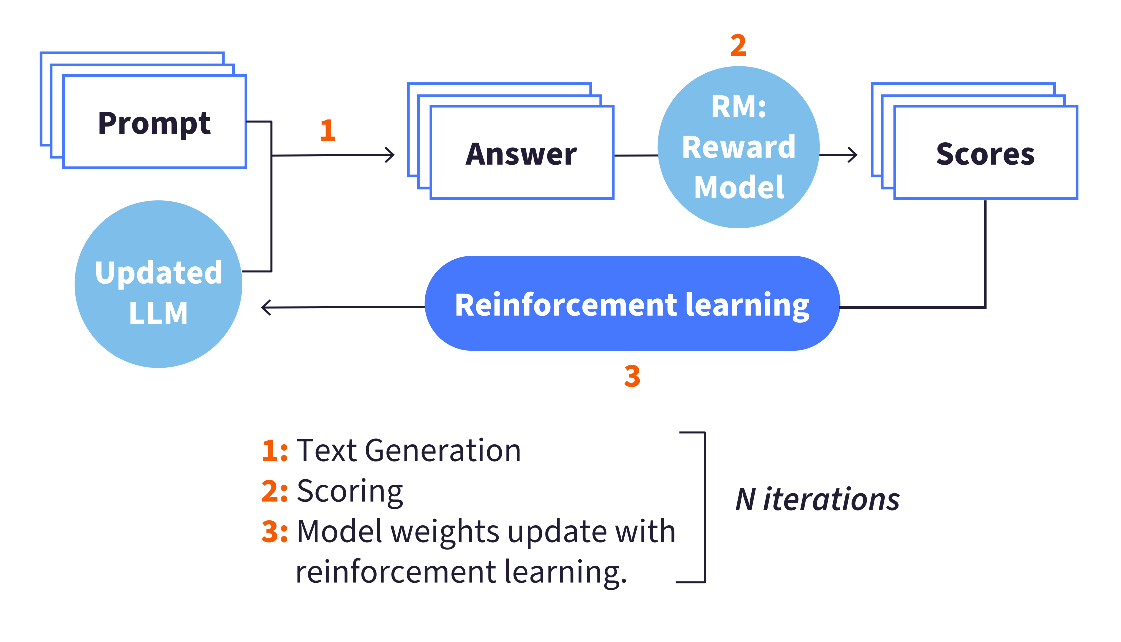
Preference Fine-Tuning (Part 2)

FINE-TUNING WITH RL & REWARD MODEL

The LLM weights are updated to create a human-aligned model via reinforcement learning, leveraging the reward model, and starting with a high-performing base model.

Goal: To align the LLM with provided instructions and human behavior.



Example:

Prompt: "A tree is..."
Iteration 1: "...a plant with a trunk." → Reward: 0.3
...
Iteration 4: "...a provider of shade and oxygen." → Reward: 1.6
...
Iteration n: "...a symbol of strength and resilience." → Reward: 2.9

As the process advances successfully, the reward will gradually increase until it meets the predefined evaluation criteria for helpfulness.

Updated model: The resulting updated model should be more aligned with human preferences.

Reinforcement learning algorithm: Proximal policy optimization (PPO) is a popular choice.

PPO ALGORITHM FOR LLMS

PPO iteratively updates the policy to **maximize the reward**, adjusting the LLM weights incrementally to **maintain proximity to the previous version** within a defined range for **stable learning.**

The **PPO objective** is used to update the LLM weights by backpropagation:

Hyperparameters

$$L^{PPO} = \underbrace{L^{POLICY}}_{\text{Policy loss}} + \underbrace{c_1 L^{VF}}_{\text{Value loss}} + \underbrace{c_2 L^{ENT}}_{\text{Entropy loss}}$$

Value Loss: Minimize it to improve return prediction accuracy.

$$\overset{\text{Value loss}}{\boxed{L^{VF}}} = \frac{1}{2}\left\| \underbrace{\overset{\text{Value function}}{\boxed{V_\theta(s)}}}_{\text{Estimated future total reward}} - \underbrace{\left(\sum_{t=0}^{T} \gamma^t r_t | s_0 = s\right)}_{\text{Actual Reward from the reward model}} \right\|_2^2$$

Policy Loss: Maximize it to get higher rewards while staying within reliable bounds.

$$L^{POLICY} = min(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}.\hat{A}_t, clip(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon).\hat{A}_t)$$

Probabilities of the next token with the updated LLM

Advantage term

Define "trust region"

Probabilities of the next token with the initial LLM

Guardrails
Keeping the policy in the "trust region"

π_θ Model's probability distribution over tokens

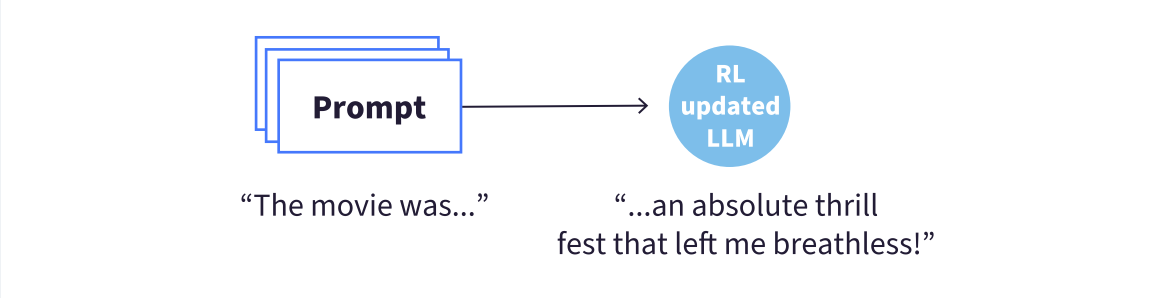
Entropy Loss: Maximize it to promote and sustain model creativity.

$$L^{ENT} = entropy(\pi_\theta(.|s_t))$$

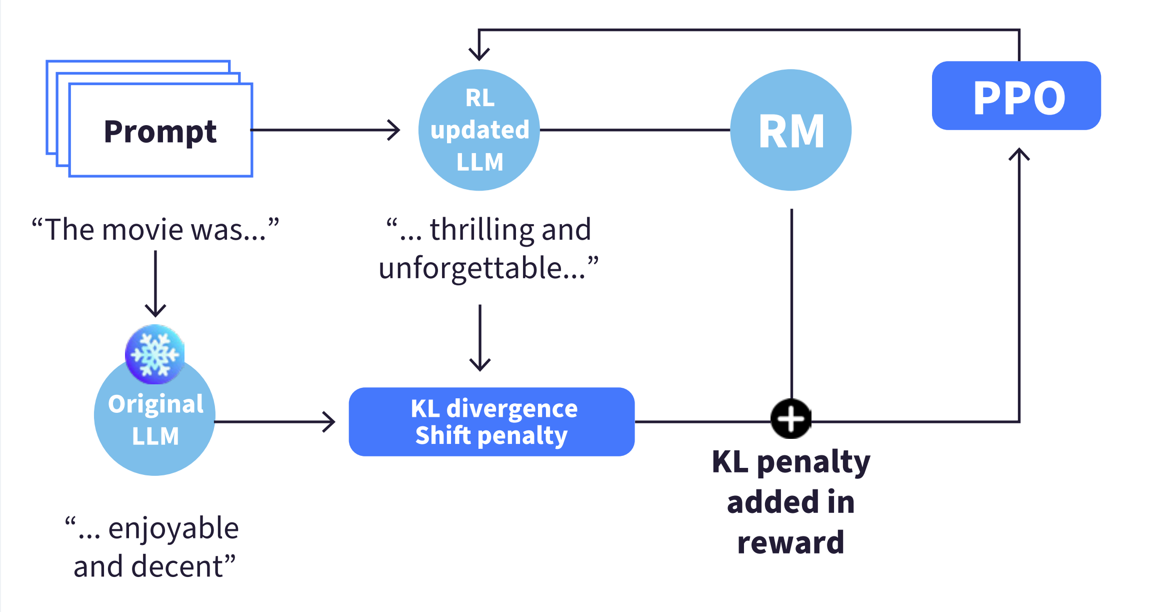
The higher the entropy, the more creative the policy.

REWARD HACKING

The agent **learns to cheat the system** by maximizing rewards at the expense of alignment with desired behavior.



To prevent reward hacking, **penalize RL updates** if they significantly deviate from the frozen original LLM, using **KL divergence.**



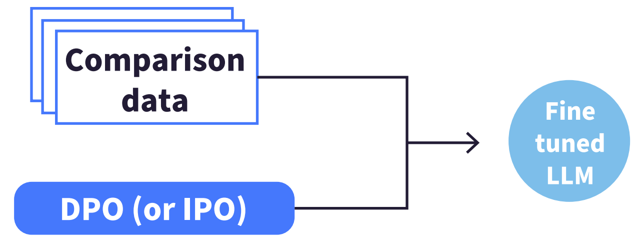
DIRECT PREFERENCE OPTIMIZATION

An **RLHF** pipeline is **difficult to implement:**

- Need to train a reward model
- New completions needed during training
- Instability of the RL algorithm

Direct Preference Optimization (**DPO**) is a simpler and more stable **alternative to RLHF**. It solves the same problem by minimizing a training loss directly based on the preference data (without reward modeling or RL).

Identity Preference Optimization (**IPO**) is a variant of DPO less prone to overfitting.



RL FROM AI FEEDBACK

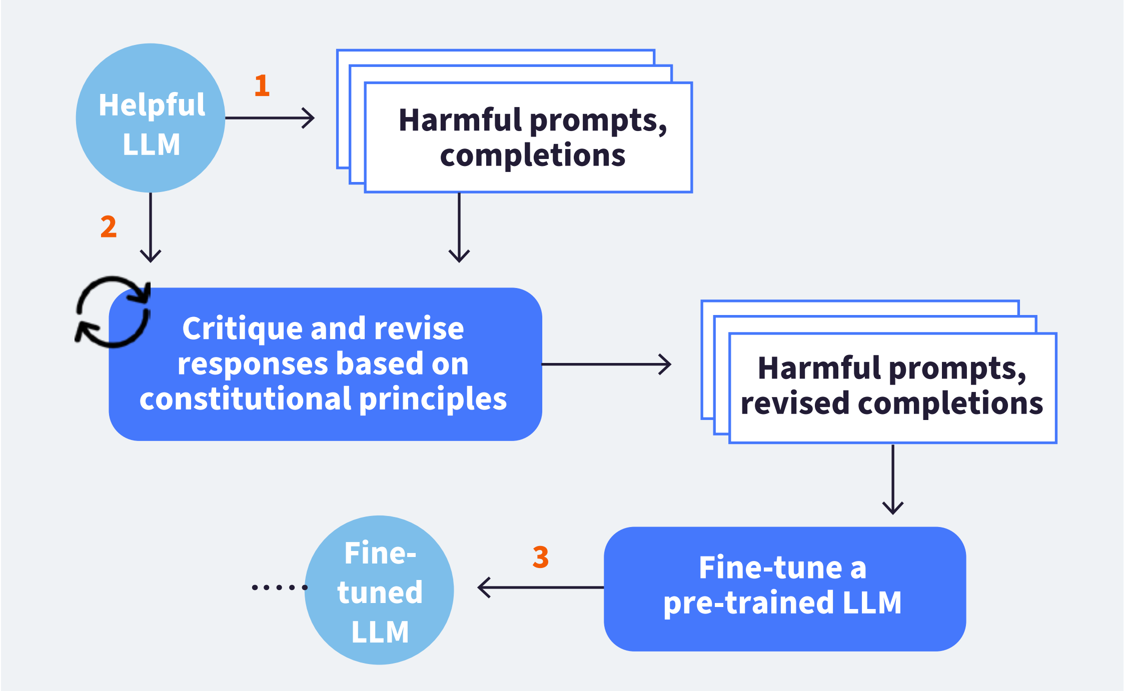
Obtaining the reward model is labor-intensive; scaling through AI-supervision is more precise and requires fewer human labels.

Constitutional AI (Bai, Yuntao, et al., 2022)

Approach that relies on a **set of principles** governing AI behavior, along with a small number of examples for few-shot prompting, collectively forming the "**constitution**."

Example of constitutional principle: "Please choose the response that is the most helpful, honest, and harmless."

1. Supervised Learning Stage



2. Reinforcement Learning (RL) Stage - RLAIF