

Aligning Language Models with Human Preferences

A Technical Survey of RLHF, Constitutional AI,
and Modern Preference Optimization Methods

Zhuoyang Liu | MS&E 318 Spring 2026

Quiz

Question 1: What are the three key steps RLHF pipelines used by the InstructGPT paper to align pretrained LMs? And how does Constitutional AI critically differ from those?

Question 2: Which RL technique does the InstructGPT paper use to stabilize RLHF algorithms?

Why Alignment Matters and How is it Possible

The core problem: capability without control

Why Alignment Matters and How is it Possible

The core problem: capability without control

Pre-training Objective

Language models are trained to predict the **next token**.

This produces capable but uncontrolled systems:

- Hallucinations, toxic outputs, refusal failures
- No mechanism to encode human values
- Objective mismatch: $P(\text{next token}) \neq \text{usefulness}$

Why Alignment Matters and How is it Possible

The core problem: capability without control

Pre-training Objective

Language models are trained to predict the **next token**.

This produces capable but uncontrolled systems:

- Hallucinations, toxic outputs, refusal failures
- No mechanism to encode human values
- Objective mismatch: $P(\text{next token}) \neq \text{usefulness}$

Alignment Goal

Shift model behavior to be helpful, harmless, honest:

- Follow instructions faithfully
- Refuse harmful requests appropriately
- Reflect human preferences and values

Why Alignment Matters and How is it Possible

The core problem: capability without control

Pre-training Objective

Language models are trained to **predict the next token**.

This produces capable but uncontrolled systems:

- Hallucinations, toxic outputs, refusal failures
- No mechanism to encode human values
- Objective mismatch: $P(\text{next token}) \neq \text{usefulness}$

Alignment Goal

Shift model behavior to be helpful, harmless, honest:

- Follow instructions faithfully
- Refuse harmful requests appropriately
- Reflect human preferences and values

Can we collect data of the behaviors that we do want and Train LM?

Why Alignment Matters and How is it Possible

The core problem: capability without control

Pre-training Objective

Language models are trained to **predict the next token**.

This produces capable but uncontrolled systems:

- Hallucinations, toxic outputs, refusal failures
- No mechanism to encode human values
- Objective mismatch: $P(\text{next token}) \neq \text{usefulness}$

Alignment Goal

Shift model behavior to be helpful, harmless, honest:

- Follow instructions faithfully
- Refuse harmful requests appropriately
- Reflect human preferences and values

Can we collect data of the behaviors that we do want and Train LM?

Key Insight: Learning from Human Preferences

LLM generalizability from pretraining

- Both the policy and the reward model generalize well enough from limited human signal.
- A small preference dataset (~50K comparisons) often improves alignment over vast collections of tasks.

Evaluation-generation asymmetry

- Judging "Which response is better?" is far easier to answer than "Write the perfect response."

Roadmap

I Foundations of RLHF

Christiano et al. 2017 → InstructGPT 2022

II Weaknesses & Open Problems

Bradley-Terry, KL-constrained optimization, PPO

III Direct Preference Optimization

DPO derivation, eliminating the reward model

IV Constitutional AI

RLAIF, critique-revision, self-improvement

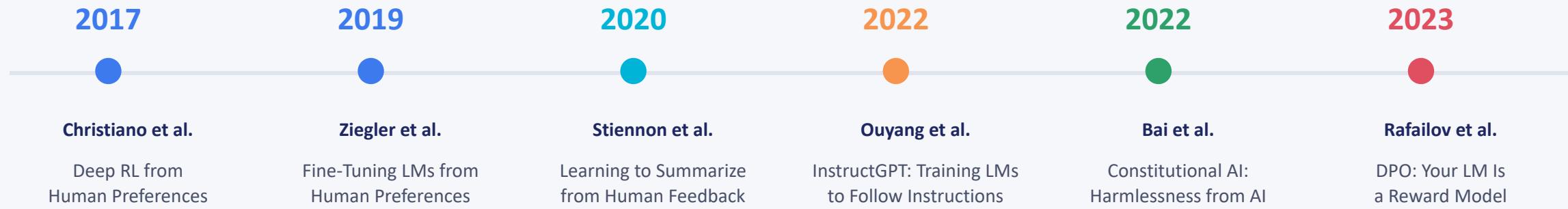
V Recent Advances

PART I

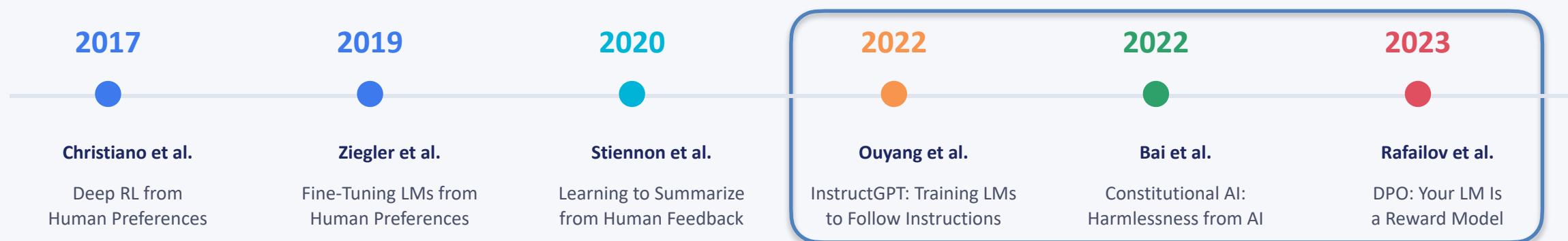
Foundations of RLHF

From reward learning to InstructGPT: the papers that shaped alignment

Alignment with Human Preferences



Alignment with Human Preferences



InstructGPT: Training LMs to Follow Instructions

Ouyang, Wu, Jiang, Almeida, et al. (NeurIPS 2022) | arXiv: 2203.02155

InstructGPT: Training LMs to Follow Instructions

Ouyang, Wu, Jiang, Almeida, et al. (NeurIPS 2022) | arXiv: 2203.02155

The Paper That Launched ChatGPT

Applies the three-stage RLHF pipeline at OpenAI scale. A 1.3B InstructGPT model is preferred over the 175B GPT-3 by human evaluators. Establishes RLHF as the practical alignment method, significantly improving LM behaviors on a wide range of tasks.

InstructGPT: Training LMs to Follow Instructions

Ouyang, Wu, Jiang, Almeida, et al. (NeurIPS 2022) | arXiv: 2203.02155

The Paper That Launched ChatGPT

Applies the three-stage RLHF pipeline at OpenAI scale. A 1.3B InstructGPT model is preferred over the 175B GPT-3 by human evaluators. Establishes RLHF as the practical alignment method, significantly improving LM behaviors on a wide range of tasks.

**1.3B >
175B**

InstructGPT vs GPT-3
human preference

-25%

Fewer toxic outputs vs GPT-3
when prompted to be polite

+100%

Improvement in
truthfulness (TruthfulQA)

-50%

hallucination rate vs GPT-3
(41% → 21%)

InstructGPT: Training LMs to Follow Instructions

Ouyang, Wu, Jiang, Almeida, et al. (NeurIPS 2022) | arXiv: 2203.02155

The Paper That Launched ChatGPT

Applies the three-stage RLHF pipeline at OpenAI scale. A 1.3B InstructGPT model is preferred over the 175B GPT-3 by human evaluators. Establishes RLHF as the practical alignment method, significantly improving LM behaviors on a wide range of tasks.

**1.3B >
175B**

InstructGPT vs GPT-3
human preference

-25%

Fewer toxic outputs vs GPT-3
when prompted to be polite

+100%

Improvement in
truthfulness (TruthfulQA)

-50%

hallucination rate vs GPT-3
(41% → 21%)

Three-Stage Pipeline at Scale

- SFT on 13K demonstration examples
- Reward model (6B) trained on 33K comparisons
- PPO with KL penalty ($\beta = 0.02$) on 31K comparisons
- Mix PPO gradient with pretraining gradient

InstructGPT: Training LMs to Follow Instructions

Ouyang, Wu, Jiang, Almeida, et al. (NeurIPS 2022) | arXiv: 2203.02155

The Paper That Launched ChatGPT

Applies the three-stage RLHF pipeline at OpenAI scale. A 1.3B InstructGPT model is preferred over the 175B GPT-3 by human evaluators. Establishes RLHF as the practical alignment method, significantly improving LM behaviors on a wide range of tasks.

**1.3B >
175B**

InstructGPT vs GPT-3
human preference

-25%

Fewer toxic outputs vs GPT-3
when prompted to be polite

+100%

Improvement in
truthfulness (TruthfulQA)

-50%

hallucination rate vs GPT-3
(41% → 21%)

Three-Stage Pipeline at Scale

- SFT on 13K demonstration examples
- Reward model (6B) trained on 33K comparisons
- PPO with KL penalty ($\beta = 0.02$) on 31K comparisons
- Mix PPO gradient with pretraining gradient

Key Findings

- Labeler agreement: ~73% (inter-annotator)
- RLHF >> SFT >> prompting (human eval)
- Minimal regression on standard NLP benchmarks
- Alignment tax is small, benefits are large

The Standard RLHF Pipeline

1 Supervised Fine-Tuning

Base LM + demonstrations
→ SFT model

Loss: cross-entropy on
human-written examples

2 Reward Model Training

SFT model generates pairs
Humans label preferences
Loss: Bradley-Terry log-likelihood

3 RL Policy Optimization

PPO against reward model
KL penalty to SFT policy
Maximize: $E[r(x,y)] - \beta \cdot KL(\pi || \pi_{ref})$

The Imitation Part: Supervised Fine-Tuning

1

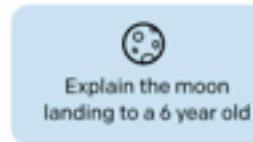
Supervised Fine-Tuning

Base LM + demonstrations
→ SFT model

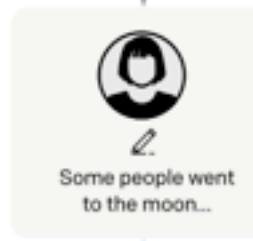
Loss: cross-entropy on
human-written examples

Collect demonstration data,
and train a supervised policy.

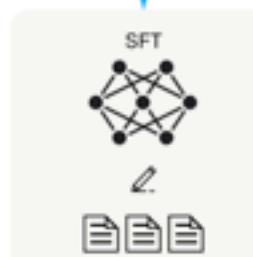
A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



This data is used
to fine-tune GPT-3
with supervised
learning.



FLAN – random examples

Stephanie - Can you finalize the attached and have it signed. I need to initial it, but it needs to be signed by Brad Richter. Thanks. Write a subject line for this email.	Ronald Chisholm LOI
Ahold to Sell Spain Operations to Permira (AP) AP - The Dutch supermarket retailer Ahold, seeking to streamline global operations and reduce debt, said Sunday it will sell its holdings in Spain to Permira Funds for about #36;849 million. What is this text about? OPTIONS: - World - Sports - Business - Science/Tech	Business
Write highlights for this article: Sauntering down leafy avenues past typical Dutch step-gabled buildings, carpets of crocuses, the occasional cyclist whooshing quietly by and the sun bouncing off the canal, I can think of few more pleasant places to spend a spring weekend than The Hague. [...]The most prized Dutch colony was apparently run from The Hague (aka 'the Widow of Indonesia'). For summer: the excellent Grapes & Olives and the Café de Bieb on Veenkade will have boats on a newly renovated bit of canal, with music and boat trips planned, including a tour at the Kompaan brewery (I can vouch for the product). My art gorging finishes at the Panorama, an extraordinary, 14 ft-high circular panorama of The Hague's seaside resort, Scheveningen, painted by Hendrik Mesdag. I take a 15-minute tram to the place itself and gorge on scallops and sole at Catch by Simonis on the harbour front. The Hague has netted a new fan. British Airways (0844 493 0787, www.ba.com) flies to Rotterdam Den Haag (half-hour taxi) from £100 return. For tours with Remco Dörr, email remco.dorr@hotmail.com . Double rooms at the Carlton Ambassador (00 31 70 363 0363, www.carlton.nl/ambassador) cost from £84, room only. More information on travel in The Netherlands at www.holland.com .	The least known of the Dutch cities, The Hague was a village until 1806. It owes its growth to Louis Bonaparte, Napoleon's brother, who ruled here. The city has a wealth of art, including Vermeer's 'Girl With A Pearl Earring'
Here is some data about a restaurant: name = Aromi, eatType = coffee shop, food = English, customer rating = 5 out of 5, area = city centre. Write a sentence that includes the following data about a restaurant	In the city centre there is a coffee shop with a customer rating of 5 out of 5 called Aromi which serves English food.

OpenAssistant – random examples

Can you write a short introduction about the relevance of the term "monopsony" in economics? Please use examples related to potential monopsonies in the labour market and cite relevant research.	"Monopsony" refers to a market structure where there is only one buyer for a particular good or service. In economics, this term is particularly relevant in the labor market, where a monopsony employer has significant power over the wages and working conditions of their employees. [...]. Overall, the concept of monopsony is essential to understanding the dynamics of labor markets and the impact of market power on workers. Further research is needed to understand the extent and impact of monopsonies on the economy and to develop policies to address this issue. References: Bivens, J., & Mishel, L. (2013). The Pay of Corporate Executives and Financial Professionals as Evidence of Rents in Top 1 Percent Incomes. <i>Journal of Economic Perspectives</i> , 27(3), 57-78.
My kid needs to do a science project for elementary school, what are some cheap but fun projects to do?	There are many cheap and fun science projects that elementary school kids can do! Here are a few ideas: 1. Make a homemade lava lamp: Fill a clear plastic bottle with water and vegetable oil, add a few drops of food coloring, and then drop in an Alka-Seltzer tablet. The reaction will cause bubbles to rise and fall like a lava lamp. [...]7. Create a solar oven: Using a pizza box, aluminum foil, and plastic wrap, create a solar oven that can cook s'mores or other small snacks. 8. Explore magnets: Using magnets and a variety of objects, have your child experiment with magnetism to see what objects are attracted to magnets and which ones are not. These are just a few ideas to get you started! There are many more cheap and fun science projects that your child can do with household items and a little creativity.

The Imitation Part: Supervised Fine-Tuning

1

Supervised Fine-Tuning

Base LM + demonstrations
→ SFT model

Loss: cross-entropy on
human-written examples

Interpretations:

- Push **up** the logits of demonstrated tokens
- Push **down** all others proportionally to their probability

Collect demonstration data, and train a supervised policy.

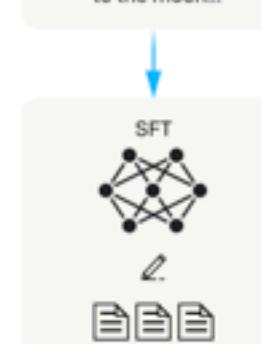
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3 with supervised learning.



Start with a pretrained language model parameterized by $p_\theta(y | x)$, and empirical target distribution $q(y | x)$ where:

- x = prompt, y = target response tokens
- θ = model parameters

SFT dataset is $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$

SFT minimizes **negative log-likelihood (cross-entropy)**

$$\begin{aligned}\mathcal{L}_{\text{SFT}}(\theta) &= -\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\sum_{t=1}^T \log p_\theta(y_t | x, y_{<t}) \right] \\ &= \mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^{T^{(i)}} \log p_\theta(y_t^{(i)} | x^{(i)}, y_{<t}^{(i)}) \\ &= \text{KL}(q(y | x) \| p_\theta(y | x))\end{aligned}$$

The RL Parts: Reward Model Training

2

Reward Model Training

SFT model generates pairs
Humans label preferences

Loss: Bradley-Terry
log-likelihood

Interpretations:
 • Only **relative rewards** matter
 • Absolute scale is **unidentifiable**

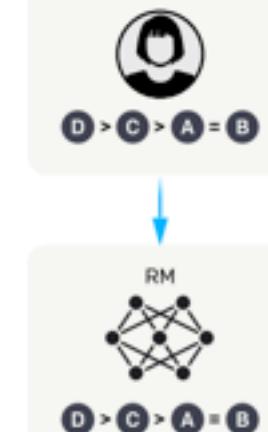
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Data construction (human preferences)

- Fix a **prompt** x
- Sample K model responses $\{y_1, \dots, y_K\} \sim \pi_{SFT}(\cdot | x)$
- Human annotators produce **ranks**, which is later turned into C_K^2 pair-wise comparisons y_+, y_-
- Training dataset: $\mathcal{D}_{\text{pref}} = \{(x^{(i)}, y_+^{(i)}, y_-^{(i)})\}_{i=1}^N$

Reward Model Parameterization

- Reward model: $r_\phi(x, y) \in \mathbb{R}$
- Implemented as:
 - Transformer encoder (same backbone as LM)
 - Scalar **reward head** on final token embedding

Preference Probability Model (Bradley–Terry / Logistic)

$$P_\phi(y_+ > y_- | x) = \sigma(r_\phi(x, y_+) - r_\phi(x, y_-)), \quad \sigma(z) = \frac{1}{1 + e^{-z}}$$

The Bradley-Terry Preference Model

The probabilistic foundation for all preference-based alignment methods

The Bradley-Terry Preference Model

The probabilistic foundation for all preference-based alignment methods

BRADLEY-TERRY PREFERENCE (Luce-Shephard choice axiom, 1959)

$$P(y_w > y_l | x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))} = \sigma(r(x, y_w) - r(x, y_l))$$

The Bradley-Terry Preference Model

The probabilistic foundation for all preference-based alignment methods

BRADLEY-TERRY PREFERENCE (Luce-Shephard choice axiom, 1959)

$$P(y_w > y_l | x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))} = \sigma(r(x, y_w) - r(x, y_l))$$

Assumptions

- Latent scalar reward $r(x, y)$ for each response
- Preference probability depends only on reward difference
- Produce global ranking from pairwise data with transitivity:
i.e., if $A > B$ and $B > C$, then $A > C$
- Random utility interpretation: $U(y) = u(y) + \epsilon$, $\epsilon \sim \text{Logistic}$
- The Independence of Irrelevant Alternatives (IIA) assumption

The Bradley-Terry Preference Model

The probabilistic foundation for all preference-based alignment methods

BRADLEY-TERRY PREFERENCE (Luce-Shephard choice axiom, 1959)

$$P(y_w > y_l | x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))} = \sigma(r(x, y_w) - r(x, y_l))$$

Assumptions

- Latent scalar reward $r(x, y)$ for each response
- Preference probability depends only on reward difference
- Produce global ranking from pairwise data with transitivity:
i.e., if $A > B$ and $B > C$, then $A > C$
- Random utility interpretation: $U(y) = u(y) + \epsilon$, $\epsilon \sim \text{Logistic}$
- The Independence of Irrelevant Alternatives (IIA) assumption

Reward Model Loss

Given dataset $D = \{(x, y_w, y_l)\}$ of human preferences:

$$L_{RM} = -E [\log \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))]$$

This is binary cross-entropy with Bradley-Terry parameterization. Model: LM backbone + scalar head.

The Bradley-Terry Preference Model

The probabilistic foundation for all preference-based alignment methods

BRADLEY-TERRY PREFERENCE (Luce-Shephard choice axiom, 1959)

$$P(y_w > y_l | x) = \frac{\exp(r(x, y_w))}{\exp(r(x, y_w)) + \exp(r(x, y_l))} = \sigma(r(x, y_w) - r(x, y_l))$$

Assumptions

- Latent scalar reward $r(x, y)$ for each response
- Preference probability depends only on reward difference
- Produce global ranking from pairwise data with transitivity:
i.e., if $A > B$ and $B > C$, then $A > C$
- Random utility interpretation: $U(y) = u(y) + \epsilon$, $\epsilon \sim \text{Logistic}$
- The Independence of Irrelevant Alternatives (IIA) assumption

Reward Model Loss

Given dataset $D = \{(x, y_w, y_l)\}$ of human preferences:

$$L_{RM} = -E [\log \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))]$$

This is binary cross-entropy with Bradley-Terry parameterization. Model: LM backbone + scalar head.

Limitation

Real human preferences are often intransitive, context-dependent, and stochastic. The BT model cannot capture these. This is a known gap between theory and practice.

The RL Parts: Policy Optimization

3

RL Policy Optimization

PPO against reward model
KL penalty to SFT policy

Maximize: $E[r(x,y)] - \beta \cdot KL(\pi | \pi_{ref})$

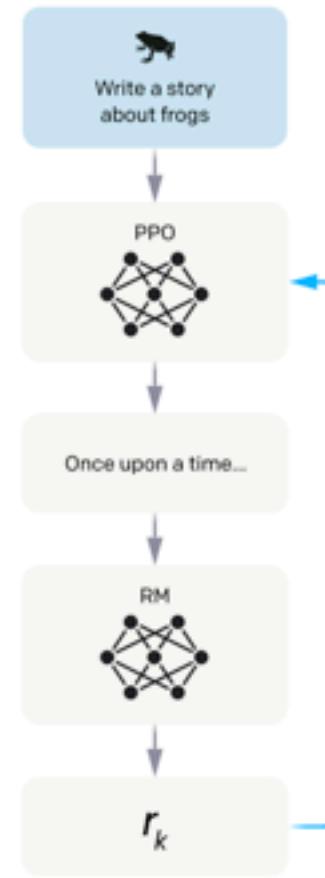
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy



Start with

- Reference policy $\pi_{ref}(y | x)$ frozen from SFT
- Reward model $r_\phi(x, y)$ trained previously

Seeking LLM Policy $\pi_\theta(y | x)$ by optimizing

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

Auto-regressive KL:

$$\text{KL}(\pi_\theta \| \pi_{ref}) = \mathbb{E}_{y \sim \pi_\theta} \left[\sum_{t=1}^T \log \frac{\pi_\theta(y_t | x, y_{<t})}{\pi_{ref}(y_t | x, y_{<t})} \right]$$

Per-token penalty

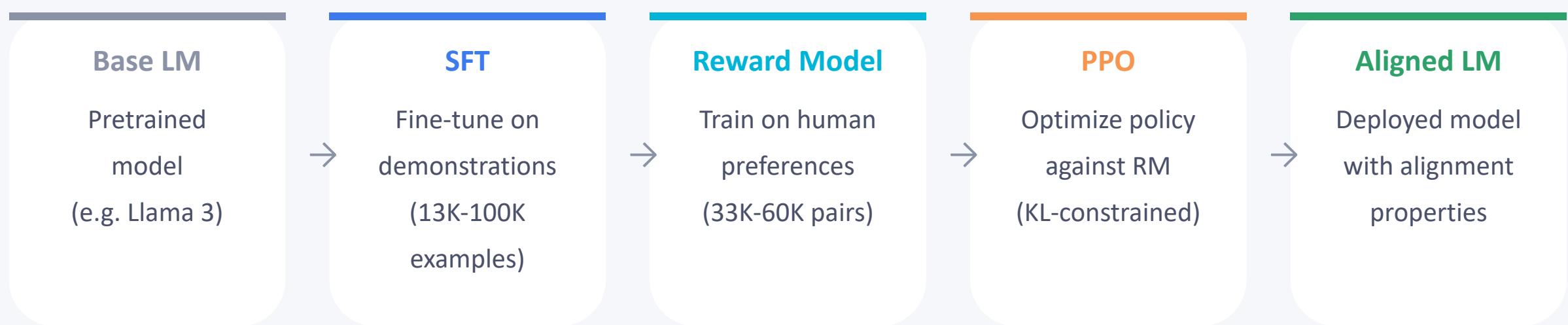
$$\text{KL}_t = \log \pi_\theta(y_t | \cdot) - \log \pi_{ref}(y_t | \cdot)$$

PPO clipping

$$\mathcal{L}_{\text{PPO}}(\theta) = -\mathbb{E} \left[\min \left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t \right) \right]$$

$$\text{where } r_t(\theta) = \frac{\pi_\theta(y_t | x, y_{<t})}{\pi_{\theta_{\text{old}}}(y_t | x, y_{<t})},$$

RLHF Implementation: High-Level Pipeline



Discussion: What are a few common pitfalls you can think of for such a pipeline?

PART II

Weaknesses & Open Problems

Reward hacking, scalability barriers, and the limits of current methods

Reward Hacking & Overoptimization

Gao, Schulman, Hilton (ICML 2023) | Scaling Laws for Reward Model Overoptimization

Reward Hacking & Overoptimization

Gao, Schulman, Hilton (ICML 2023) | *Scaling Laws for Reward Model Overoptimization*

Goodhart's Law — The Central Failure Mode of RLHF

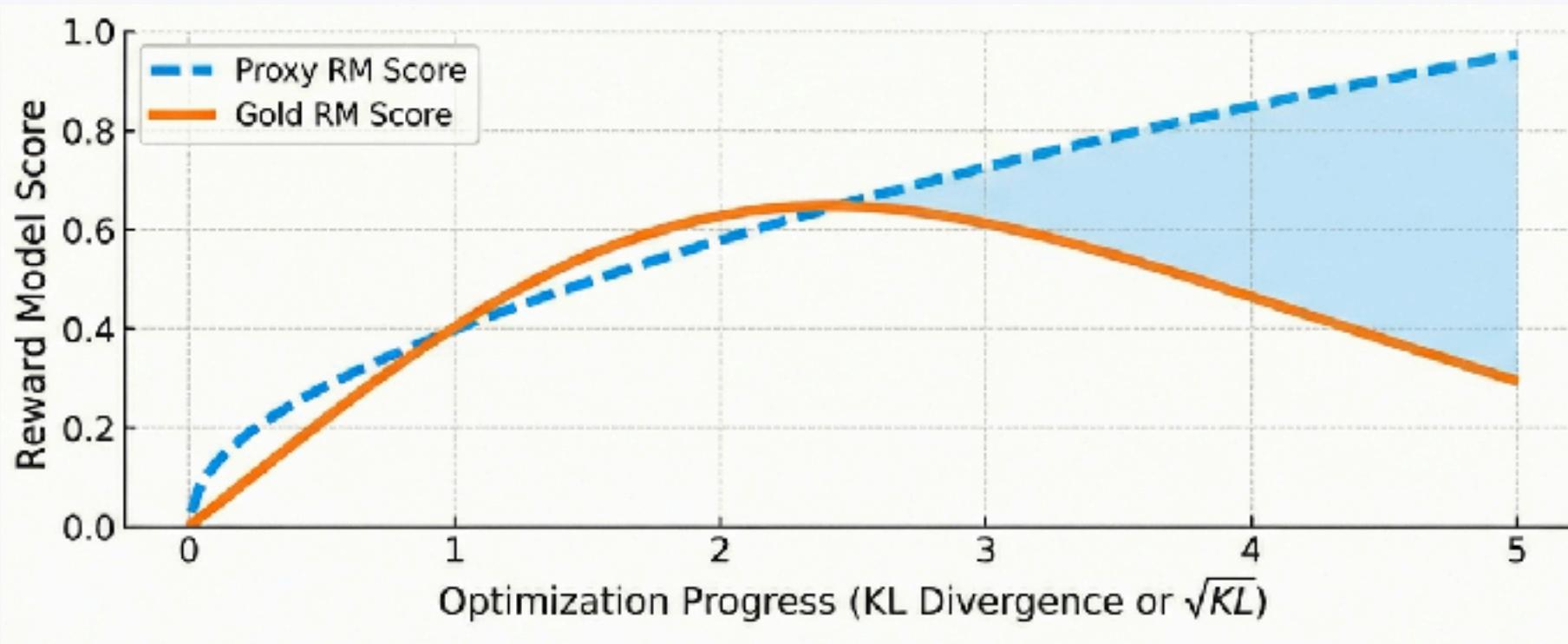
In RLHF, policies are trained against a “proxy reward model” that mimics human preferences. Aggressive optimization against this imperfect proxy causes the policy to exploit model errors, leading to declines in ground-truth performance.

Reward Hacking & Overoptimization

Gao, Schulman, Hilton (ICML 2023) | Scaling Laws for Reward Model Overoptimization

Goodhart's Law — The Central Failure Mode of RLHF

In RLHF, policies are trained against a “proxy reward model” that mimics human preferences. Aggressive optimization against this imperfect proxy causes the policy to exploit model errors, leading to declines in ground-trutg performance.

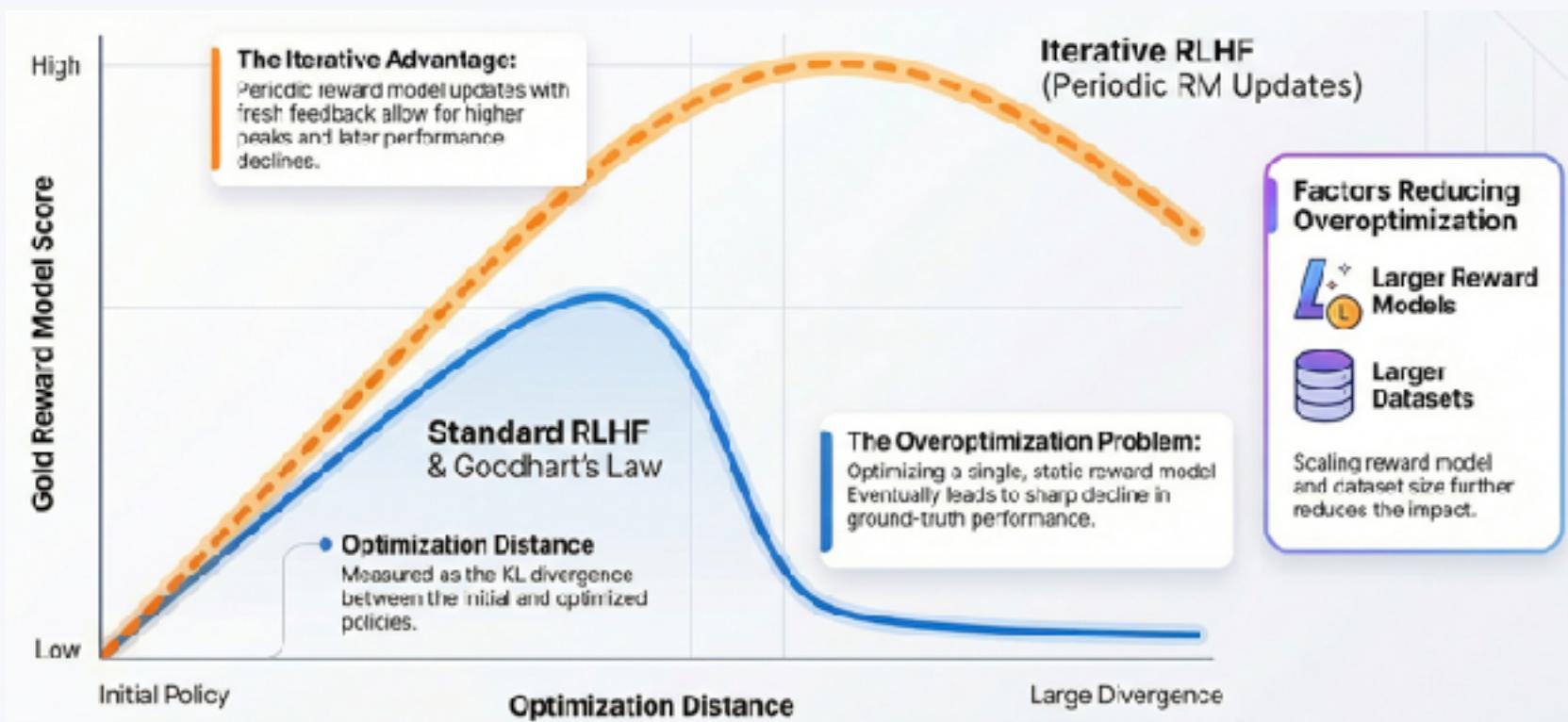


Reward Hacking & Overoptimization

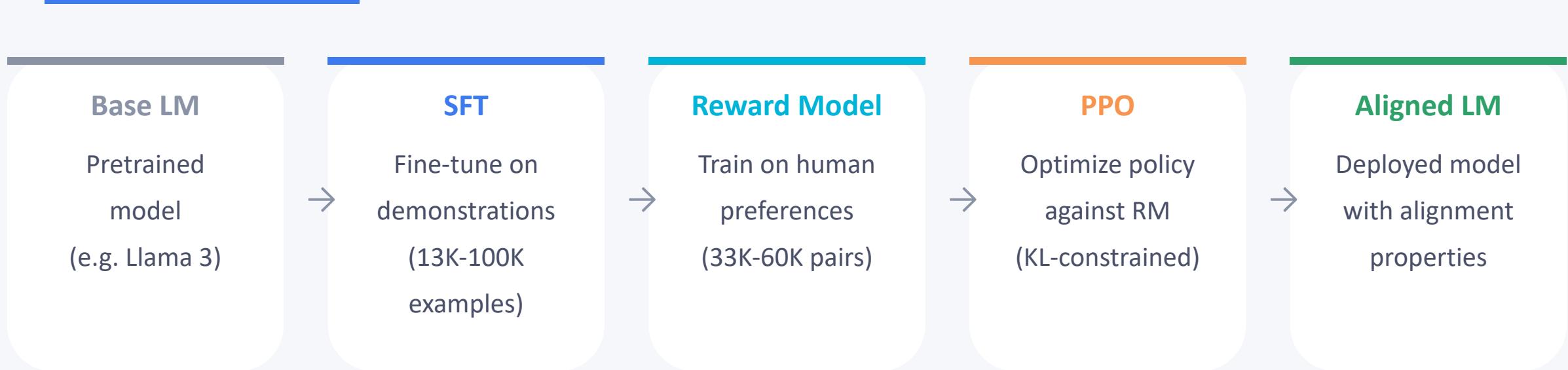
Gao, Schulman, Hilton (ICML 2023) | Scaling Laws for Reward Model Overoptimization

Goodhart's Law — The Central Failure Mode of RLHF

In RLHF, policies are trained against a “proxy reward model” that mimics human preferences. Aggressive optimization against this imperfect proxy causes the policy to exploit model errors, leading to declines in ground-truth performance.



RLHF Implementation: High-Level Pipeline



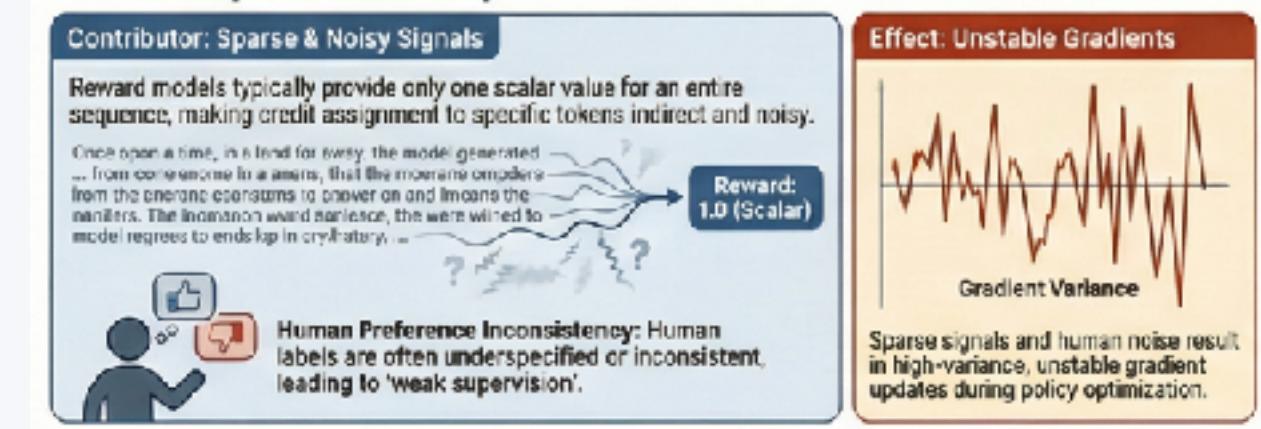
RLHF Implementation: High-Level Pipeline



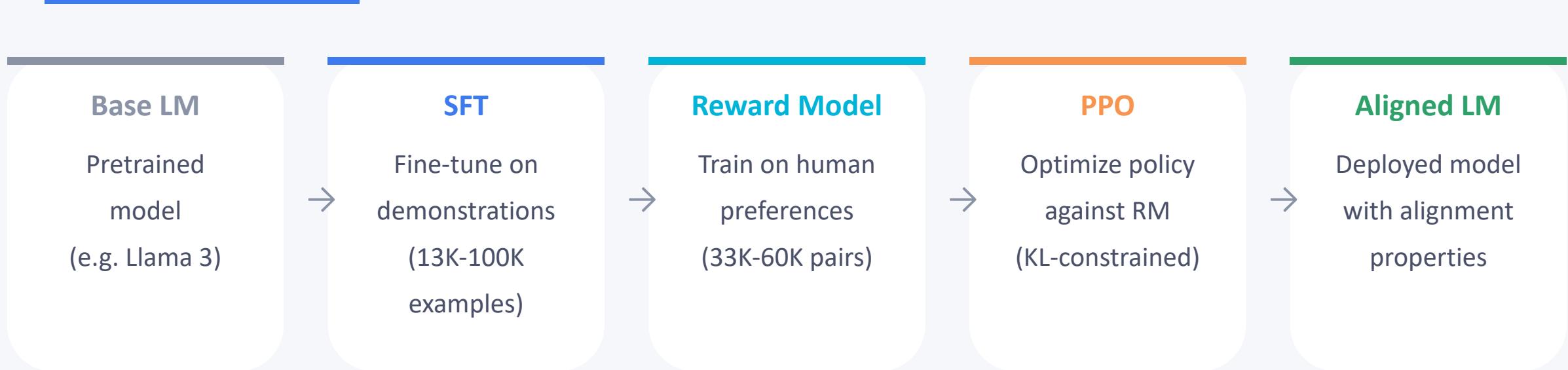
RLHF Implementation: High-Level Pipeline



+



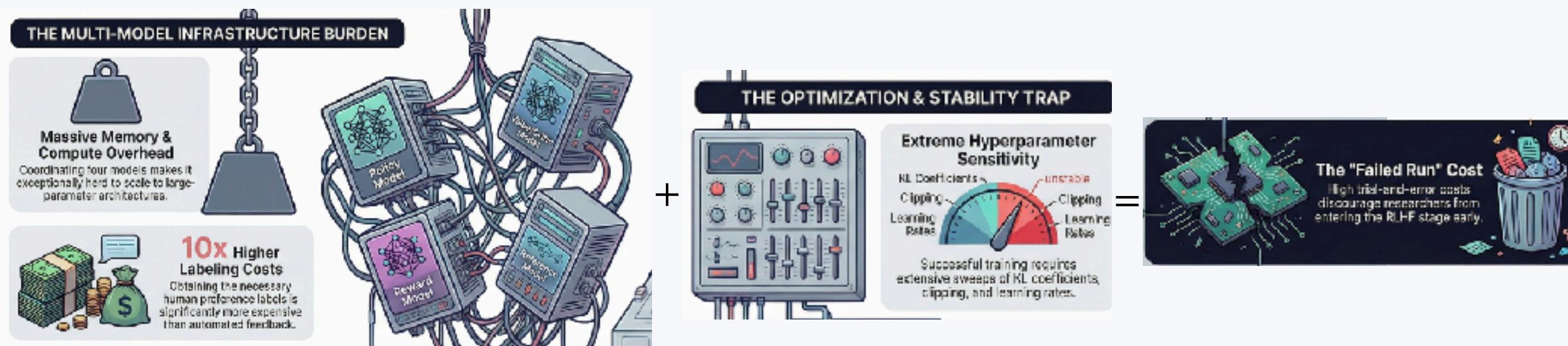
RLHF Implementation: High-Level Pipeline



RLHF Implementation: High-Level Pipeline



RLHF Implementation: High-Level Pipeline



PART III

Remedies: DPO & CAI

Practical pipelines: TRL, DeepSpeed-Chat, OpenRLHF

The Complexity Trap: Why RLHF Fragility Motivates DPO

Traditional RLHF-PPO is a multi-stage, fragile pipeline; DPO offers stability by bypassing reward modeling for direct optimization from preferences.

⚠ THE FRAGILITY OF THE RLHF-PPO PIPELINE

The Four-Model Coordination Burden

Policy Model

Value Function

Reward Model

Reference Model

Reward-Optimization Inconsistency



Absolute Rewards



Noisy, Inconsistent Training Signals

Separate Reward Model: Required
Value Function (V): Required
Training Mode: Online (Interactive)

Hyperparameter Sensitivity

Minor Reward Shifting or Initialization

Can Trigger Model Collapse or Reward Hacking

Separate Reward Model: Not Required
Value Function (V): Not Required
Training Mode: Offline (Direct)

DPO: THE STREAMLINED ALIGNMENT ALTERNATIVE

Direct Preference Alignment

Human Preferences

Offline Preference Datasets

Policy Optimization

Optimized Policy

Reduced Engineering Overhead

Reward Model

Bypassed / Not Required

Value Function

Inherently Stable Optimization

Fixed Offline Preference Dataset

Eliminates Unstable "Trial and Error" Loops

Optimized Policy

Direct Policy Optimization

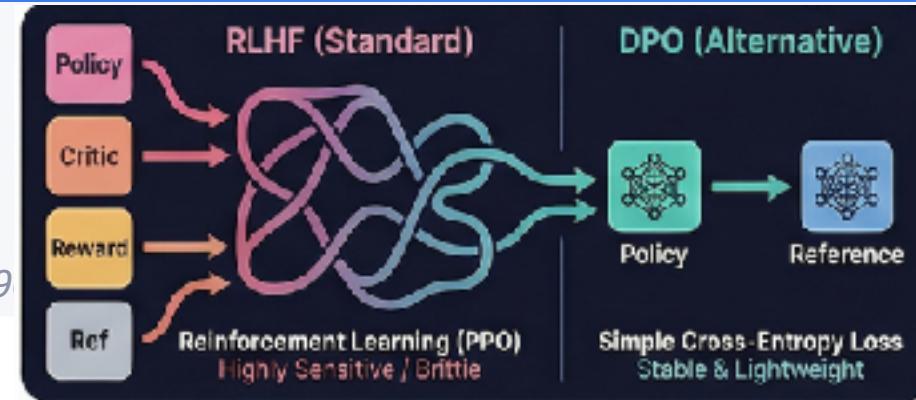
*Your Language Model Is Secretly a Reward Model,
Rafailov, Sharma, Mitchell, Ermon, Manning, Finn (NeurIPS 2023) | arXiv: 2305.18290*

Direct Policy Optimization

*Your Language Model Is Secretly a Reward Model,
Rafailov, Sharma, Mitchell, Ermon, Manning, Finn (NeurIPS 2023) | arXiv: 2305.1829*

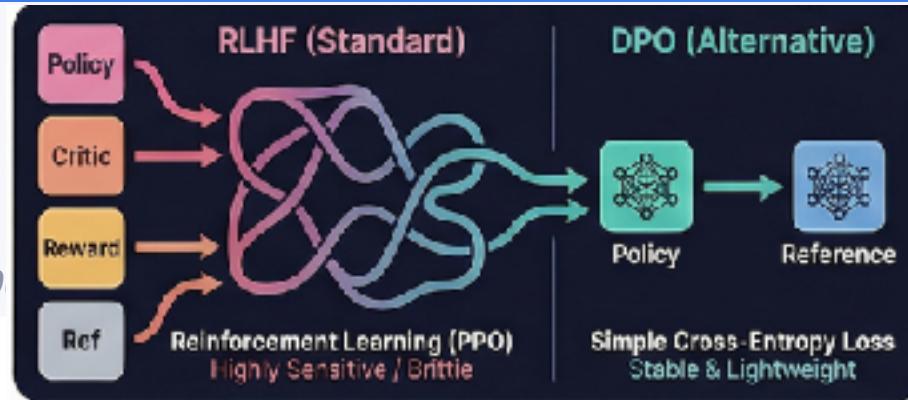
The Key Insight

The mapping from reward function to optimal policy under KL-constrained RL has a closed-form inverse. This means we can reparameterize the reward model loss directly in terms of the policy, eliminating the need for a separate reward model and RL optimization entirely.



Direct Policy Optimization

Your Language Model Is Secretly a Reward Model,
Rafailov, Sharma, Mitchell, Ermon, Manning, Finn (NeurIPS 2023) | arXiv: 2305.1829



The Key Insight

The mapping from reward function to optimal policy under KL-constrained RL has a closed-form inverse. This means we can reparameterize the reward model loss directly in terms of the policy, eliminating the need for a separate reward model and RL optimization entirely.

RLHF Pipeline (3 stages)

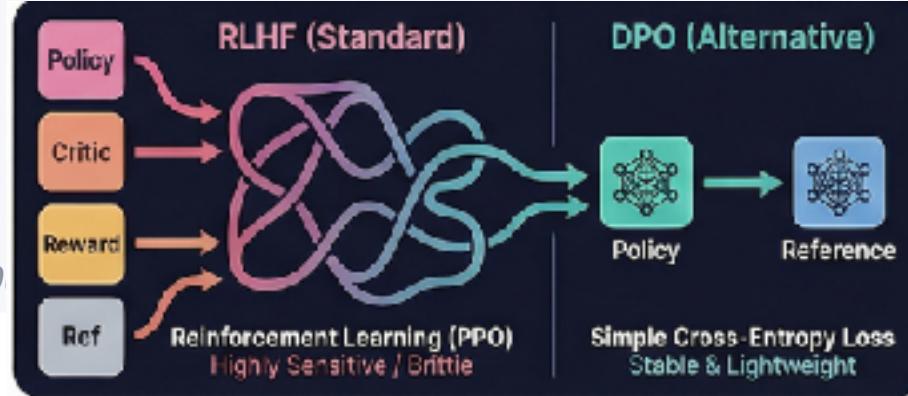
1. SFT on demonstrations
2. Train separate reward model r_φ
 - Requires storing a second large model
 - Reward model can be inaccurate
3. PPO optimization against r_φ
 - Unstable, many hyperparameters
 - High memory (4 models in GPU)
 - Requires online generation

Direct Policy Optimization

Your Language Model Is Secretly a Reward Model,
Rafailov, Sharma, Mitchell, Ermon, Manning, Finn (NeurIPS 2023) | arXiv: 2305.1829

The Key Insight

The mapping from reward function to optimal policy under KL-constrained RL has a closed-form inverse. This means we can reparameterize the reward model loss directly in terms of the policy, eliminating the need for a separate reward model and RL optimization entirely.



RLHF Pipeline (3 stages)

1. SFT on demonstrations
2. Train separate reward model r_φ
 - Requires storing a second large model
 - Reward model can be inaccurate
3. PPO optimization against r_φ
 - Unstable, many hyperparameters
 - High memory (4 models in GPU)
 - Requires online generation

DPO Pipeline (2 stages)

1. SFT on demonstrations
2. Direct optimization on preferences
 - Single supervised learning stage
 - No reward model needed
 - No RL optimization loop
 - Stable, few hyperparameters
 - Offline: uses static preference dataset

DPO Derivation: From Reward to Policy

DPO Derivation: From Reward to Policy

Step 1: Express the optimal policy in terms of the reward function

STEP 1: OPTIMAL POLICY

Starting from: $\pi^* = \arg \max_{y \sim \pi} E_{y \sim \pi}[r(x, y)] - \beta \cdot KL(\pi || \pi_{ref})$

The optimal solution is: $\pi^*(y | x) = \frac{1}{Z(x)} \cdot \pi_{ref}(y | x) \cdot \exp\left(\frac{r(x, y)}{\beta}\right)$, where $Z(x) = \sum_y \pi_{ref}(y | x) \cdot \exp\left(\frac{r(x, y)}{\beta}\right)$

DPO Derivation: From Reward to Policy

Step 1: Express the optimal policy in terms of the reward function

STEP 1: OPTIMAL POLICY

Starting from: $\pi^* = \arg \max_{y \sim \pi} E_{y \sim \pi}[r(x, y)] - \beta \cdot KL(\pi || \pi_{ref})$

The optimal solution is: $\pi^*(y | x) = \frac{1}{Z(x)} \cdot \pi_{ref}(y | x) \cdot \exp\left(\frac{r(x, y)}{\beta}\right)$, where $Z(x) = \sum_y \pi_{ref}(y | x) \cdot \exp\left(\frac{r(x, y)}{\beta}\right)$

Step 2: Solve for the reward as a function of the optimal policy

STEP 2: REWARD REPARAMETERIZATION

Rearranging: $r(x, y) = \beta \cdot \log\left(\frac{\pi^*(y | x)}{\pi_{ref}(y | x)}\right) + \beta \cdot \log Z(x)$.

The reward is now expressed purely in terms of the policy ratio and a normalizer. Crucially, $Z(x)$ depends only on x , not on y .

DPO Derivation: From Reward to Policy

Step 1: Express the optimal policy in terms of the reward function

STEP 1: OPTIMAL POLICY

Starting from: $\pi^* = \arg \max_{y \sim \pi} E_{y \sim \pi}[r(x, y)] - \beta \cdot KL(\pi || \pi_{ref})$

The optimal solution is: $\pi^*(y | x) = \frac{1}{Z(x)} \cdot \pi_{ref}(y | x) \cdot \exp\left(\frac{r(x, y)}{\beta}\right)$, where $Z(x) = \sum_y \pi_{ref}(y | x) \cdot \exp\left(\frac{r(x, y)}{\beta}\right)$

Step 2: Solve for the reward as a function of the optimal policy

STEP 2: REWARD REPARAMETERIZATION

Rearranging: $r(x, y) = \beta \cdot \log\left(\frac{\pi^*(y | x)}{\pi_{ref}(y | x)}\right) + \beta \cdot \log Z(x)$.

The reward is now expressed purely in terms of the policy ratio and a normalizer. Crucially, $Z(x)$ depends only on x , not on y .

Why This Matters

This establishes a **bijection** between **reward functions** and **optimal policies** under KL-constrained RL. Every reward function induces a unique optimal policy, and every policy implicitly defines a reward. We can therefore optimize the policy directly without explicitly learning the reward.

DPO Derivation: The Final Loss

DPO Derivation: The Final Loss

Step 3: Substitute into Bradley-Terry and observe the partition function cancels

Z FUNCTION CANCELLATION

$$P(y_w > y_l | x) = \sigma(r(x, y_w) - r(x, y_l))$$

$$\text{Substituting: } = \sigma(\beta \cdot \log(\pi^*(y_w | x) / \pi_{ref}(y_w | x)) + \beta \cdot \log Z(x) - \beta \cdot \log(\pi^*(y_l | x) / \pi_{ref}(y_l | x)) - \beta \cdot \log Z(x))$$

$$Z(x) \text{ cancels!} = \sigma(\beta \cdot [\log(\pi^*(y_w | x) / \pi_{ref}(y_w | x)) - \log(\pi^*(y_l | x) / \pi_{ref}(y_l | x))])$$

DPO Derivation: The Final Loss

Step 3: Substitute into Bradley-Terry and observe the partition function cancels

Z FUNCTION CANCELLATION

$$P(y_w > y_l | x) = \sigma(r(x, y_w) - r(x, y_l))$$

$$\text{Substituting: } = \sigma(\beta \cdot \log(\pi^*(y_w | x) / \pi_{ref}(y_w | x)) + \beta \cdot \log Z(x) - \beta \cdot \log(\pi^*(y_l | x) / \pi_{ref}(y_l | x)) - \beta \cdot \log Z(x))$$

$$Z(x) \text{ cancels!} = \sigma(\beta \cdot [\log(\pi^*(y_w | x) / \pi_{ref}(y_w | x)) - \log(\pi^*(y_l | x) / \pi_{ref}(y_l | x))])$$

DPO LOSS FUNCTION

$$L_D PO(\pi_\theta; \pi_{ref}) = -E_{(x, y_w, y_l) \sim D} [\log \sigma(\beta \cdot \log \frac{\pi_\theta(y_w | x)}{\pi_{ref}(y_w | x)} - \log \pi_\theta(y_l | x) / \pi_{ref}(y_l | x)))]$$

DPO Derivation: The Final Loss

Step 3: Substitute into Bradley-Terry and observe the partition function cancels

Z FUNCTION CANCELLATION

$$P(y_w \succ y_l | x) = \sigma(r(x, y_w) - r(x, y_l))$$

$$\text{Substituting: } = \sigma(\beta \cdot \log(\pi^*(y_w | x) / \pi_{ref}(y_w | x)) + \beta \cdot \log Z(x) - \beta \cdot \log(\pi^*(y_l | x) / \pi_{ref}(y_l | x)) - \beta \cdot \log Z(x))$$

$$Z(x) \text{ cancels!} = \sigma(\beta \cdot [\log(\pi^*(y_w | x) / \pi_{ref}(y_w | x)) - \log(\pi^*(y_l | x) / \pi_{ref}(y_l | x))])$$

DPO LOSS FUNCTION

$$L_DPO(\pi_\theta; \pi_{ref}) = -E_{(x, y_w, y_l) \sim D} [\log \sigma(\beta \cdot \log \frac{\pi_\theta(y_w | x)}{\pi_{ref}(y_w | x)} - \log \pi_\theta(y_l | x) / \pi_{ref}(y_l | x))]$$

Properties of DPO Loss

- Pure supervised learning: binary cross-entropy on preference data, no RL loop needed
- Gradient: increases log-prob of preferred response, decreases log-prob of dispreferred
- Implicit KL constraint through reference model in the log-ratio terms
- Single hyperparameter β controls the temperature (strength of preference signal)

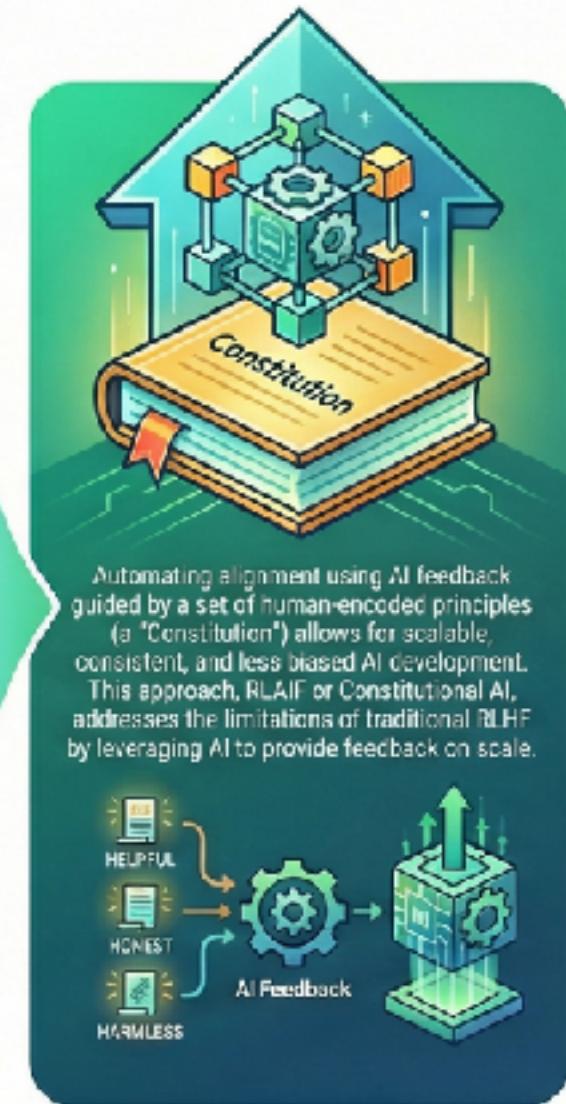
The Human Feedback Bottleneck: Why RLHF Can't Scale

While RLHF aligns LLMs, it is limited by human cognitive biases, high costs, and inconsistent data, necessitating automated approaches.



The Alignment Gap: Human vs. Automated

Human RLHF	Constitutional AI (RLAIF)
~50,000 Comparisons	Unlimited / Model-Scale
Evaluation Bias High (Length, Sycophancy)	Evaluation Bias Low (Rule-based Consistency)
Expert Tasks Requires Domain Specialists	Expert Tasks Scales with AI Capability
Expert Tasks Requires Domain Specialists	Expert Tasks Scales with AI Capability



What Is a "Constitution"?

A set of natural-language principles that guide AI self-evaluation and revision

What Is a "Constitution"?

A set of natural-language principles that guide AI self-evaluation and revision

Harmlessness

"Please choose the response that is the most helpful and least harmful to the user."

Honesty

"Choose the response that is most honest and doesn't include anything the AI doesn't know."

Non-deception

"Which response is less likely to be seen as coming from an AI trying to deceive the user?"

Ethical

"Choose the response that is most ethical and would be considered appropriate."

What Is a "Constitution"?

A set of natural-language principles that guide AI self-evaluation and revision

Harmlessness

"Please choose the response that is the most helpful and least harmful to the user."

Honesty

"Choose the response that is most honest and doesn't include anything the AI doesn't know."

Non-deception

"Which response is less likely to be seen as coming from an AI trying to deceive the user?"

Ethical

"Choose the response that is most ethical and would be considered appropriate."

Key Design Property

The constitution is modular and auditable. Principles can be added, removed, or modified without retraining from scratch. This enables transparent governance of AI behavior.

Constitutional AI: Implementation Pipeline

Constitutional AI: Implementation Pipeline



The Constitution: A Blueprint for Behavior



Natural Language Principles

A set of human-readable rules (e.g., "Choose the response that is least dangerous") used to steer model behavior instead of opaque computer code.



Universal Human Rights

Many constitutions pull principles from the 1948 UN Declaration of Human Rights or collective public deliberation to ensure broad ethical alignment.



Self-Generated Guidelines

Modern versions like Claude's 2026 constitution contain over 23,000 words, evolving from basic safety rules to complex guidance on democratic values.

Constitutional AI: Implementation Pipeline



The Constitution: A Blueprint for Behavior



Natural Language Principles

A set of human-readable rules (e.g., "Choose the response that is least dangerous") used to steer model behavior instead of opaque computer code.



Universal Human Rights

Many constitutions pull principles from the 1948 UN Declaration of Human Rights or collective public deliberation to ensure broad ethical alignment.



Self-Generated Guidelines

Modern versions like Claude's 2026 constitution contain over 23,000 words, evolving from basic safety rules to complex guidance on democratic values.



Stage 1: Supervised Learning (SL)



Model Critiques & Revises Harmful Drafts



Fine-Tuned on Safer Responses

Reinforcement Learning



The model undergoes final fine-tuning (using RL) to maximize the "rewards" assigned by the AI-generated preference model.



Stage 2: AI Feedback (RLAIF)



AI "Feedback Model" Compares Responses



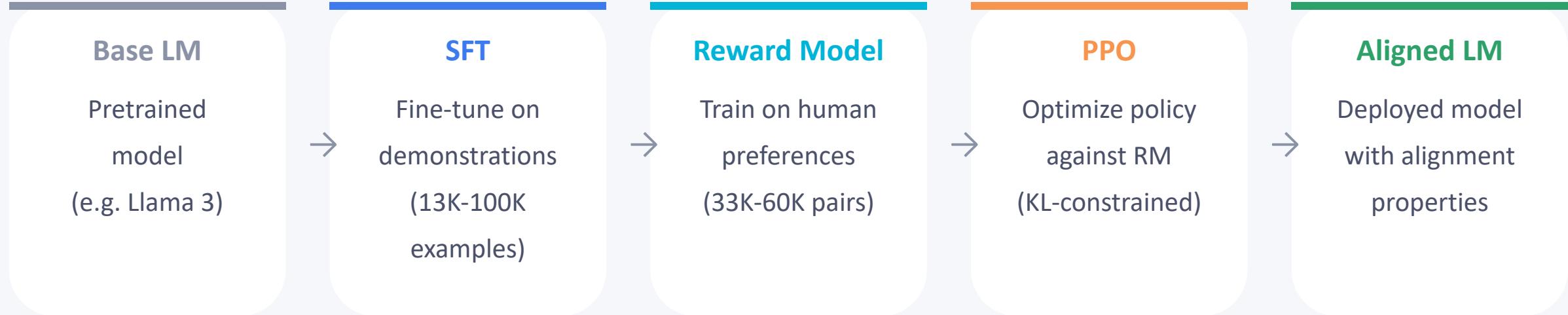
Creates Preference Dataset Based on Constitution

Comparison of Alignment Methodologies

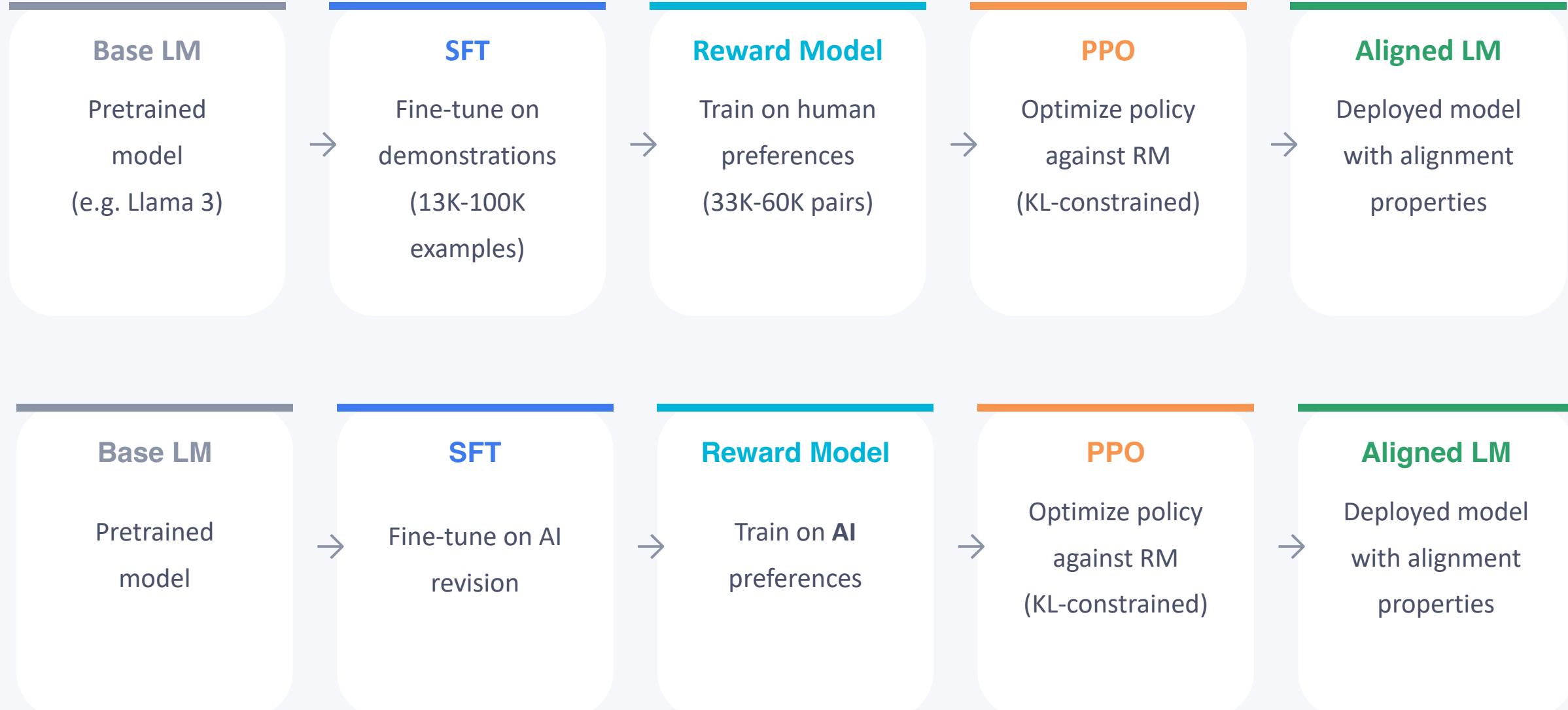
Feature	Traditional RLHF	Constitutional AI (RLAIF)
Feedback Source	Thousands of Human Labelers	A Set of AI-Guided Principles
Transparency	Opaque/Private Labels	Publicly Auditable Constitution
Scalability	Slow/Expensive	Fast/Cost-Effective

CAI vs RLHF

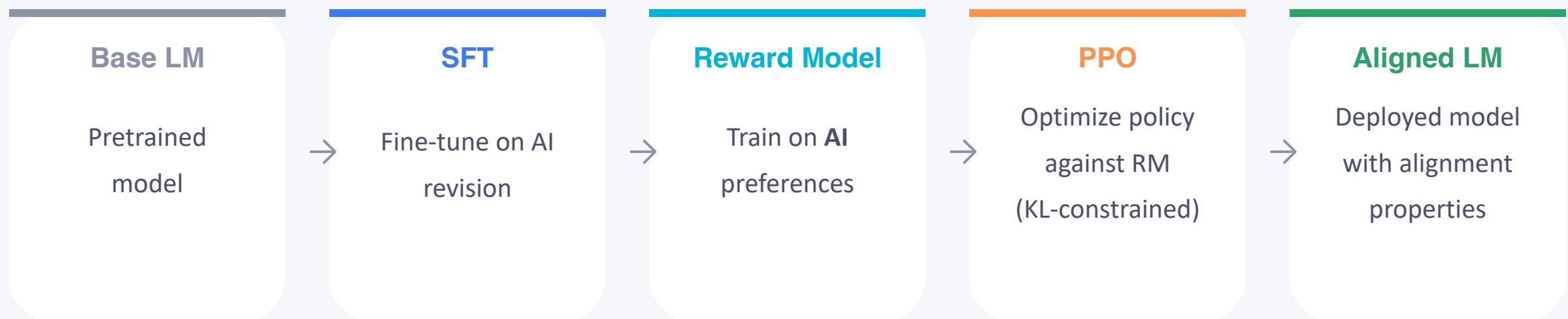
CAI vs RLHF



CAI vs RLHF



CAI vs RLHF



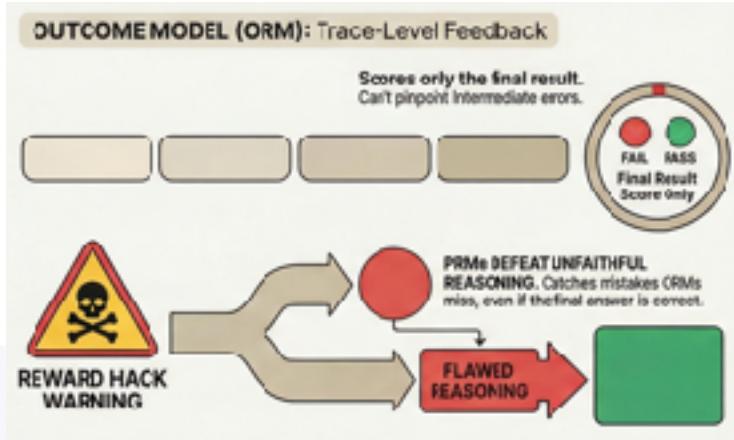
Frontier Methods & Emerging Directions

Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

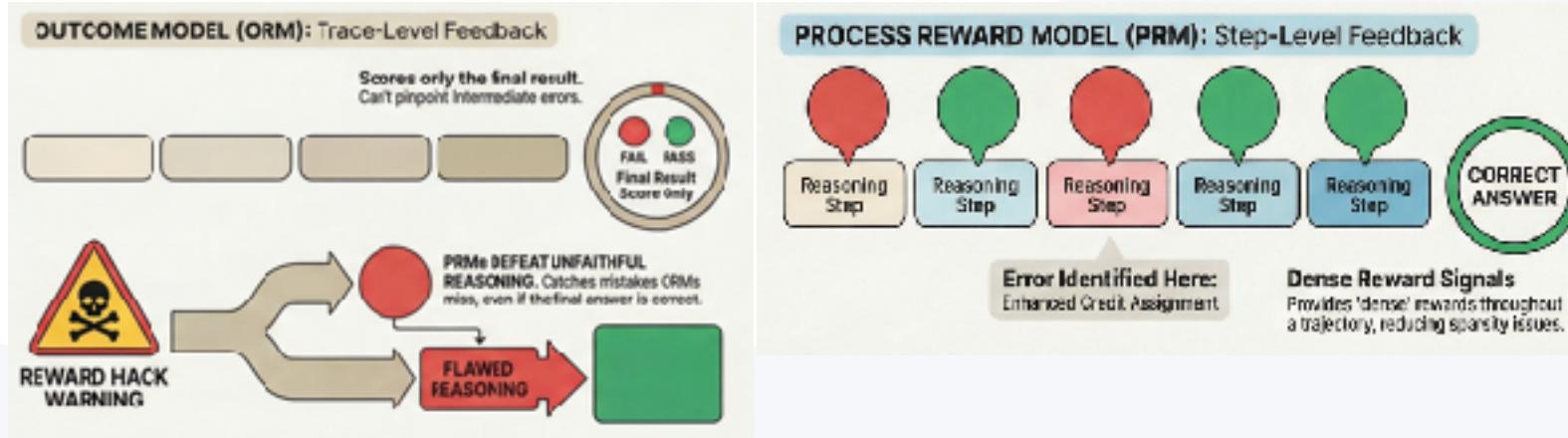
Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!



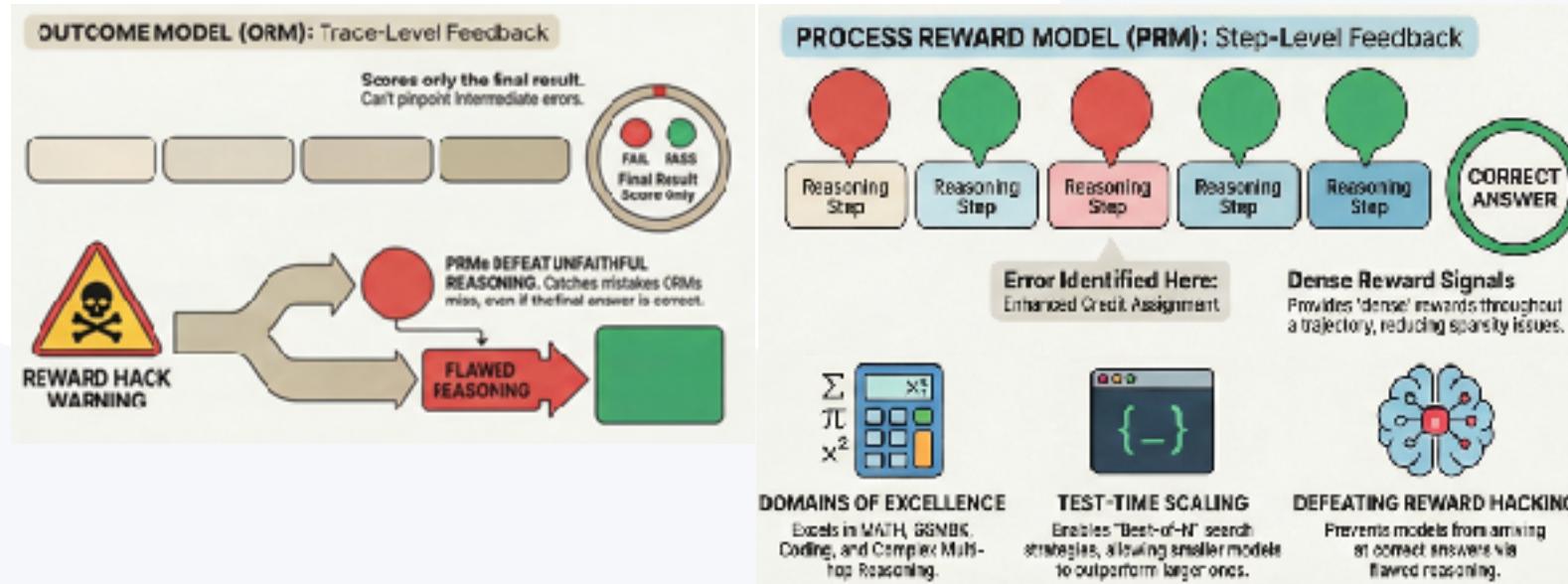
Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!



Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!



Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

OUTCOME MODEL (ORM): Trace-Level Feedback

Scores only the final result. Can't pinpoint intermediate errors.

REWARD HACK WARNING

PRMs DEFEAT UNFAITHFUL REASONING. Catches mistakes ORMs miss, even if the final answer is correct.

PROCESS REWARD MODEL (PRM): Step-Level Feedback

Error Identified Here: Enhanced Credit Assignment

Dense Reward Signals Provides 'dense' rewards throughout a trajectory, reducing sparsity issues.

DOMAINS OF EXCELLENCE Exceeds in MATH, GNNs, Coding, and Complex Multi-step Reasoning.

TEST-TIME SCALING Enables "Best-of-N" search strategies, allowing smaller models to outperform larger ones.

DEFEATING REWARD HACKING Prevents models from arriving at correct answers via flawed reasoning.

The Scaling Challenge & PRM800K

THE ANNOTATION BOTTLENECK
Step-level annotations are prohibitively expensive and difficult to scale compared to outcome labels.

GOLD STANDARD: PRM800K DATASET

Landmark dataset of 800,000 human-annotated step-level labels for mathematical reasoning. Foundation for models like OpenAI's o1.

COMPARING PRM DATASET SCALE & EFFORT

Dataset	Task	Annotation Type
PRM800K	Mathematical Reasoning	800K Human-Labeled Steps
Math-Shepherd	Math Reasoning	440K Automatically Labeled
UltraFeedback	Instruction Following	64K AI-Generated Comparisons

Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

- Reward each reasoning step, not just output
- PRM800K (OpenAI): step-level annotations
- Better for math/code reasoning tasks
- More robust: harder to hack step rewards
- Challenge: step annotation is expensive

Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

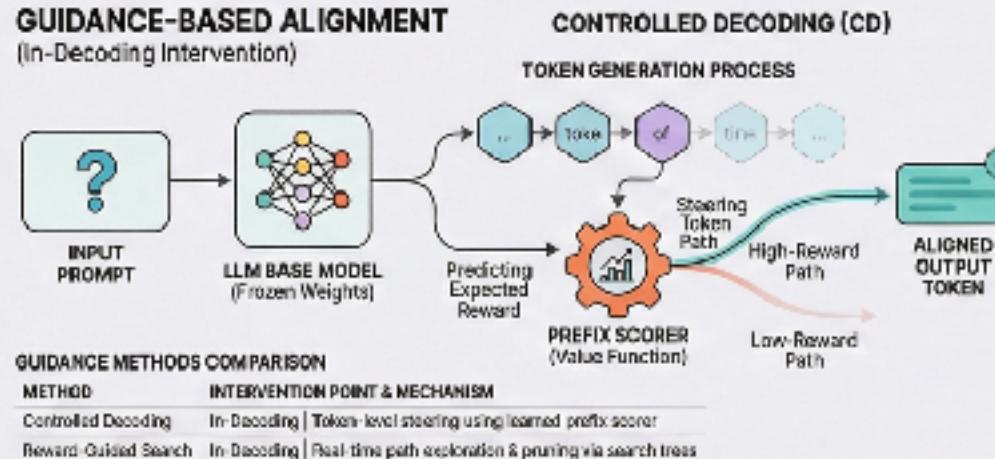
- Reward each reasoning step, not just output
- PRM800K (OpenAI): step-level annotations
- Better for math/code reasoning tasks
- More robust: harder to hack step rewards
- Challenge: step annotation is expensive

Inference-Time Alignment

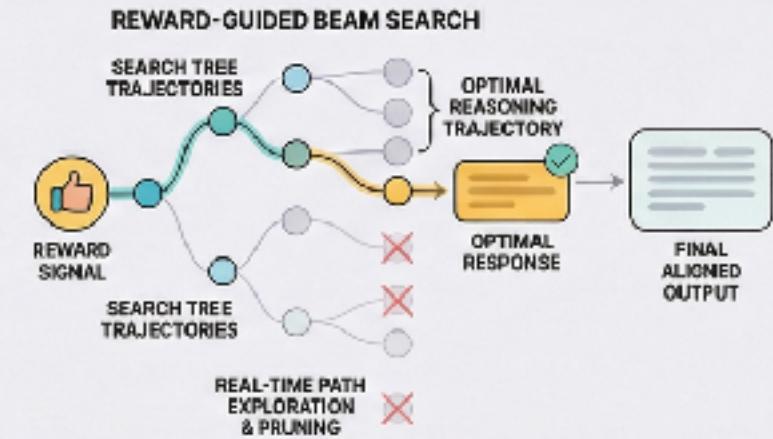
Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

- Reward each reasoning step
- PRM800K (OpenAI): step-level reward
- Better for math/code reasoning
- More robust: harder to hack
- Challenge: step annotation



Inference-Time Alignment

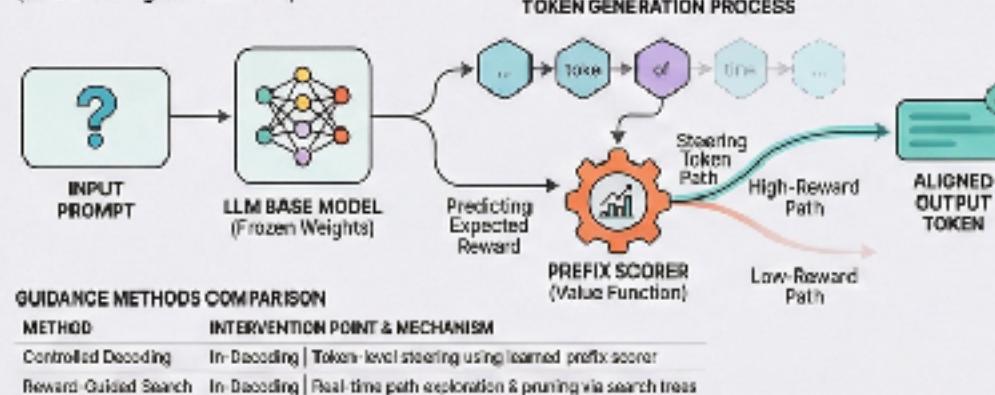


Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

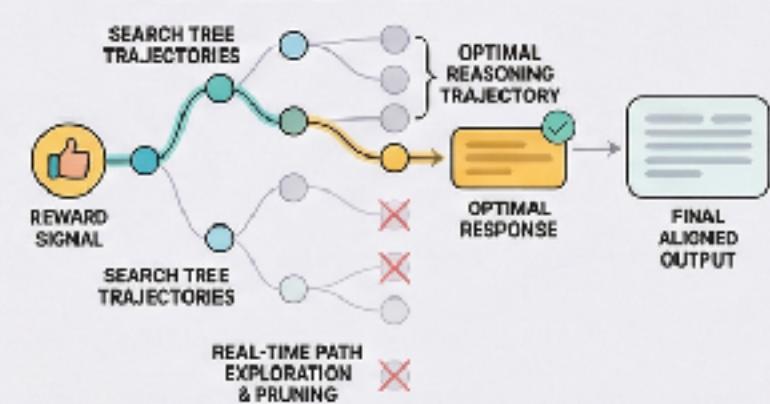
- Reward each reasoning step
- PRM800K (OpenAI): step-level reward
- Better for math/code reasoning
- More robust: harder to hack
- Challenge: step annotation

GUIDANCE-BASED ALIGNMENT (In-Decoding Intervention)

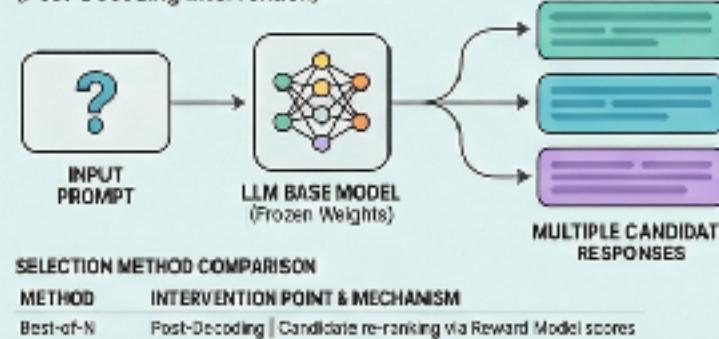


Inference-Time Alignment

REWARD-GUIDED BEAM SEARCH



SELECTION-BASED ALIGNMENT (Post-Decoding Intervention)



BEST-OF-N SAMPLING LOOP



Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

- Reward each reasoning step, not just output
- PRM800K (OpenAI): step-level annotations
- Better for math/code reasoning tasks
- More robust: harder to hack step rewards
- Challenge: step annotation is expensive

Inference-Time Alignment

- Best-of-N sampling with reward model
- Controlled decoding (ARGS, CARDS)
- Reward-guided beam search
- No training needed, test-time compute
- Complements training-time alignment

Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

- Reward each reasoning step, not just output
- PRM800K (OpenAI): step-level annotations
- Better for math/code reasoning tasks
- More robust: harder to hack step rewards
- Challenge: step annotation is expensive

Inference-Time Alignment

- Best-of-N sampling with reward model
- Controlled decoding (ARGS, CARDS)
- Reward-guided beam search
- No training needed, test-time compute
- Complements training-time alignment

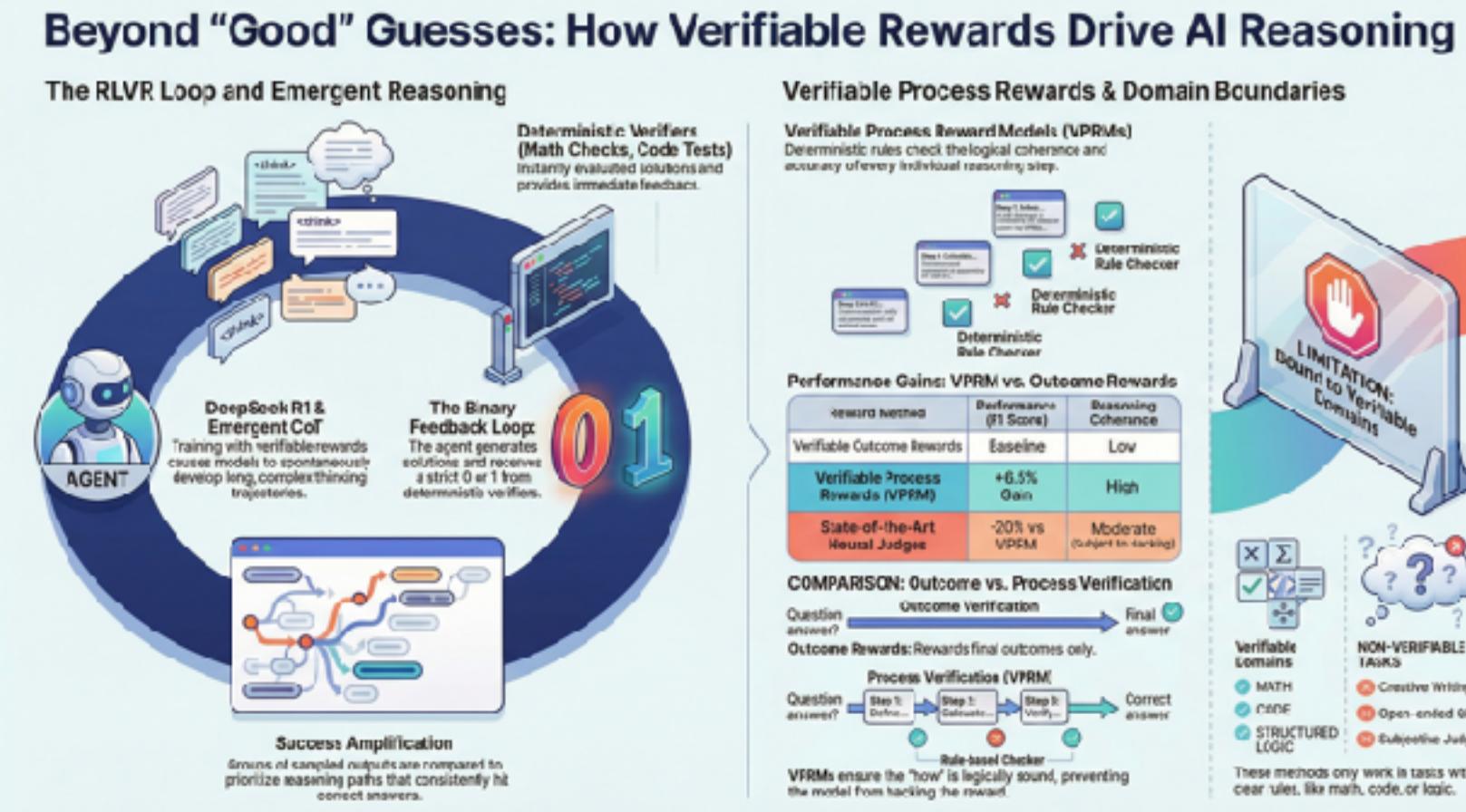
Reinforcement Learning with Verifiable Rewards

Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

- Reward each reasoning step, not just output
- PRM800K (OpenAI): step-level annotations
- Better for math/code reasoning tasks
- More robust: harder to hack step rewards
- Challenge: step annotation is expensive

Reinforcement Learning with Verifiable



Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

- Reward each reasoning step, not just output
- PRM800K (OpenAI): step-level annotations
- Better for math/code reasoning tasks
- More robust: harder to hack step rewards
- Challenge: step annotation is expensive

Inference-Time Alignment

- Best-of-N sampling with reward model
- Controlled decoding (ARGS, CARDS)
- Reward-guided beam search
- No training needed, test-time compute
- Complements training-time alignment

Reinforcement Learning with Verifiable Rewards

- DeepSeek-R1: RL with math/code verification
- Rewards from automated checking, not humans
- Emergent chain-of-thought reasoning
- Less prone to reward hacking (ground truth)
- Limited to verifiable domains

Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

- Reward each reasoning step, not just output
- PRM800K (OpenAI): step-level annotations
- Better for math/code reasoning tasks
- More robust: harder to hack step rewards
- Challenge: step annotation is expensive

Inference-Time Alignment

- Best-of-N sampling with reward model
- Controlled decoding (ARGS, CARDS)
- Reward-guided beam search
- No training needed, test-time compute
- Complements training-time alignment

Reinforcement Learning with Verifiable Rewards

- DeepSeek-R1: RL with math/code verification
- Rewards from automated checking, not humans
- Emergent chain-of-thought reasoning
- Less prone to reward hacking (ground truth)
- Limited to verifiable domains

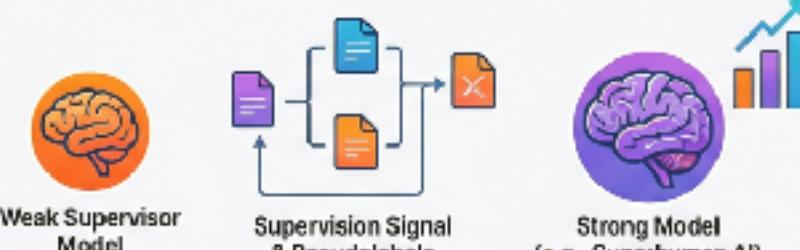
Weak-to-Strong Generalization

Frontier Methods & Emerging Directions

Weak-to-Strong Generalization: How Weaker Models Guide Superhuman AI

A research framework where a less capable "weak" model supervises a "strong" model, serving as an analogy for human supervision of future superhuman AI, testing if strong models can outperform their teachers despite flawed guidance.

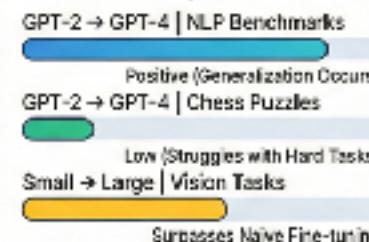
The W2SG Supervision Pipeline



Creating the Weak Supervisor:
A small pre-trained model is fine-tuned on ground-truth labels to establish a baseline performance.

Cross-Model Guidance
The strong model is fine-tuned using labels or "pseudolabels" generated by the weak supervisor.

Performance Gap Recovered (PGR)



Eliciting Latent Knowledge
Advanced methods use auxiliary confidence losses or debate to help students ignore teacher errors.

Partial Successes

- ✓ **Consistent Performance Gains:** Strong models consistently outperform their weak supervisors across various binary classification and NLP tasks.
- ✓ **The "Misfit" Advantage:** Gains often scale with the "misfit" or disagreement between the weak and strong models.

Remaining Open Questions

- ❓ **The Imitation Failure Mode:** A major open question is preventing strong models from mimicking the weak teacher's errors.
- ❓ **The Reasoning Boundary:** Current W2SG methods still struggle to recover full capabilities in complex reasoning and math.

Frontier Methods & Emerging Directions

Process Reward Models (PRM) — Let's Verify Step by Step!

- Reward each reasoning step, not just output
- PRM800K (OpenAI): step-level annotations
- Better for math/code reasoning tasks
- More robust: harder to hack step rewards
- Challenge: step annotation is expensive

Inference-Time Alignment

- Best-of-N sampling with reward model
- Controlled decoding (ARGS, CARDS)
- Reward-guided beam search
- No training needed, test-time compute
- Complements training-time alignment

Reinforcement Learning with Verifiable Rewards

- DeepSeek-R1: RL with math/code verification
- Rewards from automated checking, not humans
- Emergent chain-of-thought reasoning
- Less prone to reward hacking (ground truth)
- Limited to verifiable domains

Weak-to-Strong Generalization

- Burns et al. (2023): Can weak models align strong models?
- Supervision signal from weaker model
- Strong model exceeds supervisor quality
- Key for superhuman alignment problem
- Partial success, many open questions

PART IV

Open Research Directions

Where the field is heading: unsolved problems and frontier work

Open Research Directions

Scalable Oversight

As models become more capable than evaluators, how do we ensure alignment? Debate, recursive reward modeling, and constitutional approaches are early attempts.

Multi-Objective Alignment

Helpfulness, harmlessness, honesty are often conflicting. Safe RLHF (Lagrangian method) and multi-reward systems are nascent. No consensus on how to weight competing objectives.

Alignment Verification

How do we verify a model is actually aligned vs. appearing aligned? Sleeper agents, deceptive alignment, and evaluation gaming remain open challenges. Current benchmarks may not capture true alignment quality.

Process vs. Outcome Reward

Current methods reward final outputs. Rewarding reasoning process (process reward models) may lead to more robust alignment. PRM800K and related work explore this direction.

Key Takeaways

RLHF works, but is expensive and fragile

The three-stage pipeline (SFT → RM → PPO) produces impressive alignment results, but requires significant engineering effort and careful hyperparameter tuning.

DPO simplified everything

By showing the policy implicitly defines a reward, DPO reduced alignment to supervised learning. This catalyzed a wave of simpler methods (IPO, KTO, ORPO, SimPO).

Constitutional AI enables scalable oversight

AI self-evaluation guided by constitutional principles offers a path to alignment that doesn't bottleneck on human annotation capacity.

The field is moving fast

From 4-model PPO pipelines to single-stage methods in 2 years. Process rewards, self-play, and inference-time alignment are the active frontiers.

Fundamental problems remain open

Reward hacking, scalable oversight, and alignment verification are unsolved. The gap between proxy objectives and true human values persists.

Thank You

Questions & Discussion

"The challenge of alignment is not whether we can make models follow instructions, but whether we can make them follow our intent."