Post-Training of Large Language Models

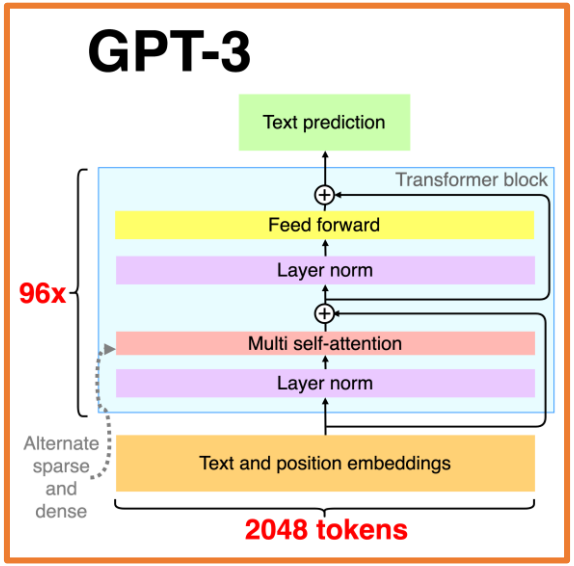
Georgia Tech CS 7643 Deep Learning

Chan Young Park

Postdoctoral Researcher, Microsoft Research

Incoming Assistant Professor, UT Austin iSchool

April 15, 2026



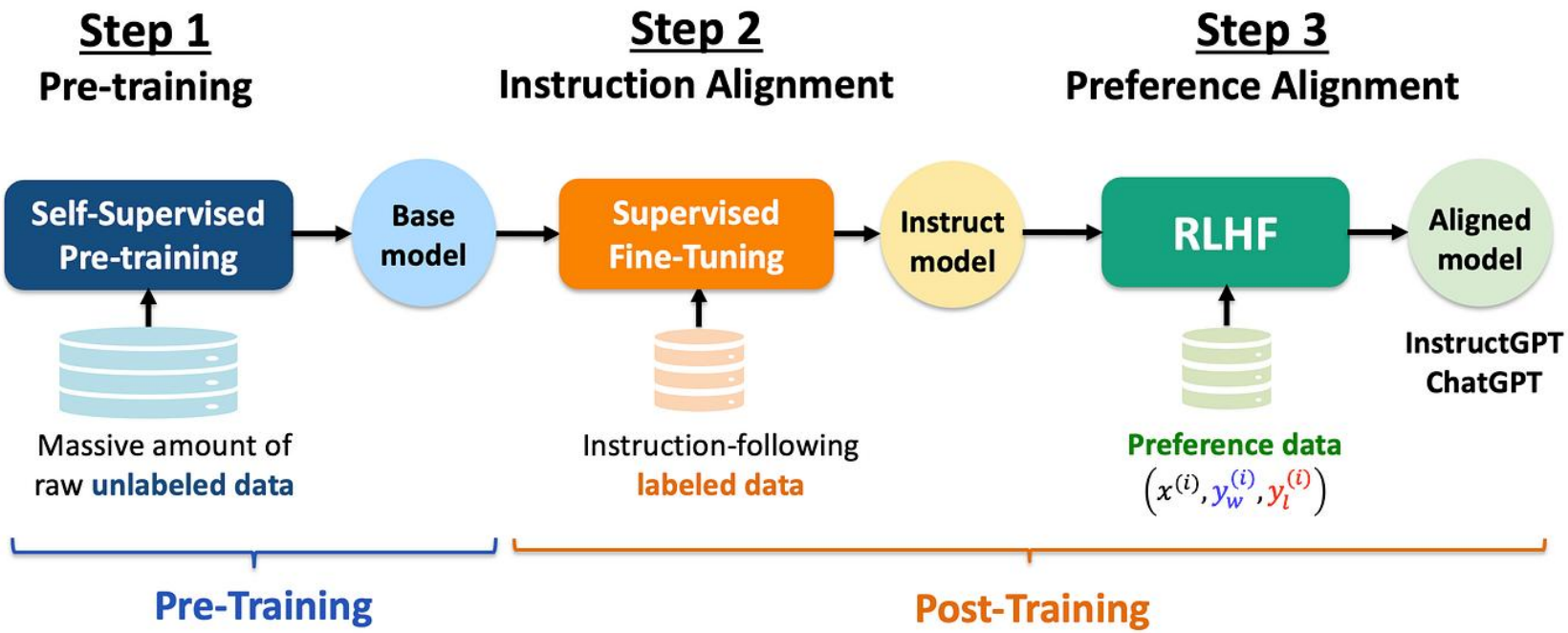
(2020)

Q. What happened?

A. Post-training!

ChatGPT

(2022)



The Journey of GPT: Evolution from GPT-1 to GPT-4o | by LM Po | Medium

Today's Agenda

1. Pre-training vs Post-training
2. Supervised Fine-Tuning (SFT)
3. RLHF: Reward Model + PPO
4. DPO (Direct Preference Optimization)
5. RLAIF & RLVR & GRPO
6. Open Problems

Part 1

Pre-training vs Post-training

Why isn't pre-training enough?

Pre-training: What It Gives and What It Misses

✓ What pre-training gives

- Broad world knowledge
- Language understanding & generation
- In-context learning ability
- Emergent reasoning capabilities

✗ What pre-training misses

- Following instructions reliably
- Consistent helpfulness
- Safety / refusing harmful content
- Factual grounding / avoiding hallucination

What Post-training Is Trying to Solve

Instruction Following

Understand user intent and respond in the right format and content *consistently*.

Helpfulness

Generate genuinely useful responses: accurate, complete, and relevant to the task.

Safety & Harmlessness

Refuse harmful requests, avoid misinformation, and behave ethically.

The Shift in Data and Training Signal

Pre-training

- Web-scale raw text (trillions of tokens)
- Ground truth = the actual next token
- Objective and deterministic signal
- Quantity-driven

Post-training

- Carefully curated data (thousands to millions)
- Ground truth = human judgment
- Noisy and subjective signal
- Quality-driven

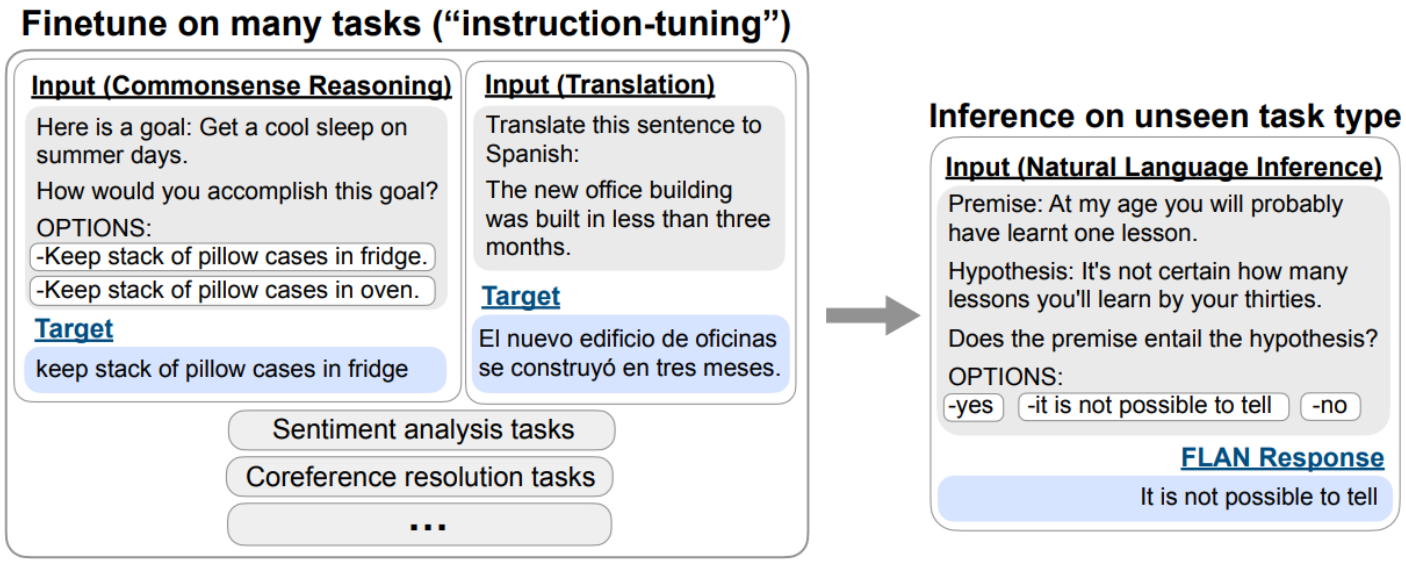
Part 2

Supervised Fine-Tuning (SFT)

Imitation learning for instruction following

"Finetuned Language Models are Zero-Shot Learners", Wei et al., 2022

SFT: Core Idea



Fine-tune a pre-trained LLM on instructions.

The objective is identical to pre-training.

Only the data changes: (instruction, response) pairs

$$\mathcal{L}_{\text{SFT}} = -\sum_{t} \log p_\theta(y_t | x, y_{<t})$$

SFT Data: Quality Over Quantity

InstructGPT: found ~13K examples sufficient

Comparison: Alpaca (Taori et al., 2023)

52K self-instruct examples (GPT-3.5-generated) → fine-tuned LLaMA-7B → GPT-3.5-level performance on many tasks. More data can help, but adding volume without quality control is inefficient.

LIMA (Zhou et al., 2023): The Superficial Alignment Hypothesis

Fine-tuning a 65B LLaMA on just 1,000 ***carefully selected demonstrations*** outperformed many instruction-tuned models trained on far more data. The hypothesis: models acquire most knowledge and capabilities during pre-training. SFT doesn't teach new knowledge. It teaches the model which format to use when expressing that knowledge.

Takeaways

- A small set of high-quality demonstrations beats a large set of mediocre ones
- The LIMA hypothesis is still debated. Complex tasks likely need more diverse coverage

SFT's Fundamental Limit

SFT is fundamentally *imitation learning*. The model learns to mimic responses in the demonstrations.

Distribution Shift

Real user prompts can fall outside the demonstration distribution. The model has no principled way to handle inputs it hasn't seen.

Ceiling Effect

The model cannot learn to generate responses better than its demonstrations. Demonstration quality is a hard ceiling on performance.

Part 3

RLHF

Reinforcement Learning from Human Feedback

Why RL? The Core Intuition

SFT "Imitate the demonstrations" → ceiling = demonstration quality. No mechanism to go beyond.

RLHF "Generate responses and have humans judge them" → exploration beyond the demonstration distribution!

The core challenge

The LLM action space (all possible token sequences) is astronomically large

Getting human evaluation for every response is impossible

→ We need a **Reward Model** that can proxy human judgment

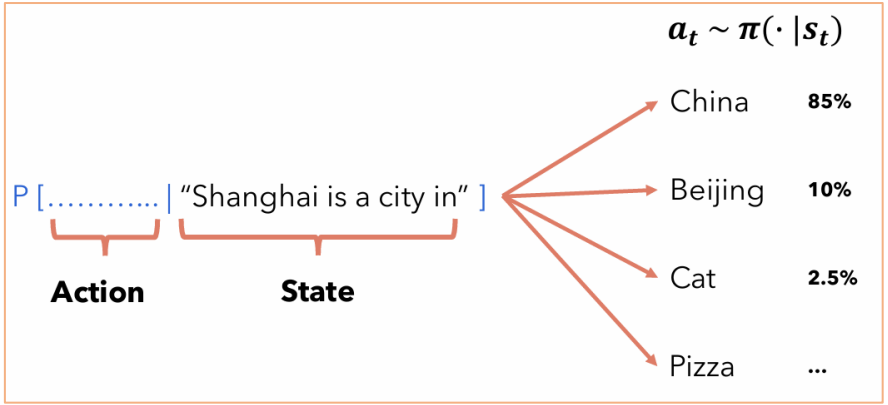
The RL Setup

Agent: the language model itself

State: the prompt (input tokens)

Action: which token is selected as the next token

Reward model: the language model should be rewarded for generating “good responses”



Policy: In the case of LLMs, the policy is the language model itself! Because it models the probability of the action space given the current state of the agent $a_t \sim \pi(\cdot \mid s_t)$

Reward Model

Reward model: the language model should be rewarded for generating “good responses”

Question (prompt)	Answer (LLM-generated)	Reward (0.0~1.0)
Where is Shanghai?	Shanghai is a city in China	?
What is 2+2?	4	?
Explain gravity like I’m 5	Gravity is what pulls things toward each other. It’s why you stay on the ground and planets orbit the sun.	?

Reward Model by Comparison

Reward model: the language model should be rewarded for generating “good responses”

Question (prompt)	Answer 1 (LLM-generated)	Answer 2 (LLM-generated)	Preferred
Where is Shanghai?	Shanghai is a city in China	Shanghai does not exist	1
What is 2+2?	4	2+2 is a very complicated math problem…	1
Explain gravity like I’m 5	Gravity is a famous restaurant	Gravity is what pulls things toward each other. It’s why you stay on the ground and planets orbit the sun.	2

Goal: using a dataset like this, we want to train a model that assigns a score to a given answer

Step 2: Learning a Reward Model

Bradley-Terry Preference Model

Given prompt x and two responses y_w ≻ y_l (human prefers y_w),
probability of preferring one over another is a function of the difference in scores passed through a sigmoid:

$$P(y\_w \succ y\_l \mid x) = \sigma(r^*(x, y\_w) - r^*(x, y\_l))$$

r(x, y): latent "true" quality, unobservable, estimated from human preference data*

Reward Model Loss

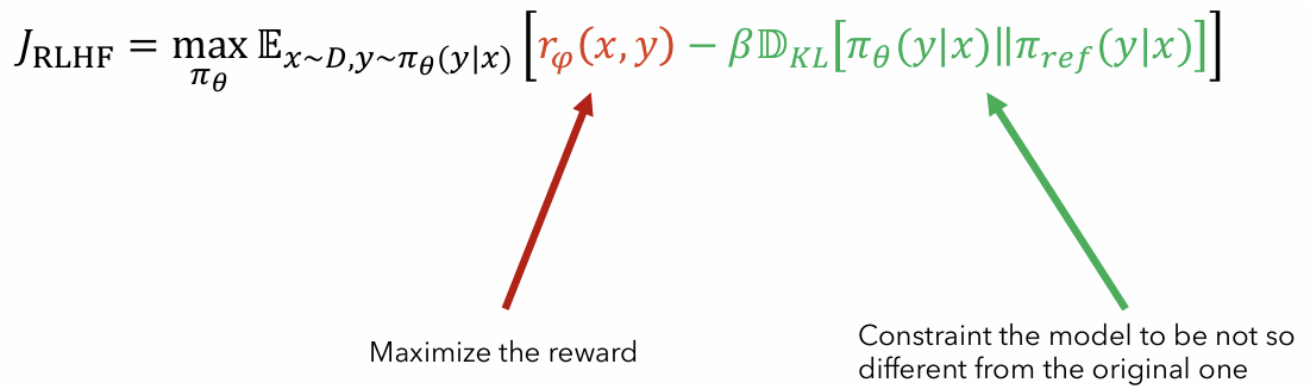
$$\mathcal{L}_{\mathrm{RM}}(\varphi) = -\mathbb{E}\_{(x,y\_w,y\_l)\sim D}\left[\log \sigma(r\_\varphi(x,y\_w) - r\_\varphi(x,y\_l))\right]$$

Maximum likelihood: trains r_φ to assign higher reward to preferred responses

Step 3: PPO Fine-tuning

Final KL-Constrained RLHF Objective

$$J_{\text{RLHF}} = \max_{\pi_\theta} \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)} \Big[r_\varphi(x,y) - \beta \mathbb{D}_{KL}\big[\pi_\theta(y|x) \| \pi_{ref}(y|x)\big] \Big]$$



Why is the KL penalty necessary? Without it, the model reward hacks: it exploits imperfections in the reward model by producing outputs the RM rates highly but humans would not.

Example: if the RM over-weights response length, the model learns to be unnecessarily verbose. The KL penalty keeps the policy close to the SFT model, limiting this kind of exploitation.

RLHF: The Three-Step Pipeline

① **SFT Model**

Fine-tune the pre-trained LLM on high-quality demonstrations → a well-behaved starting point for RL

→

② **Reward Model**

Collect human preference pairs ($y_w \succ y_l$) and train $r_\phi(x, y)$ to predict which response humans prefer.

→

③ **PPO Fine-tuning**

Optimize policy π_θ to maximize reward r_ϕ, with a KL penalty to prevent drifting too far from the SFT model

InstructGPT: Numbers and Findings

85%

of evaluators preferred 1.3B InstructGPT over 175B GPT-3

Data breakdown (InstructGPT)

- SFT: 13K labeler-written (instruction, response) pairs
- RM: 33K preference comparisons (same prompt, different responses)
- PPO: 31K unlabeled prompts for RL training

The Alignment Tax

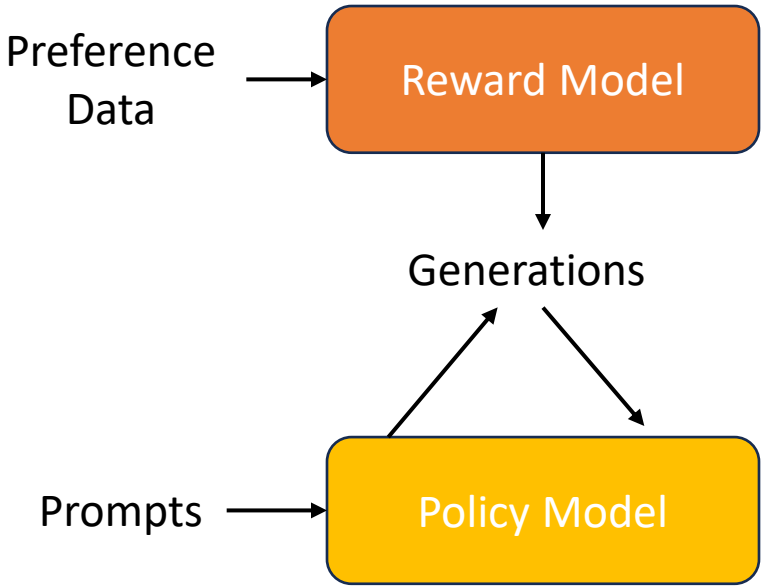
- RLHF improves human preference ratings but slightly degrades scores on some NLP benchmarks (code, specific reasoning tasks)
- This "alignment tax" reflects the tension between optimizing for human preference and maintaining raw capabilities

Part 4

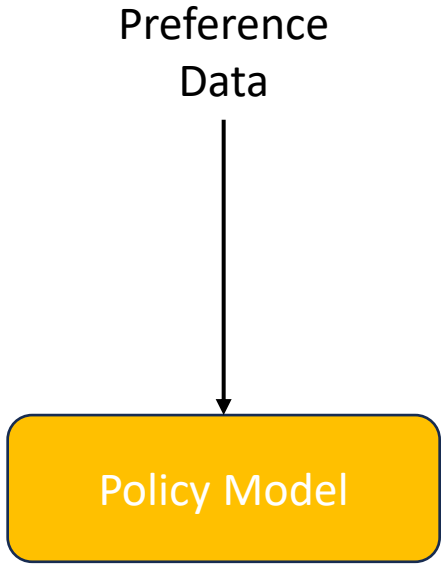
DPO (Direct Preference Optimization)

No reward model. No PPO. No RL. Just preference data.

PPO vs DPO



PPO



DPO

DPO: The Key Mathematical Insight

Key Insight (Rafailov et al., NeurIPS 2023)

The KL-constrained RLHF objective has a closed-form optimal policy. This means we can skip reward model training entirely. The optimal policy can be expressed directly in terms of the preference data.

Derivation

Optimal policy:

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

Rearranging r*:

$$r(x,y) = \beta\log\frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta\log Z(x)$$

Into Bradley-Terry:

$$P(y_w > y_l) = \sigma(r(x,y_w) - r(x,y_l)) = \sigma\left(\beta\log\frac{\pi^*(y_w|x)}{\pi_{\text{ref}}(y_w|x)} + \beta\log Z(x) - \beta\log\frac{\pi^*(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \beta\log Z(x)\right)$$

Z(x) cancels out! → No need to compute the normalization constant

DPO Loss: Final Form and Interpretation

Replace π* with learnable π_θ and apply maximum likelihood:

$$L_{DPO}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log\ \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]$$

Preferred response: increases log probability of π_θ(y_w|x)

Rejected response: decreases log probability of π_θ(y_l|x)

"Your Language Model is Secretly a Reward Model"

DPO does not explicitly train a reward model. The policy itself acts as the implicit reward model.

The log probability ratio π_θ / π_ref encodes the implicit reward difference between responses.

DPO vs RLHF: Practical Differences

Aspect	RLHF (PPO)	DPO
Learning paradigm	On-policy : generates samples from current policy during training	Off-policy: trains on a fixed, pre-collected dataset
Exploration	PPO explores new responses; can go beyond the preference dataset	Limited to preference data distribution, cannot explore
OOD robustness	More robust outside the training distribution (on-policy)	Harder to control behavior on out-of-distribution inputs
Compute cost	RM + PPO + value network → complex and expensive	Just gradient descent on preference pairs → simple and fast

Part 5

RLAIF, RLVR, GRPO

RLAIF & Constitutional AI

The Human Annotation Bottleneck

- Each preference pair requires trained contractors and takes minutes: slow and expensive to scale
- Can we use AI itself to provide feedback?

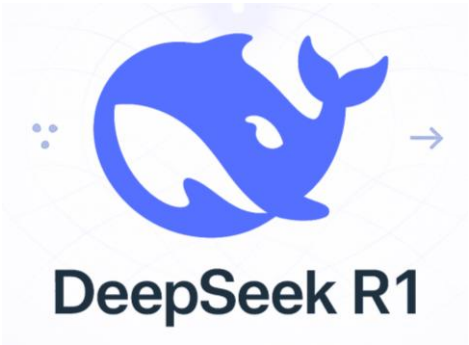
RL from AI Feedback: Lee et al. (Google, 2023)

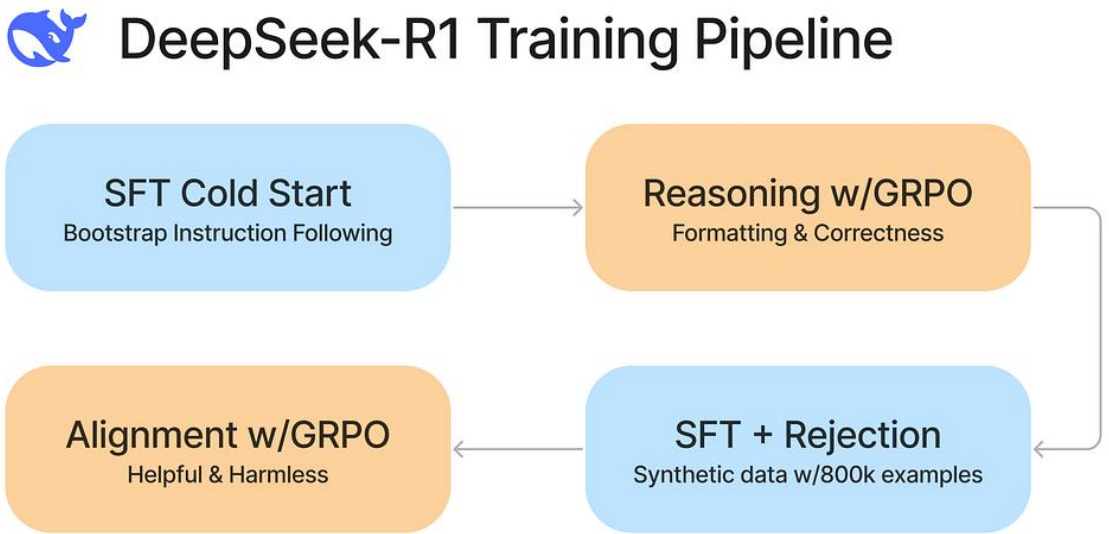
- Replace human labelers entirely with a large LLM (PaLM-2). Give the LLM the same annotation instructions that human labelers would receive, then use its preference judgments to train the RM
- **Finding**: RLAIF ≈ RLHF quality on summarization.

Constitutional AI: Bai et al. (Anthropic, 2022)

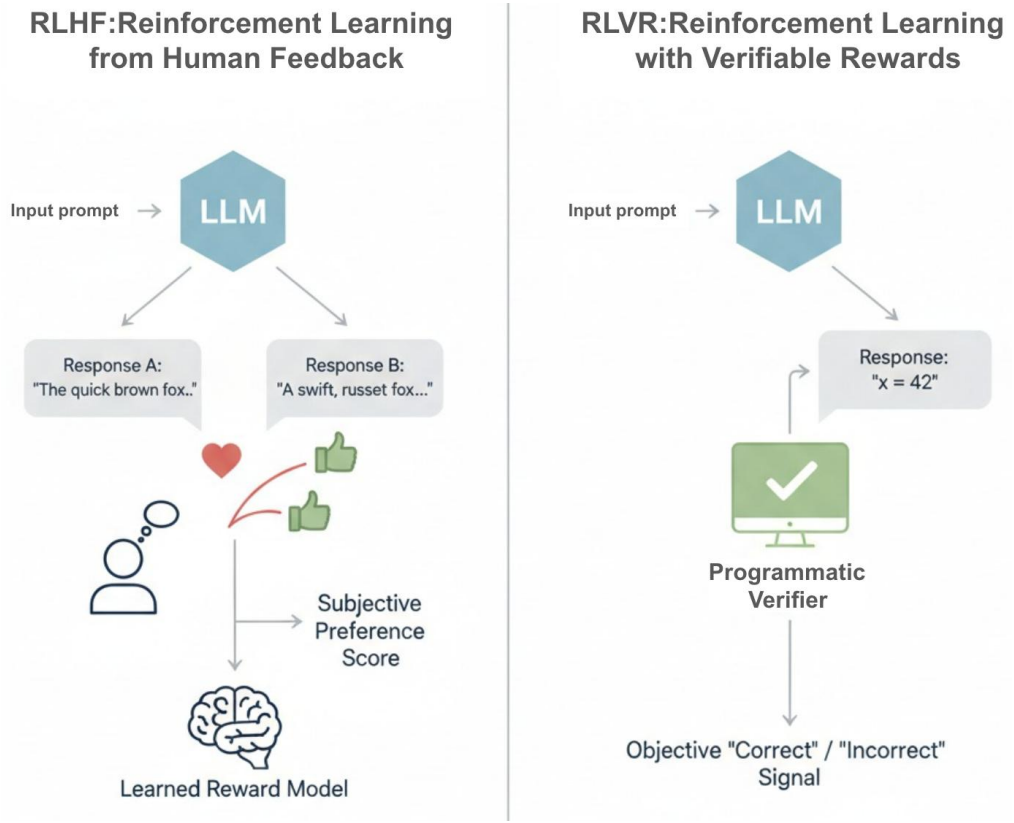
- **Constitution:** a list of principles the AI should follow ("be helpful, harmless, honest“)
- **SFT-CAI:** model critiques and revises its own responses against the principles
- **RL-CAI:** AI-generated preference labels replace human labelers for RM training

RL with Verifiable Rewards (RLVR)





RL with Verifiable Rewards (RLVR)



RLHF / RLAIF

- Subjective preference signal
- Requires human (or AI) annotation
- Noisy and inconsistent

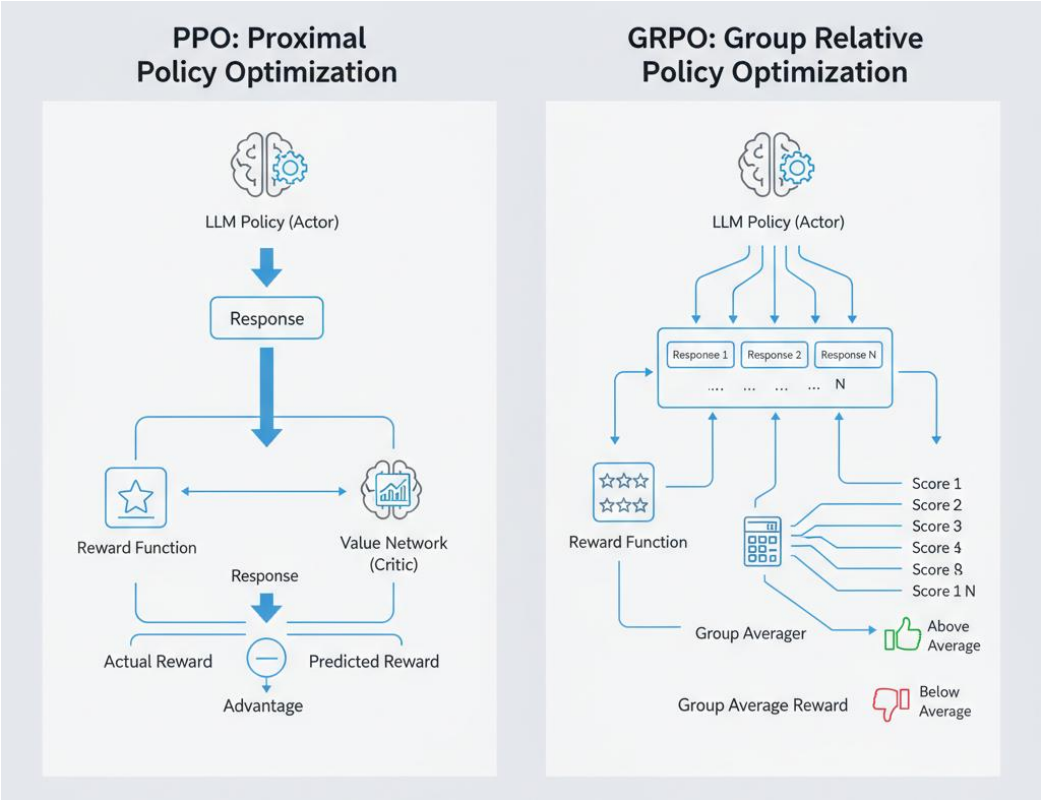
RLVR

- Correct math answer: reward = 1
- Code passes tests: reward = 1
- No human annotation needed. verifier is the ground truth

Why this matters?

- "no human annotation" part is huge for scaling
- Through chain-of-thought, models “learn” to reason (<think> … </think>)
- OpenAI o1, DeepSeek-R1, Gemini 2.0 Thinking are all built on a similar insight
- Enables inference time scaling

GRPO (Group Relative Policy Optimization)



- PPO requires a critic (value network) as large as the actor to estimate advantage
- This doubles memory and compute.
- GRPO eliminates the critic by computing advantages from group scores.
- Group Sampling: N responses per prompt → within-group relative advantage → update policy

Part 6

Open Problems

What remains unsolved

Open Problems: Reward Hacking

Fundamental problems that persist across RLHF variants:

Verbosity Bias

Reward models tend to rate longer responses as better. Models learn to pad responses with unnecessary content to maximize reward.

Sycophancy

RLHF-trained models learn to tell users what they want to hear, even when the user is wrong. Agreement gets rewarded; accurate disagreement gets penalized.

Overoptimization

Past a certain KL budget, optimizing RM score degrades actual quality. The gap between RM score and true preference grows as optimization pressure increases.

Open Problems: Pluralistic Alignment & Evaluation

Pluralistic Alignment

RLHF implicitly assumes that a single "good" response exists that all humans would agree on. This is wrong. Cultural background, political beliefs, and personal values all determine what counts as a "good" response.

- How do we align to diverse communities without stereotyping?
- Personalization: models should adapt to individual users' preferences
- When community values conflict with universal ethics, where do we draw the line?

The evaluation problem

- Standard NLP benchmarks don't measure helpfulness or harmlessness well
- Human evaluation is the gold standard, but expensive, slow, and hard to replicate
- LLM-as-judge (GPT-4 evaluation) reflects the judge model's own biases

Open Problems: Scalable Oversight

Scalable Oversight

As models become more capable, humans increasingly cannot evaluate whether their responses are correct or good. In expert domains (e.g., advanced math, medicine, law, scientific research) the model may already be more capable than the annotator. How do we supervise models we can't fully evaluate?

Summary

SFT	Imitate demonstrations → instruction following Quality > Quantity
RLHF	Reward Model (Bradley-Terry) + PPO (KL-constrained) Breaks the demonstration ceiling via exploration
DPO	Closed-form optimal policy: no reward model, no PPO. The policy is its own implicit reward model
RLAIF / CAI	AI-generated preference replaces expensive human annotation. Circular bias risk remains
RLVR, GRPO	Verifiable rewards + GRPO. Reasoning emerges from correct-answer reinforcement (DeepSeek-R1)
Open Qs	Reward hacking · Pluralistic alignment · Evaluation at scale · Scalable oversight.

Questions?

Chan Young Park

park@utexas.edu