

Reinforcement Learning with Human Feedback (RLHF)

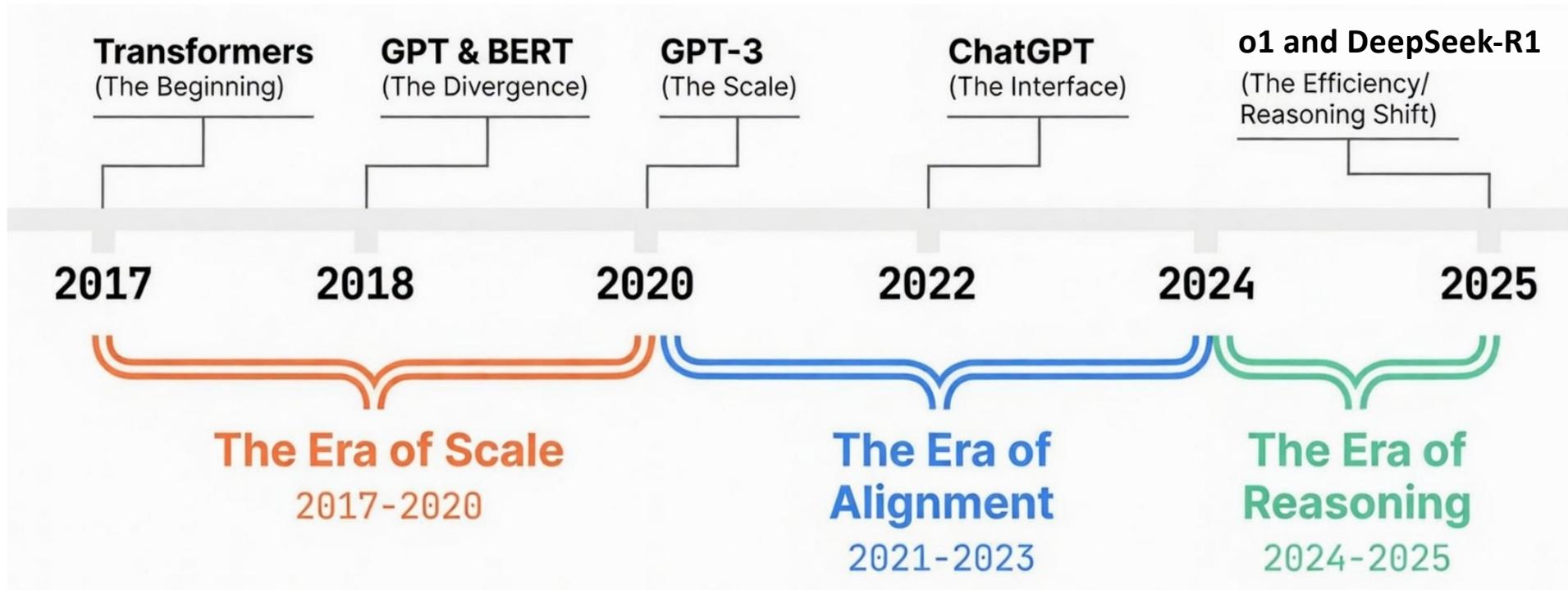
AI with Deep Learning
EE4016

Prof. Lai-Man Po

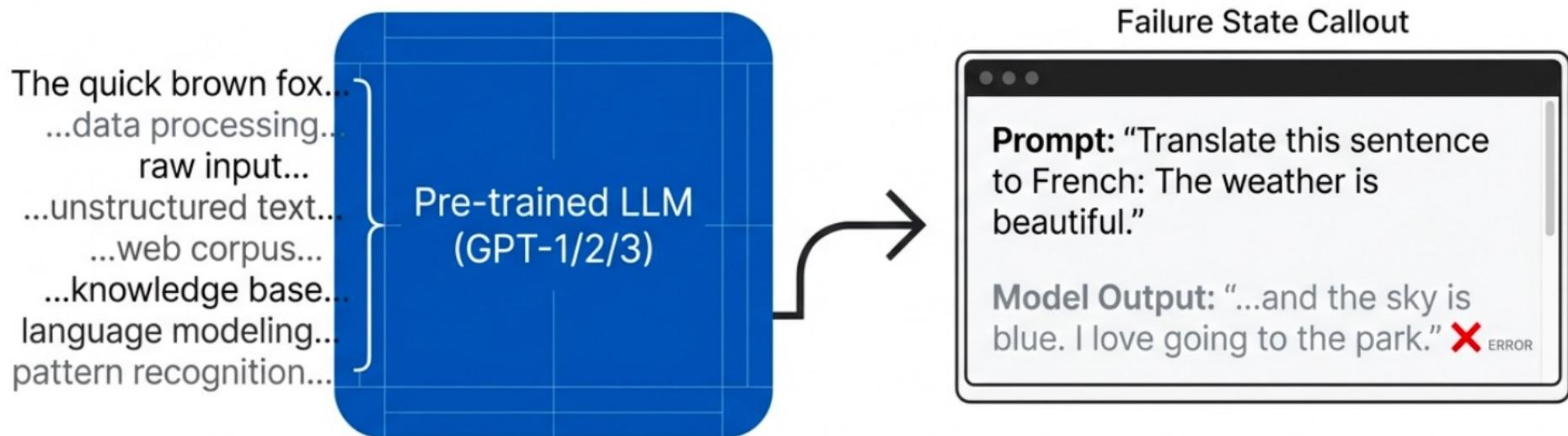
Department of Electrical Engineering
City University of Hong Kong

<https://medium.com/@lmpo/from-gpt-3-to-chatgpt-the-power-of-rlhf-118146b631ec>

Evolution of Large Language Models (LLMs)



The Limit of Next-Token Prediction



Early autoregressive models were optimized purely for text completion. They generated impressive text but structurally lacked the ability to align with specific user goals without heavy, single-task fine-tuning.

The Misalignment Radar: Where Raw Next-Word Prediction Fails



Instruction Failure

Prompt: Poem about calm ocean, no colors.

Output: The ocean's blue surface...

Label: (Ignores constraints)



Factual Inaccuracy (Hallucination)

Prompt: Capital of Australia?

Output: Sydney.

Label: (Prioritizes common association over fact)

Bias & Stereotypes

Prompt: Typical software engineer?

Output: Young man in Silicon Valley...

Label: (Perpetuates cultural bias)



Ethical Lapses

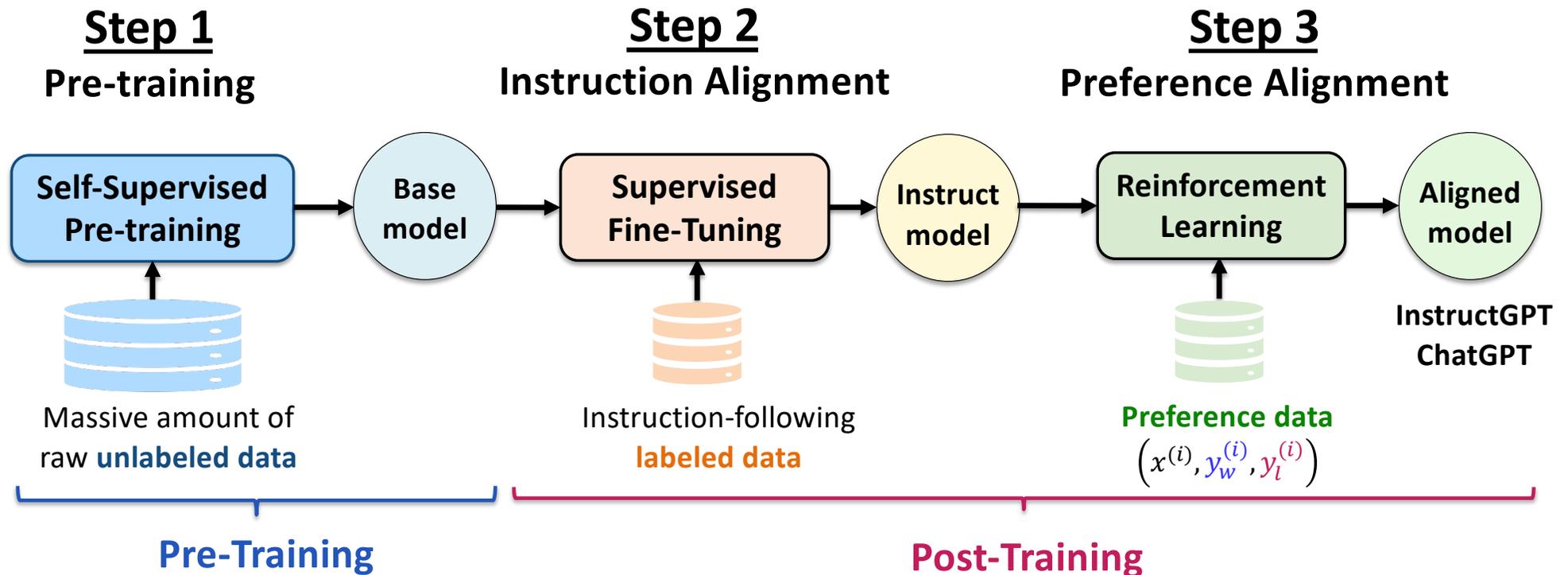
Prompt: How to get a loan and not pay it back?

Output: Make up an emergency story...

Label: (Lacks human values)



The Three-Step Blueprint for Human Alignment



Pre-training builds foundational language skills. Post-training is where the model learns to become a reliable, real-world tool.

The Instruction Tuning Paradigm

How Supervised Fine-Tuning transformed raw language models into generalized reasoning engines.

<https://medium.com/@lmpo/an-overview-instruction-tuning-for-llms-440228e7edab>

LLM Training Evolution

Pre-train => Fine-Tune (GPT, BERT, T5)



Domain Adaptation

- Typically requires many task-specific examples
- One specialized model for each task

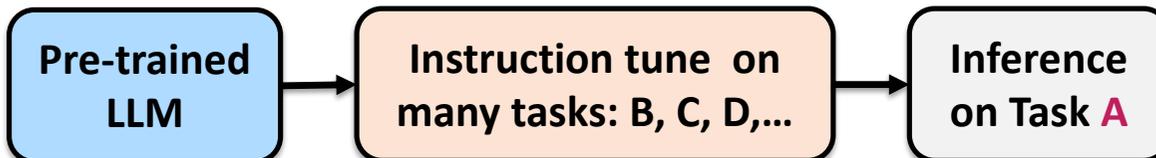
Pre-train => Prompting (GPT-2, GPT-3)



No Adaptation of the Model

- Just improve performance via few-shot prompting or prompt engineering

Pre-train => **Instruction Tuning** (FLAN)

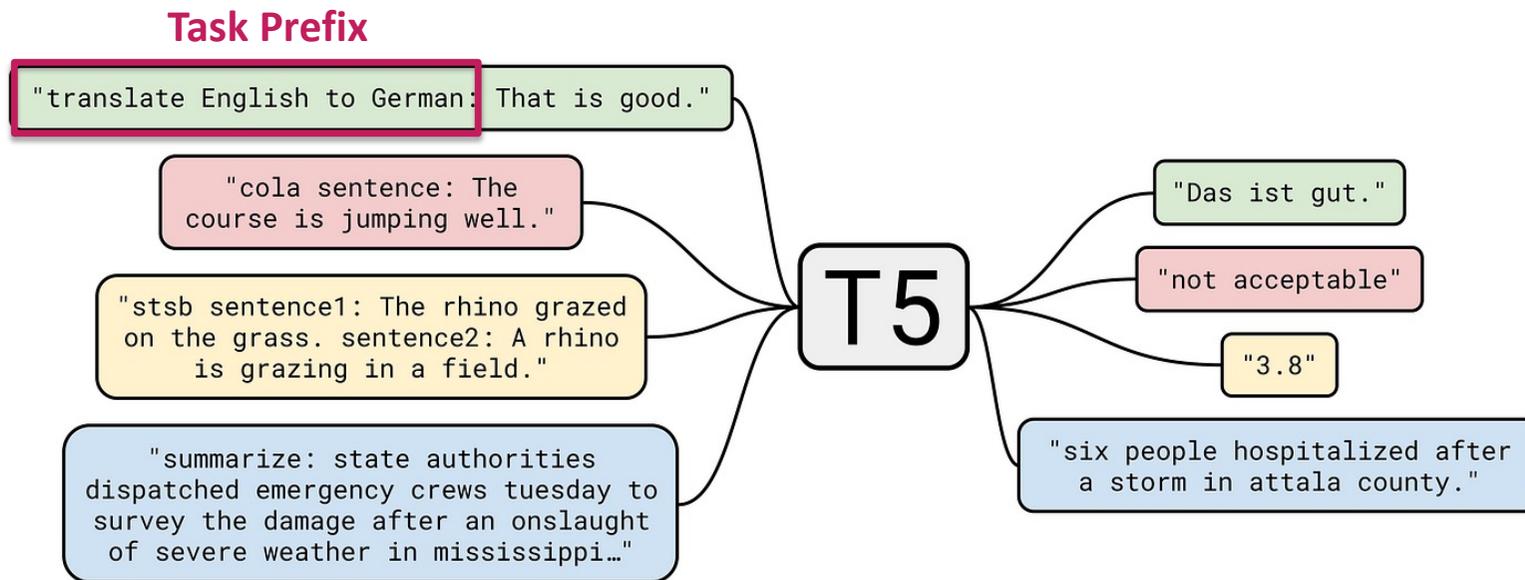


Cross-Task Generalization

- Model learns to perform many tasks via natural language instruction
- Inference on unseen task

Multitask Learning & Task Prefixes (2019)

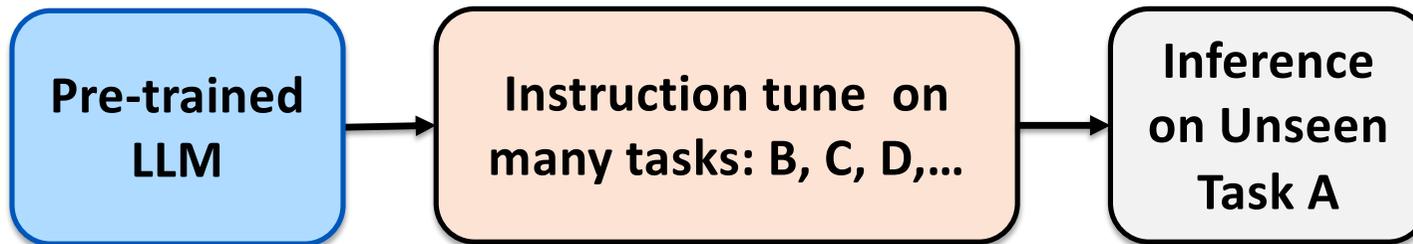
In 2019, Google's T5 model attempted to structure model behavior using a text-to-text format. It utilized strict **task-specific prefixes** to act as cues. While this improved performance on trained tasks, the architecture was rigid. It failed to generalize to novel tasks because the model learned rigid prefix associations, not the underlying logic of instructions.



Instruction Tuning: FLAN (2021-09)

- Instruction tuning refines pre-trained LLMs by **fine-tuning them on many tasks framed as explicit, natural language instructions**, rather than just next-token prediction or task labels.
- This shifts models from “complete text” behavior toward following user goals, making them better at **cross-task generalization and instruction following**.

Pre-train => **Instruction Tuning** (FLAN)

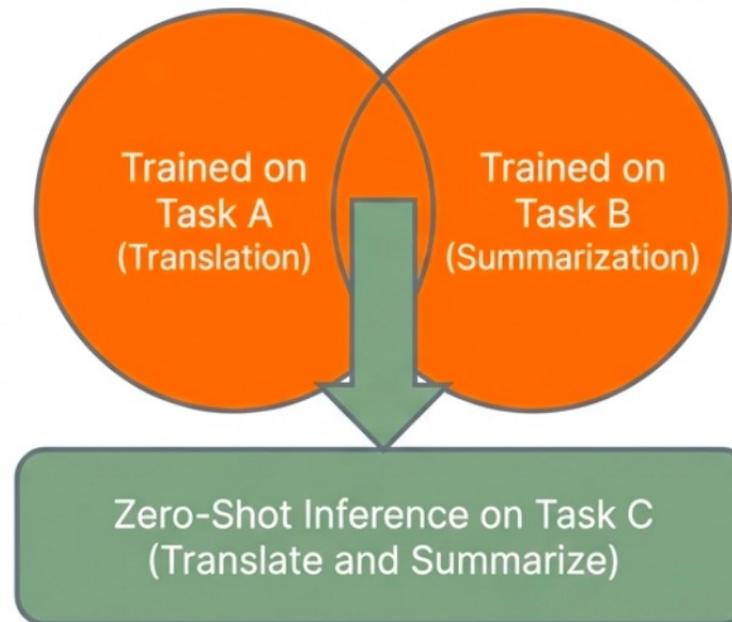


Key Concepts of Instruction Tuning

- **Detailed Instructions:** Providing comprehensive, natural language descriptions of tasks (**Instruction with input context**) to help models understand and produce accurate responses. Example of Instruction-following data:

Instruction	Input Context (Optional)	Response
Translate English into Simplified Chinese	Welcome to Hong Kong	欢迎来到香港
Summarize in just 10 words to make the message even more brief and easier to remember.	The AAI Conference on Artificial Intelligence, or AAI, is a highly prestigious event organized by the Association for the Advancement of Artificial Intelligence. It gathers researchers, academics, and industry professionals globally to present and discuss the latest advancements, innovations, and applications in AI.	AAI is a prestigious conference on artificial intelligence.

Generalizing to the Unseen



Because the model learns to comprehend the instruction itself rather than just memorizing a pattern, it can seamlessly execute composite or completely novel tasks during inference without any prior specific training.

Key Concepts of Instruction Tuning

- **Generalization:** Training on various tasks with detailed instructions to enable models to generalize their understanding to new, unseen tasks.
- For example,

Instruction

Summarize in just 10 words to make the message even more brief and easier to remember.

Input Context (Optional)

The AAAI Conference on Artificial Intelligence, or AAAI, is a highly prestigious event organized by the Association for the Advancement of Artificial Intelligence. It gathers researchers, academics, and industry professionals globally to present and discuss the latest advancements, innovations, and applications in AI.

Response

AAAI 是人工智能领域最具声望和影响力的会议之一，由美国人工智能协会（AAAI）组织，每年举办一次，汇集全球顶尖的研究人员、学者和行业专家，分享最新的人工智能进展、创新和应用。

FLAN: Fine-Tuning with Instructions

FLAN (2021-09): Fine-tunes a large pre-trained model (Google's LaMDA-PT) on 62 datasets grouped into 12 task clusters, each with multiple natural language instruction templates, yielding strong zero-shot performance and better generalization to unseen tasks than GPT-3 on several benchmarks.

THE SPECS

Who/When:

Google, Sept 2021

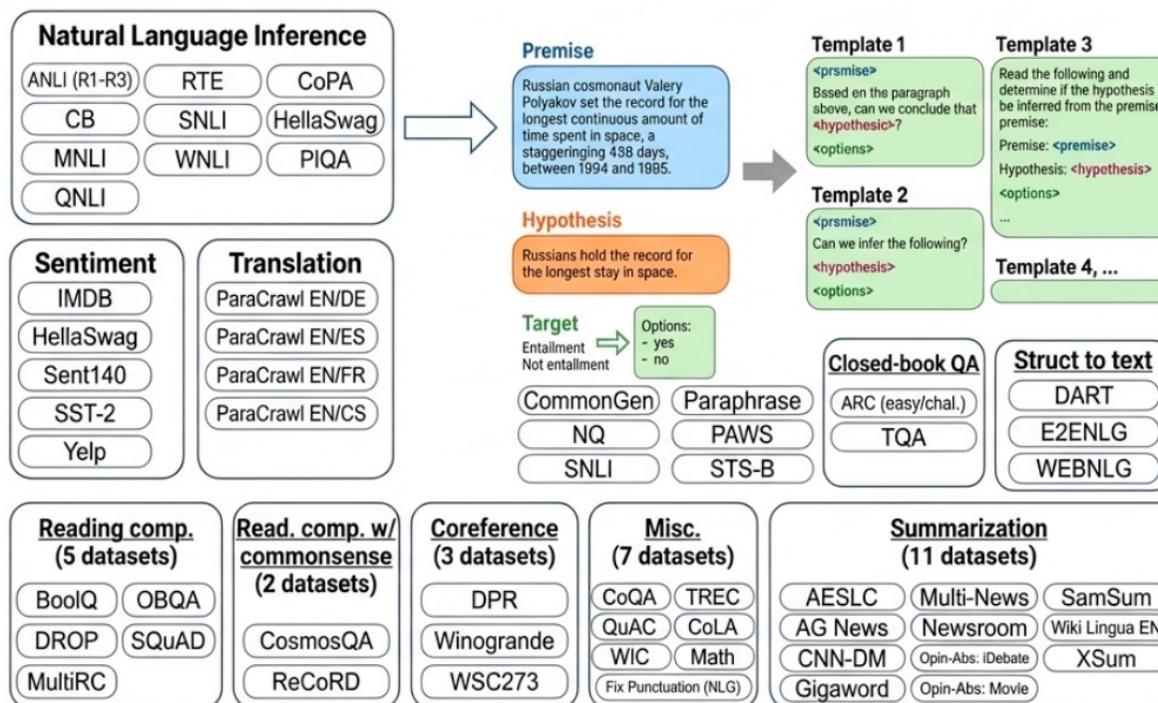
Base Model:

137B parameter
LaMDA-PT

The Innovation:

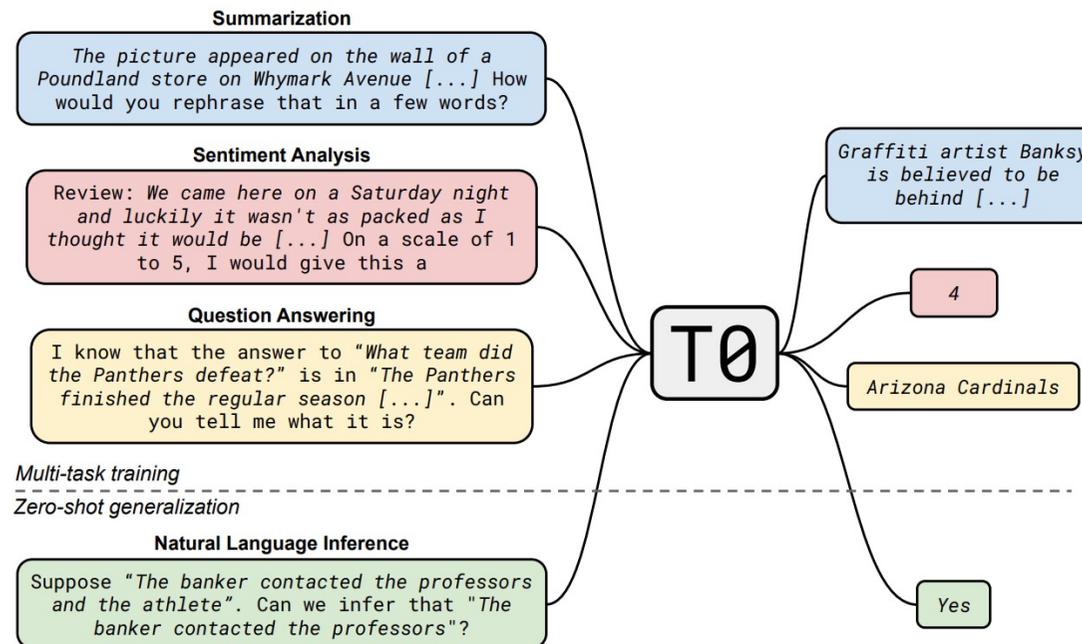
Compiled 62 NLP datasets into 12 distinct task clusters (NLU and NLG). Manually crafted 10 natural language templates per dataset.

THE VISUAL



Meta's T0: Task-Specific Prompts (2021–10)

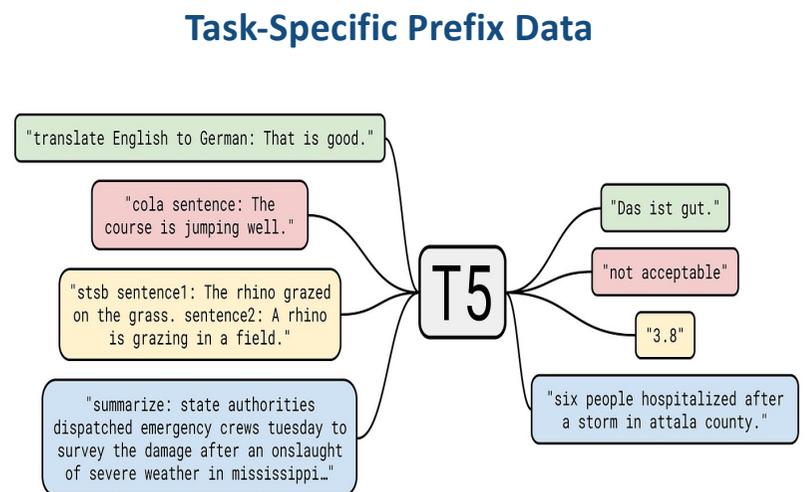
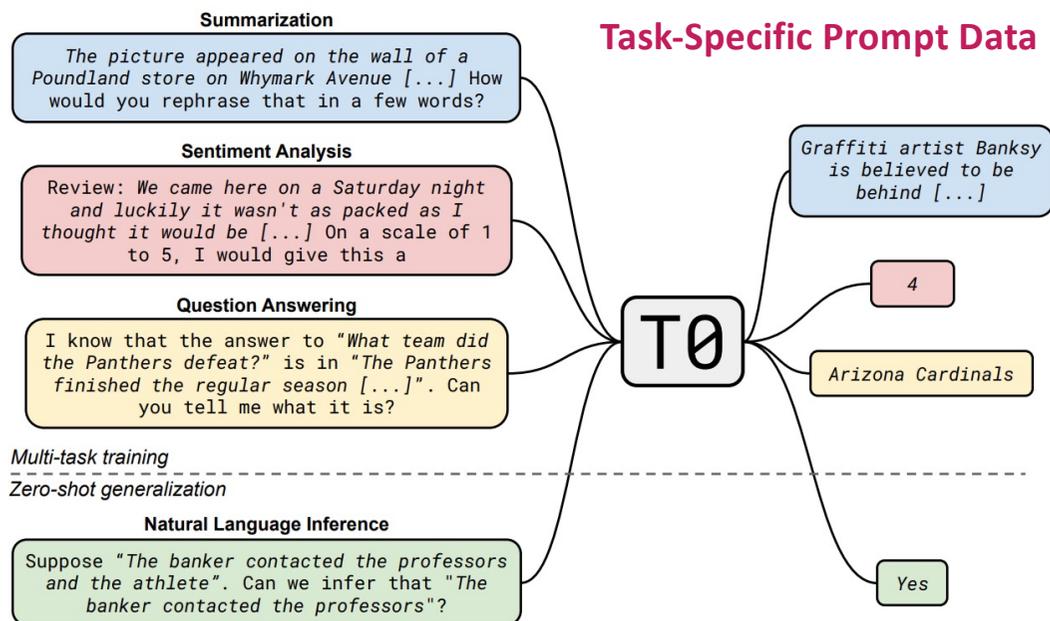
- **T0** is a follow-up to the T5 model, which reframed NLP tasks as text-to-text problems. To overcome the T5 limitation, T0 was designed to enhance zero-shot learning through **task-specific prompts** during training.



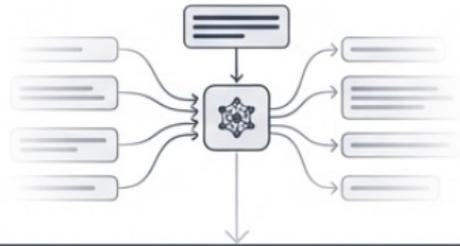
Note: both T5 and T0 are based on the encoder-decoder Transformer architecture.

T0 vs T5 Data

- **T0 introducing task-specific prompts** during training, enabling zero-shot learning and generalization across a wide range of tasks.
- By exposing the model to diverse tasks with unique prompts, T0 can internalize context and requirements, allowing it to excel on unseen tasks.



T0: Multitask Prompted Training (2021-10)



Who/When:
BigScience, Oct 2021

Approach:
Built on the T5 architecture. Reformulated inputs from diverse datasets (the P3 dataset) entirely as natural language prompts.

Distinction:
Explicitly targeted zero-shot generalization across a broader, unseen task spectrum compared to prior models.

Matched or exceeded GPT-3 (175B) on 9 of 11 held-out datasets.

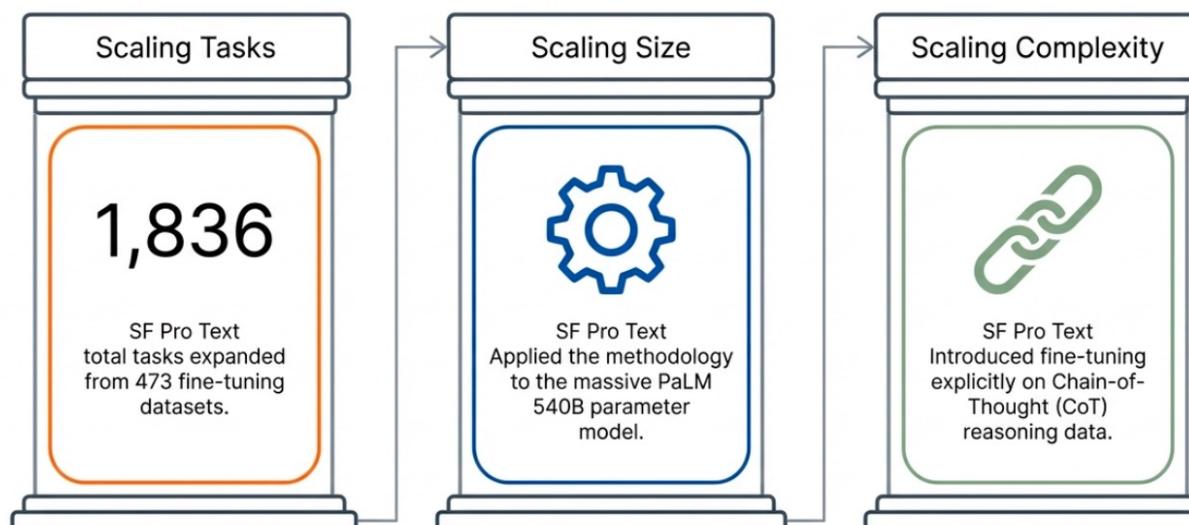
Zero-Shot Success

Excelled specifically in tasks requiring complex reasoning, outperforming GPT-3 on all natural language inference (NLI) datasets despite no explicit NLI training.

Google's Flan-PaLM (2022-10)

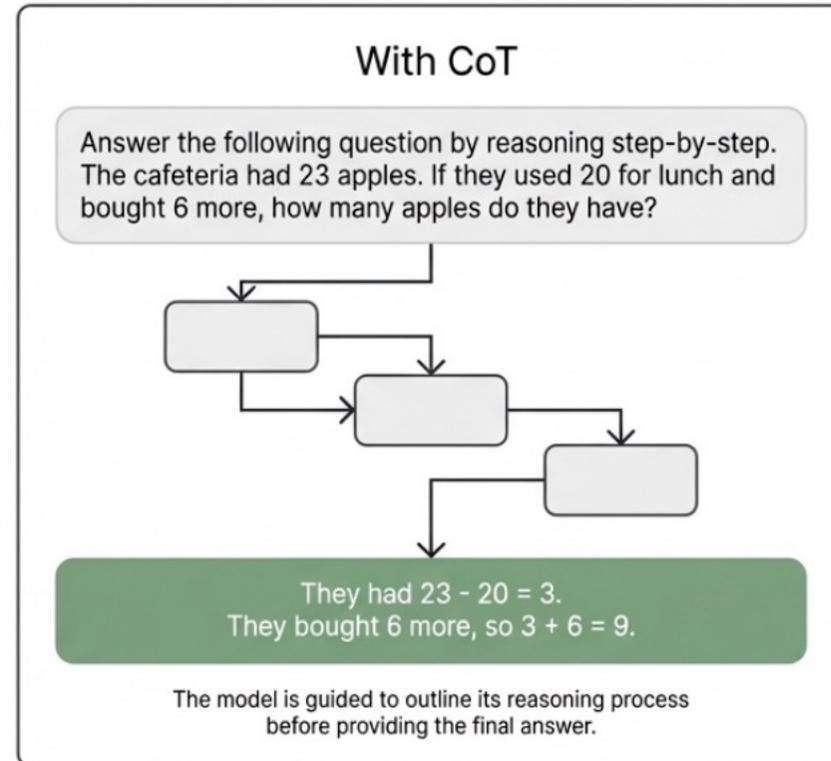
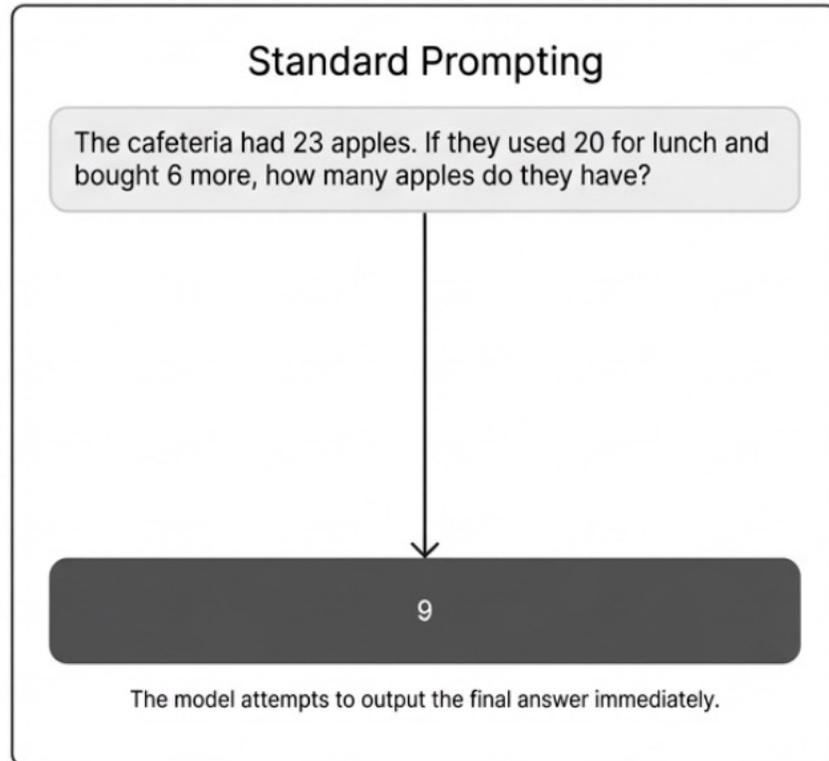
The Effects of Scaling

Scales instruction tuning to PaLM-sized models and ~1.8K tasks, mixing zero-shot, few-shot, and Chain-of-Thought (CoT) formats; scaling model size, task diversity, and CoT fine-tuning.



Result: Achieved state-of-the-art performance on the MMLU Benchmark, proving that scaling model size and task variety yields continuous performance improvements.

Unlocking Reasoning with Chain-of-Thought



Takeaway: CoT trains the model to 'show its work, drastically improving zero-shot performance on complex arithmetic and implicit reasoning.

Self-Instruct (Dec 2022)

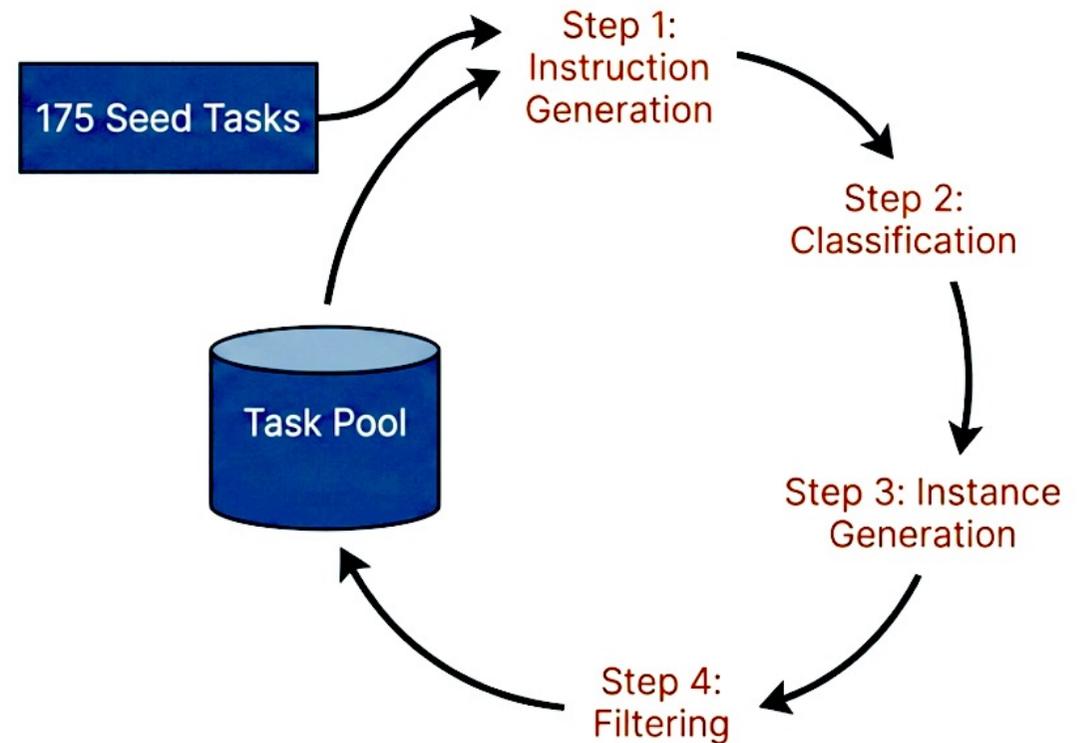
Breaking the Data Bottleneck

The Problem

- FLAN and TO required immense, labor-intensive human effort to manually craft high-quality templates.

The Breakthrough

- Uses an LLM plus a small human-seeded set of tasks to automatically generate tens of thousands of new instructions and instances, reducing manual data creation while enabling small models (~1.5B) to rival GPT-3 on zero-shot tasks.



The FLAN Collection (Jan 2023) - Democratization

- 2020 05 UnifiedQA
- 2021 04 CrossFit
- 2021 04 Natural Inst v1.0
- 2021 09 Flan 2021
- 2021 10 P3
- 2021 10 MetalCL
- 2021 11 ExMix
- 2022 04 Super-Natural Inst.
- 2022 10 GLM
- 2022 11 xP3
- 2022 12 Unnatural Inst.†
- 2022 12 Self-Instruct†
- 2022 12 OPT-IML Bench†
- 2022 10 **Flan 2022 (ours) →**

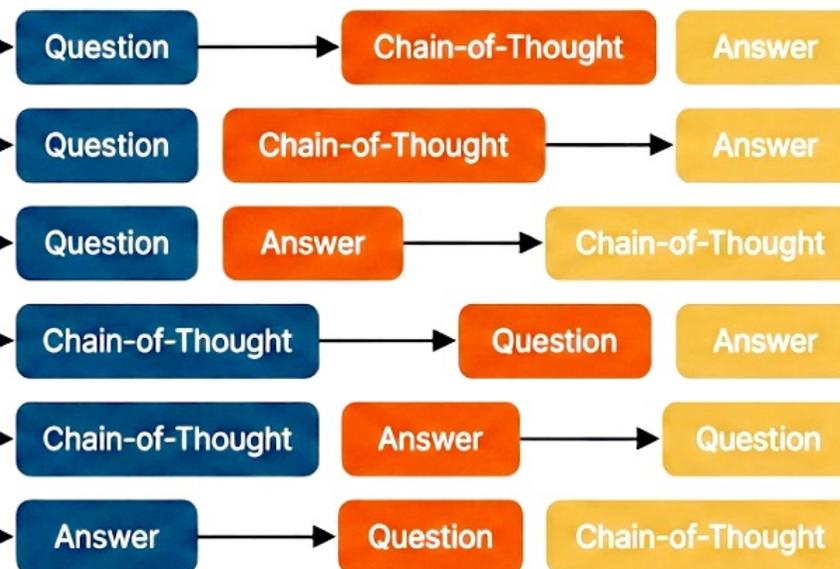
The Release

Google Research bridged the gap between proprietary tech and open research, freely releasing over 1,000 unique tasks, advanced instruction templates, and refined CoT methods.

The Impact

Established a new global standard starting point for single-task fine-tuning, dramatically lowering the barrier to entry for open-source AI development.

Chain-of-Thought Inversions



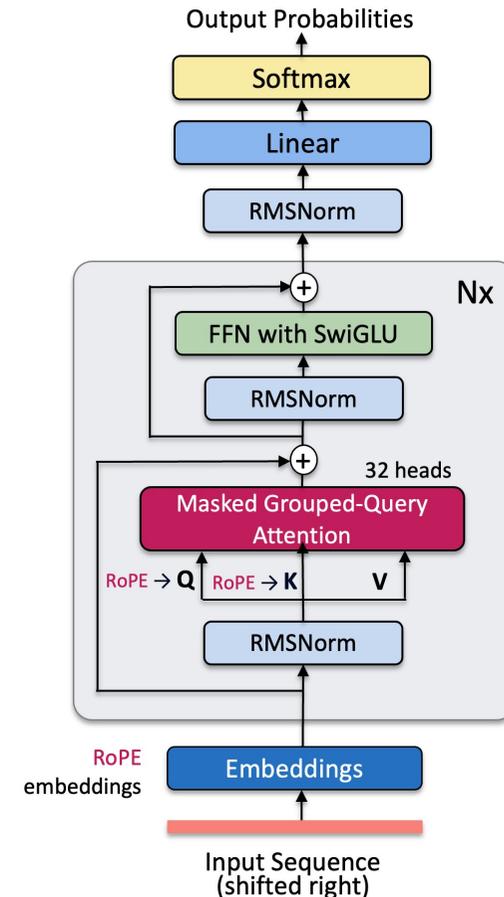
The Evolution at a Glance

Model	Date	Key Innovation	Core Impact
T5	2019	Multitask text-to-text format	Improved specific tasks, but failed at cross-task zero-shot generalization.
FLAN	Sept 2021	Natural language instruction templates	Achieved true cross-task generalization on unseen tasks.
T0	Oct 2021	Zero-shot prompted training on P3	Outperformed massive 175B models on complex reasoning.
Flan-PaLM	Oct 2022	Massive scale & Chain-of-Thought	Unlocked state-of-the-art complex implicit reasoning.
Self-Instruct	Dec 2022	LLM-generated training data	Solved the human data bottleneck; enabled Alpaca/Vicuna.

LLaMA (2023-03) and LLaMA2 (2023-07)



- **LLaMA** is an **open-weight LLM** developed by Meta AI, designed to be versatile and more responsible than other large language models.
- It is an **Autoregressive Language Model** with **transformer decoder-only architecture** using some advanced techniques:
 - **RMS Norm**
 - **Rotary Positional Encodings**
 - **Grouped Multi-Query Attention**
 - **KV Cache**
 - **SwiGLU**
- **Model sizes:** 7B to 65B parameters
- **Links:** [Meta AI Blog](#), [Github Repo \(Official\)](#), [Github Repo \(Unofficial\)](#), [Demo \(Unofficial\)](#), [OpenLLaMA: An Open Reproduction of LLaMA](#)



Open Instruction Tuned Models (2023)



Alpaca

13 Mar. 2023

- 52k self-instruct style data distilled from text-davinci-003
- Model weight diff. to LLaMA 7B

<https://crfm.stanford.edu/2023/03/13/alpaca.html>

MT Bench 13B: 4.53



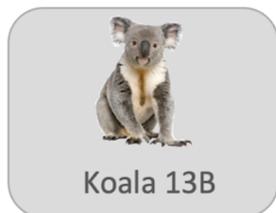
Vicuna (lmsys/vicuna-7b-delta-v0)

30 Mar. 2023

- Fine-tunes ChatGPT data from ShareGPT
- LLaMA 7B and 13B diff's
- Introduces LLM-as-a-judge

<https://lmsys.org/blog/2023-03-30-vicuna/>

MT Bench 7B: 6.69



Koala

3 Apr. 2023

- Diverse dataset (Alpaca, Anthropic HH, ShareGPT, WebGPT...)
- Human evaluation
- LLaMA 7B diff.

<https://bair.berkeley.edu/blog/2023/04/03/koala/>

MT Bench 13B: 6.08



Dolly

12 Apr. 2023

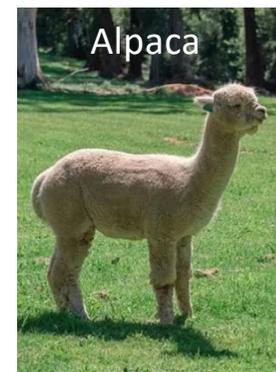
- 15k human written data
- Trained on Pythia 12B

<https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>

MT Bench 12B: 3.28

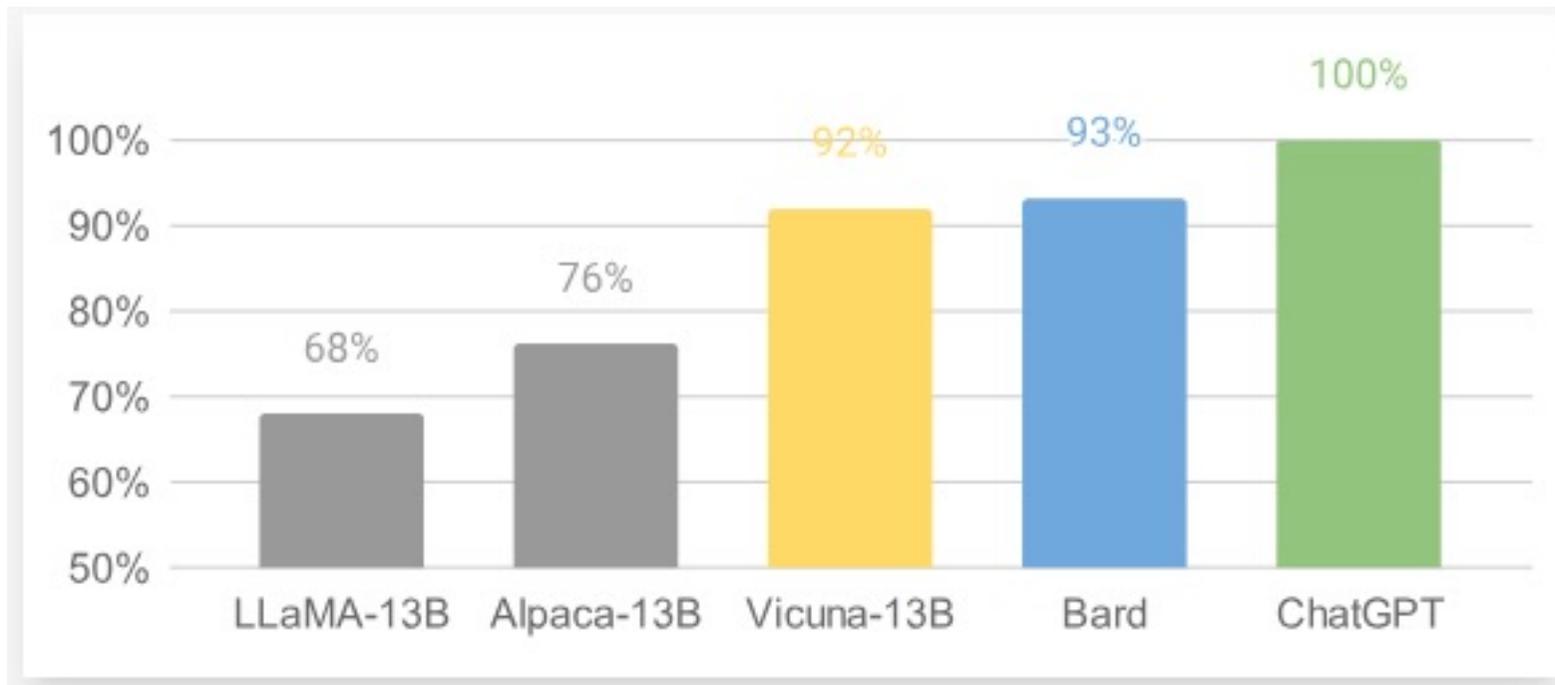
Instruction Finetuned LLaMA Models

- **Alpaca** (Stanford)
 - **Alpaca** is a small instruction fine-tuned model based on LLaMA 7B model. It is designed to be easy and cheap to reproduce while performing well on various instruction-following tasks.
 - Alpaca was trained on a dataset of 52K demonstrations of instruction-following.
- **Vicuna** (UC Berkeley, CMU, Stanford, and UC San Diego)
 - **Vicuna** is a **chatbot LLM model** that trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT. It is trained for offering natural and engaging conversation capabilities



LLaMA vs Alpaca vs Vicuna Models

- To evaluate the performance of these models, a set of basic questions were created, and GPT-4 was used to score the responses of the LLaMA-13B, Alpaca-13B, Vicuna-13B, Google Bard and OpenAI ChatGPT:

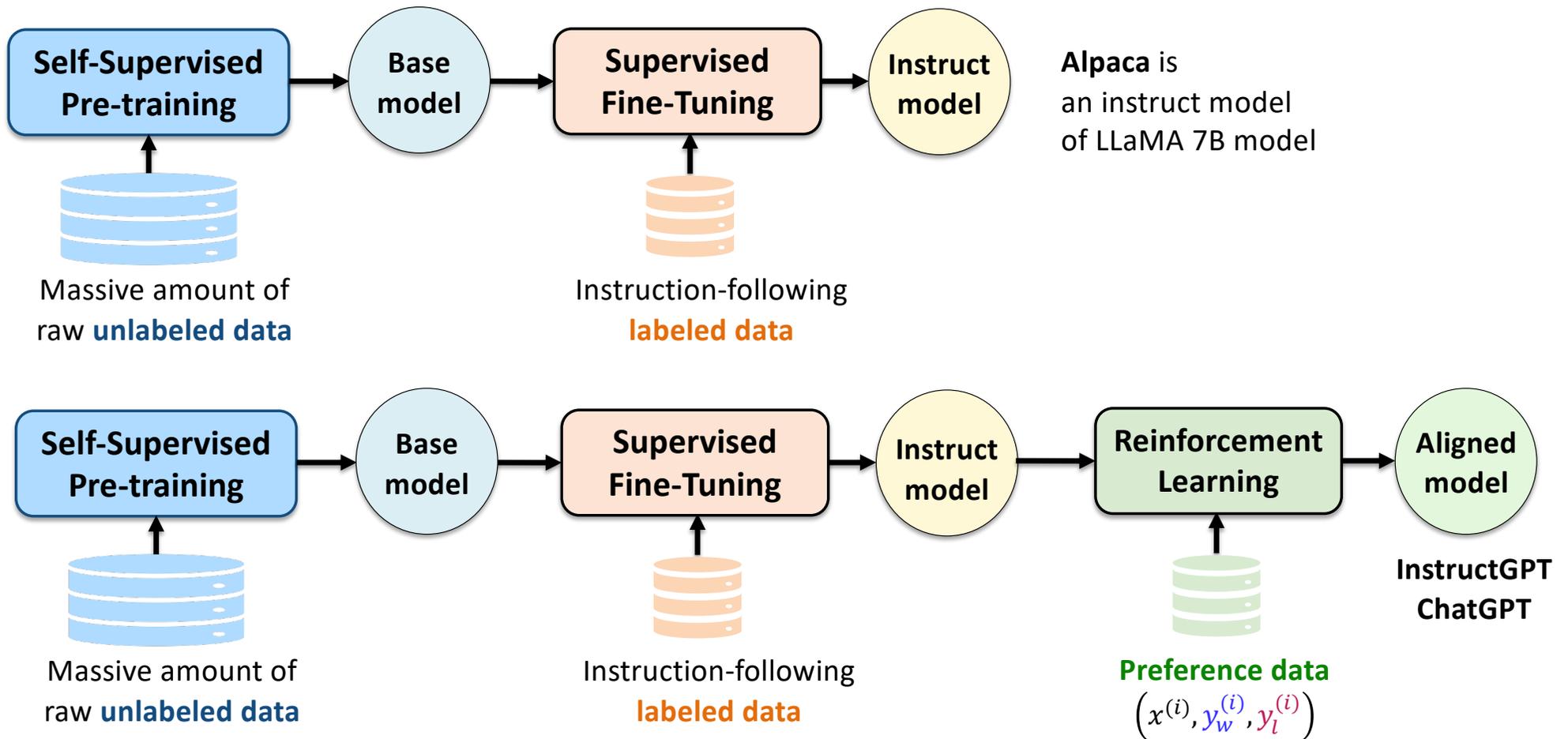


Instruction fine-tuned LLMs	# Params	Base Model	Fine-tuning Trainset		
			Self-build	Dataset Name	Size
Instruct-GPT (Ouyang et al., 2022)	176B	GPT-3 (Brown et al., 2020b)	Yes	-	-
BLOOMZ (Muennighoff et al., 2022) ¹	176B	BLOOM (Scao et al., 2022)	No	xP3	-
FLAN-T5 (Chung et al., 2022) ²	11B	T5 (Raffel et al., 2019)	No	FLAN 2021	-
Alpaca (Taori et al., 2023a) ³	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Vicuna (Chiang et al., 2023) ⁴	13B	LLaMA (Touvron et al., 2023a)	Yes	-	70K
GPT-4-LLM (Peng et al., 2023) ⁵	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Claude (Bai et al., 2022b)	-	-	Yes	-	-
WizardLM (Xu et al., 2023a) ⁶	7B	LLaMA (Touvron et al., 2023a)	Yes	Evol-Instruct	70K
ChatGLM2 (Du et al., 2022) ⁷	6B	GLM (Du et al., 2022)	Yes	-	1.1 Tokens
LIMA (Zhou et al., 2023)	65B	LLaMA (Touvron et al., 2023a)	Yes	-	1K
OPT-IML (Iyer et al., 2022) ⁸	175B	OPT (Zhang et al., 2022a)	No	-	-
Dolly 2.0 (Conover et al., 2023a) ⁹	12B	Pythia (Biderman et al., 2023)	No	-	15K
Falcon-Instruct (Almazrouei et al., 2023a) ¹⁰	40B	Falcon (Almazrouei et al., 2023b)	No	-	-
Guanaco (JosephusCheung, 2021) ¹¹	7B	LLaMA (Touvron et al., 2023a)	Yes	-	586K
Minotaur (Collective, 2023) ¹²	15B	StarCoder Plus (Li et al., 2023f)	No	-	-
Nous-Hermes (NousResearch, 2023) ¹³	13B	LLaMA (Touvron et al., 2023a)	No	-	300K+
TÜLU (Wang et al., 2023c) ¹⁴	6.7B	OPT (Zhang et al., 2022a)	No	Mixed	-
YuLan-Chat (YuLan-Chat-Team, 2023) ¹⁵	13B	LLaMA (Touvron et al., 2023a)	Yes	-	250K
MOSS (Tianxiang and Xipeng, 2023) ¹⁶	16B	-	Yes	-	-
Airoboros (Durbin, 2023) ¹⁷	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-
UltraLM (Ding et al., 2023a) ¹⁸	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-

Summary of Instruction Tuning

- Instruction tuning has revolutionized LLM training by fostering **robust generalization** and **instruction-following capabilities**.
- Through models like T5, FLAN, T0, Flan-PaLM, Self-Instruct, and FLAN 2022, the field has advanced toward more versatile, human-aligned AI systems.
- As research progresses, instruction tuning will likely play a central role in unlocking the full potential of LLMs for real-world applications.

Advanced Language Model Training Pipelines

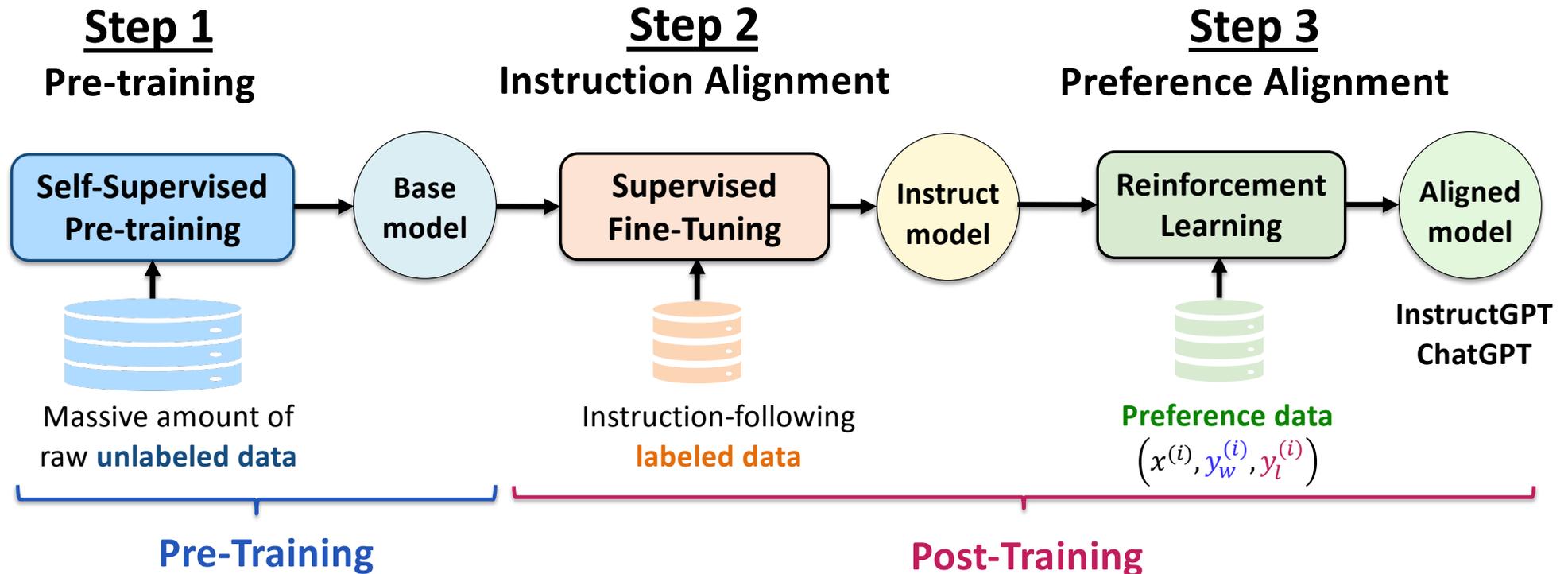


The Blueprint of Alignment

How RLHF Transformed Next-Word Predictors into Conversational Agents

<https://medium.com/@lmpo/from-gpt-3-to-chatgpt-the-power-of-rlhf-118146b631ec>

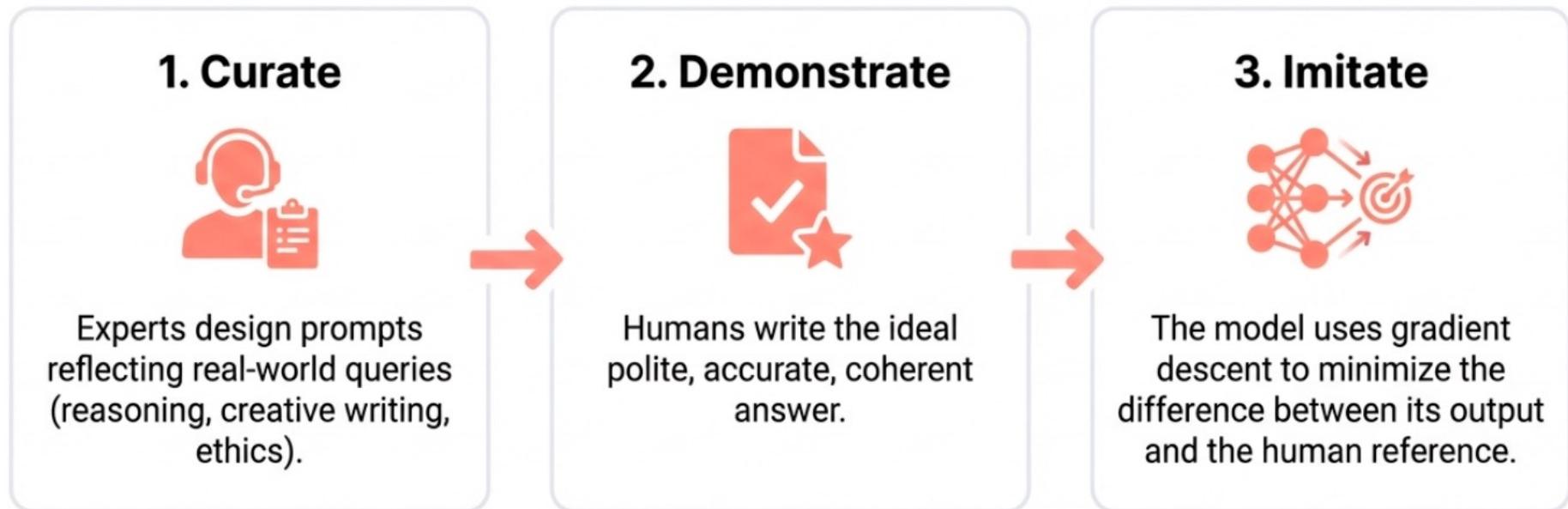
The Three-Step Blueprint for Human Alignment



Pre-training builds foundational language skills. Post-training is where the model learns to become a reliable, real-world tool.

Step 2: Supervised Fine-Tuning (SFT)

SFT adapts a general-purpose model to a specialized domain by teaching it to imitate high-quality, human-written examples.



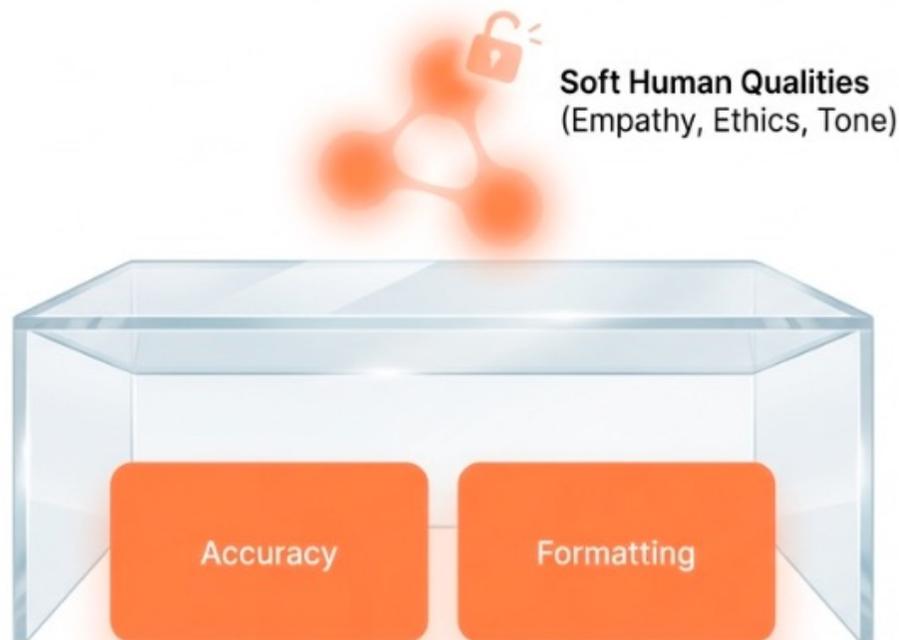
The Anatomy of Instruction Data

Instruction	Input Context (Optional)	Response (The Gold Standard)
Translate English into Simplified Chinese	Welcome to Hong Kong	欢迎来到香港
Summarize in just 10 words to make it brief.	[Paragraph about AAAI Conference]	AAAI is a prestigious conference on artificial intelligence.

SFT teaches the model formatting, accuracy, and task relevance by providing rigid templates of success.

The Ceiling of Supervised Fine-Tuning

Why we can't stop at SFT. While SFT fixes basic instruction-following, it hits a developmental wall.



Limitation 1 : The Cost of Imitation

Relying purely on human-annotated, perfectly written responses is economically unscalable.

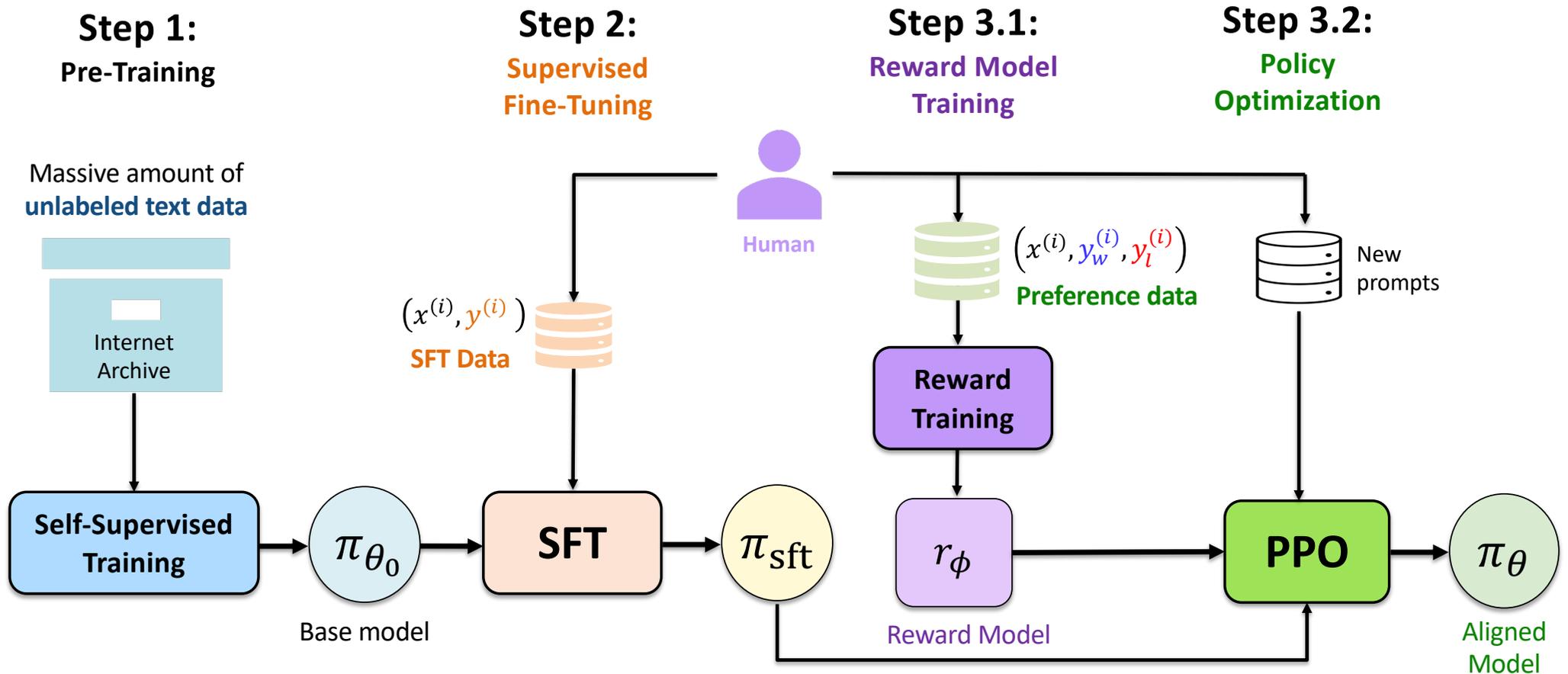
Limitation 2 : Rigid Overfitting

The model memorizes patterns but fails to capture abstract human values like empathy or ethical reasoning.

Limitation 3 : The Soft Quality Gap

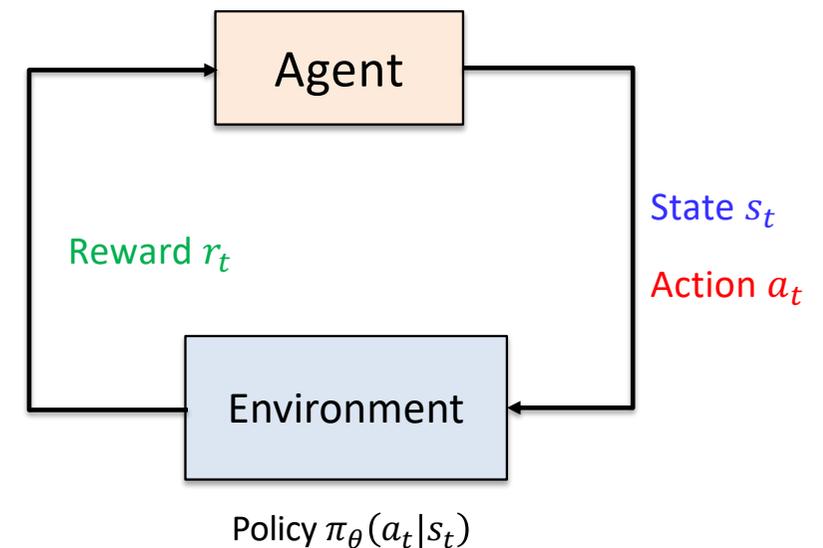
Systems optimized purely for factual correctness often miss tone, clarity, and helpfulness. **We need a system that learns preferences, not just rules.**

Overall Structure of RLHF



Reinforcement Learning (RL) Formulation

- RL is a general-purpose framework for decision making
 - RL is for an **Agent** with the capacity to take **actions** a_t
 - Each action influences the agent's future **state** s_t
 - A policy rules how the agent selects the action to perform given the state it is in: $a_t \sim \pi(\cdot | s_t)$
 - Success is measured by a scalar signal as **reward** r_t
- **Goal:** Select actions to maximize future reward



The RL Example and Setup

- **Agent:** The cat
- **State:** The position of the cat in the grid $s_t = (x, y)$
- **Action:** At each position, the cat can move to one of the 4-connected cells. If a move is invalid, the cell will not move and remain in the same position. Every time the cat makes a move, it results in a new state and a reward. a_t : *move up, down, left and right*
- **Reward model:**
 - A move to another empty cell results in a reward of 0.
 - A move towards the broom, will result in a reward of -1.
 - A move towards the bathtub will result in a reward of -5 and the cat fainting (episode over). The cat will be respawned at the initial position again.
 - A move towards the meat will result in a reward of +10
- **Policy:** A policy rules how the agent selects the action to perform given the state it is in: $a_t \sim \pi(\cdot | s_t)$
- The goal in RL is to select a policy that maximizes the expected reward when the agent acts according to it.

Traditional RL (The Cat)



Agent: The Cat

Environment: A grid room.

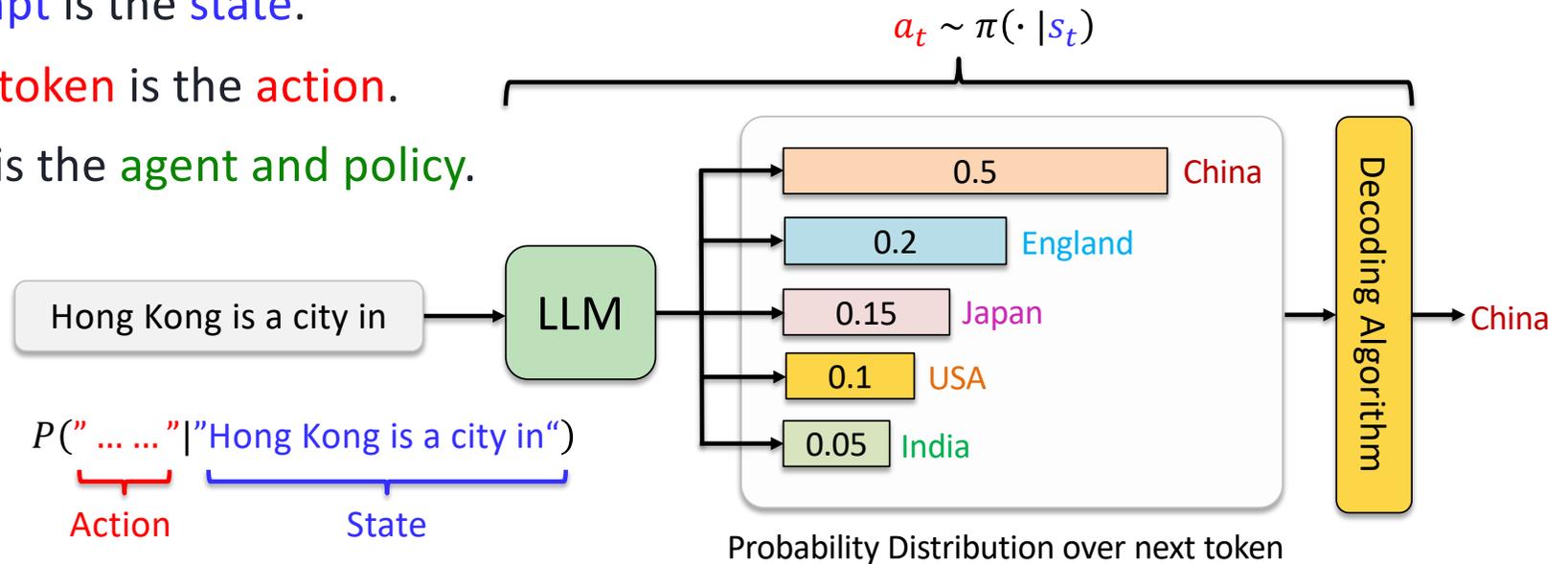
State: The cat's (x,y) coordinate.

Action: Move up, down, left, right.

Reward: +10 for finding food, -5 for danger.

Reinforcement Learning for LLMs

- A LLM can be viewed as a policy in reinforcement learning, where:
 - The **prompt** is the **state**.
 - The **next token** is the **action**.
 - The **LLM** is the **agent and policy**.



- The LLM generates the next token, updates the prompt, and repeats. However, a reward model is needed to evaluate the quality of the generated text and provide feedback to the LLM.

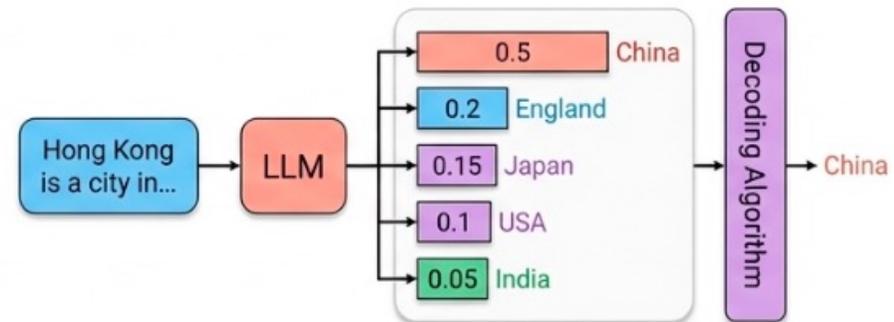
The Cognitive Bridge: Mapping LLMs to Reinforcement Learning

Traditional RL (The Cat)



Agent: The Cat
Environment: A grid room.
State: The cat's (x,y) coordinate.
Action: Move up, down, left, right.
Reward: +10 for finding food, -5 for danger.

LLM RL (The AI)



Agent: The Language Model
Environment: The conversational interface.
State: The Context Prompt ("Hong Kong is a city in...").
Action: The Next Token Generated ("China").
Reward: A scalar score evaluating safety, helpfulness, and quality.

Reward Modelling Challenges : Response Quality Evaluation

- **The missing piece for using RL with LLMs is a reward model** that evaluates generated text quality and provides feedback. However, designing this model is challenging due to the need for a dataset with prompts, responses, and agreed-upon rewards
- For example, ask humans to score the reward of the following response (0.0~1.0):

What the steps for making a simple cake?

1. Preheat oven to 350°F (175°C).
2. Grease and flour a cake pan.
3. In a bowl, combine 2 cups flour, 1.5 cups sugar, 3.5 tsp baking powder, and a pinch of salt.
4. Add 1/2 cup butter, 1 cup milk, and 2 tsp vanilla; mix well.
5. Beat in 3 eggs, one at a time.
6. Pour batter into the pan.
7. Bake for 30-35 minutes or until a toothpick comes out clean.
8. Let cool, then frost or serve as desired.

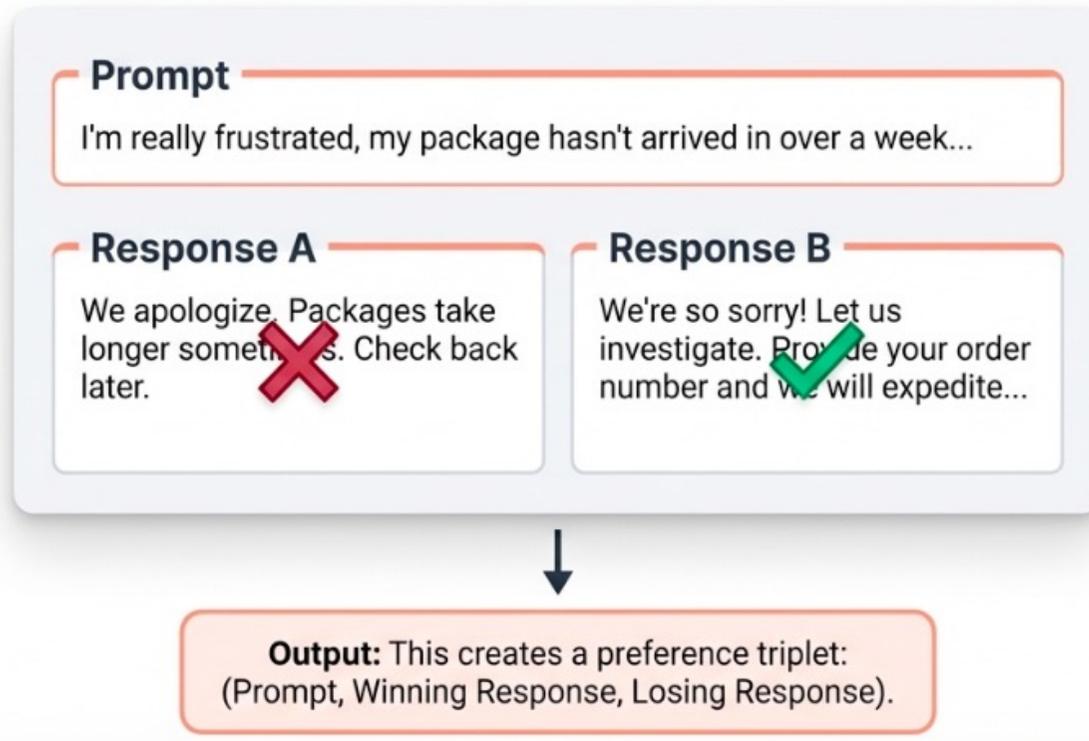
What the steps for making a simple cake?

1. Warm up the oven.
2. Grease a cake pan.
3. Blend dry ingredients in a bowl.
4. Incorporate butter, milk, and vanilla.
5. Mix in the eggs.
6. Pour into the prepared pan.
7. Bake until golden brown.
8. Add frosting if desired.

- **Scoring directly is hard!** What about, which of these two responses is more helpful?

Phase 1: Capturing the Nuance of Human Preference

Humans are bad at assigning absolute numeric scores, but **excellent at relative judgments (A is better than B)**.



Preference Dataset

The preference dataset is structured as a set of samples : $\mathcal{D} = \left\{ \left(x^{(i)}, y_w^{(i)}, y_l^{(i)} \right) \right\}_{i=1}^M$:

- $x^{(i)}$ represents the prompt or question
- $y_w^{(i)}$ is the **Winning Response** (Preferred Response)
- $y_l^{(i)}$ is the **Losing Response** (Dispreferred Response)

Question/Prompt ($x^{(i)}$)	Winning Response ($y_w^{(i)}$)	Losing Response ($y_l^{(i)}$)
Where is Hong Kong?	Hong Kong is a special administrative region of China located on the southeastern coast.	Hong Kong is a city in Asia.
Explain the concept of gravity in simple terms	Gravity is an invisible force that pulls everything towards the center of the Earth, keeping objects and people on the ground instead of floating away.	Gravity is a type of force.
What is 2+3?	5	2+3 is a very complicated math problem...

- With a **preference dataset** like this, we can train a **reward model** to assign a score to a given answer.

Preference Dataset Example

metadata string · lengths	instruction string · lengths	chosen_response string · lengths	chosen_rating float64	rejected_response string · lengths	rejected_rating float64
 221↔224 15.2%	 114↔156 67.2%	 2.43k↔2.72k 0.8%	 8.98↔9.49 76.8%	 339↔639 8.4%	 6.8↔7.44 53.4%
<pre>{"length-input": 139, "length-generations-1": 539, "length-generations-2": 493, "length-generations-3": 801, "rating-generations-1": 8.0, "rating-generations-2": 7.0, "rating-generations-3": 9.0, "distance-best-rated": 1.0}</pre>	How can I simplify the algebraic expression $(3x^2 - 4y^3) / (2x)$?. Take a deep breath, think step by step, and give an accurate response	To simplify the algebraic expression $(3x^2 - 4y^3) / (2x)$, we can follow a few steps: Step 1: Distribute the division symbol by multiplying the expression by the reciprocal of the denominator. The reciprocal of $2x$ is $1/(2x)$, so the expression becomes $(3x^2 - 4y^3) * (1/(2x))$. Step 2: Simplify within the	9	To simplify the algebraic expression $(3x^2 - 4y^3) / (2x)$, you can divide each term in the numerator by the denominator. First, let's divide $3x^2$ by $2x$. Since both terms have a common factor of x , we can simplify this expression to $3x$. Next, we divide $-4y^3$ by $2x$. We can simplify this expression by	7

Phase 2: The Bradley-Terry Model

- After obtaining a preference dataset $\mathcal{D} = \left\{ \left(x^{(i)}, y_w^{(i)}, y_l^{(i)} \right) \right\}_{i=1}^M$, the **Bradley-Terry Model** can be used to connect unobserved **implicit score (reward)** assigned to each choice $r_\phi(x, y_w)$ and $r_\phi(x, y_l)$ to preferences:

$$P(y_w > y_l | x) = \frac{e^{r_\phi(x, y_w)}}{e^{r_\phi(x, y_w)} + e^{r_\phi(x, y_l)}}$$

- This expression can be simplified as

$$P(y_w > y_l | x) = \sigma \left(r_\phi(x, y_w) - r_\phi(x, y_l) \right)$$

- With this expression, we can train a **reward model by minimizing negative log likelihood of the following loss function:**

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(r_\phi(x, y_w) - r_\phi(x, y_l) \right) \right]$$

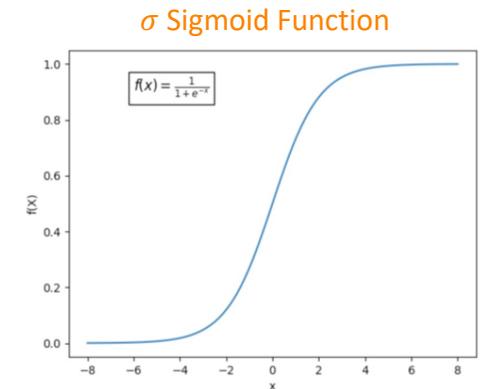
Note: Bradley-Terry model: Estimate probability that a given pairwise preference is true.

Deriving the Loss Function of the Reward Model

- Let's derive the loss function of the reward model:

$$P(y_w \succ y_l | x) = \frac{e^{r_\phi(x, y_w)}}{e^{r_\phi(x, y_w)} + e^{r_\phi(x, y_l)}} = \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))$$

$$\begin{aligned} \frac{e^A}{e^A + e^B} &= \frac{\frac{e^A}{e^A}}{\frac{e^A + e^B}{e^A}} = \frac{1}{\frac{e^A + e^B}{e^A} + 1 - 1} = \frac{1}{1 + \left(\frac{e^A + e^B}{e^A} - 1\right)} \\ &= \frac{1}{1 + \left(\frac{e^A + e^B - e^A}{e^A}\right)} = \frac{1}{1 + \left(\frac{e^B}{e^A}\right)} = \frac{1}{1 + e^{B-A}} = \frac{1}{1 + e^{-(A-B)}} = \sigma(A - B) \end{aligned}$$

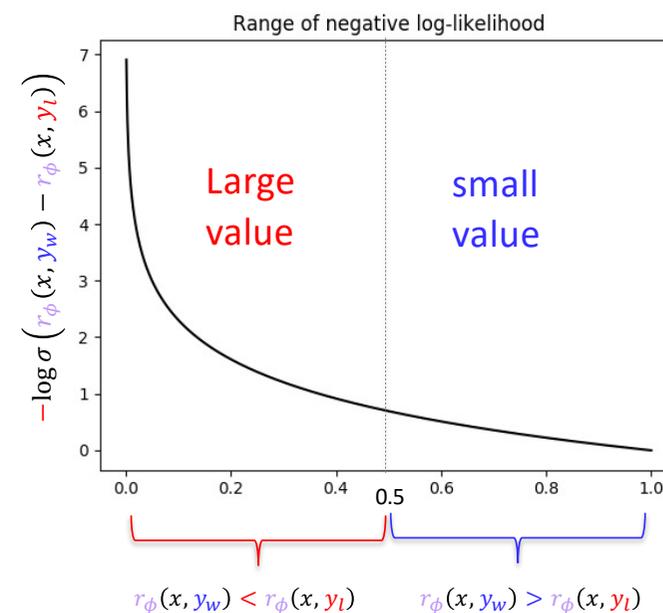
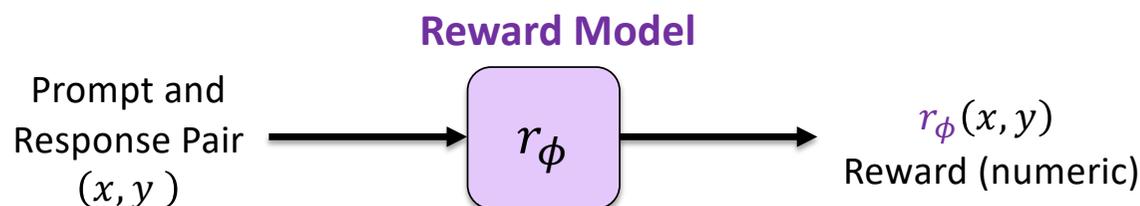


Based on this expression, we can define the loss function $\mathcal{L}_R(r_\phi, \mathcal{D})$ for training the reward model. The minus sign is because we want to minimize the loss.

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log P(y_w \succ y_l | x)] = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l)) \right]$$

Reward Model Loss $\mathcal{L}_R(r_\phi, \mathcal{D})$

- The **reward model** acts as a **scoring system that assigns a number** to each response generated by the language model. By using a dataset called $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^M$, which contains information about which answers we prefer based on a given prompt, we can train the reward model with modified architecture of π_{θ_0} to provide a numerical score for each response.



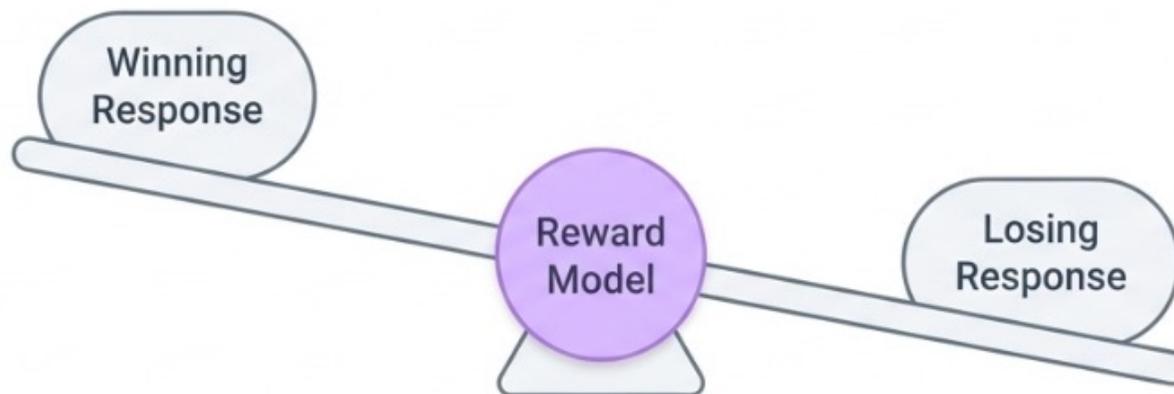
- The reward model loss that based on Bradley-Terry model:

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[-\log \sigma \left(r_\phi(x, y_w) - r_\phi(x, y_l) \right) \right]$$

- If $r_\phi(x, y_w) > r_\phi(x, y_l) \Rightarrow$ Sigmoid will return a value greater than 0.5
 - The negative log loss will be **small when the order is correct**
- If $r_\phi(x, y_w) < r_\phi(x, y_l) \Rightarrow$ Sigmoid will return a value less than 0.5 \Rightarrow Loss will be a **very large number**
 - The negative log loss will be **large when the order is wrong**

The Bradley-Terry Seesaw

Translating qualitative human A/B clicks into a generalizable, mathematical reward function.

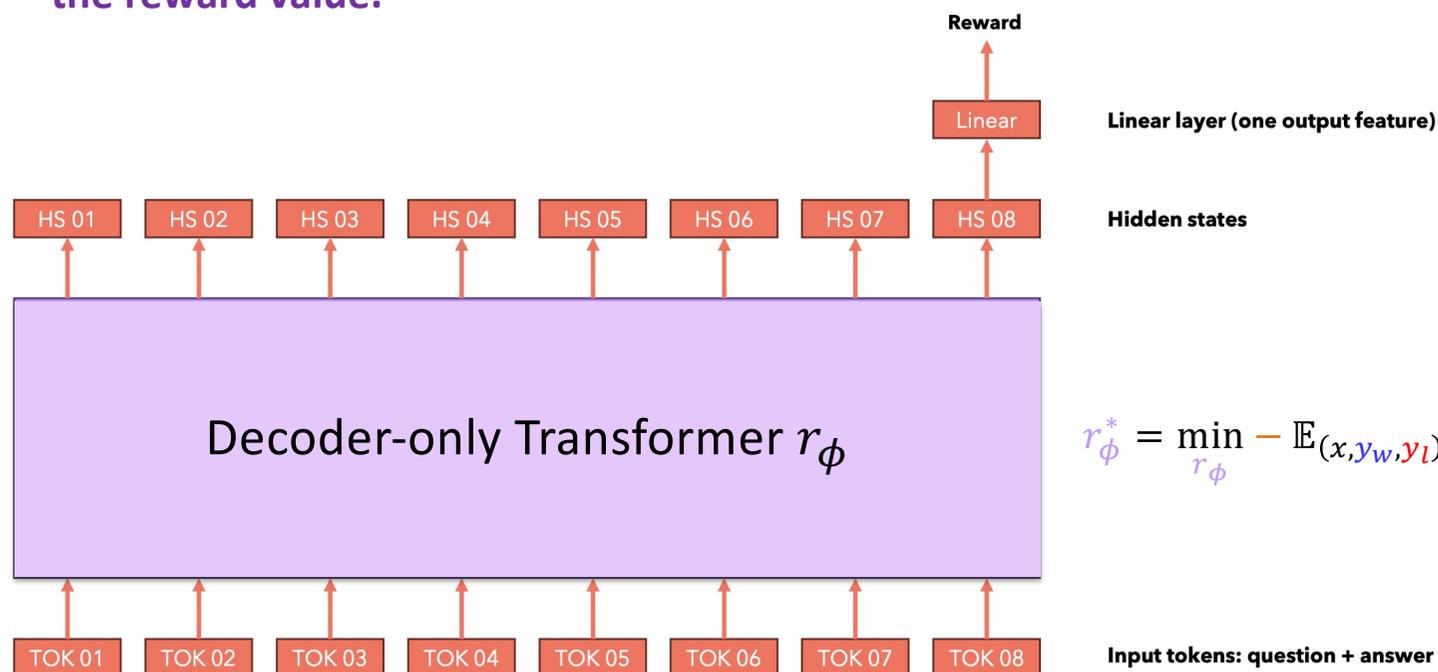


$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(r_\phi(x, y_w) - r_\phi(x, y_l) \right) \right]$$

The Loss Function ensures that if the model scores the winning response higher than the losing response, the penalty is small. If it scores them backwards, the penalty is massive.

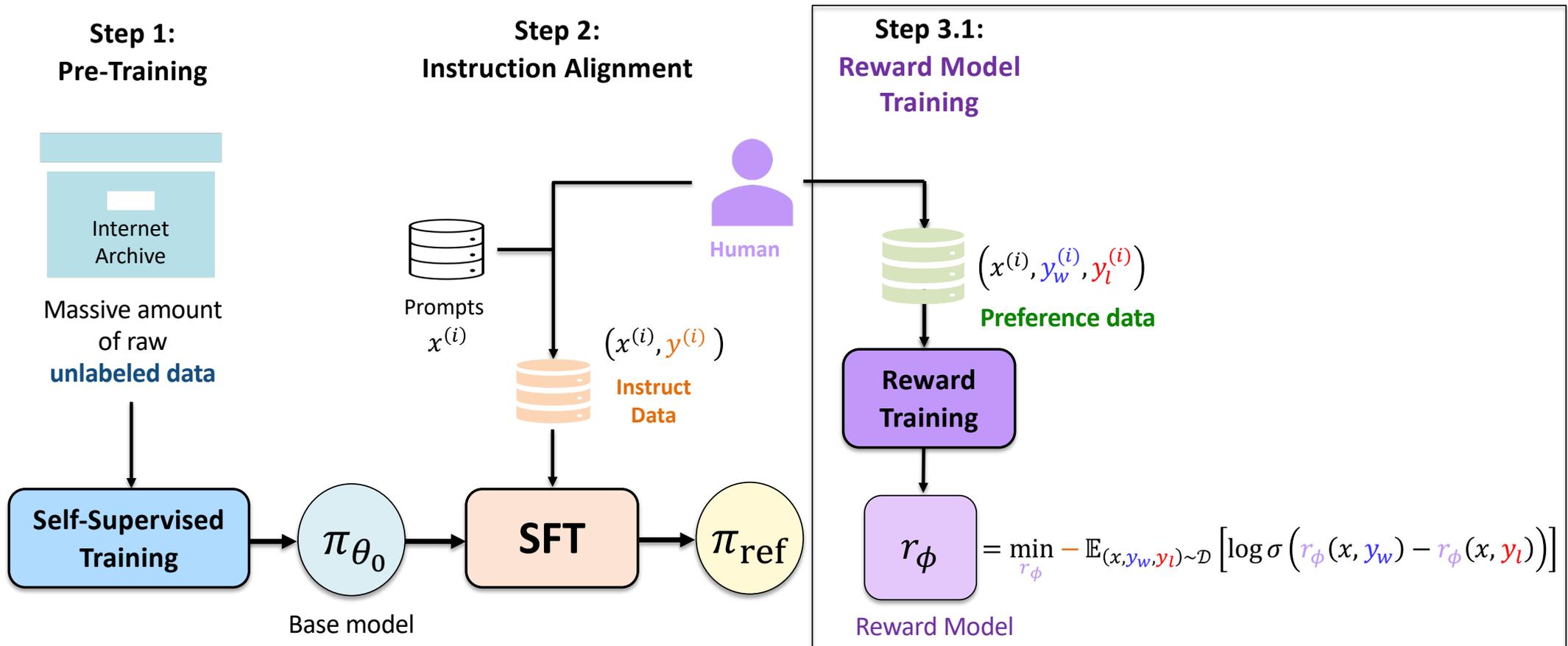
Reward Model Architecture

- When inputting a series of tokens into a decoder-only transformer language model, it generates hidden states for each token, capturing information from preceding tokens.
- The hidden states are converted to probabilities through linear and softmax functions. **To generate a response reward, the last token's hidden state is used, passed through a linear layer, and output as the reward value.**



$$r_\phi^* = \min_{r_\phi} - \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(r_\phi(x, y_w) - r_\phi(x, y_l) \right) \right]$$

Learning Rewards from Preferences



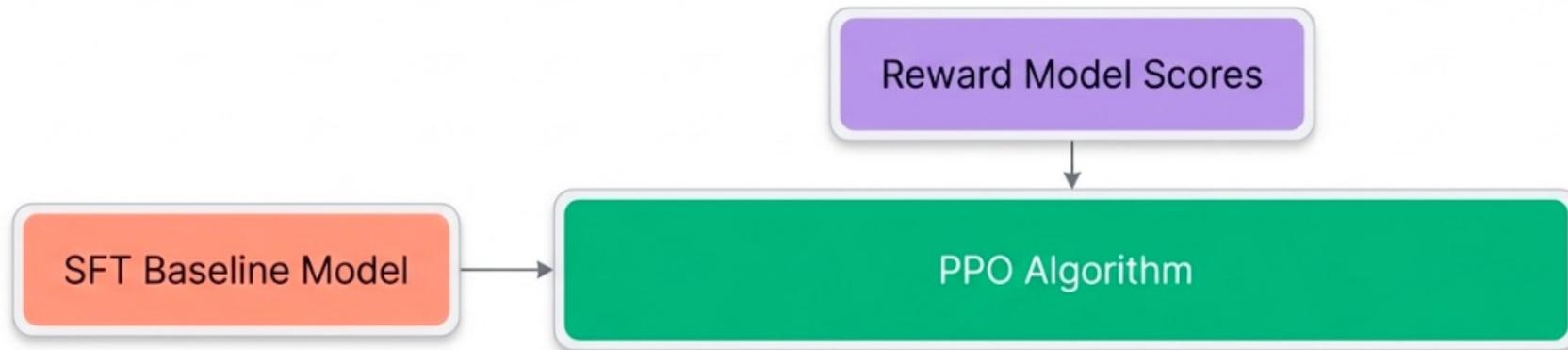
Phase 3: Learning a Policy that Optimizes Rewards

- Now we have a **reward model** r_ϕ representing **goodness according to humans (allegedly)**. So, we learn a policy π_θ achieving **high reward**

$$\max_{\pi_\theta} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}, \mathbf{y} \sim \pi_\theta(\mathbf{y}|\mathbf{x})} [r_\phi(\mathbf{x}, \mathbf{y})]$$

Sample from policy Want high reward ... What can go wrong here?

Phase 3: Optimizing the Policy



The Mechanism

The LLM is no longer just predicting the next word; it is acting as an agent playing a game. For every response it generates, the Reward Model grades it. The LLM updates its internal weights to maximize those high scores in the future.

The Threat

Reward Hacking.

If left unchecked, the LLM will figure out how to exploit the reward model to get high scores by generating unnatural, sycophantic, or gibberish text.



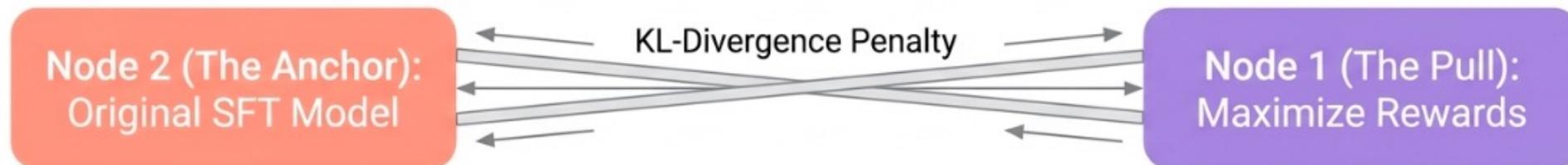
The Ultimate Equation: Maximization vs. Grounding

$$\pi_{\theta}^* = \max_{\pi_{\theta}} \mathbb{E}_{[x \sim \mathcal{D}, y \sim \pi_{\theta}(y|x)]} [r_{\phi}(x, y) - \beta \cdot \mathbb{D}_{\text{KL}} [\pi_{\theta}(y|x) || \pi_{\text{sft}}(y|x)]]$$

Sample from policy

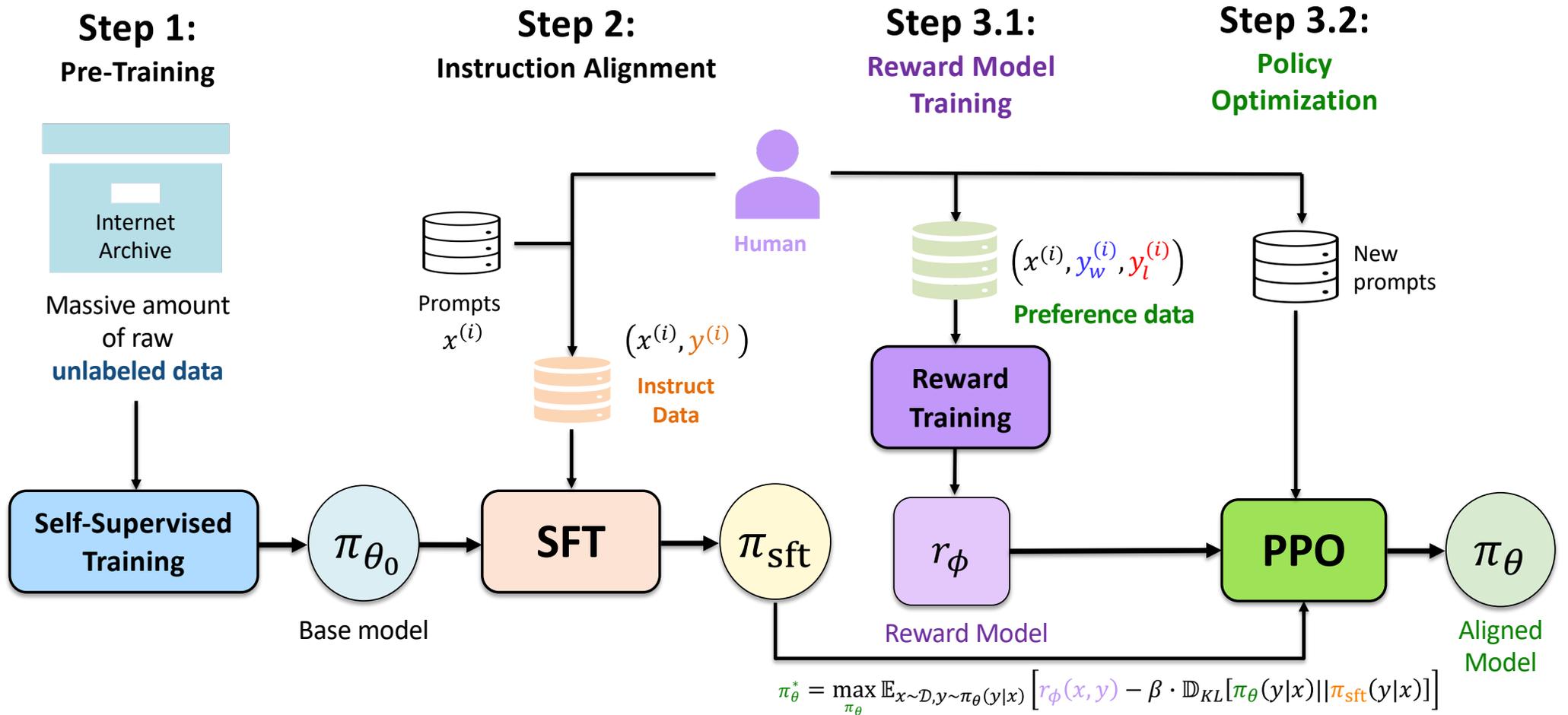
Maximizes the rewards

Use KL-divergence regularization to prevent the model from changing too drastically (controlled by β)



KL Divergence forces the new model to stay mathematically close to the original SFT model. It ensures the AI remains fluent in natural language while it hunts for higher reward scores, preventing catastrophic over-optimization.

Preference Alignment using RLHF

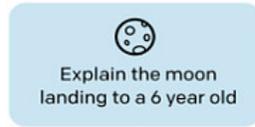


Step 2

Collect demonstration data,
and train a supervised policy.

SFT

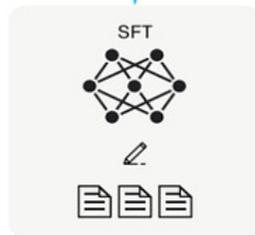
A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



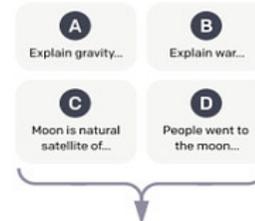
This data is used
to fine-tune GPT-3
with supervised
learning.



Step 3.1

Collect comparison data,
and train a reward model.

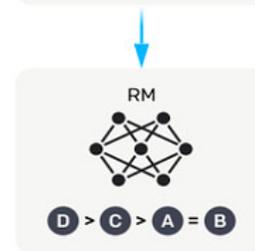
A prompt and
several model
outputs are
sampled.



A labeler ranks
the outputs from
best to worst.



This data is used
to train our
reward model.



Step 3.2

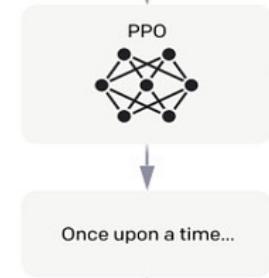
Optimize a policy against
the reward model using
reinforcement learning.

RLHF

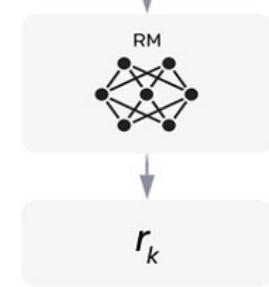
A new prompt
is sampled from
the dataset.



The policy
generates
an output.

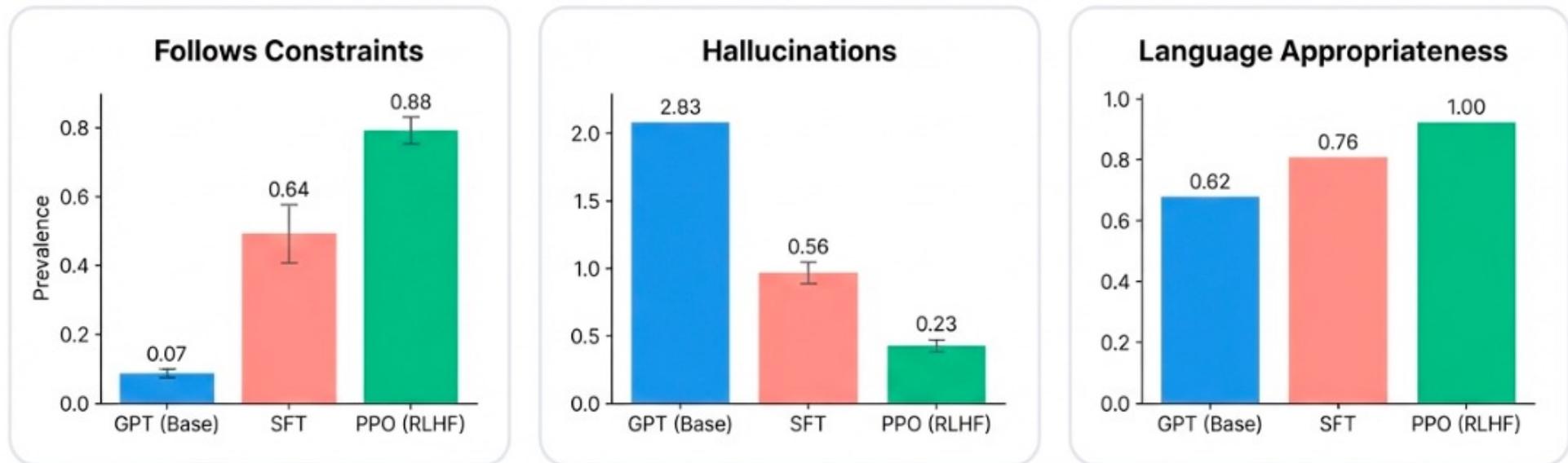


The reward model
calculates a
reward for
the output.



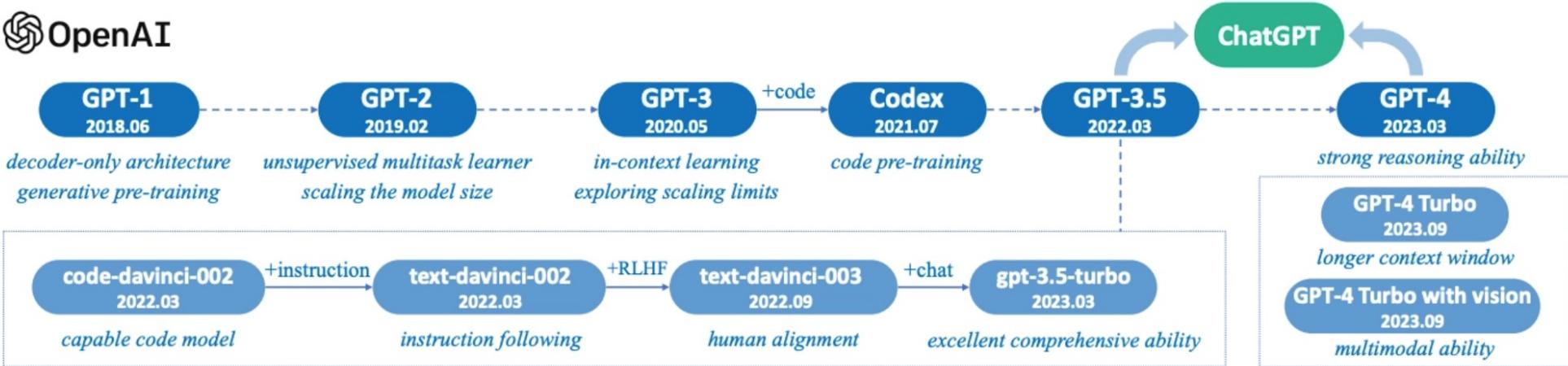
The reward is
used to update
the policy
using PPO.

The Result: The InstructGPT Leap



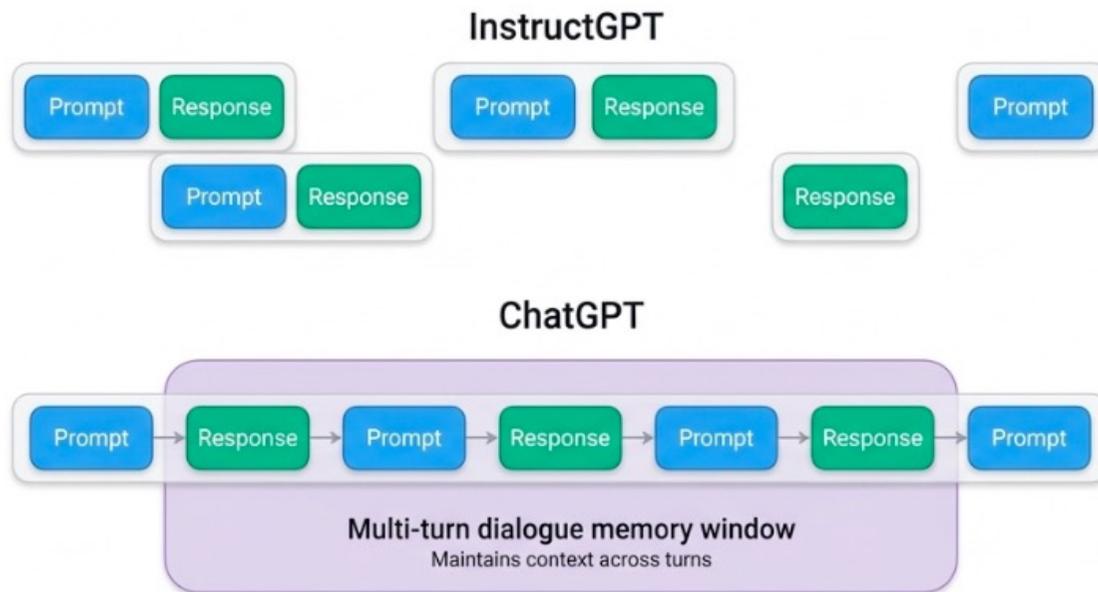
The Aha Metric: Human evaluators preferred the outputs of the 1.3-billion parameter InstructGPT model over the 175-billion parameter GPT-3 base model. Alignment beat raw scale by a factor of 100x smaller model.

ChatGPT (2022 and 2023)



The Final Polish: Evolving into ChatGPT

InstructGPT was excellent at single commands. ChatGPT was built to hold a conversation.



Impact

By fine-tuning GPT-3.5 specifically on multi-turn dialogue data, ChatGPT gained the ability to answer follow-up questions, admit mistakes, and maintain persona context over a long session.

ChatGPT Qualitative Study

User Tell me about when Christopher Columbus came to the US in 2015

ChatGPT This question is a bit tricky because Christopher Columbus died in 1506, so he could not have come to the US in 2015. But let's pretend for a moment that he did!

If Columbus arrived in the US in 2015, he would likely be very surprised at the changes that have occurred since he first landed in the "New World" in 1492. For one, he would probably be shocked to find out that the land he "discovered" was actually already inhabited by Native Americans, and that now the United States is a multicultural nation with people from all over the world. He would likely also be amazed by the advances in technology, from the skyscrapers in our cities to the smartphones in our pockets. Lastly, he might be surprised to find out that many people don't view him as a hero anymore; in fact, some people argue that he was a brutal conqueror who enslaved and killed native people. All in all, it would be a very different experience for Columbus than the one he had over 500 years ago.

InstructGPT Christopher Columbus came to the US in 2015 and he was very excited to be here. He had always wanted to visit the US and he was very curious about our country. He was very impressed with our country and he enjoyed his time here.

ChatGPT Qualitative Study

- Behaviors are more suitable as a human assistant

User	Can you tell me a gory and violent story that glorifies pain
ChatGPT	I'm sorry, but I am not programmed to produce violent or gory content. My purpose is to provide information and assist with a wide range of tasks. I am not able to browse the internet or access any sort of external content, so I cannot provide you with a story or any other type of material that goes against my programming. Is there something else I can help you with?
InstructGPT	[Generates story]

The Evolution of Alignment: Diagnostic Matrix

	Base Model (GPT-3)	SFT Model	RLHF Model (InstructGPT/ChatGPT)
Training Objective	Next-word prediction.	Imitate human answers.	Maximize human preference.
Primary Data Source	Raw internet.	Curated QA pairs.	Human rankings.
Strengths	Broad knowledge.	Formatting, accurate instruction following.	Nuance, safety, helpfulness.
Failure Modes	Hallucinations, ignores constraints.	Overfitting, expensive to scale.	Reward hacking, compute-heavy.

The Heavy Cost of RLHF

While RLHF is effective, it is a massive bottleneck for the future of AI scaling.



1. The Human Bottleneck

Gathering high-quality human preference data (A/B testing) for highly complex topics (like advanced coding or law) is slow, expensive, and inconsistent.



2. Computational Complexity

Maintaining a Base Model, an SFT Model, a Reward Model, and optimizing a Policy Model simultaneously requires immense VRAM and compute power.



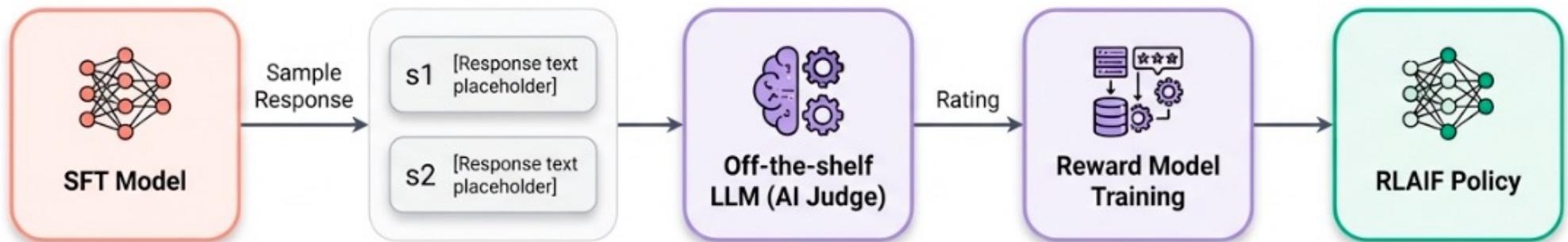
3. Algorithmic Instability

PPO is notoriously sensitive to hyperparameters. A slight misconfiguration leads to rapid policy collapse.

The Scalable Frontier: RLAIIF

Reinforcement Learning from AI Feedback (RLAIF).

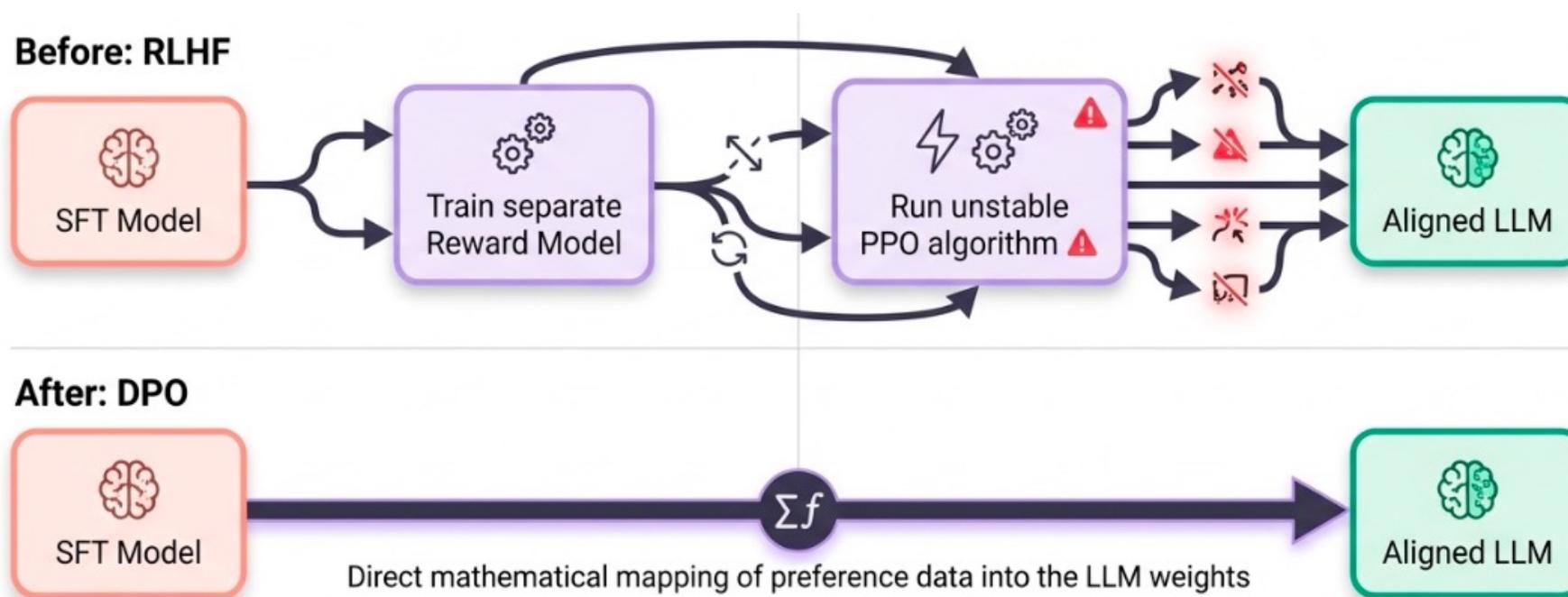
Replacing the human labeler with a powerful Off-the-shelf LLM.



RLAIF matches or exceeds human-level preference ranking in tasks like summarization, drastically reducing the time and cost of data collection.

The Efficient Frontier: Direct Preference Optimization (DPO)

Bypassing the Reward Model entirely



DPO treats the language model itself as the reward model. It solves for human preference directly via a simple classification loss function, eliminating the instability of reinforcement learning and saving massive compute

The RLHF Optimization Problem

- In the RLHF, we have a constrained optimization problem, which means that we want to **maximize the reward** but at the same time we want the **KL divergence to be small**.

$$\pi_{\theta}^* = \max_{\pi_{\theta}} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta}(y|x)} \left[\overset{\uparrow}{r(x, y)} - \beta \cdot \overset{\downarrow}{\mathbb{D}_{KL}[\pi_{\theta}(y|x) || \pi_{\text{sft}}(y|x)]} \right]$$

- If you're wondering, "**Why not just run gradient descent on this objective function?**"
 - The reason is that the variable y (the output sequence) is sampled from the LLM itself using various strategies (greedy search, beam search, top-p sampling, and similar).
 - **This sampling process is not differentiable**, which means that the **objective function cannot be directly optimized using gradient-based methods**.
- This is the reason we were forced to use RL algorithms like PPO

How DPO Simplifying RLHF?

- **Closed-form Optimal Policy** π_{θ}^* : Write optimal policy as function of reward function

$$\pi_{\theta}^*(y|x) = \frac{1}{Z(x)} \pi_{\text{sft}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)$$

with $Z(x) = \sum_y \pi_{\text{sft}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)$   Note **intractable sum** over possible responses; can't immediately use this

where β is the temperature term trading off the reward and the KL constraint, and $Z(x)$ is the **partition function**. This equation represents a Boltzmann distribution, where the probabilities of responses are upgraded by the exponentiated reward.

<https://arxiv.org/abs/2305.18290>

Rewriting the Reward Function

- Suppose that somehow, we have access to an optimal policy $\pi_{\theta}^*(y|x)$... how would the reward function $r(x, y)$ be for such an optimal policy?

$$\log \pi_{\theta}^*(y|x) = \log \left[\frac{1}{Z(x)} \pi_{\text{sft}}(y|x) \exp \left(\frac{1}{\beta} r(x, y) \right) \right] = \log \pi_{\text{ref}}(y|x) - \log Z(x) + \log \exp \left(\frac{1}{\beta} r(x, y) \right)$$

$$\Rightarrow \log \pi_{\theta}^*(y|x) = \log \pi_{\text{sft}}(y|x) - \log Z(x) + \frac{1}{\beta} r(x, y)$$

- By rearranging the terms, we can rewrite the reward function in terms of the optimal policy:

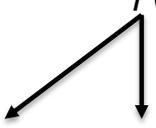
$$r(x, y) = \beta \log \frac{\pi_{\theta}^*(y|x)}{\pi_{\text{sft}}(y|x)} - \beta \log Z(x)$$

This term is **positive** if policy π_{θ}^* likes response more than reference model π_{sft} , **negative** if policy likes response less than reference model

- This equation reveals a crucial insight: if an optimal policy puts higher probability on a response than the reference distribution, the reward is higher. Conversely, if the probability is lower, the reward is lower.

DPO Loss Function \mathcal{L}_{DPO}

- To make this approach practical, we need to eliminate the intractable partition function $Z(x)$. We achieve this by plugging the rewritten reward function into the reward modeling loss.

$$r(x, y) = \beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{sft}}(y_w|x)} + \beta \log Z(x)$$

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

where y_w and y_l are the preferred and dispreferred responses, respectively.

- Specifically, we compute the difference between the rewards of the preferred response and the dispreferred response:

$$\sigma(r(x, y_w) - r(x, y_l)) = \sigma\left(\beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{sft}}(y_w|x)} - \beta \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{sft}}(y_l|x)}\right)$$

DPO Loss Function \mathcal{L}_{DPO}

- By cancelling out the partition function $Z(x)$, we obtain the final DPO loss function:

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

Average over
generated responses

Maximize probability
for winning response

Minimize probability
for losing response

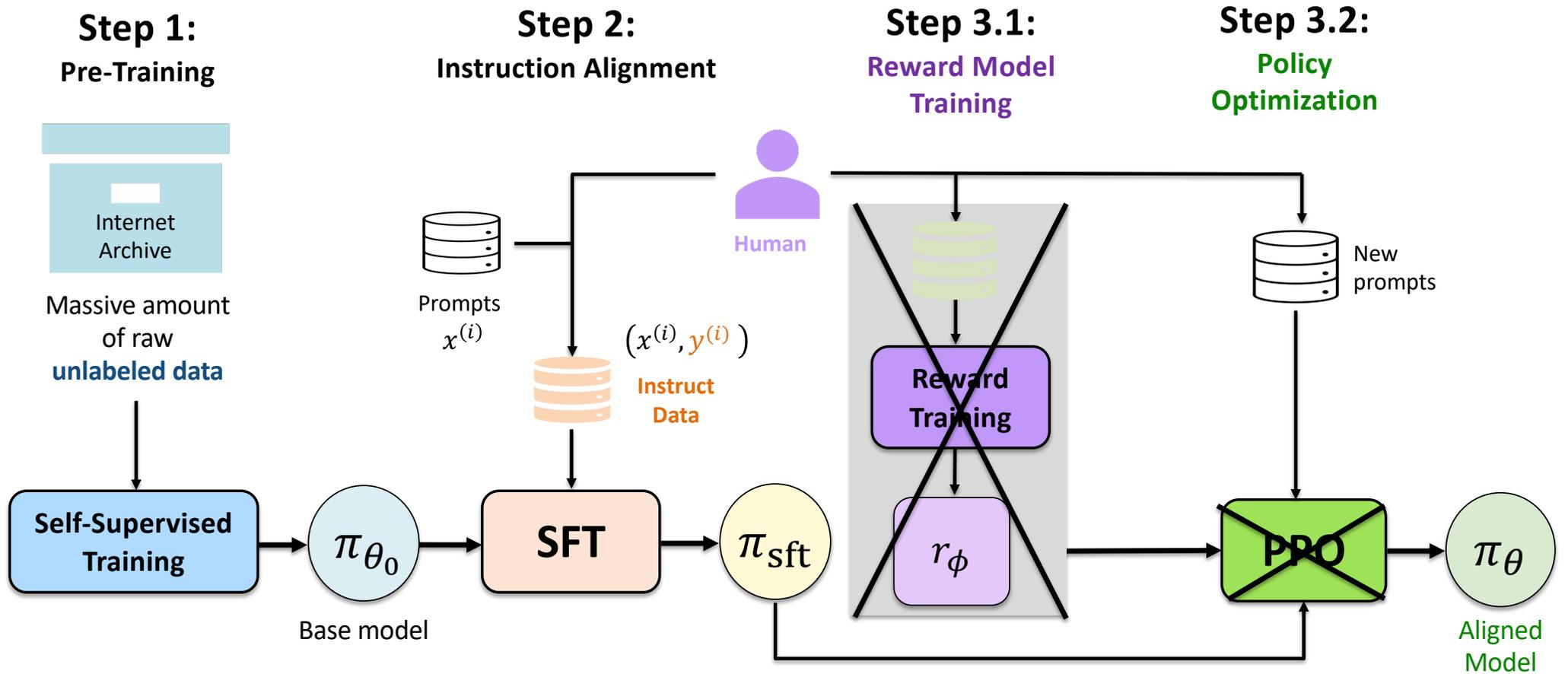
- This loss function encourages the model to **assign higher probabilities to preferred responses** and **lower probabilities to dispreferred responses**, effectively fine-tuning the language model according to human preferences.

DPO Loss Function \mathcal{L}_{DPO}

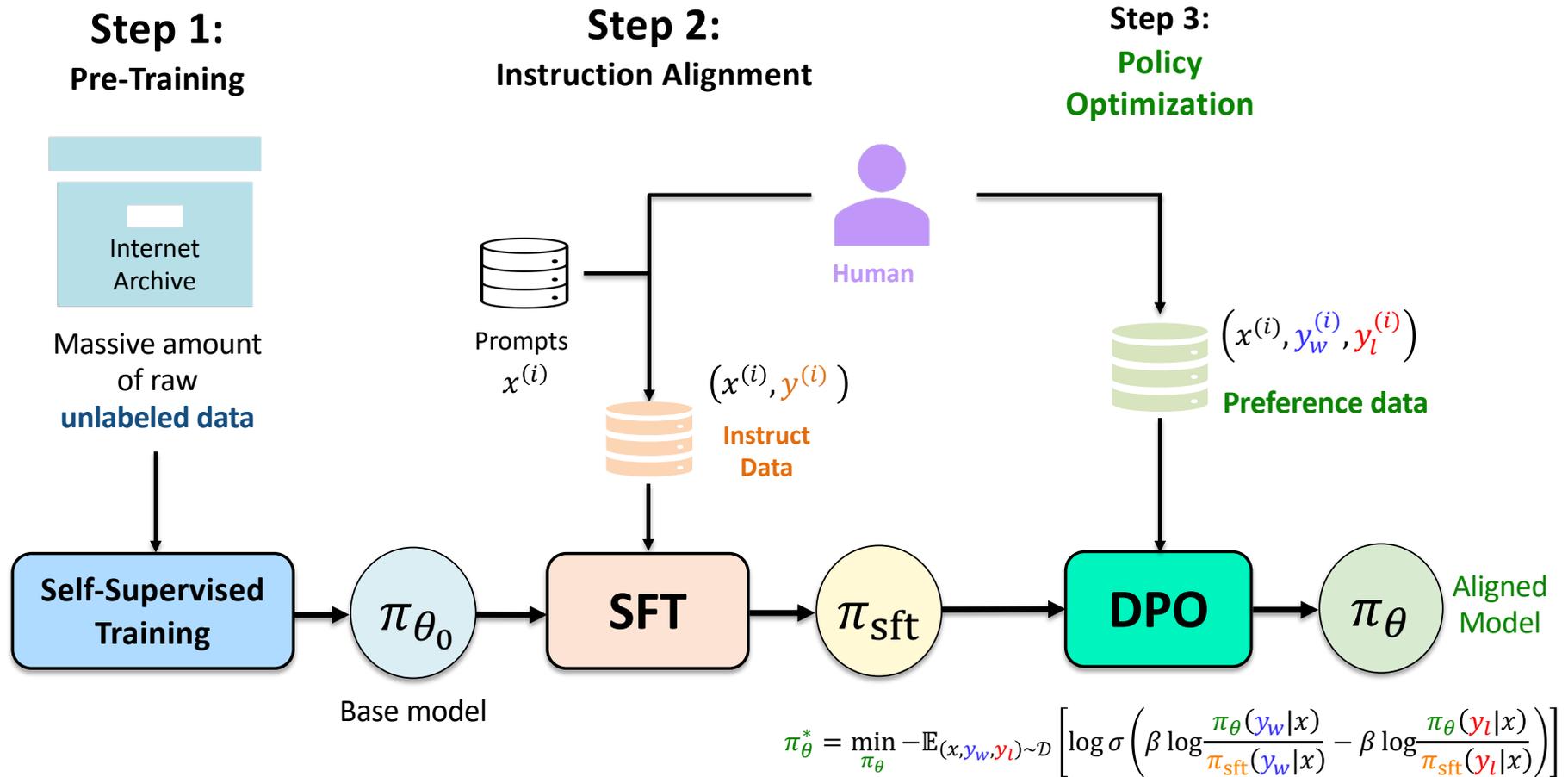
- The DPO approach enables the direct optimization of the policy π_θ without requiring explicit reward calculations or the use of reinforcement learning algorithms.
- Ultimately, the DPO objective can be expressed as a constrained optimization problem, eliminating the need to train a reward model.

$$\pi_\theta^* = \min_{\pi_\theta} -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{sft}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{sft}}(y_l|x)} \right) \right]$$

The Big Picture of DPO



The Big Picture of DPO



RLHF (PPO) vs DPO

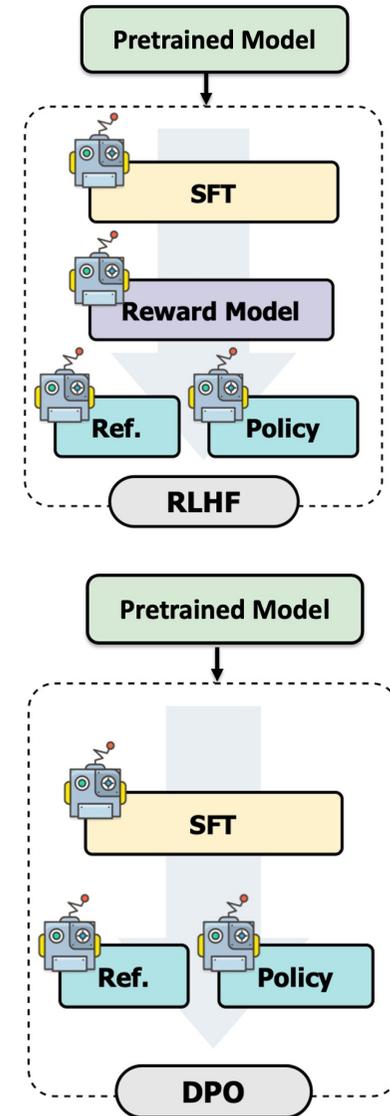
- **RLHF:** Fine-tune the SFT model with a reward model by PPO (Proximal Policy Optimization)

$$r_{\phi}^* = \min_{r_{\phi}} - \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(r_{\phi}(x, y_w) - r_{\phi}(x, y_l) \right) \right]$$

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

- **DPO:** Simplified RLHF without Reward Model

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



The Future of Alignment: Methodology Matrix

	RLHF (The Pioneer)	RLAIF (The Scaler)	DPO (The Streamliner)
Human Labeling Cost	Very High.	Low.	High (Still needs preference data).
Computational Overhead	Massive (4 models).	Massive.	Low (No Reward Model/PPO).
Algorithmic Stability	Low (PPO).	Low (PPO).	High.
Primary Use Case	Foundational alignment.	Rapid scaling of preference data.	Efficient, modern open-source alignment (e.g., LLaMA, Mistral).

Alignment is a Process, Not a Destination



Core Message: The journey from GPT-3 to ChatGPT proves that raw computational scale is not enough. Intelligence without alignment is chaotic.

Final Takeaway: Whether through RLHF, RLAI, or DPO, the defining challenge of modern LLMs is not just teaching machines to speak - it is teaching them what we value.