

LLM Alignment

Slido: <https://app.sli.do/event/3EdyaErNv3XWwj4UvwwBtv>

Yu Meng

University of Virginia

yumeng5@virginia.edu

Nov 17, 2025

Overview of Course Contents

- Week 1: Logistics & Overview
- Week 2: N-gram Language Models
- Week 3: Word Senses, Semantics & Classic Word Representations
- Week 4: Word Embeddings
- Week 5: Sequence Modeling & Recurrent Neural Networks (RNNs)
- Week 6: Language Modeling with Transformers
- Week 8: Transformer and Pretraining
- Week 9: Large Language Models (LLMs) & In-context Learning
- Week 10: Knowledge in LLMs and Retrieval-Augmented Generation (RAG)
- Week 11: LLM Reasoning
- Week 12: Reinforcement Learning for Post-Training LLMs
- **Week 13: LLM Alignment & Agents**
- Week 15 (after Thanksgiving): Project Presentations

(Recap) Reasoning: Overview

- **Reasoning** (rough definition): perform deductive, inductive, commonsense, or logical reasoning via generating or analyzing text with language models
- Deductive reasoning: draw specific conclusions from general principles or premises
 - E.g.: “All humans are mortal” + “Socrates is a human” => “Socrates is mortal”
- Inductive reasoning: make generalizations based on specific observations
 - E.g.: “The sun has risen in the east every day” => “The sun will rise in the east tomorrow”
- Commonsense reasoning: rely on world knowledge or commonsense understanding to make predictions or answer questions
 - E.g.: “If I drop a ball, what will happen?” => “It will fall”
- Mathematical/logical reasoning: follow specific rules or procedures to arrive at a correct answer
 - E.g.: “If 3 apples cost \$6, how much do 5 apples cost?” => “\$10”

(Recap) Standard Prompting vs. CoT Prompting

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

(Recap) CoT Can Be Triggered Zero-shot

Just add “Let’s think step by step” at the beginning of the answer

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let’s think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

(Recap) CoT Demo

No-CoT prompt: How many 'r' letters are there in the following word: strawberry? Answer without reasoning steps

How many 'r' letters are there in the following word: strawberry? Answer without reasoning steps

🌀 llama-3.3-70b-instruct

2

Wrong result

CoT prompt: How many 'r' letters are there in the following word: strawberry? Let's think step by step

How many 'r' letters are there in the following word: strawberry? Let's think step by step

🌀 llama-3.3-70b-instruct

To find out how many 'r' letters are in the word "strawberry", let's break it down step by step:

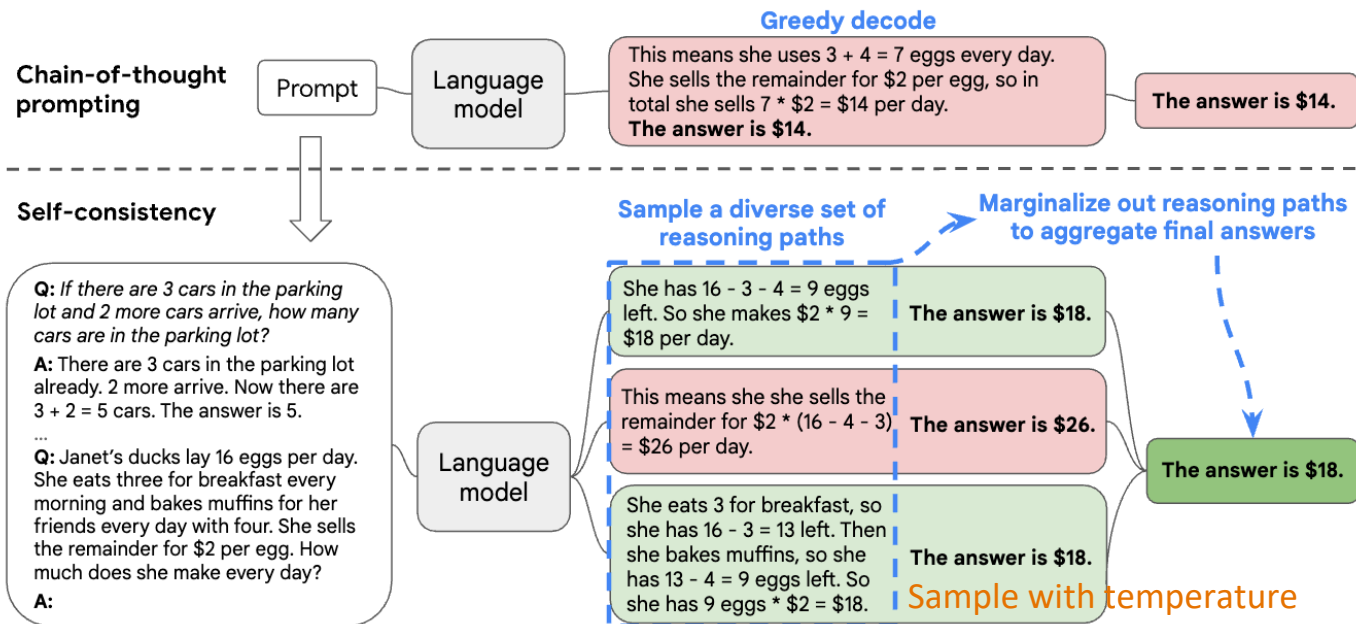
1. **Start with the word:** strawberry
2. **Separate the word into its individual letters:** s-t-r-a-w-b-e-r-r-y
3. **Identify the 'r' letters:** We see 'r' in the following positions:
 - The third letter is 'r'.
 - The eighth letter is 'r'.
 - The ninth letter is also 'r'.
4. **Count the 'r' letters:** There are 3 'r' letters in the word "strawberry".

So, there are **3** 'r' letters in the word "strawberry".

Correct result

(Recap) Self-consistency CoT

Intuition: if multiple different ways of thinking lead to the same answer, one has greater confidence that the final answer is correct



(Recap) Grade School Math (GSM8K)

8.5K high quality grade school math problems created by human problem writers

Problem: Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?

Solution: Beth bakes 4 2 dozen batches of cookies for a total of $4 \times 2 = 8$ dozen cookies
There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of $12 \times 8 = 96$ cookies
She splits the 96 cookies equally amongst 16 people so they each eat $96/16 = 6$ cookies

Final Answer: 6

Problem: Mrs. Lim milks her cows twice a day. Yesterday morning, she got 68 gallons of milk and in the evening, she got 82 gallons. This morning, she got 18 gallons fewer than she had yesterday morning. After selling some gallons of milk in the afternoon, Mrs. Lim has only 24 gallons left. How much was her revenue for the milk if each gallon costs \$3.50?

Mrs. Lim got 68 gallons - 18 gallons = 50 gallons this morning.
So she was able to get a total of 68 gallons + 82 gallons + 50 gallons = 200 gallons.
She was able to sell 200 gallons - 24 gallons = 176 gallons.
Thus, her total revenue for the milk is $\$3.50/\text{gallon} \times 176 \text{ gallons} = \616 .

Final Answer: 616

Problem: Tina buys 3 12-packs of soda for a party. Including Tina, 6 people are at the party. Half of the people at the party have 3 sodas each, 2 of the people have 4, and 1 person has 5. How many sodas are left over when the party is over?

Solution: Tina buys 3 12-packs of soda, for $3 \times 12 = 36$ sodas
6 people attend the party, so half of them is $6/2 = 3$ people
Each of those people drinks 3 sodas, so they drink $3 \times 3 = 9$ sodas
Two people drink 4 sodas, which means they drink $2 \times 4 = 8$ sodas
With one person drinking 5, that brings the total drank to $5 + 9 + 8 = 25$ sodas
As Tina started off with 36 sodas, that means there are $36 - 25 = 11$ sodas left

Final Answer: 11



(Recap) MATH

12.5K challenging competition mathematics problems

Problem: Suppose a and b are positive real numbers with $a > b$ and $ab = 8$. Find the minimum value of $\frac{a^2+b^2}{a-b}$.

Ground truth solution: We can write $\frac{a^2+b^2}{a-b} = \frac{a^2+b^2-2ab+16}{a-b} = \frac{(a-b)^2+16}{a-b} = a - b + \frac{16}{a-b}$. By AM-GM, $a - b + \frac{16}{a-b} \geq 2\sqrt{(a-b) \cdot \frac{16}{a-b}} = 8$. Equality occurs when $a - b = 4$ and $ab = 8$. We can solve these equations to find $a = 2\sqrt{3} + 2$ and $b = 2\sqrt{3} - 2$. Thus, the minimum value is $\boxed{8}$.

Problem: Right $\triangle ABC$ has legs measuring 8 cm and 15 cm. The triangle is rotated about one of its legs. What is the number of cubic centimeters in the maximum possible volume of the resulting solid? Express your answer in terms of π .

Ground truth solution: If the triangle is rotated about the shorter leg, then the radius is the longer leg and the height is the shorter leg, and the volume is $\frac{1}{3} \cdot (15^2\pi)(8) = 600\pi$ cubic centimeters. If the triangle is rotated about the longer leg, then the radius is the shorter leg and the height is the longer leg, and the volume is $\frac{1}{3}(8^2\pi)(15)$, which is $\frac{8}{15}$ of the volume we found earlier. So, the maximum possible volume is $\boxed{600\pi}$ cubic centimeters.

(Recap) AI2 Reasoning Challenge (ARC)

~8K natural science questions on commonsense knowledge/reasoning

Reasoning Type	Example
Question logic	Which item below is not made from a material grown in nature? (A) a cotton shirt (B) a wooden chair (C) a plastic spoon (D) a grass basket
Linguistic Matching	Which of the following best describes a mineral? (A) the main nutrient in all foods (B) a type of grain found in cereals (C) a natural substance that makes up rocks (D) the decomposed plant matter found in soil
Multihop Reasoning	Which property of a mineral can be determined just by looking at it? (A) luster (B) mass (C) weight (D) hardness
Comparison	Compared to the Sun, a red star most likely has a greater (A) volume. (B) rate of rotation. (C) surface temperature. (D) number of orbiting planets
Algebraic	If a heterozygous smooth pea plant (Ss) is crossed with a homozygous smooth pea plant (SS), which are the possible genotypes the offspring could have? (A) only SS (B) only Ss (C) Ss or SS (D) ss or SS
Hypothetical / Counterfactual	If the Sun were larger, what would most likely also have to be true for Earth to sustain life? (A) Earth would have to be further from the Sun. (B) Earth would have to be closer to the Sun. (C) Earth would have to be smaller. (D) Earth would have to be larger.
Explanation / Meta-reasoning	Why can steam be used to cook food? (A) Steam does work on objects. (B) Steam is a form of water. (C) Steam can transfer heat to cooler objects. (D) Steam is able to move through small spaces.

Paper: <https://arxiv.org/pdf/1803.05457>

(Recap) BIG-Bench Hard (BBH)

23 challenging tasks covering a wide range of reasoning (e.g. arithmetic, logical, spatial...)

Model Input (Navigate)

<Few shot chain-of-thought exemplars omitted for brevity>

Q: If you follow these instructions, do you return to the starting point? Turn left. Turn right. Take 5 steps. Take 4 steps. Turn around. Take 9 steps.

Options:

- Yes
- No

A: Let's think step by step.

Model Output

We start at the origin (0, 0), facing the positive y-axis.

- (1) Turn left: (0, 0), facing the negative x-axis.
 - (2) Turn right: (0, 0), facing the positive y-axis.
 - (3) Take 5 steps: (0, 5), facing the positive y-axis.
 - (4) Take 4 steps: (0, 9), facing the positive y-axis.
 - (5) Turn around: (0, 9), facing the negative y-axis.
 - (6) Take 9 steps: (0, 0), facing the negative y-axis.
- Since (0, 0) is (0, 0), we are indeed where we started. So the answer is **Yes**. ✓

Model Input (Word Sorting)

<Few shot chain-of-thought exemplars omitted for brevity>

Q: Sort the following words alphabetically: List: burley bela arapaho bacteria bock

A: Let's think step by step.

Model Output

The first letter: "burley": "b" (2). "bela": "b" (2). "arapaho": "a" (1). "bacteria": "b" (2). "bock": "b" (2).
We now have: (1) "arapaho" < (2) ["burley" ? "bela" ? "bacteria" ? "bock"]. Now let's sort this subpart ["burley" ? "bela" ? "bacteria" ? "bock"] by looking at their second letters.
The second letter: "burley": "u" (21). "bela": "e" (5). "bacteria": "a" (1). "bock": "o" (15). We now have: (1) "bacteria" < (5) "bela" < (15) "bock" < (21) "burley". Hence, we have "arapaho" < ["bacteria" < "bela" < "bock" < "burley"]. So the answer is **arapaho bacteria bela bock burley**. ✓

(Recap) American Invitational Mathematics Examination

High school math competition, where each answer is an integer from 000 to 999

Problem

Let A be the set of positive integer divisors of 2025. Let B be a randomly selected subset of A . The probability that B is a nonempty set with the property that the least common multiple of its elements is 2025 is $\frac{m}{n}$, where m and n are relatively prime positive integers. Find $m + n$.

Solution 1

We split into different conditions:

Note that the numbers in the set need to have a least common multiple of 2025, so we need to ensure that the set has at least 1 number that is a multiple of 3^4 and a number that is a multiple of 5^2 .

Multiples of 3^4 : 81, 405, 2025

Multiples of 5^2 : 25, 75, 225, 675, 2025

If the set B contains 2025, then all of the rest 14 factors is no longer important. The valid cases are 2^{14} .

If the set B doesn't contain 2025, but contains 405, we just need another multiple of 5^2 . It could be 1 of them, 2 of them, 3 of them, or 4 of them, which has $2^4 - 1 = 15$ cases. Excluding 2025, 405, 25, 75, 225, 675, the rest 9 numbers could appear or not appear. Therefore, this case has a valid case of $15 \cdot 2^9$.

If set B doesn't contain 2025 nor 405, it must contain 81. It also needs to contain at least 1 of the multiples from 5^2 , where it would be $15 \cdot 2^8$.

The total valid cases are $2^{14} + 15 \cdot (2^9 + 2^8)$, and the total cases are 2^{15} .

The answer is $\frac{2^8 \cdot (64 + 30 + 15)}{2^8 \cdot 2^7} = \frac{109}{128}$.

Desired answer: $109 + 128 = \boxed{237}$.

(Recap) Humanity's Last Exam (HLE)

2,500 challenging questions across over a hundred subjects (created by experts)

Mathematics

Question:

The set of natural transformations between two functors $F, G: \mathcal{C} \rightarrow \mathcal{D}$ can be expressed as the end

$$\mathrm{Nat}(F, G) \cong \int_A \mathrm{Hom}_{\mathcal{D}}(F(A), G(A)).$$

Define set of natural cotransformations from F to G to be the coend

$$\mathrm{CoNat}(F, G) \cong \int^A \mathrm{Hom}_{\mathcal{D}}(F(A), G(A)).$$

Let:

- $F = \mathbf{B}_*(\Sigma_4)_{*/}$ be the under ∞ -category of the nerve of the delooping of the symmetric group Σ_4 on 4 letters under the unique 0-simplex $*$ of $\mathbf{B}_*\Sigma_4$.
- $G = \mathbf{B}_*(\Sigma_7)_{*/}$ be the under ∞ -category nerve of the delooping of the symmetric group Σ_7 on 7 letters under the unique 0-simplex $*$ of $\mathbf{B}_*\Sigma_7$.

How many natural cotransformations are there between F and G ?

✉ Emily S
🏛 University of São Paulo

Computer Science

Question:

Let G be a graph. An edge-indicator of G is a function $a: \{0, 1\} \rightarrow V(G)$ such that $\{a(0), a(1)\} \in E(G)$.

Consider the following Markov Chain $M = M(G)$:

The statespace of M is the set of all edge-indicators of G , and the transitions are defined as follows:

Assume $M_t = a$.

1. pick $b \in \{0, 1\}$ u.a.r.
2. pick $v \in N(a(1 - b))$ u.a.r. (here $N(v)$ denotes the open neighbourhood of v)
3. set $a'(b) = v$ and $a'(1 - b) = a(1 - b)$
4. Set $M_{t+1} = a'$

We call a class of graphs $\mathcal{G}$ well-behaved if, for each $G \in \mathcal{G}$ the Markov chain $M(G)$ converges to a unique stationary distribution, and the unique stationary distribution is the uniform distribution.

Which of the following graph classes is well-behaved?

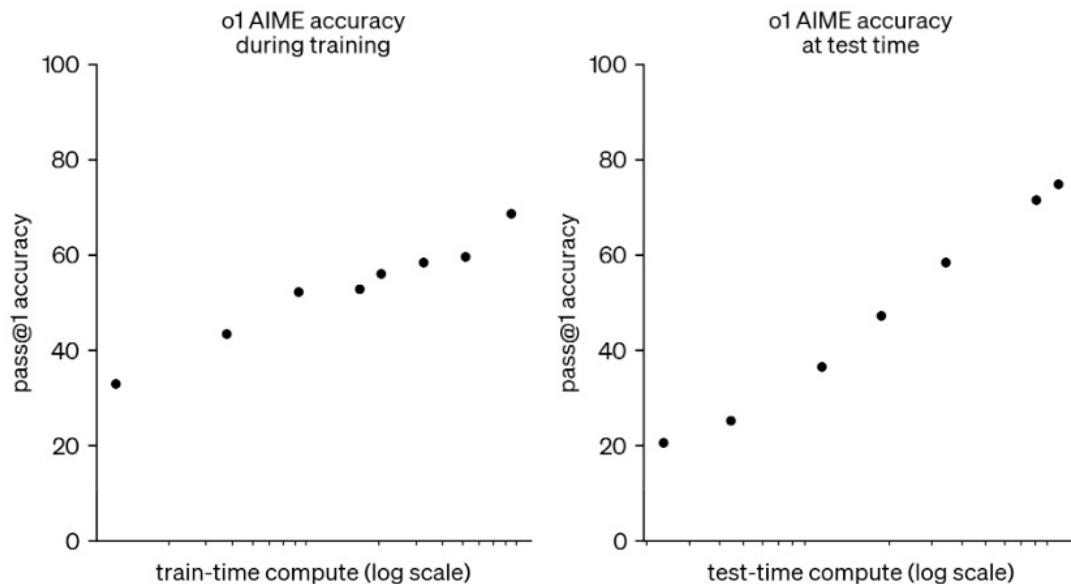
Answer Choices:

- A. The class of all non-bipartite regular graphs
- B. The class of all connected cubic graphs
- C. The class of all connected graphs
- D. The class of all connected non-bipartite graphs
- E. The class of all connected bipartite graphs.

✉ Marc R
🏛 Queen Mary University of London

(Recap) OpenAI's o1 (2024/09)

OpenAI released o1 (a reasoning model) in 2024, with remarkable performance on AIME



(Recap) DeepSeek-R1 (2025/01)

- Open-source reproduction of OpenAI's o1
- During training, DeepSeek-R1 naturally learns to solve reasoning tasks with more thinking time

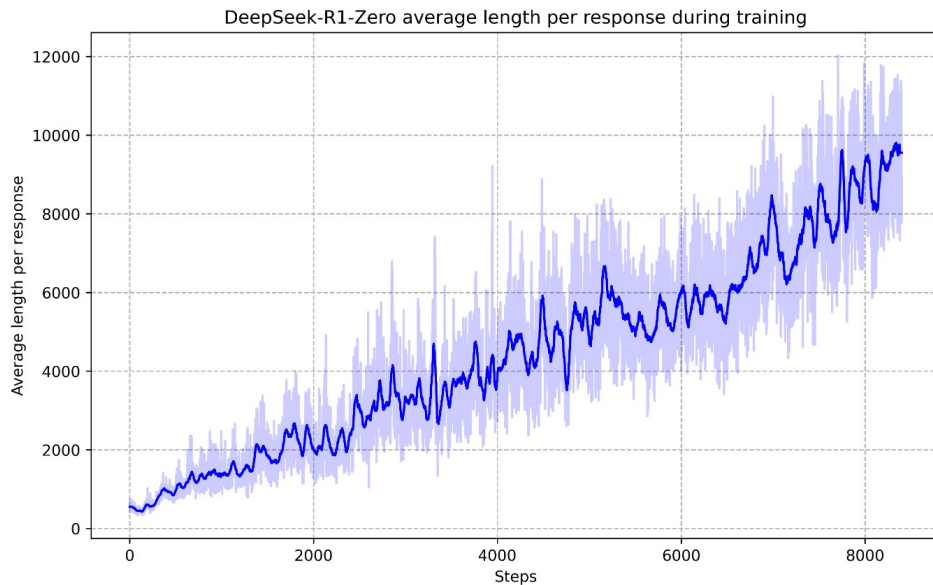


Figure source: <https://arxiv.org/pdf/2501.12948>

(Recap) The “Aha Moment” of DeepSeek-R1

The model is not explicitly taught on how to solve a problem, but rather autonomously develops advanced problem-solving strategies (e.g., reevaluating its initial approach)

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\dots$

$$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

$\dots$

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be $\dots$

We started with the equation:

$$\sqrt{a - \sqrt{a + x}} = x$$

First, let's square both sides:

$$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$$

Next, I could square both sides again, treating the equation: $\dots$

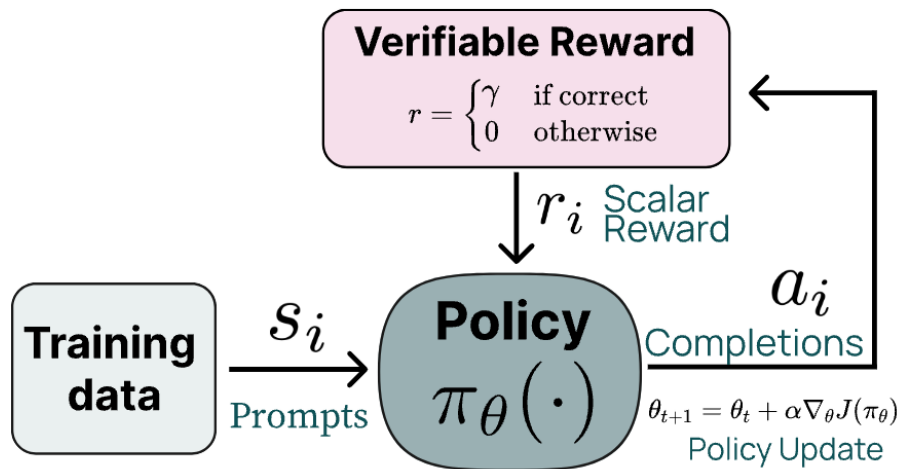
$\dots$

(Recap) Scaling Test-Time Compute

- **Test-time Scaling:** increase the compute used at inference time (e.g., generating more tokens) to generate a higher-quality answer
- **Self-consistency** (majority voting):
 - Generate multiple responses to the same prompt
 - Use majority voting to select the best answer
- **Long CoT:**
 - Longer reasoning chains (think more thoroughly) -----> **OpenAI o1 & DeepSeek-R1**
- **Beam Search/Tree Search:**
 - Explore multiple reasoning paths simultaneously
 - Backtrack when hitting dead ends
 - Prune bad branches
- **Iterative Refinement:**
 - Generate initial response and then improve it iteratively

(Recap) Reinforcement Learning with Verifiable Rewards

- The (major) post-training recipe of OpenAI's o1 & DeepSeek-R1
- **RLVR:**
 - Fine-tune the policy model (LLM) using **reinforcement learning**
 - The LLM receives a reward when its generated responses are verifiably correct



(Recap) Why RLVR?

- **Data scalability**
 - Supervised learning requires human annotators to create the full correct responses
 - RLVR only requires automatic verifiers to grade the responses
- **Imitation vs. optimization**
 - Supervised learning forces models to imitate human reasoning steps, which may be suboptimal
 - RLVR optimizes directly for correct final answers, allowing the model to discover its own efficient reasoning paths
- **Distribution mismatch**
 - Supervised learning (where data are created by humans) might cause a discrepancy from the model's own distribution
 - RLVR learns from the model's own generated sequences, matching the inference distribution



(Recap) RLVR Setup

- Generate responses using the LLM (the policy model)
- Assign rewards to the generated responses
- Maximize the expected reward

$$\max_{\theta} \mathbb{E}_{y \sim \underbrace{p_{\theta}(\cdot|x)}_{\text{LLM output probability}}} [\underbrace{r_{\text{RLVR}}(x, y)}_{\text{reward of RLVR (binary)}]}$$

Optimization with Reinforcement Learning (RL)

- Why reinforcement learning:
 - No supervised data available
 - Encourage the model to explore new possibilities (generations) guided by the rewards
- Optimization: policy gradient methods
 - Optimize the policy (LLM) by adjusting the parameters in the direction that increases expected rewards
- REINFORCE (simplest policy gradient method):

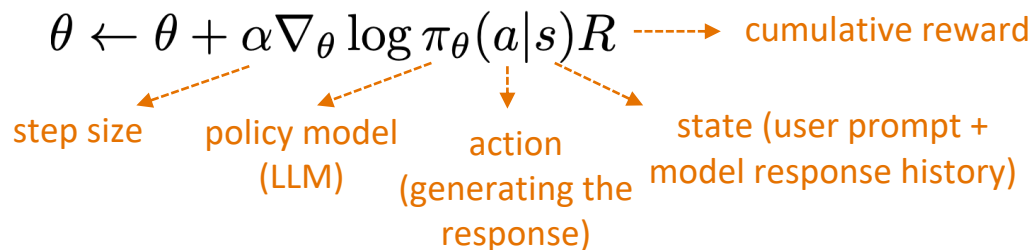
$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \log \pi_{\theta}(a|s) R$$


Diagram illustrating the components of the REINFORCE equation:

- α : step size
- $\pi_{\theta}(a|s)$: policy model (LLM)
- a : action (generating the response)
- s : state (user prompt + model response history)
- R : cumulative reward

Overview: Proximal Policy Optimization (PPO)

- A more advanced policy gradient method that improves stability and efficiency
- Clipped mechanism: PPO uses a clipped surrogate objective to ensure that policy updates are not too large, which helps maintain stability
- Advantage estimation: PPO uses Generalized Advantage Estimation (GAE) to reduce variance in the advantage estimates with a critic model, improving learning efficiency

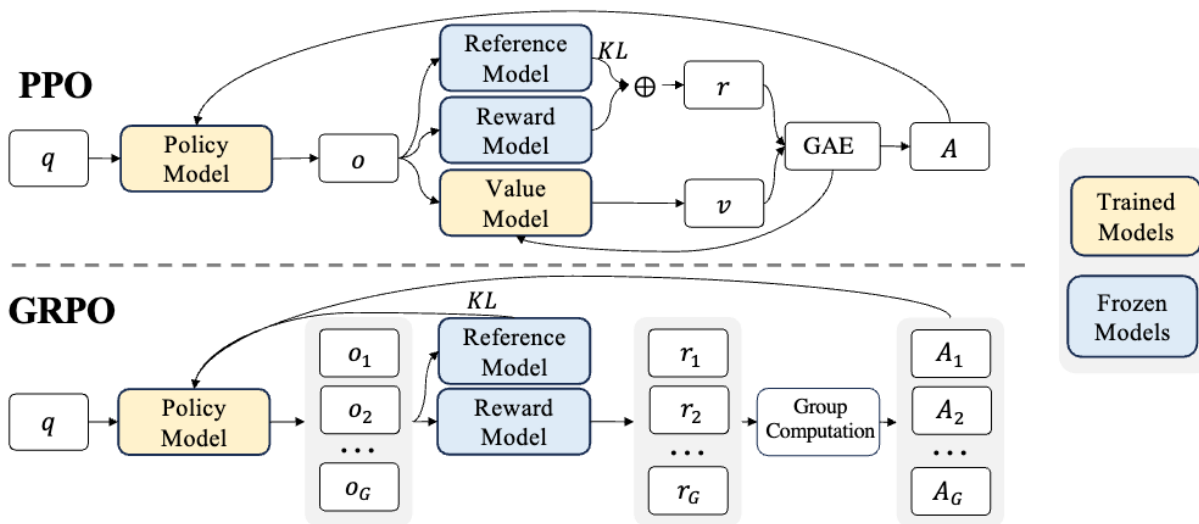
Proximal Policy Optimization Algorithms

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov
OpenAI

{joschu, filip, prafulla, alec, oleg}@openai.com

Overview: Group Relative Policy Optimization (GRPO)

- A variant of PPO that foregoes the critic model
- Advantage estimation: for each question, GRPO samples a group of outputs and use the comparative rewards to estimate advantage



Further Reading on RLVR

- [ProRL: Prolonged Reinforcement Learning Expands Reasoning Boundaries in Large Language Models](#) [Liu et al., 2025]
- [DAPO: An Open-Source LLM Reinforcement Learning System at Scale](#) [Yu et al., 2025]
- [The Surprising Effectiveness of Negative Reinforcement in LLM Reasoning](#) [Zhu et al., 2025]
- [Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?](#) [Yue et al., 2025]

Agenda

- Introduction to LLM Alignment
- Instruction Tuning
- Reinforcement Learning from Human Feedback (RLHF)

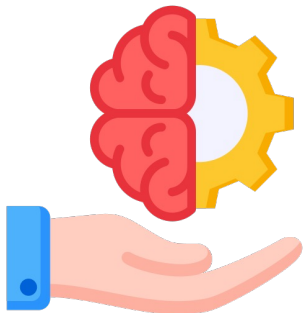
The Evolution of GPT Models: ChatGPT

- GPT-1: decoder-only Transformer pretraining
- GPT-2: language model pretraining is multi-task learning
- GPT-3: scaling up & in-context learning
- ChatGPT: language model alignment



Overview: Language Model Alignment

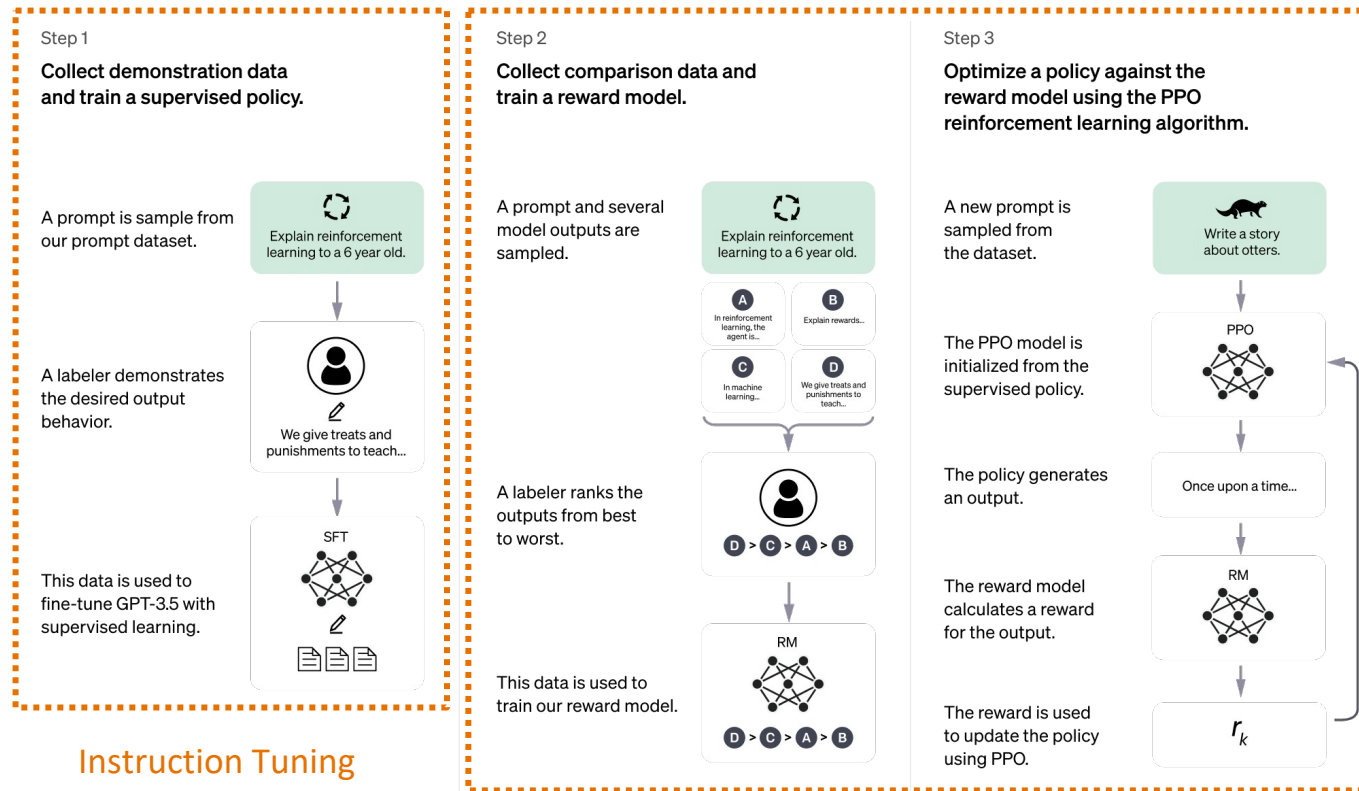
- Ensure language models behaviors are aligned with human values and intent for general tasks/applications
- “HHH” criteria (Askell et al. 2021):
 - **Helpful:** Efficiently perform the task requested by the user
 - **Honest:** Give accurate information & express uncertainty
 - **Harmless:** Avoid offensive/discriminatory/biased outputs



Language Model Alignment: Post-training

- Pretrained language models are **not** aligned
- Objective mismatch
 - Pretraining is to predict the next word in a sentence
 - Does not involve understanding human intent/values
- Training data bias
 - Text from the internet can contain biased, harmful, or misleading information
 - LMs don't distinguish between good and bad behavior in training data
- (Over-)generalization issues
 - LMs' generalization can lead to outputs that are inappropriate in specific contexts
 - Might not align with intended ethics/honesty standard

Language Model Alignment Techniques



Overview: Instruction Tuning

- Train an LM using a diverse set of tasks
 - Each task is framed as an **instruction** followed by an example of the desired output
 - The goal is to teach the model to follow specific instructions (human intent) effectively
- The resulting model can perform a variety of tasks **zero-shot** (w/o requiring in-context demonstrations)
- The instructions can also be in chat format – tuning an LM into a chatbot

🔗 meta-llama/Llama-3.2-1B

📄 Text Generation • Updated 8 days ago • 📄 1.05M • ⚡ • ❤️ 725

Pretrained (base) model

🔗 meta-llama/Llama-3.2-1B-Instruct

📄 Text Generation • Updated 8 days ago • 📄 1.31M • ⚡ • ❤️ 478

Instruction-tuned
(post-trained) model

Overview: RLHF

- Human feedback collection
 - Generate multiple responses using the model given the same prompt
 - Human evaluators rank responses of the model based on helpfulness/honesty/safety...
- Reward model training
 - A reward model is trained on human feedback data to predict the quality of responses
 - Higher reward = more preferred by human evaluators
- Policy optimization
 - Use reinforcement learning algorithms to further train the LM to maximize the reward predicted by the reward model
 - Encourage the model to produce outputs that align better with human preferences

Agenda

- Introduction to LLM Alignment
- **Instruction Tuning**
- Reinforcement Learning from Human Feedback (RLHF)

Instruction Tuning: Introduction

- **Setting:** fine-tune LLMs with task-specific instructions on diverse tasks
- **Goal:** enable LLM to better understand user prompts and generalize to a wide range of (unseen) tasks **zero-shot**

FINETUNED LANGUAGE MODELS ARE ZERO-SHOT LEARNERS

**Jason Wei*, Maarten Bosma*, Vincent Y. Zhao*, Kelvin Guu*, Adams Wei Yu,
Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le**
Google Research

Instruction Tuning: Method

- **Input:** task description
- **Output:** expected response or solution to the task
- Train LLMs to generate response tokens given prompts

$$\min_{\theta} -\log p_{\theta}(y|x)$$

Response

Prompt

Finetune on many tasks (“instruction-tuning”)

<u>Input (Commonsense Reasoning)</u>	<u>Input (Translation)</u>
Here is a goal: Get a cool sleep on summer days. How would you accomplish this goal? OPTIONS: -Keep stack of pillow cases in fridge. -Keep stack of pillow cases in oven.	Translate this sentence to Spanish: The new office building was built in less than three months.
Target keep stack of pillow cases in fridge	Target El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks

Coreference resolution tasks

...

Inference on unseen task type

Input (Natural Language Inference)

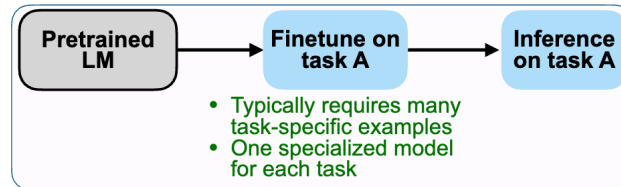
Premise: At my age you will probably have learnt one lesson.
Hypothesis: It's not certain how many lessons you'll learn by your thirties.
Does the premise entail the hypothesis?
OPTIONS:
-yes -it is not possible to tell -no

FLAN Response
It is not possible to tell

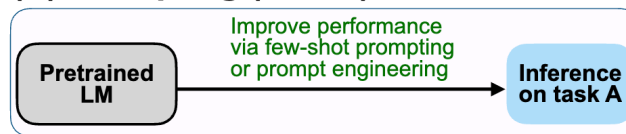
Instruction Tuning vs. Other Paradigms

- Task-specific fine-tuning does not enable generalization across multiple tasks
- In-context learning requires few-shot demonstrations
- Instruction tuning enables zero-shot cross task generalization

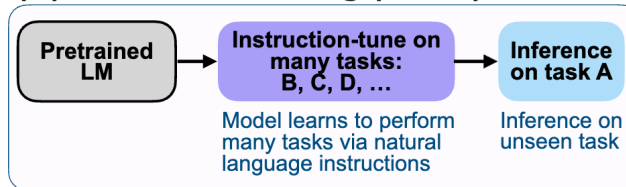
(A) Pretrain–finetune (BERT, T5)



(B) Prompting (GPT-3)



(C) Instruction tuning (FLAN)

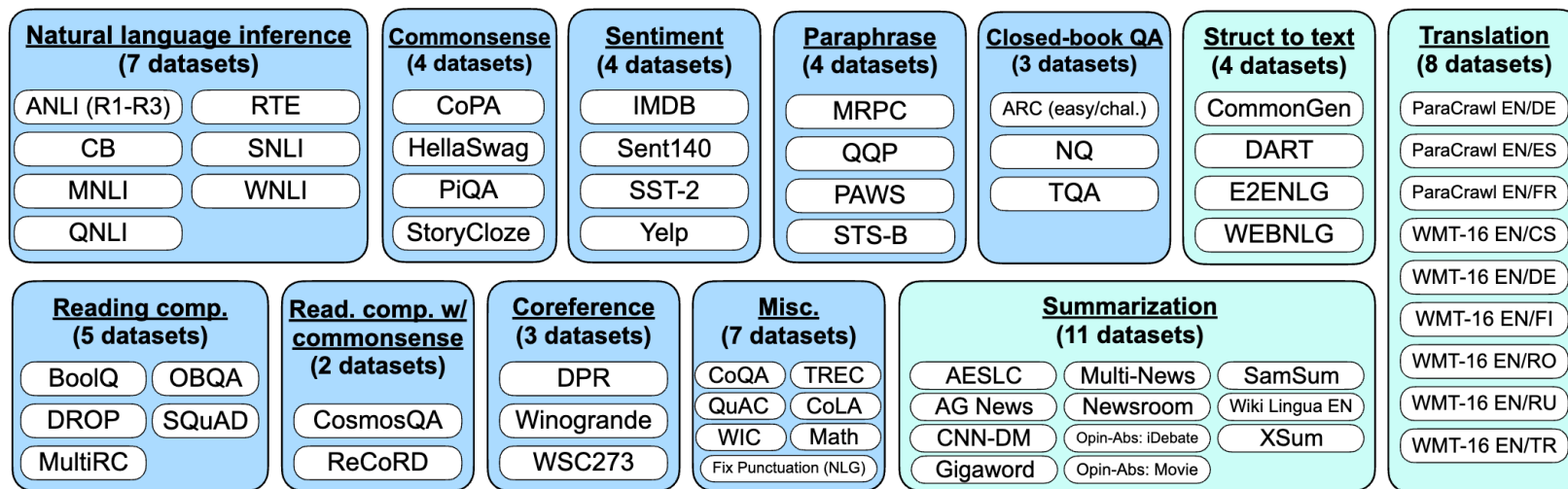


Instruction Tuning vs. Pretraining

- Both instruction tuning and pretraining are **multi-task** learning paradigms
- Supervision
 - Pretraining: self-supervised learning (raw data w/o human annotation)
 - Instruction tuning: supervised learning (human annotated responses)
- Task format
 - Pretraining: tasks are implicit (predicting next tokens)
 - Instruction tuning: tasks are explicit (defined using natural language instructions)
- Goal
 - Pretraining: teach LMs a wide range of linguistic patterns & general knowledge
 - Instruction tuning: teach LMs to follow specific instructions and perform a variety of tasks

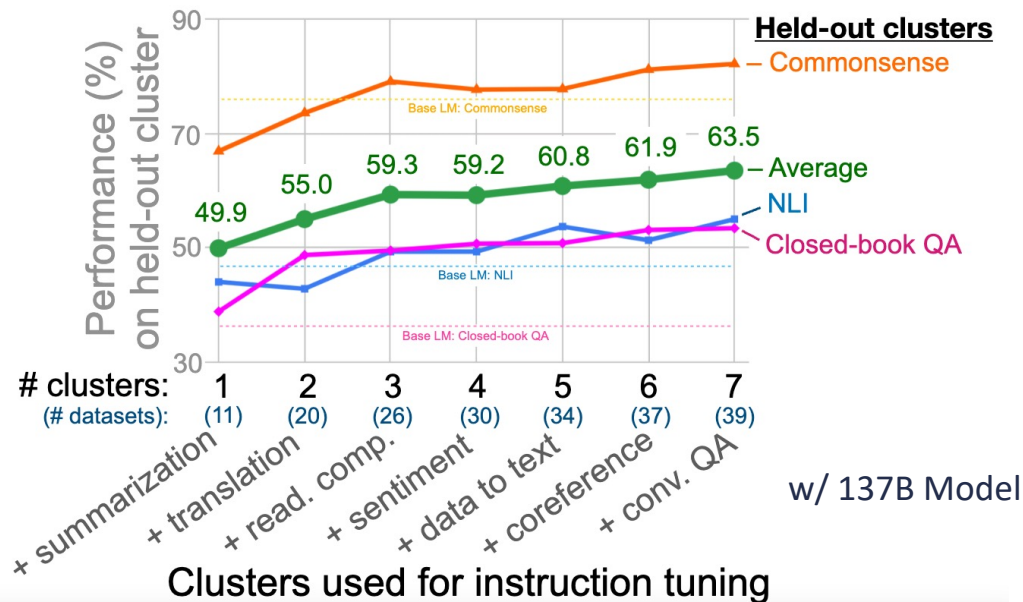
FLAN: Collection of Instruction Tuning Datasets

62 datasets (12 task clusters) covering a wide range of understanding + generation tasks



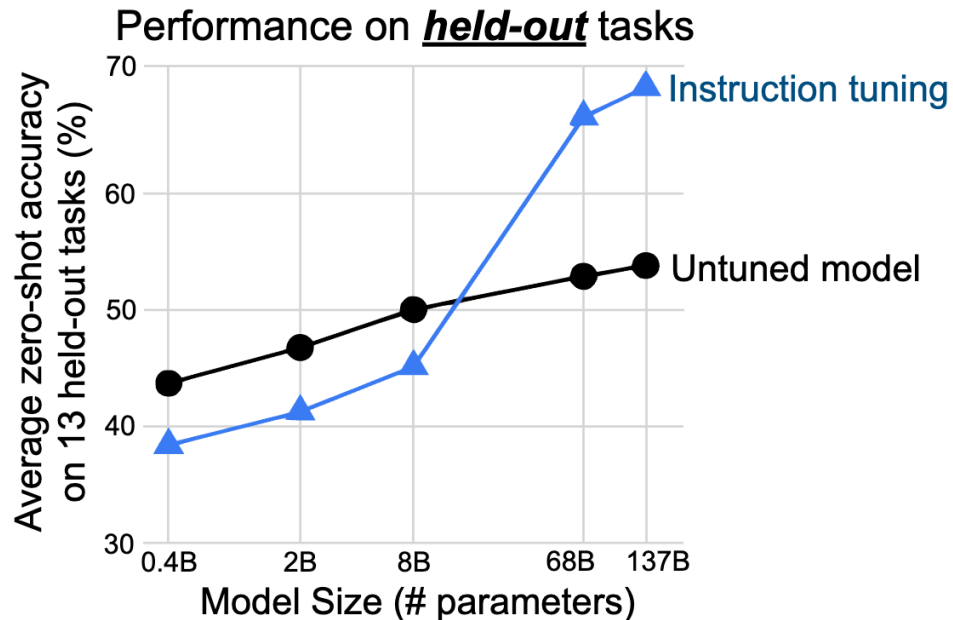
Generalization Improves with More Clusters

- Held out three clusters from instruction tuning: Commonsense, NLI, Closed-book QA
- More clusters and tasks used in instruction tuning => better generalization to unseen clusters



Instruction Tuning with Different Model Sizes

- Instruction tuning can hurt small model ($< 8\text{B}$) generalization
- Instruction tuning substantially improves generalization for large models



Chat-style Instruction Tuning

- Instruction tuning can also be used to build chatbots for multi-turn dialogue
- Instructions may not correspond strictly to one NLP task, but mimic a human-like dialogue
- Multi-turn instruction tuning training data example:

```
{"role": "user", "content": "What's the weather like today?"},  
{"role": "assistant", "content": "It's sunny with a high of 75 degrees."},  
{"role": "user", "content": "Great! What about tomorrow?"},  
{"role": "assistant", "content": "Tomorrow will be partly cloudy with a high of 72 degrees."}
```

Further Reading on Instruction Tuning

- [Multitask Prompted Training Enables Zero-Shot Task Generalization](#) [Sanh et al., 2021]
- [Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks](#) [Wang et al., 2022]
- [Self-Instruct: Aligning Language Models with Self-Generated Instructions](#) [Wang et al., 2022]
- [LIMA: Less Is More for Alignment](#) [Zhou et al., 2023]

Agenda

- Introduction to LLM Alignment
- Instruction Tuning
- Reinforcement Learning from Human Feedback (RLHF)

Limitations of Instruction Tuning & Why RLHF

- **Costly human annotations**
 - Instruction tuning requires human annotators to write down the entire expected responses
 - RLHF only relies on preference labels (which response is better?)
- **Open-ended generation**
 - Open-ended creative generation (e.g., story writing) inherently has no single “right” answer
 - RLHF uses human feedback to determine which response is more creative/appealing
- **Token-level learning**
 - Instruction tuning applies the language modeling loss -> penalizes all token mistakes equally regardless of their impact on the overall quality of the output (e.g., a grammatical error might be less critical than a factual inaccuracy)
 - RLHF uses human feedback to prioritize the error types that are more important to correct
- **Suboptimal human answers**
 - Instruction tuning may learn the suboptimal patterns written by humans
 - Identifying a better answer from a few options is usually easier than writing an optimal answer entirely

Overview: RLHF

- Human feedback collection
 - Generate multiple responses using the model given the same prompt
 - Human evaluators rank responses of the model based on helpfulness/honesty/safety...
- Reward model training
 - A reward model is trained on human feedback data to predict the quality of responses
 - Higher reward = more preferred by human evaluators
- Policy optimization
 - Use reinforcement learning algorithms to further train the LM to maximize the reward predicted by the reward model
 - Encourage the model to produce outputs that align better with human preferences

**Training language models to follow instructions
with human feedback**

RLHF Illustration

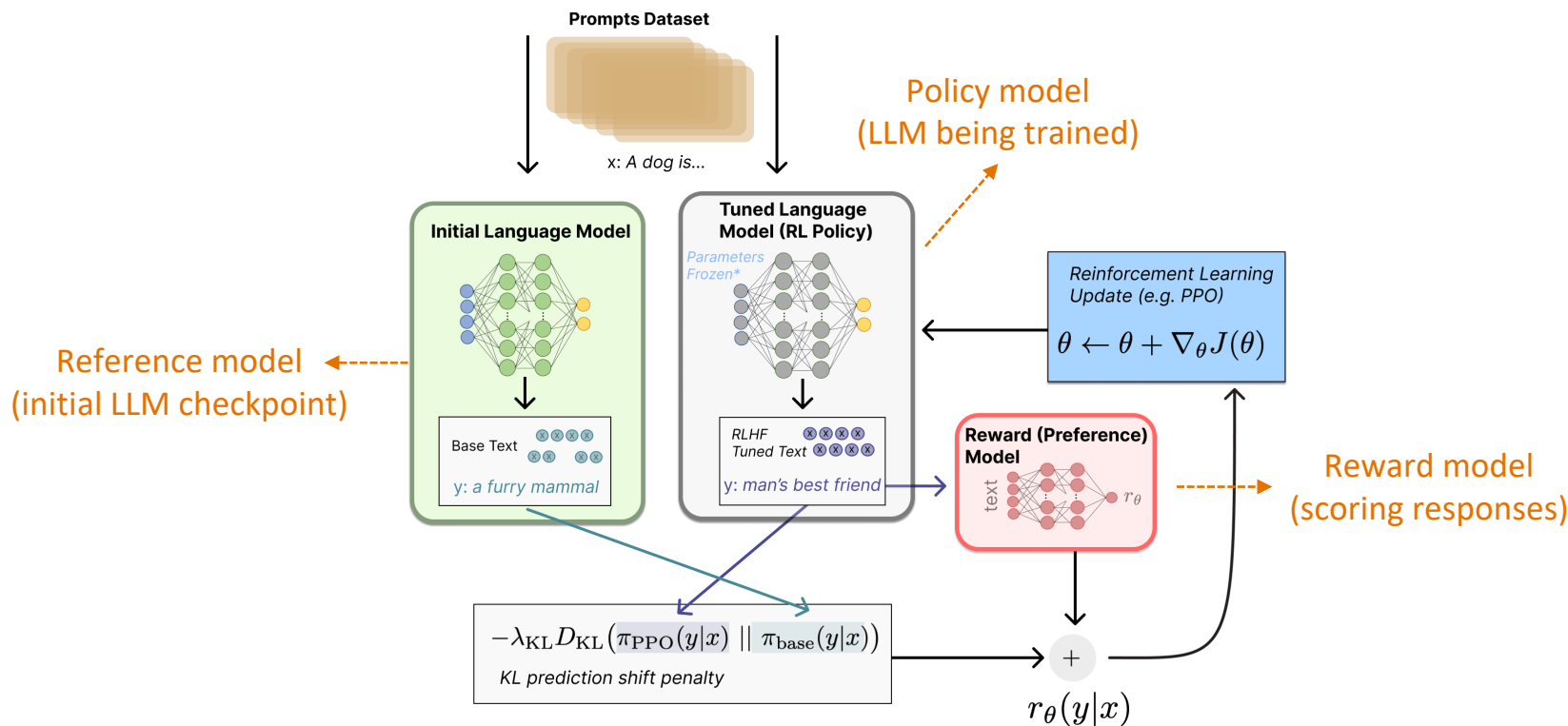


Figure source: <https://huggingface.co/blog/rlhf>

Preference Data Construction

- Goal of reward model: score the quality of LLM's output based on human feedback
- Can we directly ask human annotators to assign a scalar score (e.g., 1-10) to a single response?

What are 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.*

Different human evaluators
can be very inconsistent in
assigning absolute scores!

Preference Data with Pairwise Comparisons

Humans are better at relative judgments than absolute ones

What are 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 are the steps for making a simple cake?

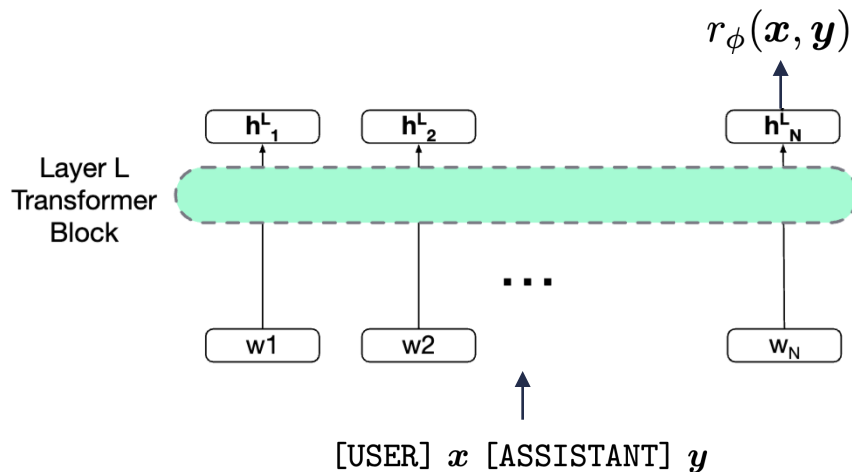
1. Warm up the oven.
2. Grease a cake pan.
3. Blend dry ingredients in a bowl.
4. Incorporate butter, milk, and vanilla.
5. Mix in the eggs.
6. Pour into the prepared pan.
7. Bake until golden brown.
8. Add frosting if desired.

Preference data: (x, y_w, y_l)



Reward Model Setup

Goal: train a reward model to assign a higher reward to y_w than y_l



Apply a linear layer at the last token representation to learn a scalar output



Reward Model Training

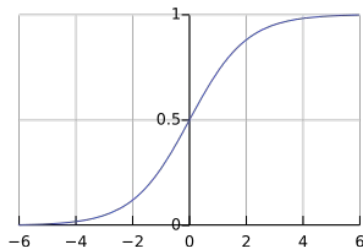
Bradley-Terry pairwise comparison objective

$$\mathcal{L}_{\text{RM}}(r_\phi) = -\mathbb{E}_{(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l) \sim \mathcal{D}} [\log \sigma(r_\phi(\mathbf{x}, \mathbf{y}_w) - r_\phi(\mathbf{x}, \mathbf{y}_l))]$$

reward of winning
response

reward of losing
response

$$y = \sigma(x)$$



Optimizing LLMs with the Reward Model

- The trained reward model serve as a proxy for human judgment (higher reward = more preferred by humans)
- Maximize the reward of generated responses from the LLM (policy model)

$$\max_{\theta} \mathbb{E}_{\underset{\text{LLM output}}{\underset{\text{probability}}{\mathbf{y} \sim p_{\theta}(\cdot | \mathbf{x})}}} [\underset{\text{reward of LLM}}{\underset{\text{generated response}}{r_{\phi}(\mathbf{x}, \mathbf{y})}}]$$

- What if our reward model is imperfect?

Issues with Naïve Optimization of Rewards

- Reward models are still only **approximations** of true human preferences
 - Can be noisy or incomplete (e.g., not well-generalized out-of-domain)
- Solely maximizing the reward leads to several issues
 - **Exploiting reward model flaws:** The LLM might learn to “hack” the reward model, finding ways to achieve high reward without actually possessing the desired behavior
 - **Mode collapse:** The LLM might converge to a narrow distribution of outputs that achieve high reward, but lack diversity and fail to generalize to different situations
 - **Loss of pretrained knowledge:** Over-optimization for the reward model can cause the LLM to unlearn desirable properties in the initial pretrained model (e.g., grammar, factuality)



Regularized Reward Optimization

- Add a penalty for drifting too far from the initial SFT checkpoint

$$\max_{\theta} \mathbb{E}_{\mathbf{y} \sim p_{\theta}(\cdot|\mathbf{x})} \left[\underbrace{r_{\phi}(\mathbf{x}, \mathbf{y})}_{\text{Maximize reward}} - \underbrace{\beta \log \left(\frac{p_{\theta}(\mathbf{y}|\mathbf{x})}{p_{\text{SFT}}(\mathbf{y}|\mathbf{x})} \right)}_{\text{Prevent deviation from the initial (SFT) model}} \right]$$

hyperparameter

- Penalize cases where $p_{\theta}(\mathbf{y}|\mathbf{x}) > p_{\text{SFT}}(\mathbf{y}|\mathbf{x})$
- In expectation, it is known as the Kullback-Leibler (KL) divergence $\text{KL}(p_{\theta}(\mathbf{y}|\mathbf{x}) || p_{\text{SFT}}(\mathbf{y}|\mathbf{x}))$

Optimization with Reinforcement Learning (RL)

- Why reinforcement learning:
 - No supervised data available (only a reward model)
 - Encourage the model to explore new possibilities (generations) guided by the reward model
- Optimization: policy gradient methods
 - Optimize the policy (LLM) by adjusting the parameters in the direction that increases expected rewards
- REINFORCE (simplest policy gradient method):

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \log \pi_{\theta}(a|s) R$$

Diagram illustrating the REINFORCE update rule components:

- α : step size
- ∇_{θ} : policy model (LLM)
- $\log \pi_{\theta}(a|s)$: action (generating the response)
- R : state (user prompt + conversation history)
- R : cumulative reward

Overview: Direct Preference Optimization (DPO)

- Overall, the RLHF framework is very complicated
 - Need to first train a reward model
 - Need to do online sampling
 - Performance is very sensitive to many hyperparameters
- Direct Preference Optimization (DPO): optimize LM parameters directly on preference data by solving a binary classification problem (without an explicit reward model)

Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Rafael Rafailov^{*†}

Archit Sharma^{*†}

Eric Mitchell^{*†}

Stefano Ermon^{†‡}

Christopher D. Manning[†]

Chelsea Finn[†]

[†]Stanford University [‡]CZ Biohub
{rafailov,architsh,eric.mitchell}@cs.stanford.edu

Paper: <https://arxiv.org/pdf/2305.18290>



Further Reading on RLHF

- [RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment](#) [Dong et al., 2023]
- [Iterative Preference Learning from Human Feedback: Bridging Theory and Practice for RLHF under KL-Constraint](#) [Xiong et al., 2023]
- [SLiC-HF: Sequence Likelihood Calibration with Human Feedback](#) [Zhao et al., 2023]
- [SimPO: Simple Preference Optimization with a Reference-Free Reward](#) [Meng et al., 2024]



Thank You!

Yu Meng

University of Virginia

yumeng5@virginia.edu