

$\Psi$

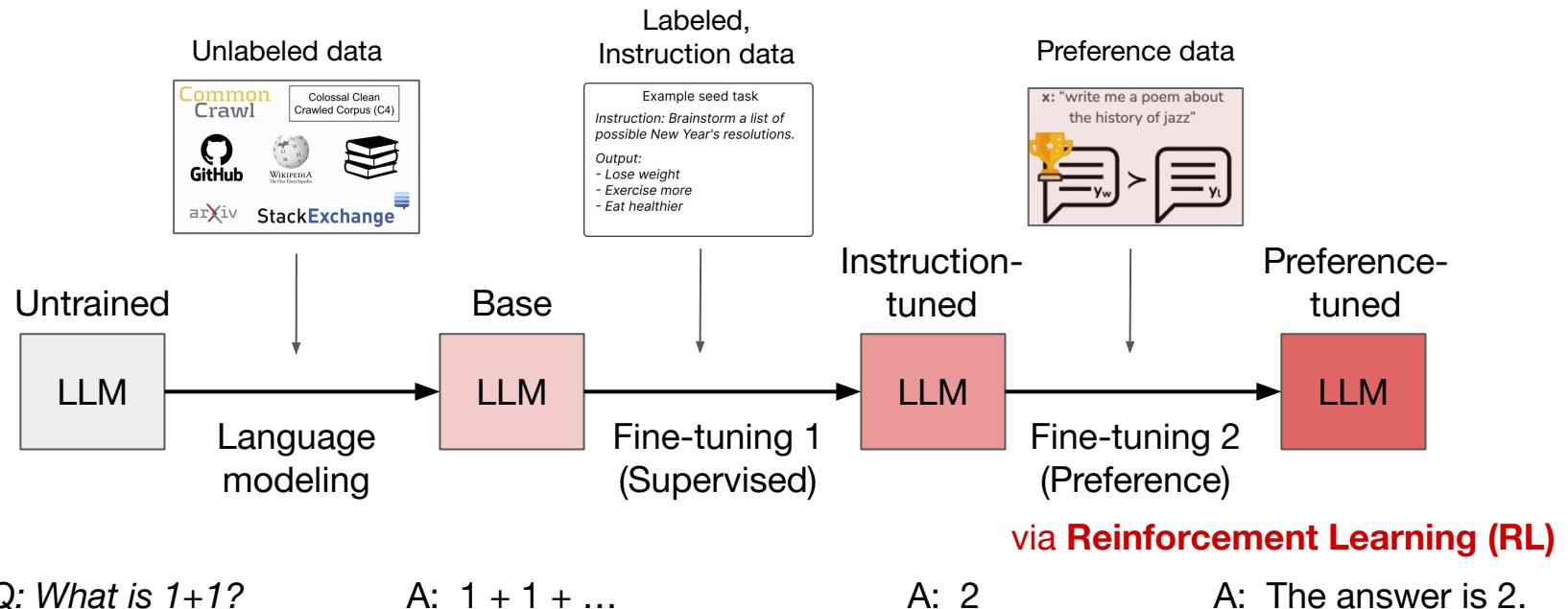
INFO-I590 Fundamentals and Applications of LLMs

**RLHF**

**Reinforcement Learning from Human Feedback**

Jisun An

# Three steps of creating a high-quality LLM



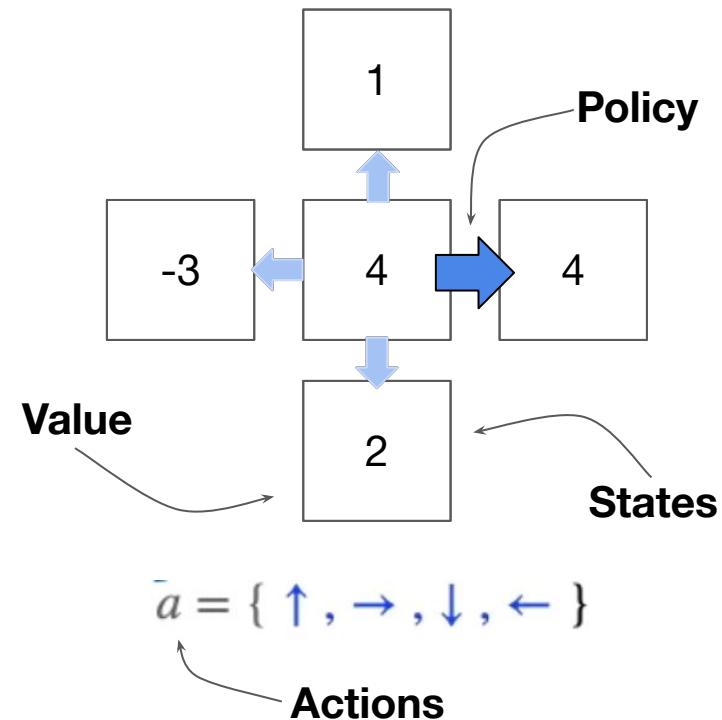
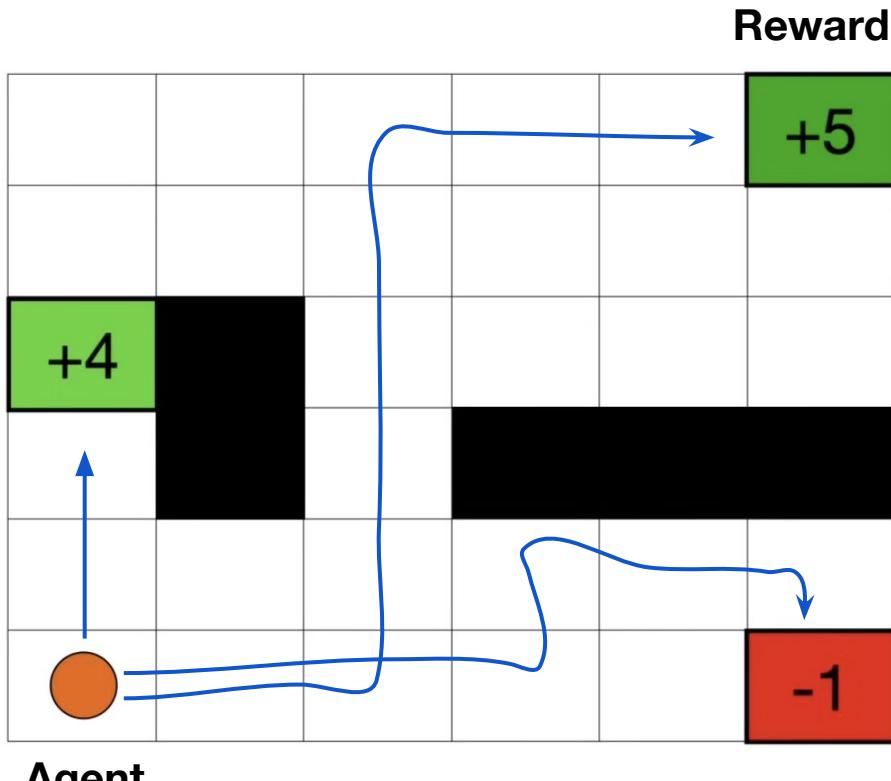
# What is RL?

## A very brief introduction

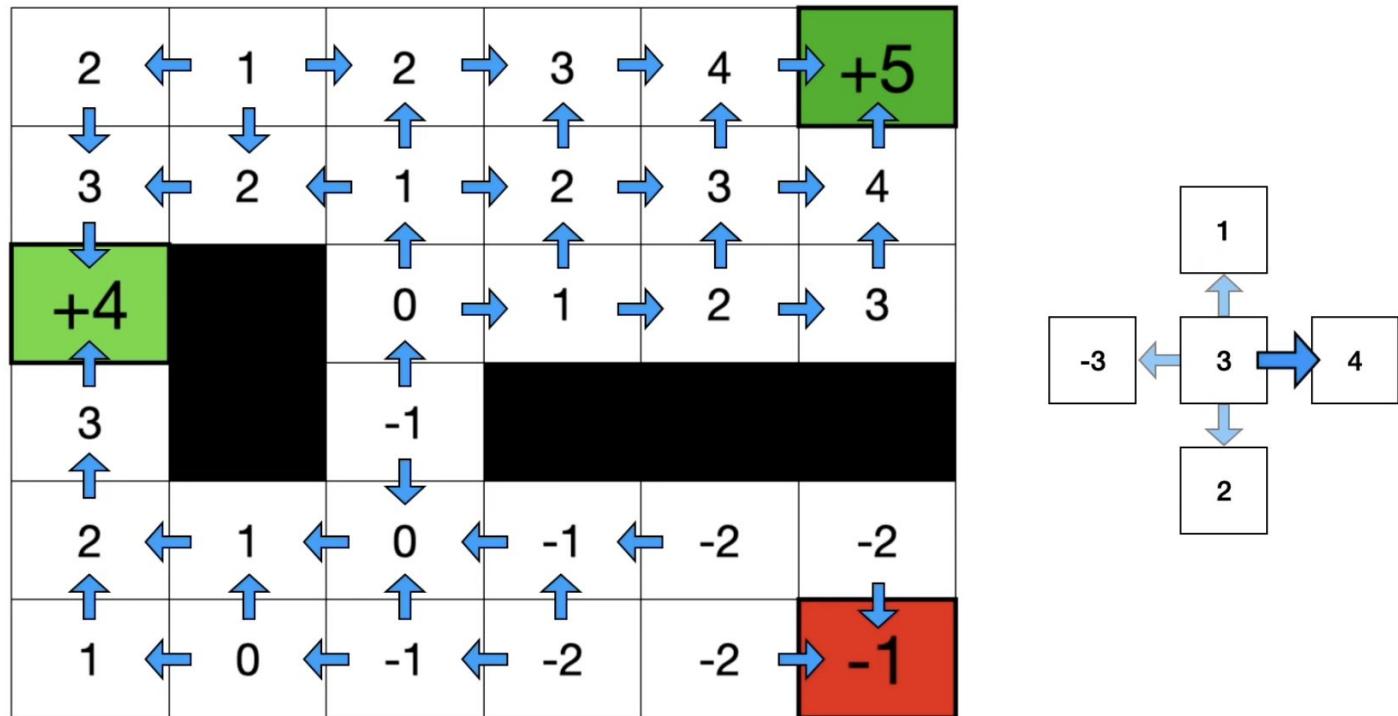
# What is Reinforcement Learning (RL)?

- RL is a type of machine learning where we have an
  - **environment X**
  - ability to make **actions A**
  - get a delayed **reward R**
- An agent learns by interacting with an environment.
- The agent receives rewards for good actions and penalties for bad ones.
- The goal is to maximize cumulative rewards over time.

# GridWorld



# Good Policy: Where is the best place to move?

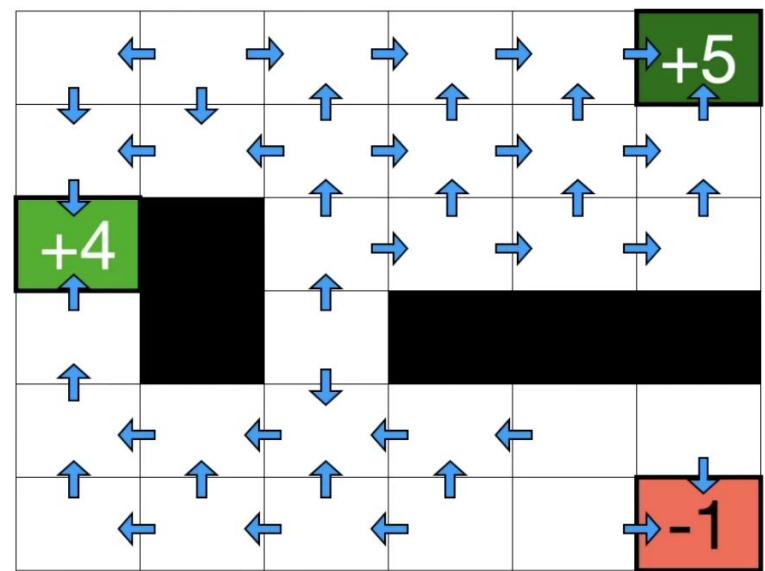


# Optimal values and policy

Values

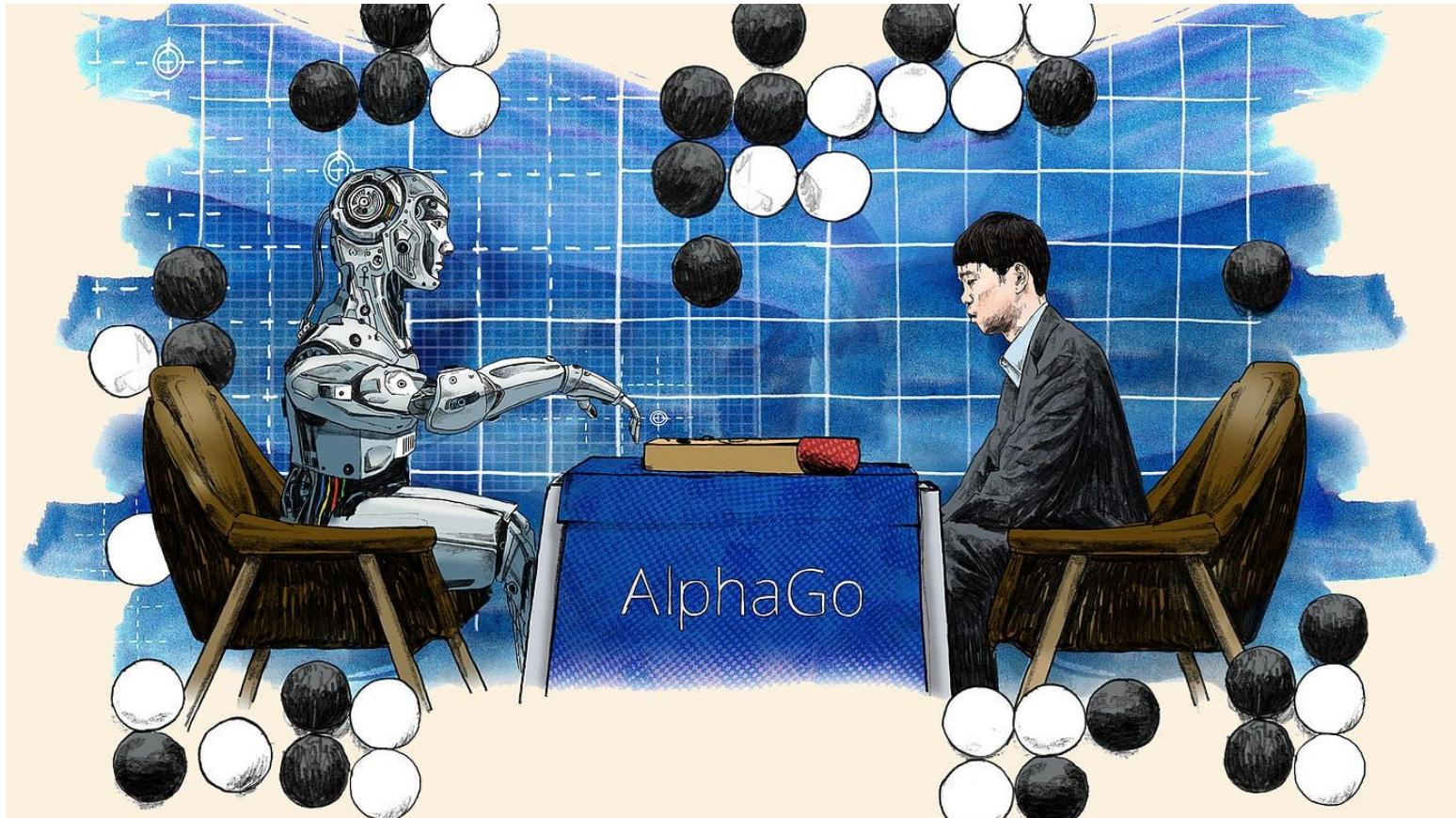
2	1	2	3	4	+5
3	2	1	2	3	4
+4		0	1	2	3
3		-1			
2	1	0	-1	-2	-2
1	0	-1	-2	-2	-1

Policy



# Why is RL Important?

- Helps in optimizing long-term decision-making
- Used in game playing (AlphaGo, Dota 2 AI), robotics, autonomous driving, and chatbot optimization



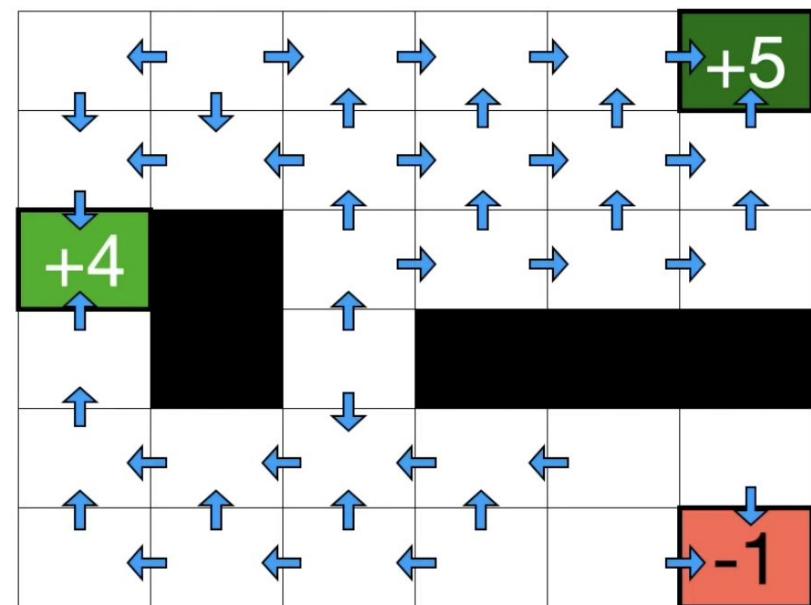
Img: <https://ai.plainenglish.io/how-deepminds-alphago-became-the-world-s-top-go-player-5b275e553d6a>

How can we learn these optimal values and policy?

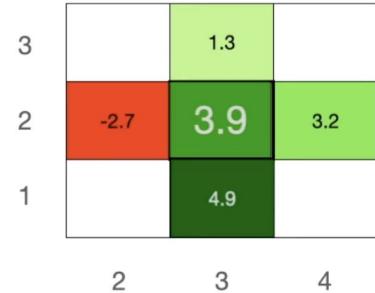
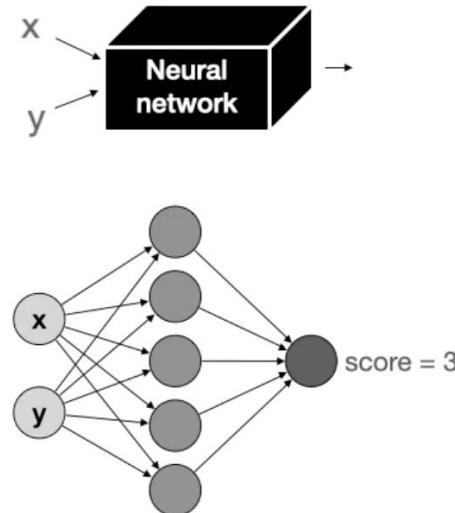
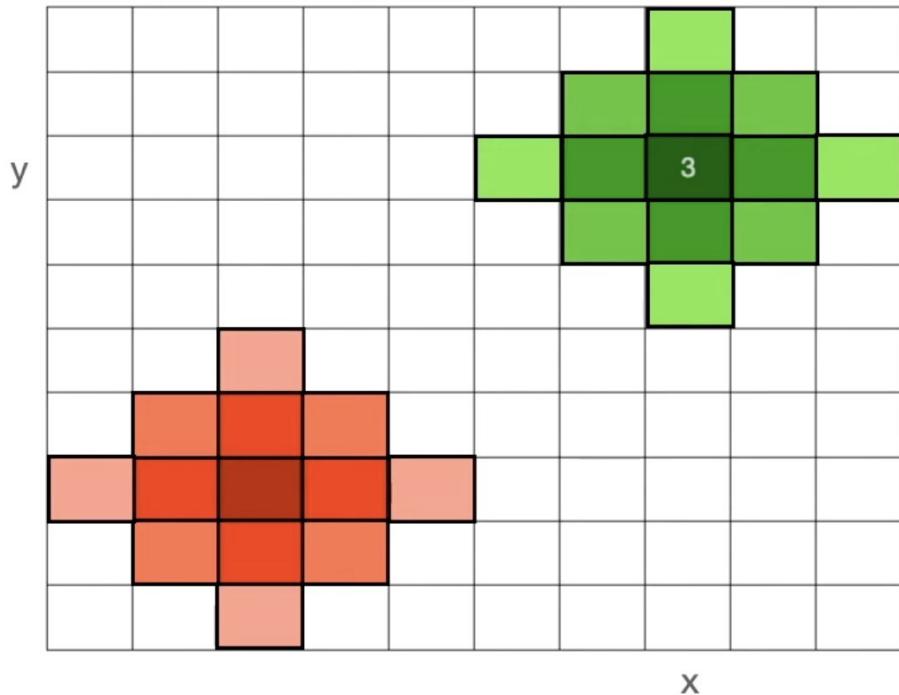
## Values

2	1	2	3	4	+5
3	2	1	2	3	4
+4		0	1	2	3
3		-1			
2	1	0	-1	-2	-2
1	0	-1	-2	-2	-1

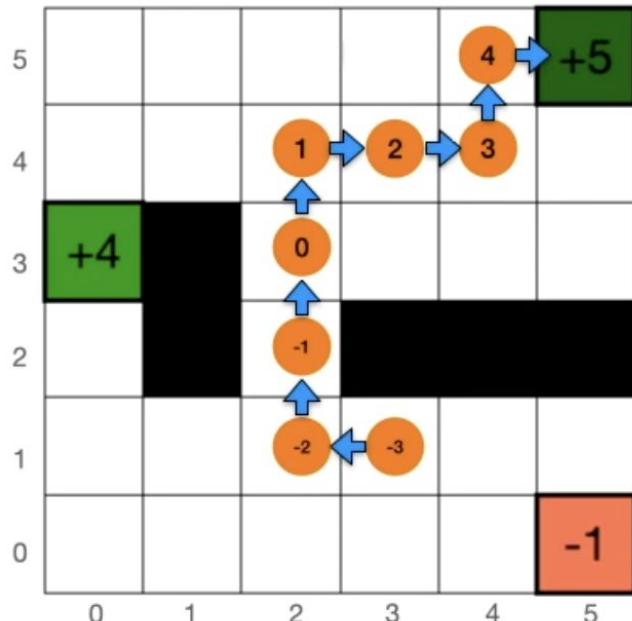
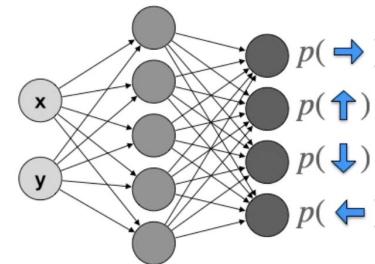
Policy



# Value neural networks



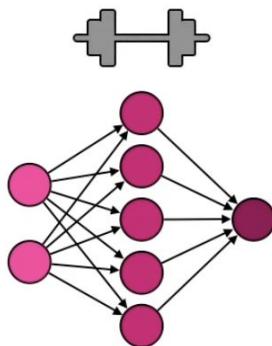
# Policy neural networks



Gain	x	y	action	Probability	Increase log loss	change
4	4	5	$\rightarrow$	$p(\rightarrow) = 0.3$	$4 \ln(p(\rightarrow))$	increase
3	4	4	$\uparrow$	$p(\uparrow) = 0.9$	$3 \ln(p(\uparrow))$	increase
2	3	4	$\rightarrow$	$p(\rightarrow) = 0.1$	$2 \ln(p(\rightarrow))$	increase
1	2	4	$\rightarrow$	$p(\rightarrow) = 0.2$	$1 \ln(p(\rightarrow))$	increase
0	2	3	$\uparrow$	$p(\uparrow) = 0.5$	$0 \ln(p(\uparrow))$	stay
-1	2	2	$\uparrow$	$p(\uparrow) = 0.4$	$-1 \ln(p(\uparrow))$	decrease
-2	2	1	$\uparrow$	$p(\uparrow) = 0.3$	$-2 \ln(p(\uparrow))$	decrease
-3	3	1	$\leftarrow$	$p(\leftarrow) = 0.7$	$-3 \ln(p(\leftarrow))$	decrease

# Value and Policy neural networks

Value  
neural  
network

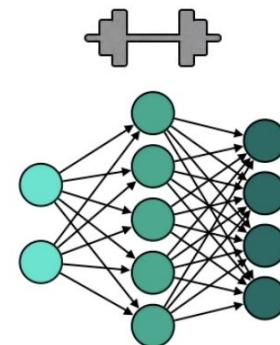


Input: state  
Output: value

Approximates  
values

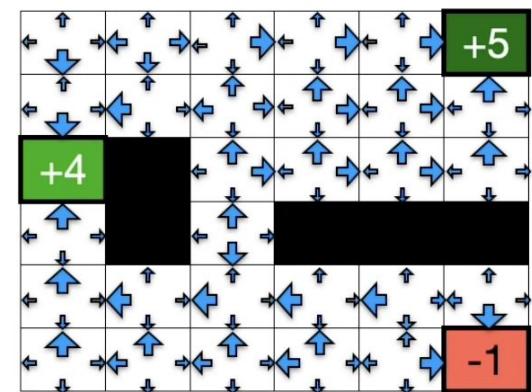
3.1	1.4	2.5	3.2	2.7	+5
2.2	-0.2	1.3	2.9	4.1	3.2
+4		-0.5	1.7	2.8	3.2
3.9		0.2			
2.6	1.5	1.1	0.3	-1.8	-2.3
1.4	0.8	0.1	-1.5	-2	-1

Policy  
neural  
network



Input: state  
Output: actions

Approximates  
policy



# RL for language

			+2		
	+3	What color is the sky?		+1	
		What color is the sky			
<start>	What				

+3

What color is the sky?  
Blue

+2

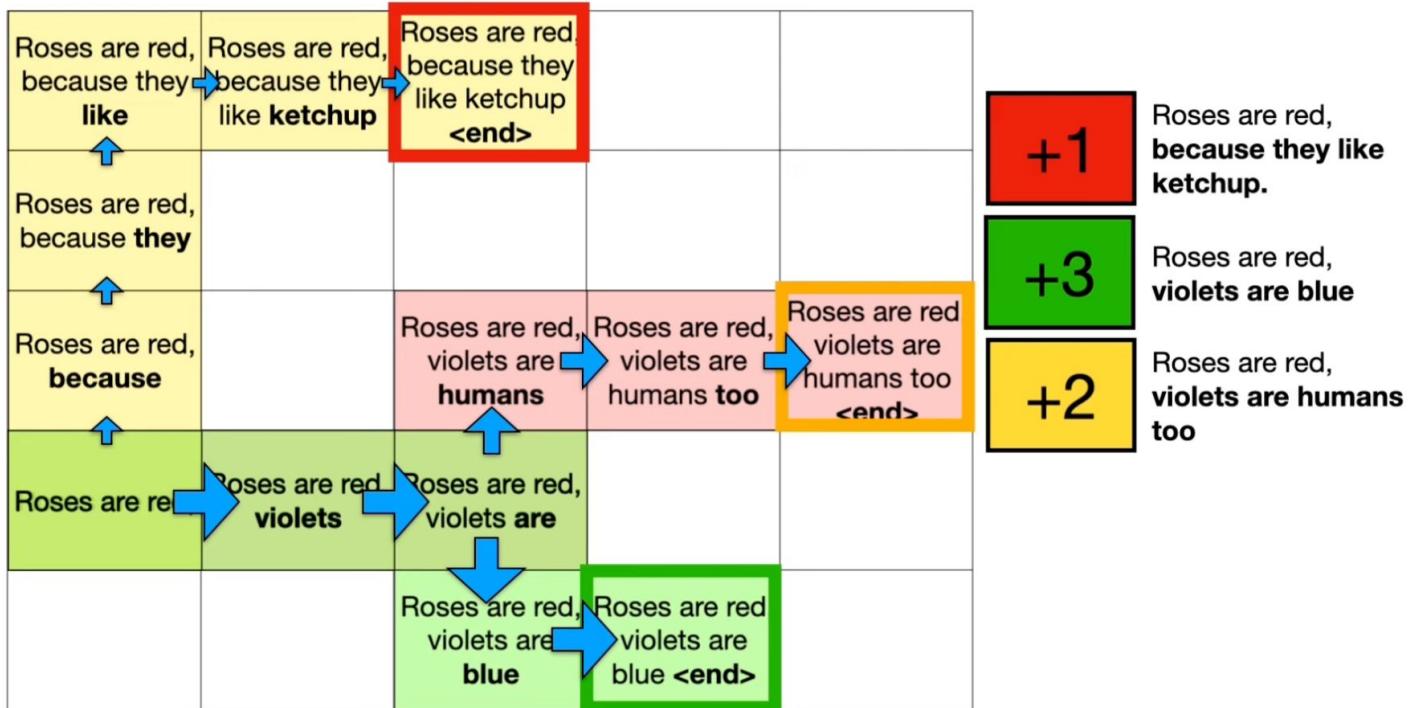
What color is the sky?  
Red

+1

What color is the sky?  
Banana



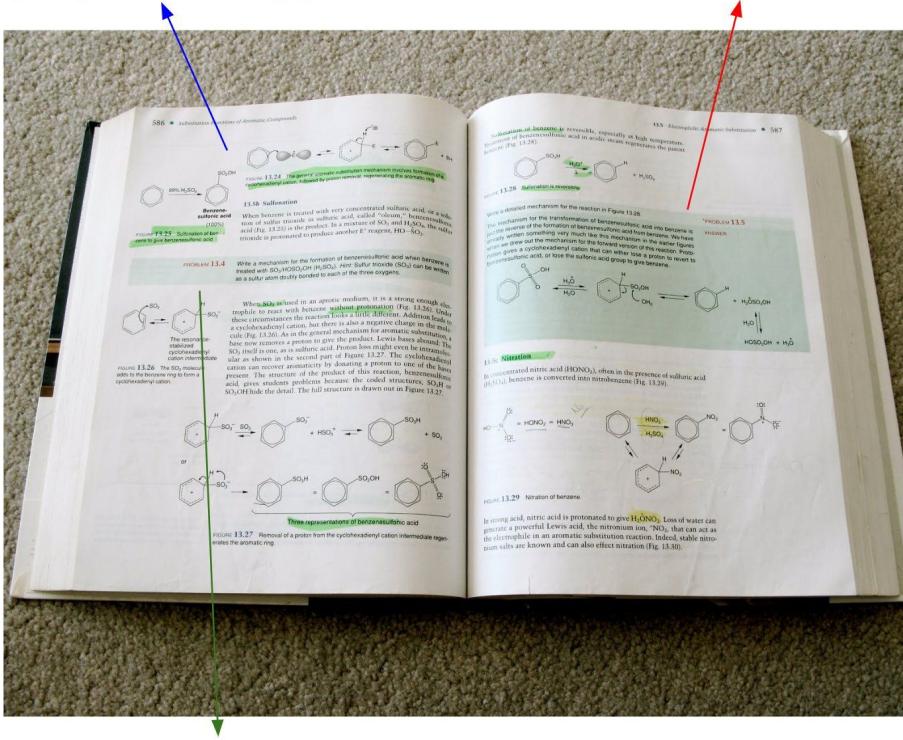
# RL for language



# RL for Language Modeling

# Why do we need RL?

exposition  $\leftrightarrow$  pretraining  
(background knowledge)



practice problems  $\leftrightarrow$  reinforcement learning  
(prompts to practice, trial & error until you reach the correct answer)

We are given problem statement (prompt) and the final answer. We want to practice solutions that take us from problem statement to the answer, and “internalize” them into the model.

Problem statement

Solution

Answer

# Which solution is better for inclusion in instructional data?

Emily buys 3 apples and 2 oranges. Each orange costs \$2. The total cost of all the fruit is \$13. What is the cost of each apple?

Answer: \$3

—

Set up a system of equations.

$X$  = price of apples

$$3*x + 2*2 = 13$$

$$3*x + 4 = 13$$

$$3*x = 9$$

$$X = 3$$

—

$$13 - 4 = 9, 9/3 = 3$$

—

The oranges cost  $2*2 = 4$ .

So the apples cost  $13 - 4 = 9$ .

There are 3 apples.

So each apple costs  $9/3 = 3$ .

—

$$(13 - 4)/3 = 3.$$

## ChatGPT o3-mini ▾

Emily buys 3 apples and 2 oranges. Each orange costs \$2. The total cost of all the fruit is \$13. What is the cost of each apple?

Reasoned for a second

Let  $x$  be the cost of one apple. Emily buys 3 apples and 2 oranges, with each orange costing \$2. The total cost is given as \$13, so we can set up the equation:

$$3x + 2(2) = 13$$

Simplify the equation:

$$3x + 4 = 13$$

Subtract 4 from both sides:

$$3x = 9$$

Divide both sides by 3:

$$x = 3$$

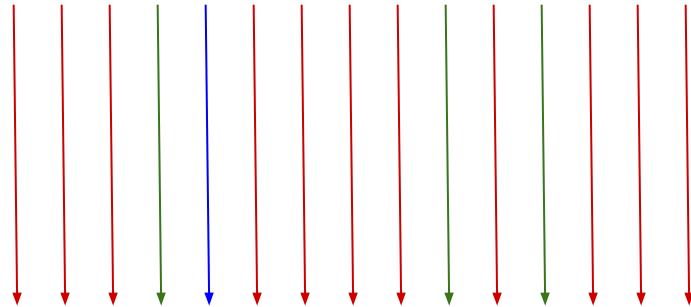
Thus, the cost of each apple is \$3.

Goals of RL:

- **Correctness: Obtain correct answers**
- Presentation: Effectively present results to humans

prompt

Emily buys 3 apples and 2 oranges. Each orange costs \$2. The total cost of all the fruit is \$13. What is the cost of each apple?



solutions

Answer: 3

We generate 15 solutions.

Only 4 of them got the right answer.

Take the top solution (each right and short).

Train on it.

Repeat many, many times.



---

# DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

[research@deepseek.com](mailto:research@deepseek.com)

## Abstract

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement learning (RL) without supervised fine-tuning (SFT) as a preliminary step, demonstrates remarkable reasoning capabilities. Through RL, DeepSeek-R1-Zero naturally emerges with numerous powerful and intriguing reasoning behaviors. However, it encounters challenges such as poor readability, and language mixing. To address these issues and further enhance reasoning performance, we introduce DeepSeek-R1, which incorporates multi-stage training and cold-start data before RL. DeepSeek-R1 achieves performance comparable to OpenAI-o1-1217 on reasoning tasks. To support the research community, we open-source DeepSeek-R1-Zero, DeepSeek-R1, and six dense models (1.5B, 7B, 8B, 14B, 32B, 70B) distilled from DeepSeek-R1 based on Qwen and Llama.

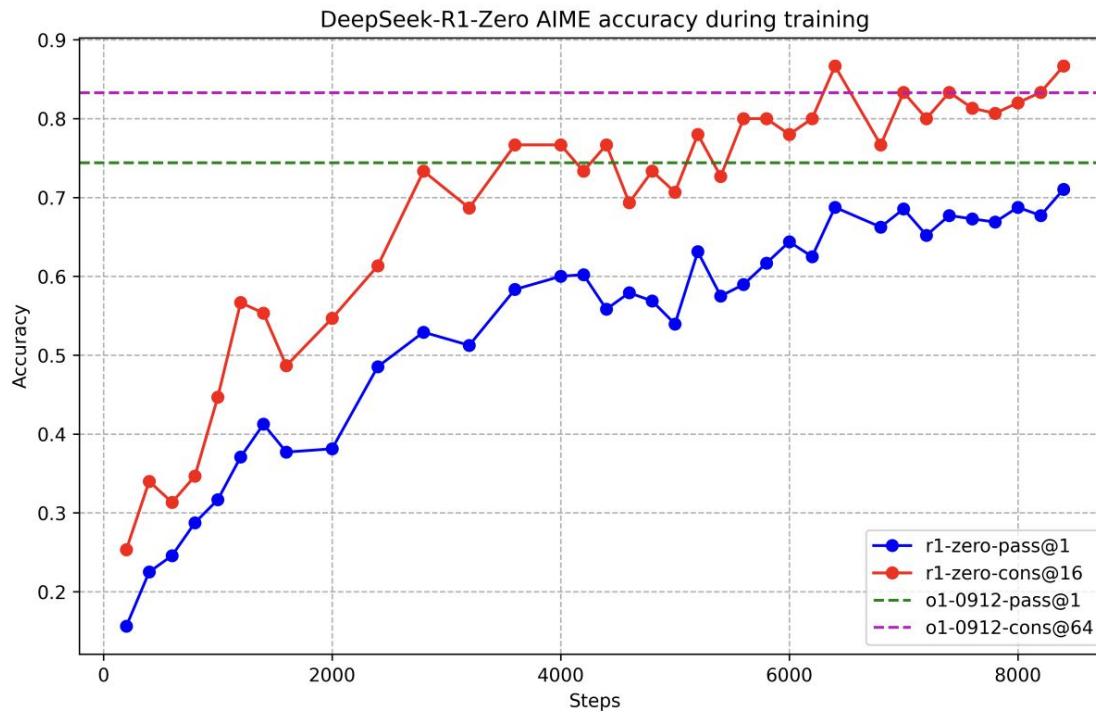


Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

# 2024 AIME I Problems

## Problem 1

Every morning Aya goes for a 9-kilometer-long walk and stops at a coffee shop afterwards. When she walks at a constant speed of  $s$  kilometers per hour, the walk takes her 4 hours, including  $t$  minutes spent in the coffee shop. When she walks at  $s + 2$  kilometers per hour, the walk takes her 2 hours and 24 minutes, including  $t$  minutes spent in the coffee shop. Suppose Aya walks at  $s + \frac{1}{2}$  kilometers per hour. Find the number of minutes the walk takes her, including the  $t$  minutes spent in the coffee shop.

[Solution](#)

## Problem 2

There exist real numbers  $x$  and  $y$ , both greater than 1, such that  $\log_x(y^x) = \log_y(x^{4y}) = 10$ . Find  $xy$ .

[Solution](#)

## Problem 3

Alice and Bob play the following game. A stack of  $n$  tokens lies before them. The players take turns with Alice going first. On each turn, the player removes 1 token or 4 tokens from the stack. The player who removes the last token wins. Find the number of positive integers  $n$  less than or equal to 2024 such that there is a strategy that guarantees that Bob wins, regardless of Alice's moves.

[Solution](#)

## Problem 4

Jen enters a lottery by selecting 4 distinct elements of  $S = \{1, 2, 3, \dots, 9, 10\}$ . Then four elements of  $S$  are drawn at random. Jen wins a prize if at least two of her numbers were drawn, and wins the grand prize if all four of her numbers are drawn. The probability that Jen wins the grand prize given that Jen wins a prize is  $\frac{m}{n}$  where  $m$  and  $n$  are relatively prime positive integers. Find  $m + n$ .

[Solution](#)

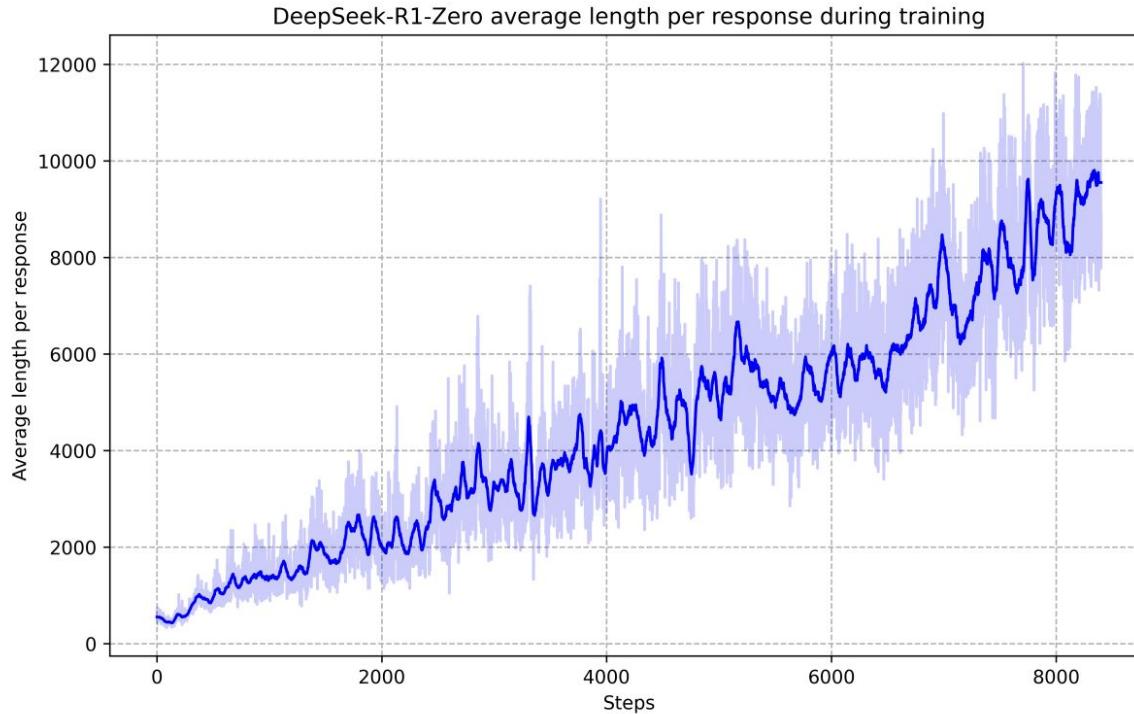


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

---

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 ...

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

Rearrange to isolate the inner square root term:

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

...

**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 ...

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: ...

...

---

Table 3 | An interesting “aha moment” of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.

UI

API



Emily buys 3 apples and 2 oranges. Each orange costs \$2. The total cost of all the fruit is \$13. What is the cost of each apple?

<think> Okay, let me try to figure out this problem. So, Emily buys 3 apples and 2 oranges. Each orange costs \$2, and the total is \$13. We need to find the cost of each apple. Hmm, okay. Let's break this down step by step.

First, let's note down what we know. The problem says there are 3 apples and 2 oranges. Each orange is \$2, so maybe I can calculate the total cost of the oranges first. If one orange is \$2, then two oranges would be 2 times \$2. Let me write that out: 2 oranges \* \$2/orange. That should be  $2 \times 2 = \$4$ . So the oranges cost \$4 in total.

Now, the total cost for all the fruit is \$13. If the oranges are \$4, then the apples must account for the remaining cost. So, subtract the cost of oranges from the total to get the cost of apples. That would be \$13 total - \$4 for oranges = \$9 for apples. Okay, so the apples cost \$9 altogether.

But Emily bought 3 apples. So if 3 apples cost \$9, then each apple must cost \$9 divided by 3. Let me write that: \$9 / 3 apples = \$3 per apple. So, each apple costs \$3? Let me check that again to be sure.

Let's verify the total cost. If each apple is \$3, then 3 apples would be  $3 \times 3 = \$9$ . Then 2 oranges at \$2 each would be  $2 \times 2 = \$4$ . Adding those together,  $\$9 + \$4 = \$13$ , which matches the total given. So that seems right. Hmm, wait, let me make sure I didn't make any mistakes in my calculations.

Starting over: 2 oranges at \$2 each is  $2 \times 2 = \$4$ . Total is \$13. So  $13 - \$4 = \$9$  left for apples. \$9 divided by 3 apples is \$3. Yes, that seems correct. Alternatively, I can set up an equation. Let me try that method to confirm.

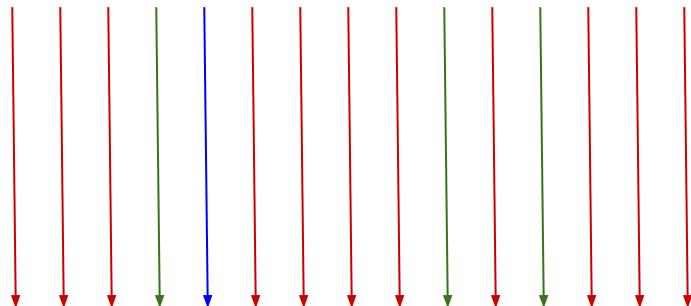
Let the cost of each apple be  $x$  dollars. Then, the total cost for apples is  $3x$ . The total cost for oranges is  $2 \times 2 = \$4$ . So the equation is  $3x + 4 = 13$ . Solving for  $x$ : subtract 4 from both sides,  $3x = 9$ , then divide by 3,  $x = 3$ . Yep, same result. So that confirms it. Each apple costs \$3. I think that's the answer. Let me just see if there's another way to approach it, but I think the equation method is straightforward.

Alternatively, maybe using ratios? Hmm, but that might complicate things. Since we have exact numbers, setting up the equation is the simplest way. Yeah, I think I'm confident with this answer. Each apple is \$3. The check also works

# Verifiable vs un-verifiable domains

prompt

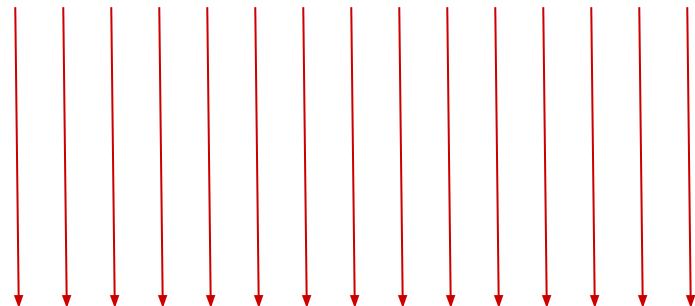
Emily buys 3 apples and 2 oranges. Each orange costs \$2. The total cost of all the fruit is \$13. What is the cost of each apple?



Answer: \$3

prompt

Write a super funny joke about pelicans



Problem: how we do score theses?

# Reinforcement Learning from Human Feedback (RLHF)

# RLHF (Reinforcement Learning from Human Feedback)

Reinforcement Learning in un-verifiable domains

---

## Fine-Tuning Language Models from Human Preferences

---

**Daniel M. Ziegler\***   **Nisan Stiennon\***   **Jeffrey Wu**   **Tom B. Brown**

**Alec Radford**   **Dario Amodei**   **Paul Christiano**   **Geoffrey Irving**

OpenAI

{dmz,nisan,jeffwu,tom,alec,damodei,paul,irving}@openai.com

---

## Training language models to follow instructions with human feedback

---

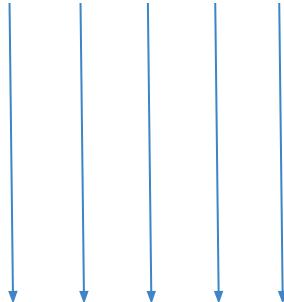
### Abstract

Making language models bigger does not inherently make them better at following a user’s intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not *aligned* with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts

# If using a typical RL approach

prompt

“Write a super funny joke about pelicans”

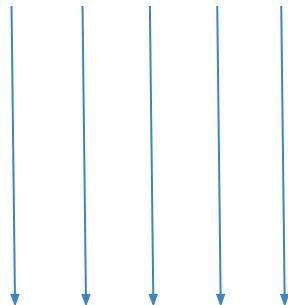


- Native approach
  - Run RL as usual, of 1,000 updates of 1,000 prompts of 1,000 rollouts (cost: 1,000,000,000 scores from humans)

# If using an RLHF

prompt

“Write a super funny joke about pelicans”



reward model scores: 0.1 0.8 0.3 0.4 0.5

human ordering: 2 1 3 5 4

- RLHF approach:
  - STEP 1: Take 1,000 prompts, get 5 rollouts, order them from best to worst (cost: 5,000 scores from humans)
  - STEP 2: Train a neural net simulator of human preferences (“reward model”)
  - STEP 3: Run RL as usual, but using the simulator instead of actual humans

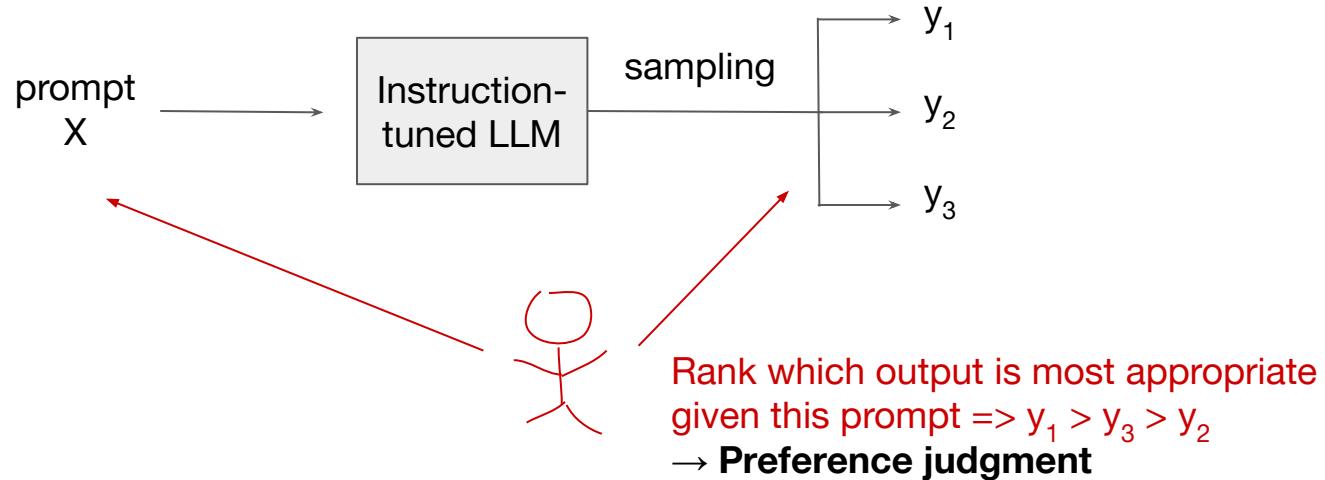
# Human feedback

- **Fluency:** how natural is the output
- **Adequacy:** in translation, how well does the output reflect the input semantics?
- **Coherence:** does the output fit coherently in a discourse?
- **Truthfulness:** does the output contain accurate information and doesn't mislead the user?
- **Helpfulness:** how effectively does the output address the user's needs or task?
- **Harmlessness:** does the output avoid causing harm, offense, or misinformation?
- etc. etc.

# RLHF: Aligning LLMs with human intents

- Start with a large pre-trained LLM
- Step 1: instruction-tuning
- Limitations of Instruction-Tuning
  - The model does not learn from negative feedback.
  - Tasks like open-ended creative generation have no right answer.
    - They allow multiple valid responses, but training data often includes only one.
  - The model struggles to recognize when it should say, "I don't know."
  - The training process does not optimize for user-specific needs or values (human preferences).
    - Language modeling penalizes all token-level mistakes equally, but some errors are worse than others.

# Human feedback: Preference ratings



- Limitation of human feedback: extremely expensive to collect

# Reward model

- Idea: Can we train a model to predict human preference judgement?
- Reward Model
  - Input: prompt x, output y;
  - Output: A scalar score representing preference
- One way to train this model is using the **Bradley-Terry pairwise preference model.**

# Bradley-Terry pairwise preference model

- $x$  is the input (e.g., a user prompt)
- $y_W$  is the preferred response
- $y_L$  is the less preferred response
- $r(x, y)$  is the reward function assigning a scalar score
- The probability that **response  $y_W$  is preferred over  $y_L$**  in terms of their reward scores:

$$P(y_W > y_L \mid x) = \frac{e^{r(x, y_W)}}{e^{r(x, y_W)} + e^{r(x, y_L)}}$$

If  $r(x, y_W) \gg r(x, y_L)$ , the probability approaches 1.  
If  $r(x, y_W) \approx r(x, y_L)$ , the probability is close to 0.5.  
If  $r(x, y_W) \ll r(x, y_L)$ , the probability approaches 0.

- The negative log-likelihood (NLL) loss of this probability is defined as:

$$\mathcal{L} = -\log \sigma(r(x, y_W) - r(x, y_L))$$

# Deriving the loss function for a preference model

$$P(y_W > y_L \mid x) = \frac{e^{r(x, y_W)}}{e^{r(x, y_W)} + e^{r(x, y_L)}}$$

Factor out  $e^{r(x, y_L)}$  from the denominator

$$P(y_W > y_L \mid x) = \frac{e^{r(x, y_W) - r(x, y_L)}}{1 + e^{r(x, y_W) - r(x, y_L)}}$$

Recognizing that this matches the sigmoid function:  $\sigma(z) = \frac{1}{1 + e^{-z}}$

$$\sigma(r(x, y_W) - r(x, y_L)) = \frac{1}{1 + e^{-(r(x, y_W) - r(x, y_L))}}$$

$$P(y_W > y_L \mid x) = \sigma(r(x, y_W) - r(x, y_L))$$

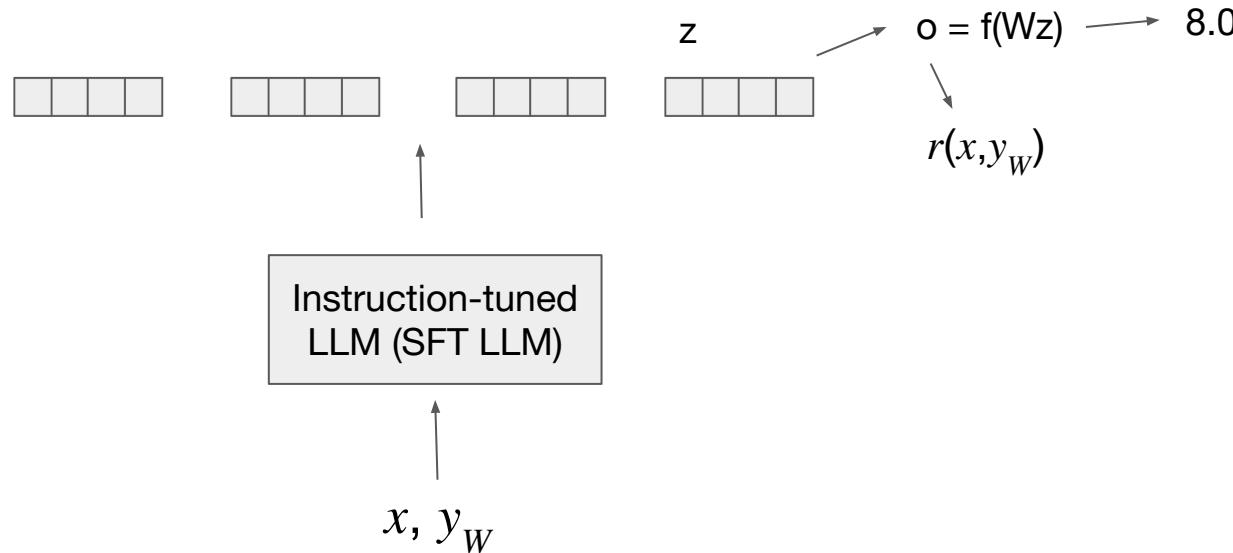
Thus, the probability of preference follows a sigmoid distribution of the score difference.

Since we want to **maximize the likelihood** that  $y_W$  is preferred over  $y_L$ , we take the **negative log-likelihood** to get a loss function:

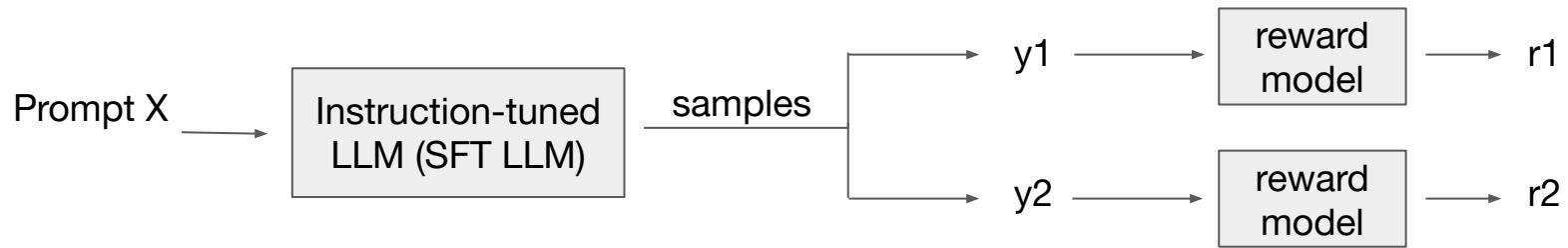
$$\begin{aligned} \mathcal{L} &= -\log P(y_W > y_L \mid x) \\ &= -\log \sigma(r(x, y_W) - r(x, y_L)) \end{aligned}$$

# Training the reward model

Intuition: good sample  $y_W$ 's reward should be greater than  $y_L$



# Using the reward model



# Aligning LLMs to human preferences using the reward model

- “best-of- $n$ ” sampling (rejection sampling)
  - Generate  $n$  samples for a given prompt, score each sample with the reward model, choose the sample with the highest reward
    - → computationally very expensive
- Fine-tuning to maximize  $p(y_w | x)$ 
  - Generate a large dataset using reward model and finetune it
    - → tends to fail to learn from negative samples
- Use **reinforcement learning** to increase  $p(y_w | x)$  slightly and decrease  $p(y_L | x)$  slightly, where the amounts are functions of  $r(x, y_w)$ ,  $r(x, y_L)$ 
  - → Allows for fine-grained control over preference alignment

# RLHF optimization

- We observe a reward only after generating a complete sequence.
- RL propagates the final reward to intermediate steps, guiding the model toward the optimal token sequence.
- The policy ( $\pi$ ) refers to the probability distribution over token sequences that the model generates.

$\pi_{\text{ref}}(y|x) \rightarrow$  SFT checkpoint (the reference model)

$\pi(y|x) \rightarrow$  current policy model (initialized to  $\pi_{\text{ref}}$ )

# RLHF optimization objective

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{x \sim D, y \sim \pi} [r(x, y) - \beta D_{KL}(\pi(y \mid x) \parallel \pi_{\text{ref}}(y \mid x))]$$

$$D_{KL}(\pi \parallel \pi_{\text{ref}}) = \sum_i \log \frac{\pi(w_i \mid w_1, \dots, w_{i-1}, x)}{\pi_{\text{ref}}(w_i \mid w_1, \dots, w_{i-1}, x)}$$

$\pi^*$  is the optimized policy model.

$r(x, y)$  is the reward function for generating output  $y$  given input  $x$ .

$\pi(y \mid x)$  is the probability distribution of the current policy model.

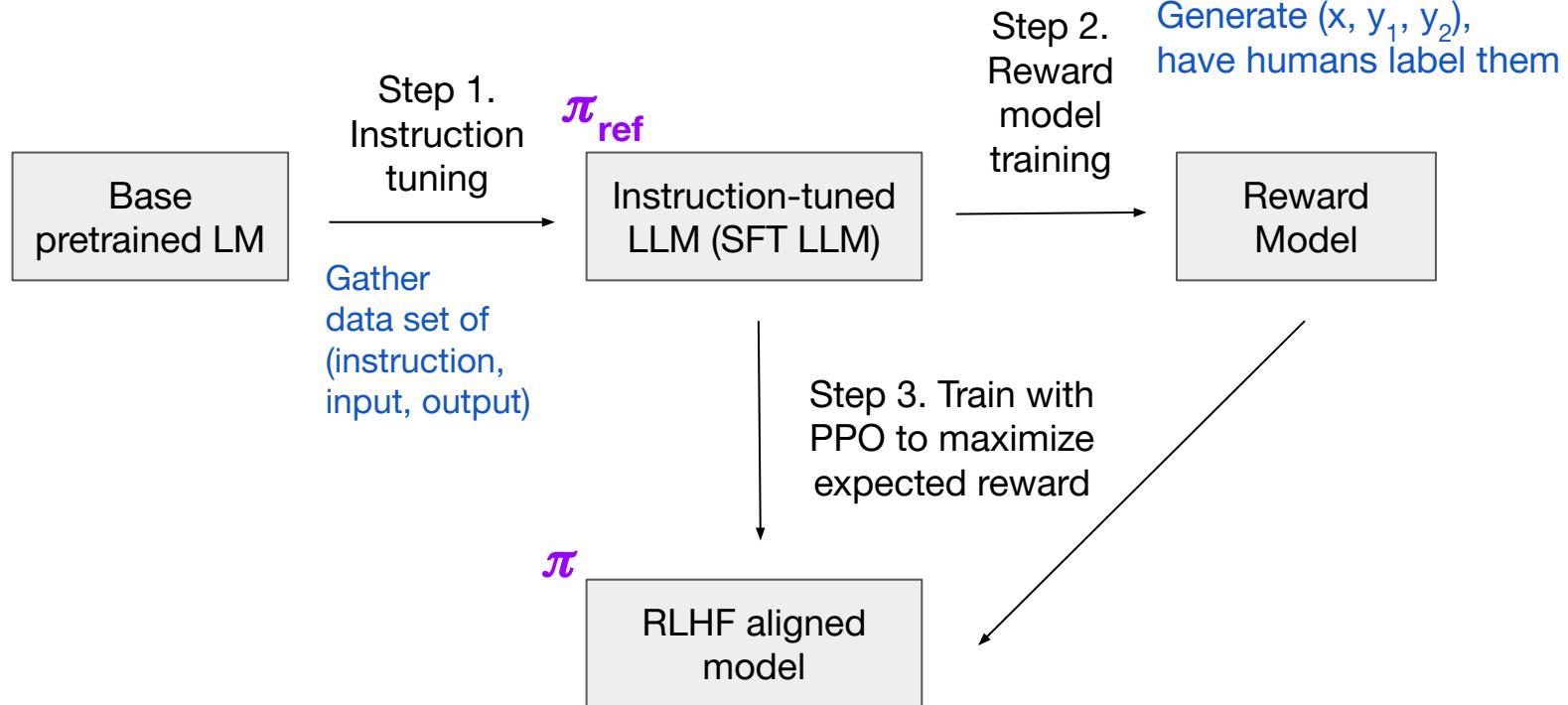
$\pi_{\text{ref}}(y \mid x)$  is the probability distribution of the SFT checkpoint (the reference model).

$D_{KL}(\pi \parallel \pi_{\text{ref}})$  is the Kullback-Leibler (KL) divergence term to regularize policy updates.

$\beta$  is a hyperparameter that controls the regularization strength.

- Optimize using Proximal Policy Optimization Algorithms (PPO) (Schulman et al. 2016) or REINFORCE (Williams et al. 1992)

# RLHF pipeline



Any Questions?