



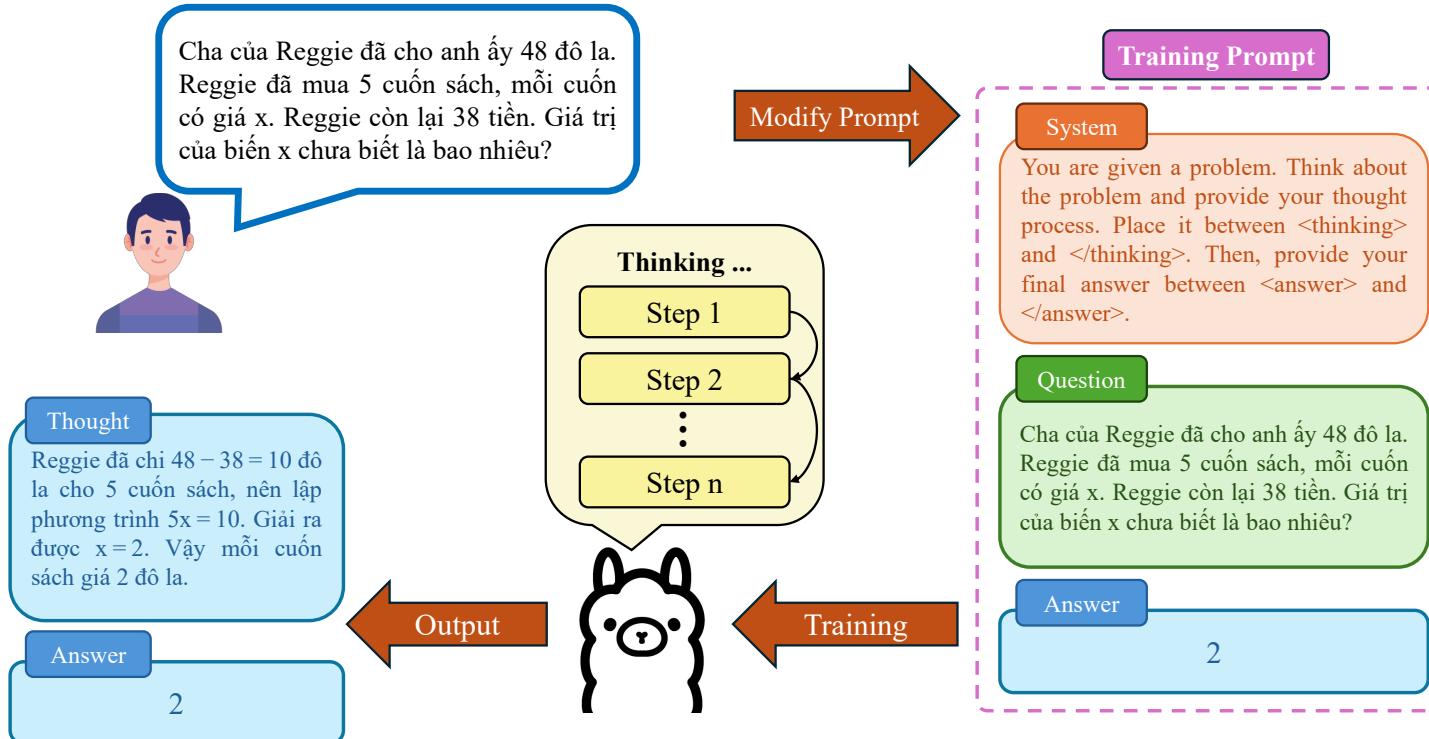
LLMs with Reasoning

LLMs Series

**Dinh-Thang Duong – TA
Truong-Binh Duong – STA**

Objectives

❖ Content



- In this study session, we will discuss about:**
1. Introduction to LLM with Reasoning.
 2. Introduction to prompting techniques to make LLMs reasoning.
 3. Introduction to LLM Reasoning.
 4. How to fine-tuning LLMs for Reasoning.
 5. How to implement an LLM Reasoning application for Math Solving.

Outline

- Introduction
- Reasoning through Prompting
- LLM Reasoning
- Math Solving with LLM Reasoning
- Question

 AI

AI VIET NAM
@aivietnam.edu.vn

Introduction

Introduction

❖ Getting Started

Textual Description (Prompt):

Write a python function that receive an image and plot it using matplotlib library.

 You
Write a python function that receive an image and plot it using matplotlib library.

 ChatGPT
To create a Python function that receives an image and plots it using the Matplotlib library, you can use the following code as an example. You'll need to have Matplotlib and a library like PIL (Pillow) installed to work with images:

```
python
import matplotlib.pyplot as plt
from PIL import Image

def plot_image(image_path):
    try:
        # Open the image using PIL (Pillow)
        img = Image.open(image_path)

        # Create a Matplotlib figure and axis
        fig, ax = plt.subplots()
```

 Copy code



Introduction

❖ What are Large Language Models (LLMs)?



You

write a python code that print "hello world"



ChatGPT

Certainly! Here's a simple Python code that prints "Hello, World!" to the console:

python

Copy code

```
print("Hello, World!")
```

You can run this code in a Python interpreter, and it will display "Hello, World!" as the output.



ChatGPT App:

1. User inputs a prompt (greetings, task...).
2. ChatGPT (Chatbot) outputs a appropriate response.

Introduction

❖ What are Large Language Models (LLMs)?

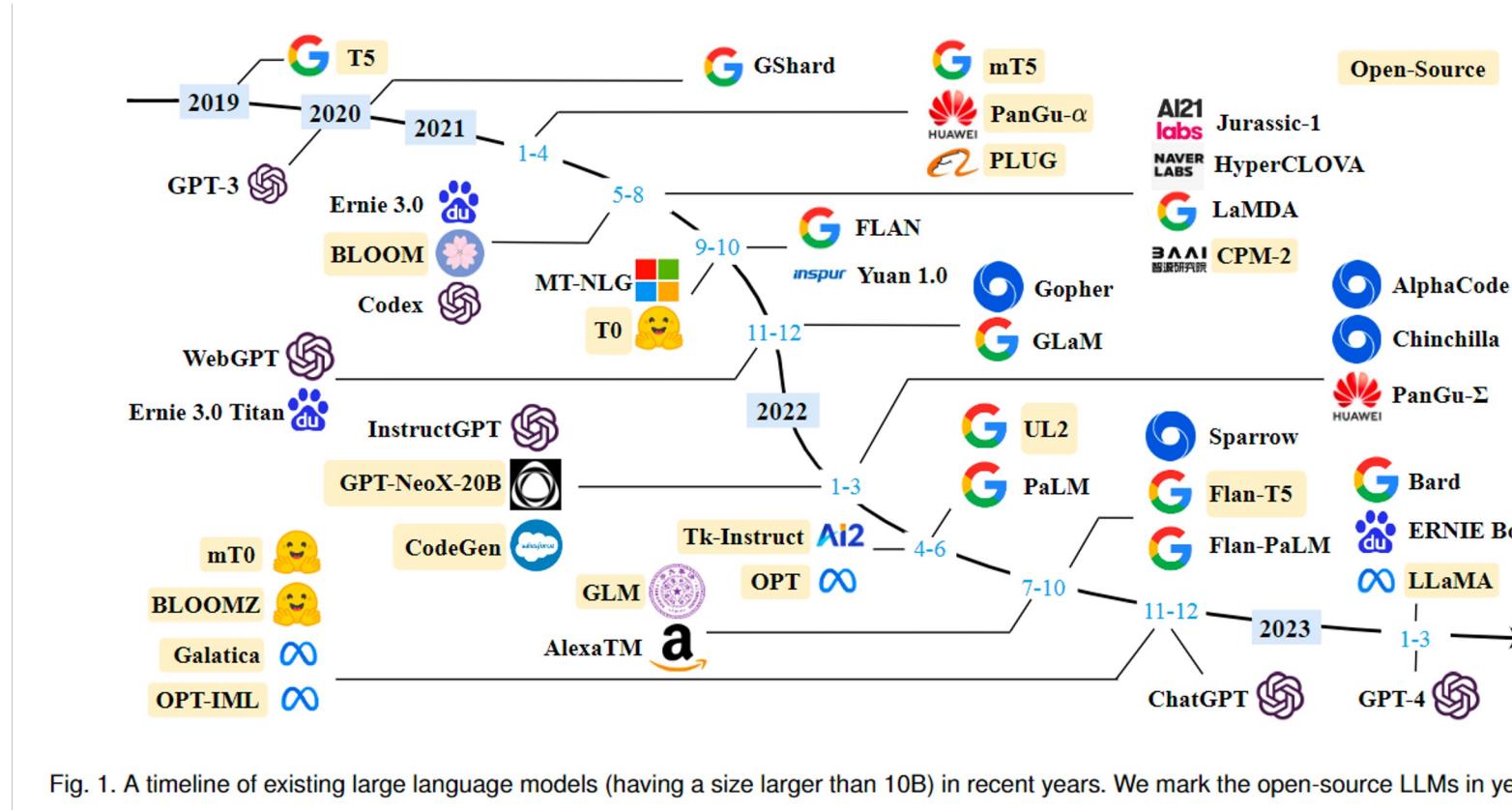


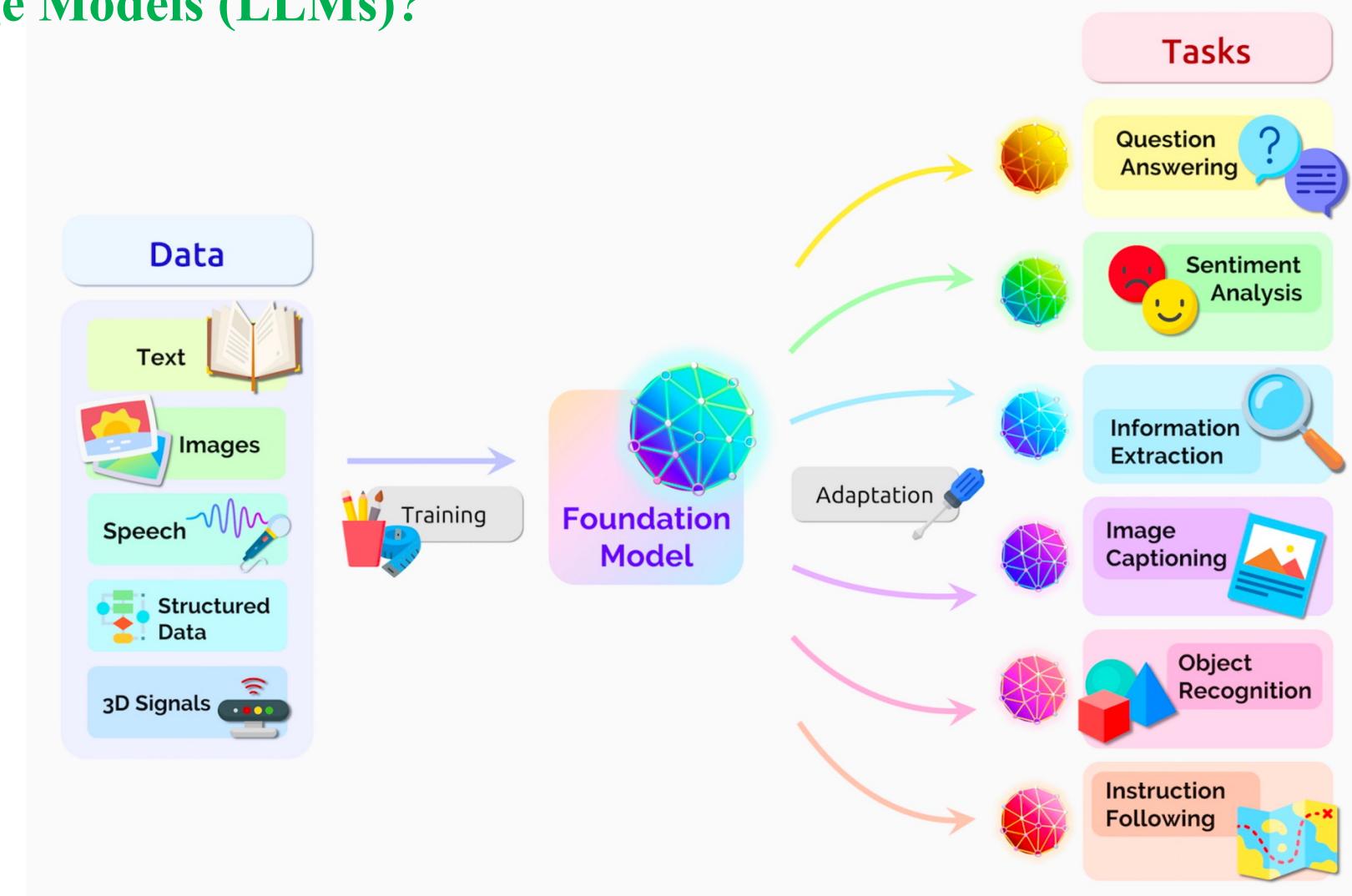
Fig. 1. A timeline of existing large language models (having a size larger than 10B) in recent years. We mark the open-source LLMs in yellow color.

LLMs (Large Language Models): AI models (language models) that were trained on a very large corpus of text. This made them capable of performing various NLP tasks with high precision.

Introduction

❖ What are Large Language Models (LLMs)?

LLMs are often pretrained on a vast majority of data and designed to be adaptable to a wide variety of tasks (Foundation models).

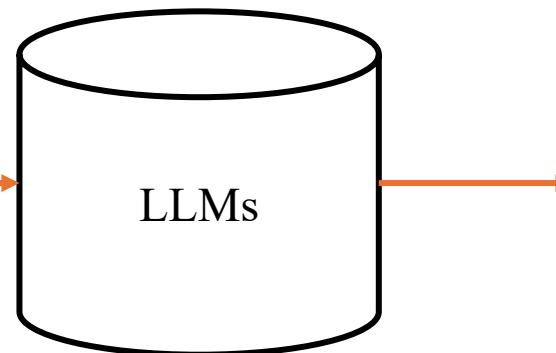


Introduction

❖ LLMs I/O

Input Text

Write a python function that receive an image and plot it using matplotlib library.



Given a “prompt”, LLMs can generate an appropriate response.

Output Text

AI You Write a python function that receive an image and plot it using matplotlib library.

ChatGPT To create a Python function that receives an image and plots it using the Matplotlib library, you can use the following code as an example. You'll need to have Matplotlib and a library like PIL (Pillow) installed to work with images:

```
python
import matplotlib.pyplot as plt
from PIL import Image

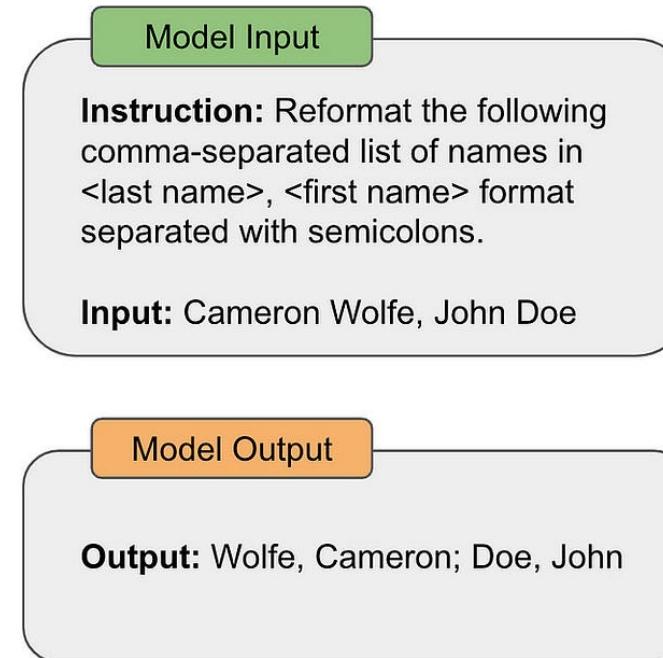
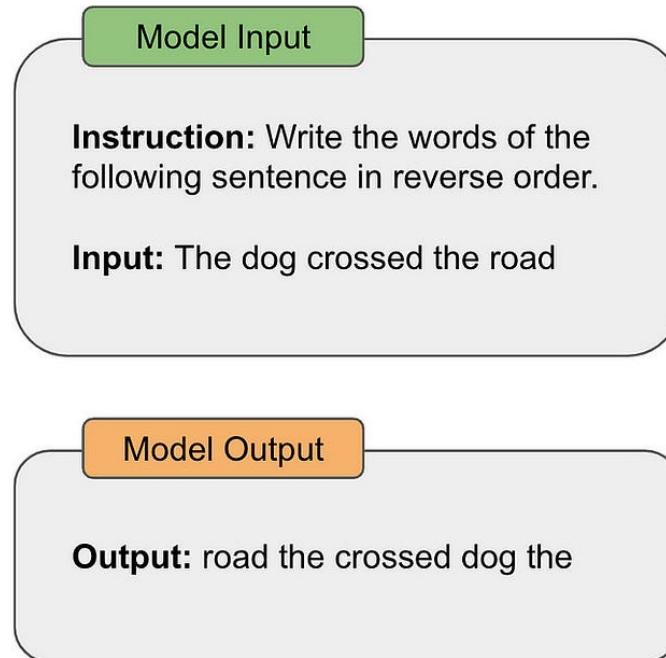
def plot_image(image_path):
    try:
        # Open the image using PIL (Pillow)
        img = Image.open(image_path)

        # Create a Matplotlib figure and axis
        fig, ax = plt.subplots()
```

Copy code

Introduction

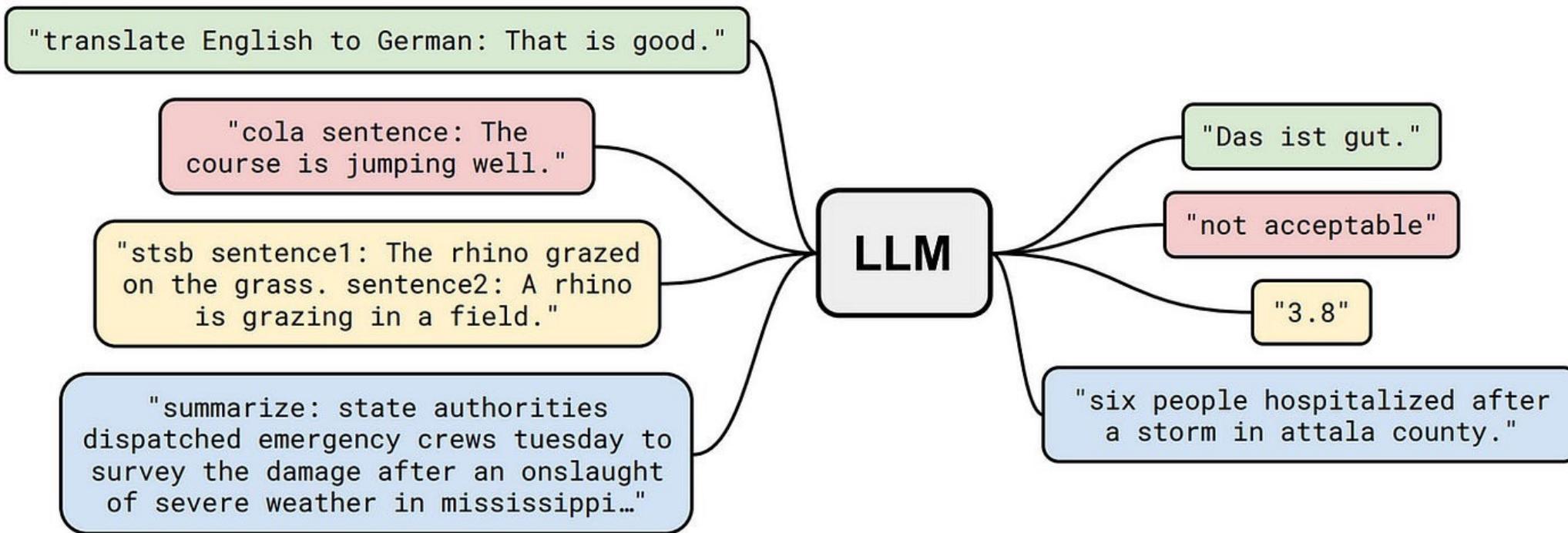
❖ Generative AI Prompting



Prompting: Refers to a process of providing an input, usually in the form of text data, to a generative AI model to generate a specific output.

Introduction

❖ Prompting in LLMs



With prompting, we can make LLMs do any task with just natural language (**zero-shot capability**).

Introduction

❖ Getting Started

<|im_start|> system

Bạn là một trợ lí AI hữu ích. Hãy trả lời người dùng một cách chính xác.

<|im_start|> user

Mô tả về thành phố Đà Lạt

<|im_start|> assistant

Đà Lạt là một thành phố đẹp như tranh vẽ nằm ở vùng cao nguyên của tỉnh Lâm Đồng, Việt Nam. Được biết đến như "Thành phố ngàn hoa", Đà Lạt nổi tiếng với những khu vườn tươi tốt, những con đường đẹp như tranh vẽ và những ngôi nhà đầy màu sắc. Thành phố này có khí hậu mát mẻ, với nhiệt độ trung bình từ 15 đến 20 độ C vào những tháng mùa đông. Vào những tháng mùa hè, nhiệt độ có thể lên tới 25 độ C. Đà Lạt nổi tiếng với các món ăn ngon, từ các món ăn truyền thống Việt Nam đến các món ăn quốc tế. Thành phố cũng nổi tiếng với các lễ hội và sự kiện văn hóa, chẳng hạn như Lễ hội hoa hồng, Lễ hội mùa đông và Lễ hội mùa xuân. Đà Lạt là một điểm đến tuyệt vời cho những người yêu thích thiên nhiên, những người đam mê ẩm thực và những người tìm kiếm một nơi yên bình để thư giãn.

Example: AI Chatbot using Large Language Models (LLMs).

Introduction

❖ LLMs problem



$241 - (-241) + 1$



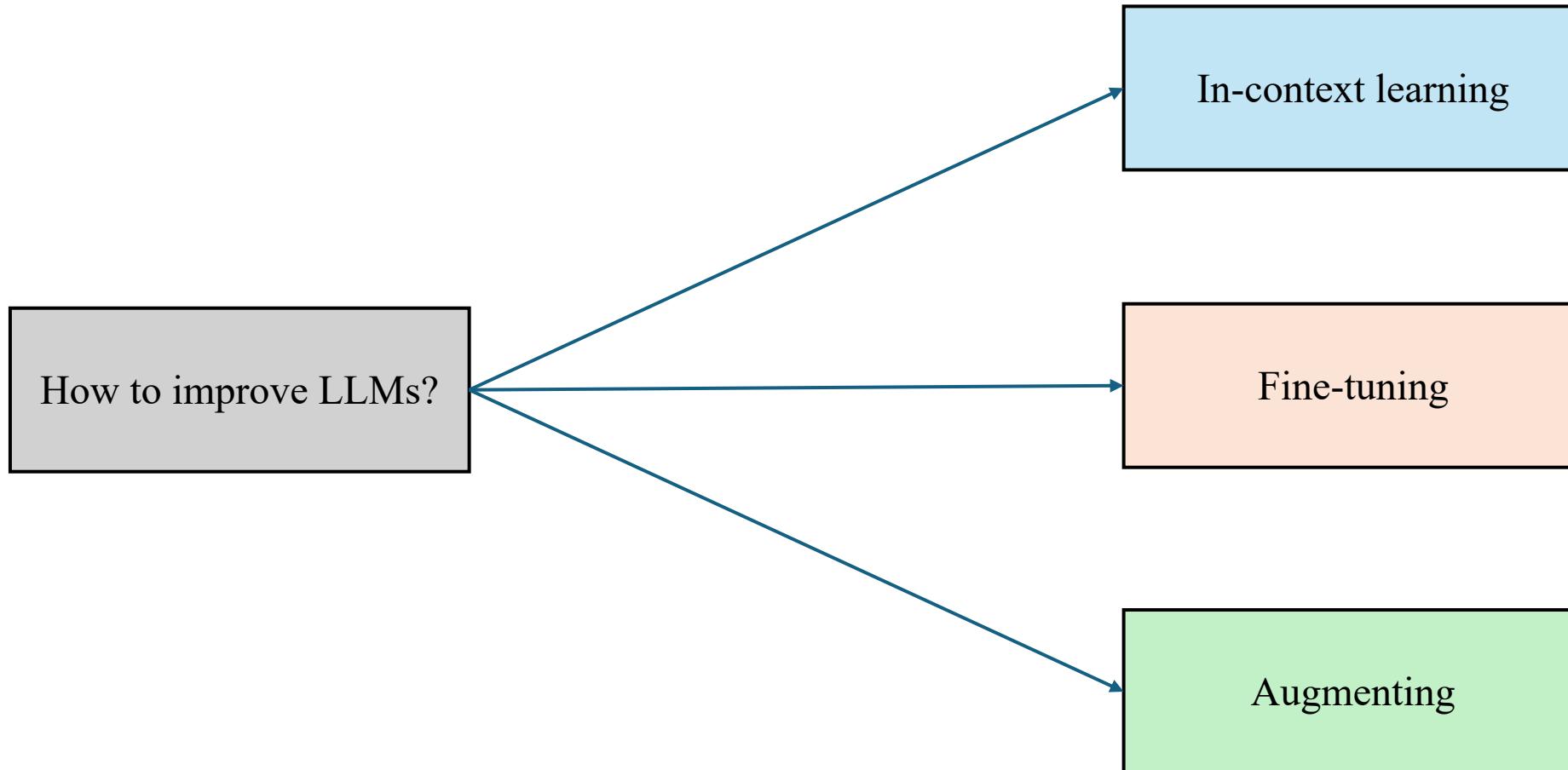
$241 - (-241) + 1$ is equivalent to $241 + 241 + 1$, which simplifies to $483 + 1$. So $241 - (-241) + 1$ is equal to 484.



Question: How can we improve LLMs on a specific task?

Introduction

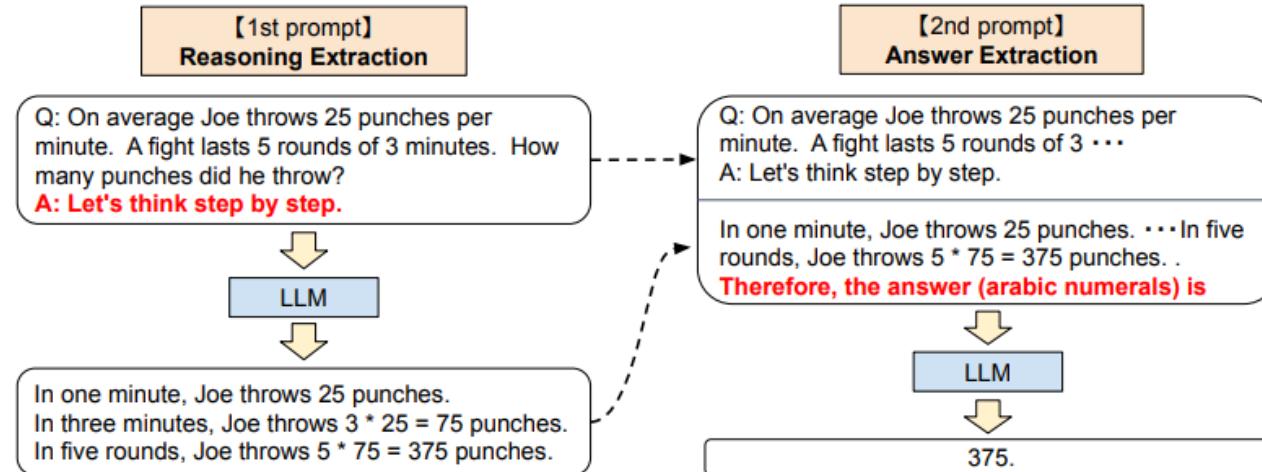
❖ How to improve LLMs on specific tasks?



Reasoning through Prompting

Reasoning through Prompting

❖ Introduction



In-context learning (ICL) in LLMs is a technique where task demonstrations are integrated into the prompt in a natural language format. This approach allows pre-trained LLMs to address new tasks without fine-tuning the model.

Reasoning through Prompting

❖ Example

Input: 2025-04-20

Output: !04!20!2025

Input: 2024-05-29

Output: !05!29!2024

Input: 2018-07-03

Output: !07!03!2018

Input: 2025-04-23

Expected Output: !04!23!2025

In-context examples

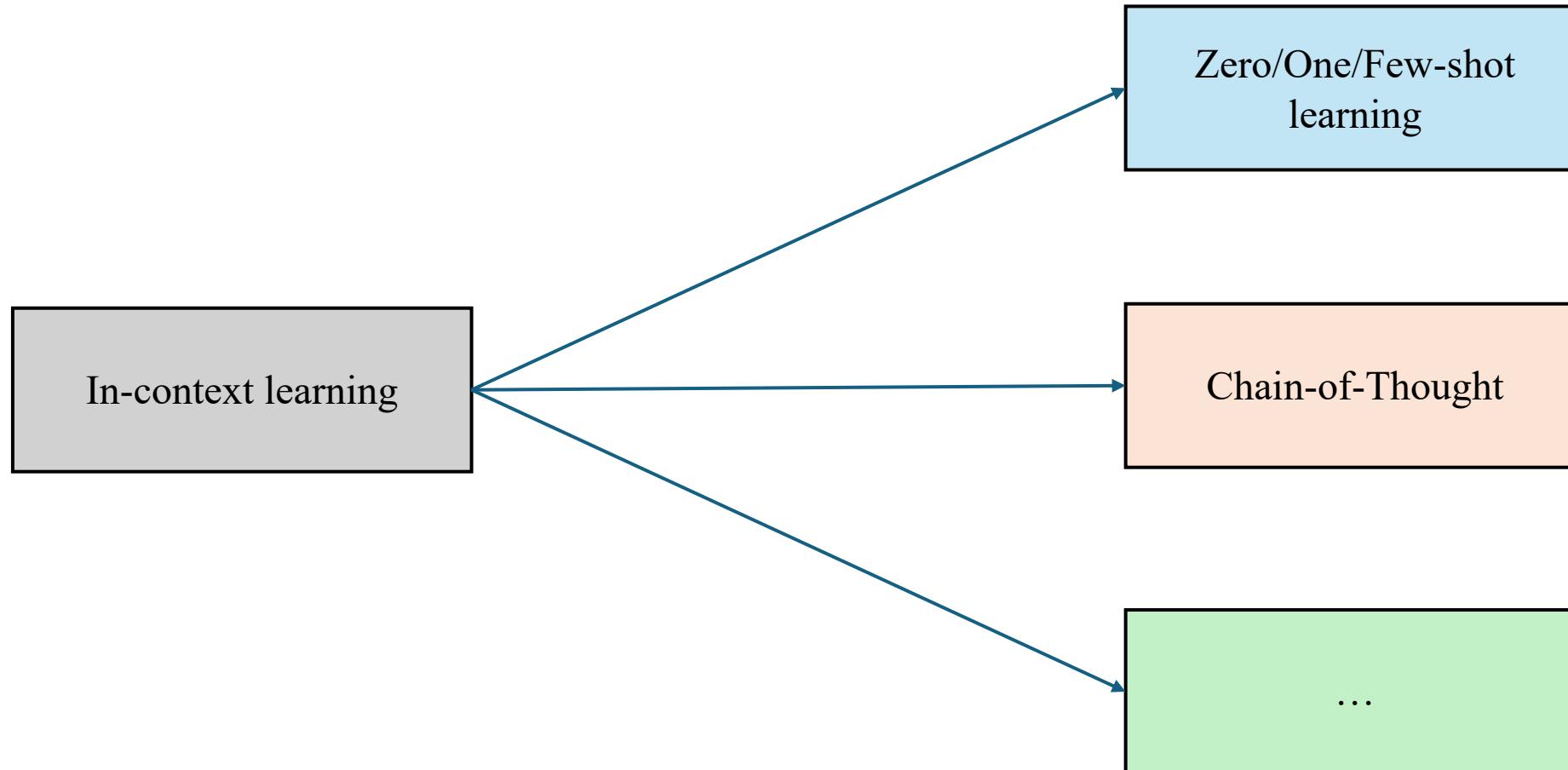
Test case

LLM Response

- ❖ Make LLMs adapt to a task using instruction and examples.

Reasoning through Prompting

❖ In-context learning type





Reasoning through Prompting

❖ Zero-shot learning

Prompt: Who won the 2014 World Cup?

GPT-4 Response: Germany won the 2014 FIFA World Cup. The tournament was held in Brazil, and Germany defeated Argentina 1–0 in the final, with the match-winning goal scored by Mario Götze during extra time

Zero-shot learning: Prompting LLMs to do a task without any examples.

Reasoning through Prompting

❖ One-shot learning

Prompt: Answer the question using the format shown in the context.

Who won the World Cup in **2014**?

Germany won the **2014** FIFA World Cup. They defeated Argentina in the final. It was a low scoring game.

Who won the world cup in **2018**?

GPT-4 Response: France won the **2018** FIFA World Cup. They defeated Croatia in the final. It was a higher scoring game compared to the **2014** final.

One-shot learning: Prompting LLMs to do a task with an example.



Reasoning through Prompting

❖ Few-shot learning

Prompt: Answer the question using the format shown in the context.

Who won the World Cup in **2014**?

Germany won the **2014** FIFA World Cup. They defeated Argentina in the final. It was a low scoring game.

Who won the World Cup in **2018**?

France won the **2018** FIFA World Cup. They defeated Croatia in the final. It was a high scoring game.

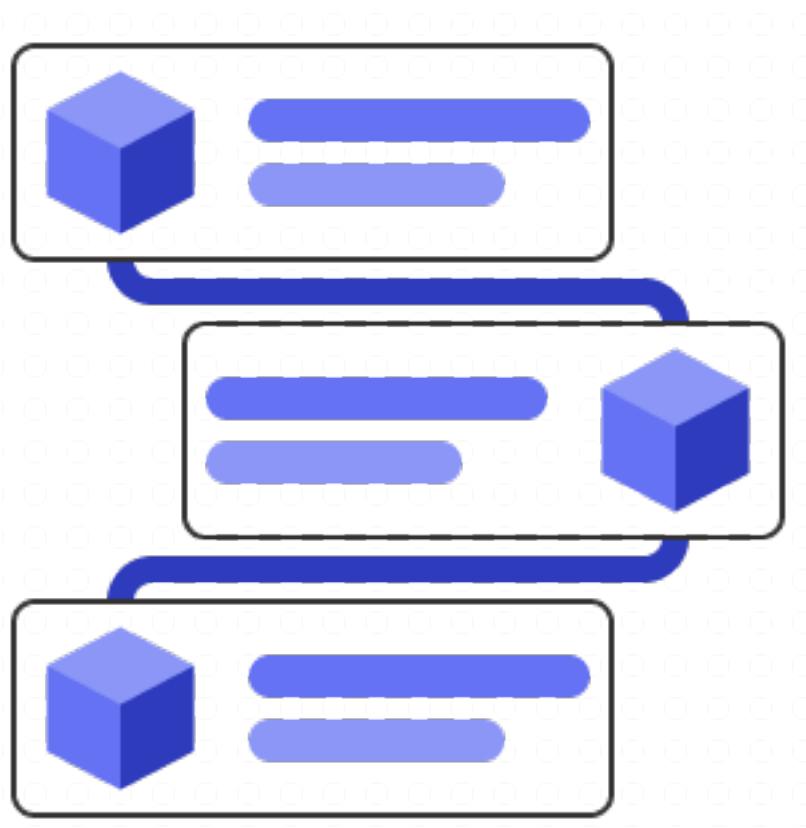
Who won the world cup in **2022**?

Few-shot learning: Prompting LLMs to do a task with more than one example.

GPT-4 Response: Argentina won the **2022** FIFA World Cup. They defeated France in the final. It was a high scoring game.

Reasoning through Prompting

❖ Chain-of-Thought Prompting



Chain-of-Thought (CoT) prompting enables complex reasoning capabilities through intermediate reasoning steps. You can combine it with few-shot prompting to get better results on more complex tasks that require reasoning before responding.

Reasoning through Prompting

❖ Chain-of-Thought

Standard Prompting

Q: An có 7 quyển sách. Bạn ấy mua thêm 4 bộ sách, mỗi bộ có 2 quyển. Hỏi An có tất cả bao nhiêu quyển sách?

A: Đáp án là 15.

Q: Một cửa hàng có 15 chiếc bánh. Bán đi 10 chiếc và nhập thêm 5 chiếc mới. Hỏi cửa hàng còn bao nhiêu chiếc bánh?

Output

A: Đáp án là 20. X

Chain-of-Thought Prompting

Q: An có 7 quyển sách. Bạn ấy mua thêm 4 bộ sách, mỗi bộ có 2 quyển. Hỏi An có tất cả bao nhiêu quyển sách?

A: An ban đầu có 7 quyển. $4 \text{ bộ} \times 2 \text{ quyển} = 8 \text{ quyển}$.
Tổng cộng là $7 + 8 = 15 \text{ quyển}$. Đáp án là 15.

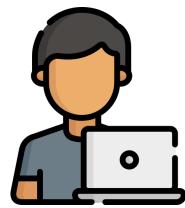
Q: Một cửa hàng có 15 chiếc bánh. Bán đi 10 chiếc và nhập thêm 5 chiếc mới. Hỏi cửa hàng còn bao nhiêu chiếc bánh?

Output

A: Cửa hàng có 15 chiếc bánh. Bán 10 chiếc còn 5 chiếc.
Nhập thêm 5 chiếc nữa, $5 + 5 = 10$. Đáp án là 10. ✓

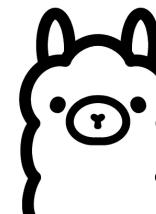
Reasoning through Prompting

❖ Chain-of-Thought

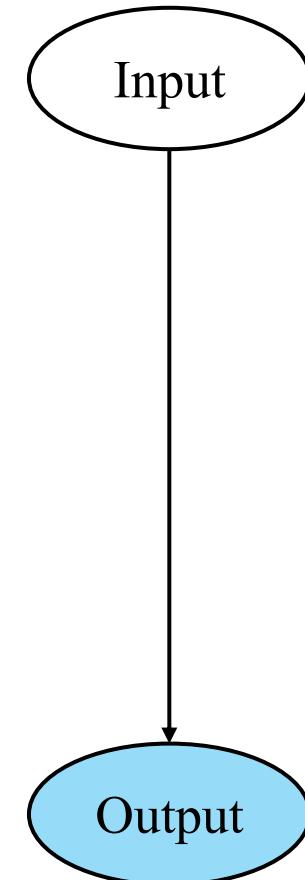


What is the smallest positive perfect cube that can be written as the sum of three consecutive integers?

Answer: 1008 ✗



Llama-3.2-3B-Instruct



Standard Prompting



Reasoning through Prompting

❖ Chain-of-Thought

```
1 math_dataset = '/mnt/VLAI_data/MATH-500/test.jsonl'  
2  
3 with open(math_dataset, 'r') as f:  
4     data = [json.loads(line) for line in f]
```

```
1 model_id = "meta-llama/Llama-3.2-3B-Instruct"  
2 pipe = pipeline(  
3     "text-generation",  
4     model=model_id,  
5     torch_dtype=torch.bfloat16,  
6     device_map="auto",  
7 )
```

```
1 question = data[6]['problem']  
2  
3 system_prompt = """You will be given a math problem and need to provide a concise factoid answer in Markdown format.  
4 Answer in the form: `Answer: <answer>`. The answer must be a single number or a mathematical expression."""  
5  
6 messages = [  
7     {"role": "system", "content": system_prompt},  
8     {"role": "user", "content": f"Problem: {question}"}  
9 ]  
10  
11 outputs = pipe(  
12     messages,  
13     max_new_tokens=100,  
14 )  
15 print(f"Problem: {question}")  
16 print(outputs[0]["generated_text"][-1]["content"].strip())
```

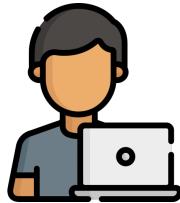
Setting `pad_token_id` to `eos_token_id`:128001 for open-end generation.

Problem: What is the smallest positive perfect cube that can be written as the sum of three consecutive integers?

Answer: 1008

Reasoning through Prompting

❖ Chain-of-Thought



Think step by step to solve this question and show your intermediate reasoning. What is the smallest positive perfect cube that can be written as the sum of three consecutive integers?

Let's consider three consecutive integers: $n, n+1, n+2$. Their sum is: $n + (n+1) + (n+2) = 3n + 3$.

We want $3n + 3$ to be a perfect cube. Try small perfect cubes: 1, 8, 27, 64, ... Check which can be written as $3n + 3$.

Start with 8: $8 = 3(1) + 5 \rightarrow$ not valid.

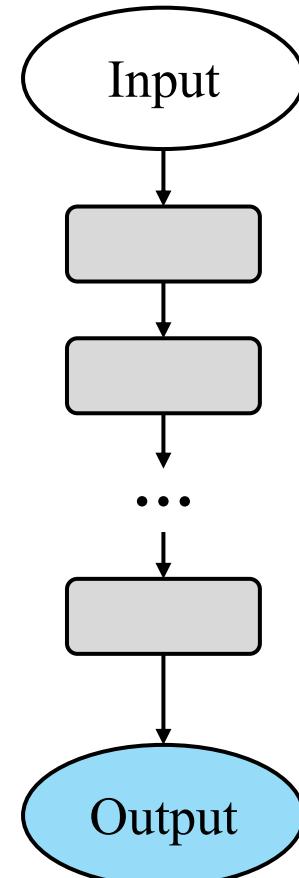
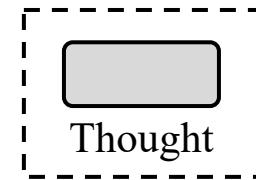
Next, 27: $27 = 3(3) + 3 \rightarrow$ valid!

So, 27 is the smallest perfect cube of $3n + 3$.

Answer: 27. 



Llama-3.2-3B-Instruct



Chain-of-Thought Prompting

Reasoning through Prompting

❖ Chain-of-Thought

```
1 system_prompt_cot = """You will be given a math problem. Think step by step to solve it and show your intermediate reasoning.
2 Answer in the form: `Answer: <single number or expression>`."""
3
4 messages = [
5     {"role": "system", "content": system_prompt_cot},
6     {"role": "user", "content": f"Problem: {question}"}
7 ]
8 out = pipe(messages, max_new_tokens=2000, do_sample=False)
9
10 print(f"Problem: {question}")
11 print(out[0]["generated_text"][-1]["content"].strip())
```

Problem: What is the smallest positive perfect cube that can be written as the sum of three consecutive integers?
To solve this problem, let's consider the general form of three consecutive integers: n , $n+1$, and $n+2$.

We want to find the smallest positive perfect cube that can be written as the sum of these three consecutive integers.

The sum of the three consecutive integers is: $n + (n+1) + (n+2) = 3n + 3$.

We need to find the smallest positive integer n such that $3n + 3$ is a perfect cube.

Let's try to find the smallest perfect cube that can be written in the form $3n + 3$.

We can start by listing some perfect cubes: 1, 8, 27, 64, 125,...

We can see that $27 = 3(3) + 3$, which is in the form $3n + 3$.

However, we need to check if 27 is the smallest perfect cube that can be written in this form.

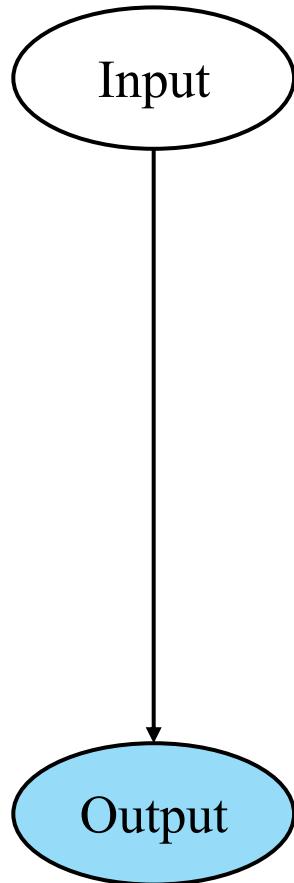
Let's check the previous perfect cube, which is 8. We can see that $8 = 3(1) + 5$, which is not in the form $3n + 3$.

Therefore, 27 is indeed the smallest positive perfect cube that can be written as the sum of three consecutive integers.

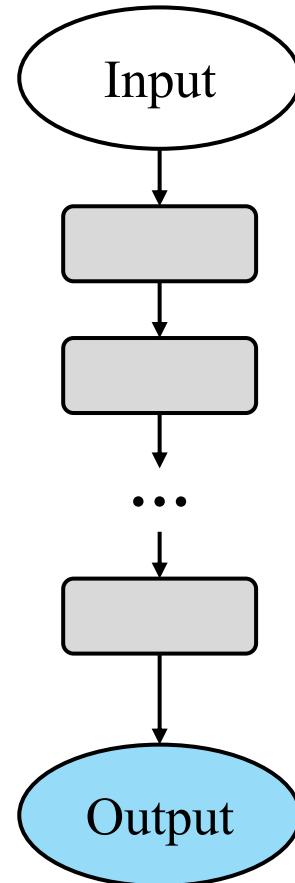
Answer: 27

Reasoning through Prompting

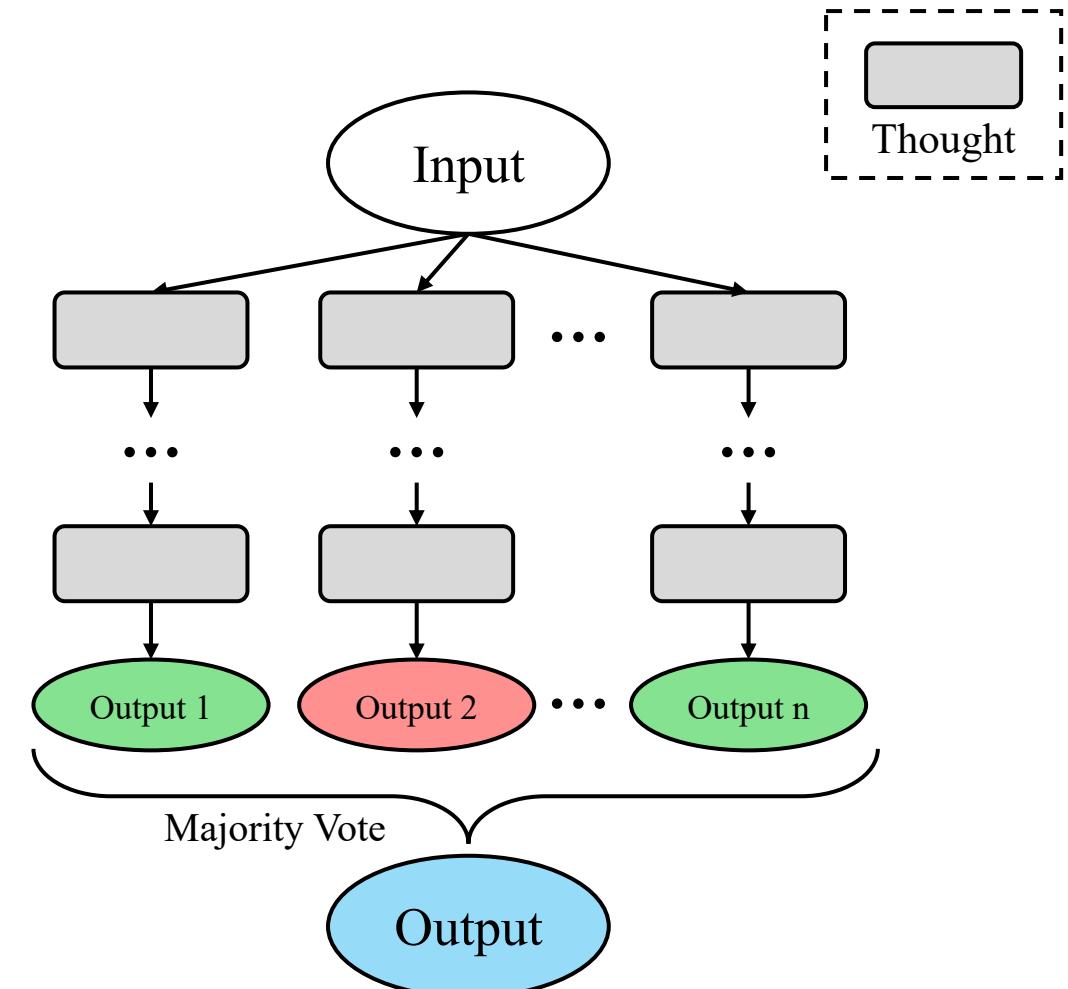
❖ Self-Consistency



Standard Prompting



Chain-of-Thought Prompting



Self-Consistency with CoT

Reasoning through Prompting

❖ Self-Consistency



What is the smallest positive perfect cube that can be written as the sum of three consecutive integers?

Let's consider ... \Rightarrow ... \Rightarrow
Answer: 27

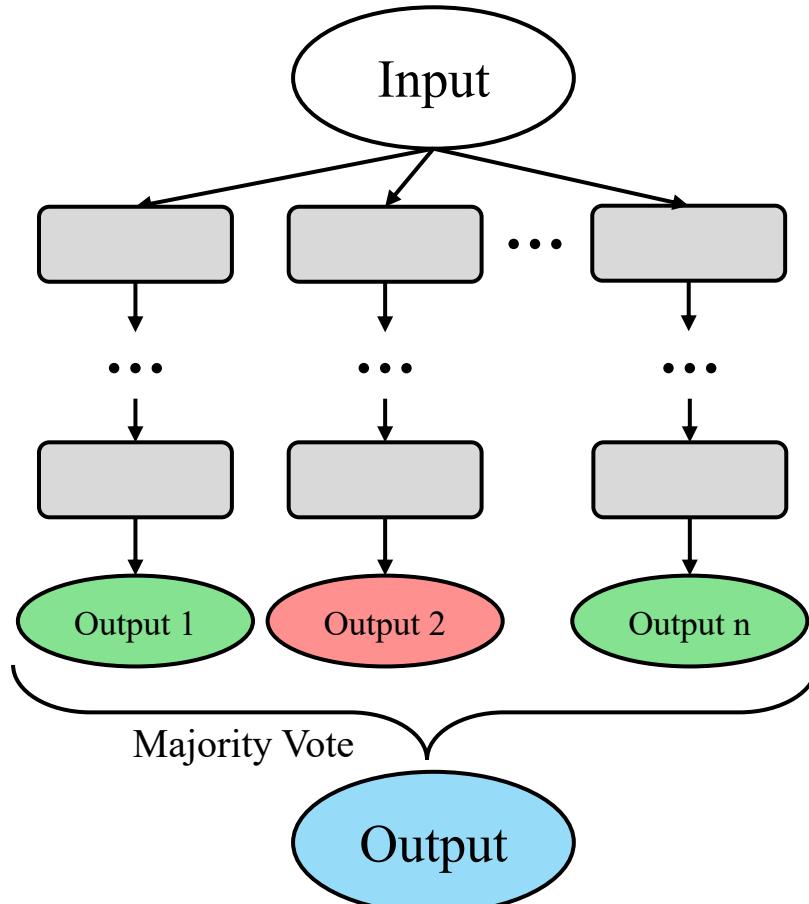
We want to... => ... =>
Answer: 9

To solve this... $\Rightarrow \dots \Rightarrow$
Answer: 27

Majority Vote: 27 ✓



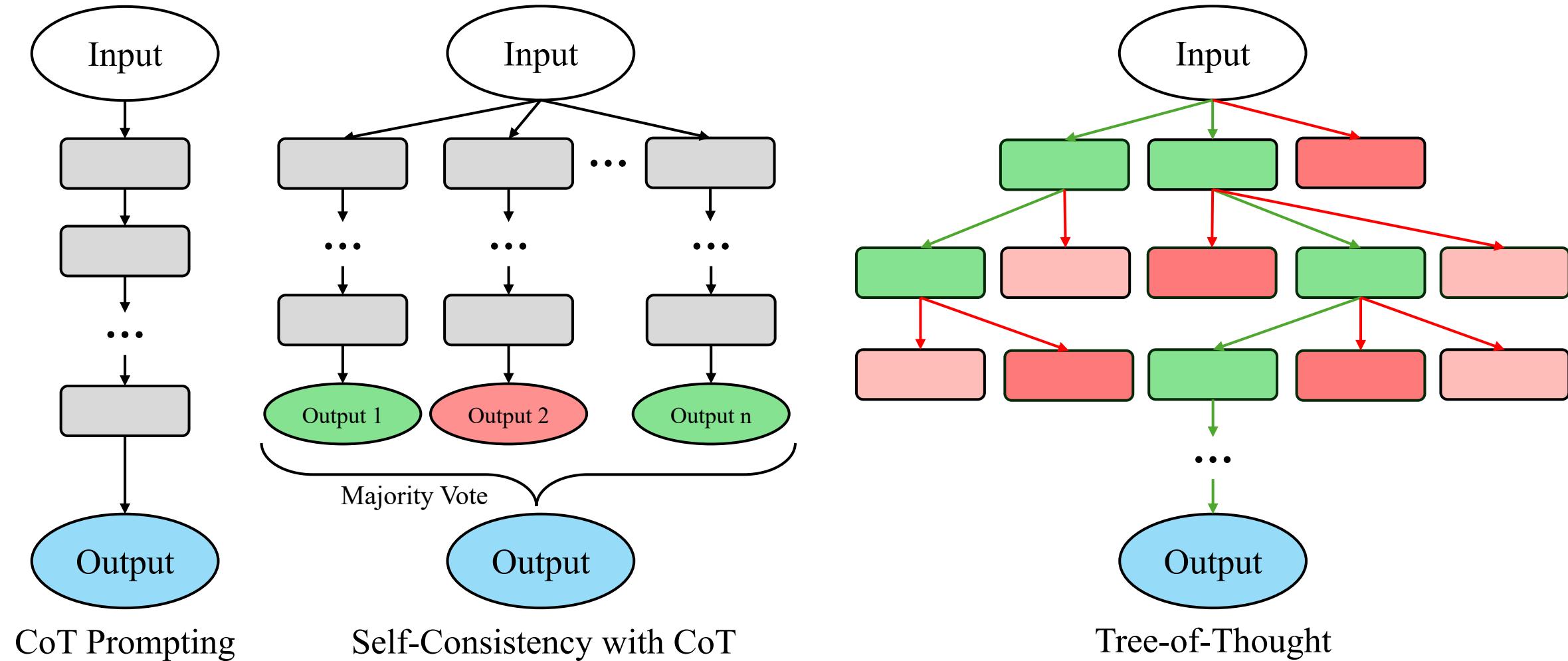
Llama-3 2-3B-Instruct



Self-Consistency with CoT

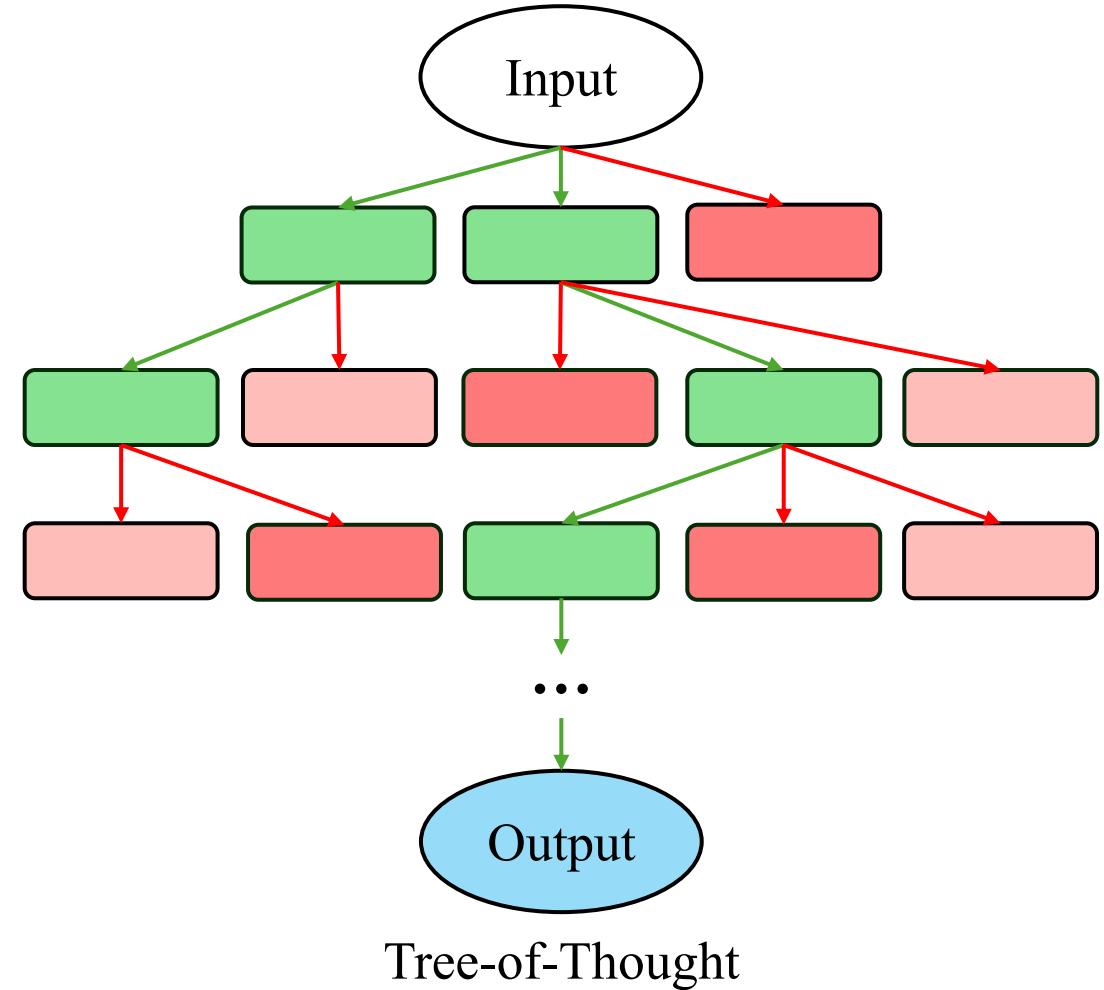
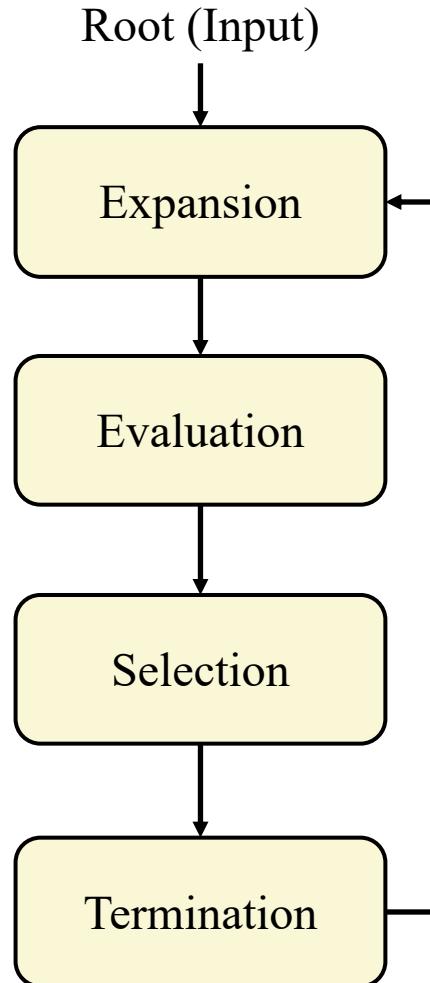
Reasoning through Prompting

❖ Tree-of-Thought



Reasoning through Prompting

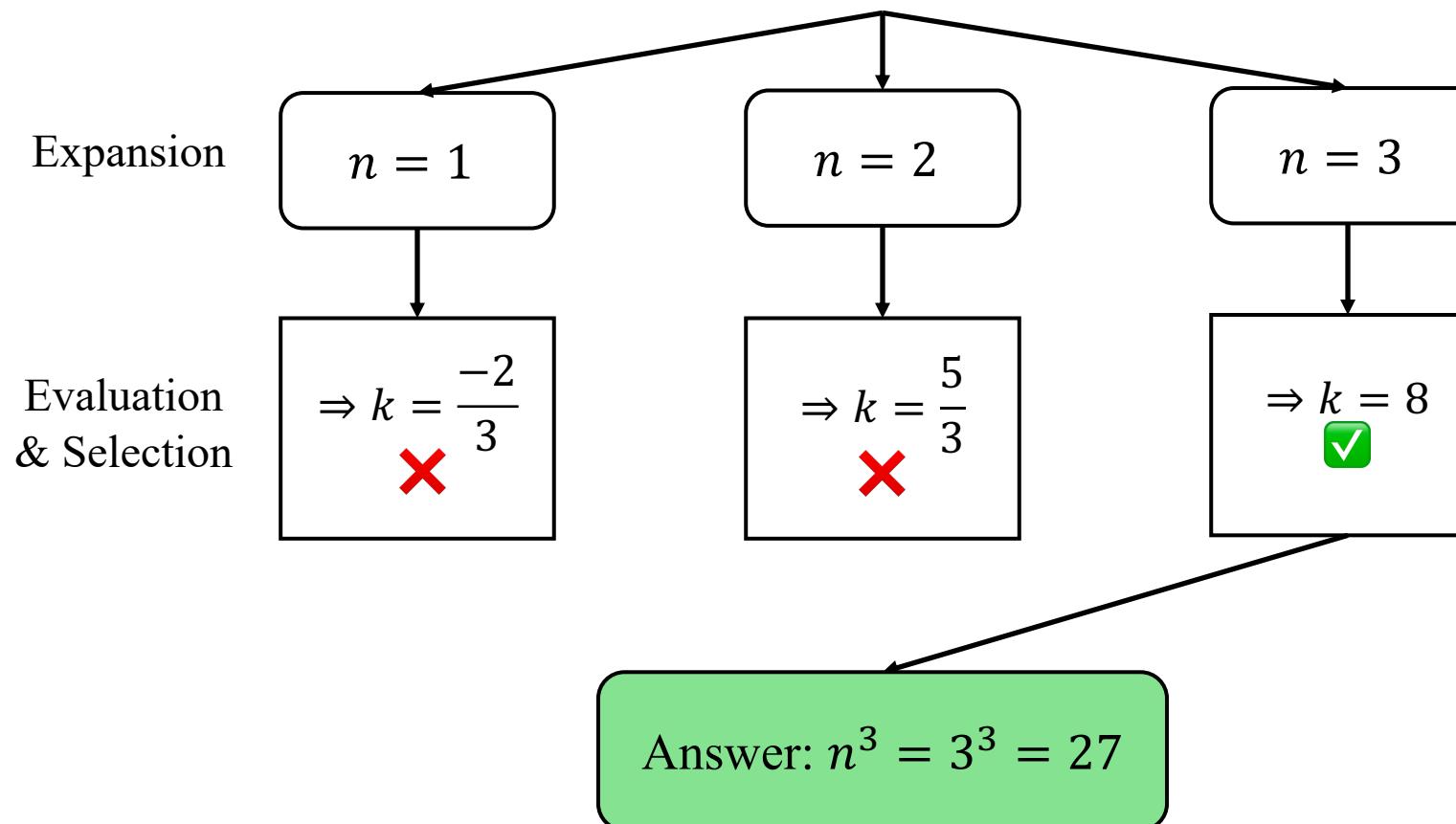
❖ Tree-of-Thought



Reasoning through Prompting

❖ Tree-of-Thought

What is the smallest positive perfect cube that can be written as the sum of three consecutive integers? \Leftrightarrow Find $n^3 = 3(k + 1), k \in \mathbb{Z}^+$





AI

AI VIET NAM
@aivietnam.edu.vn

LLM Reasoning

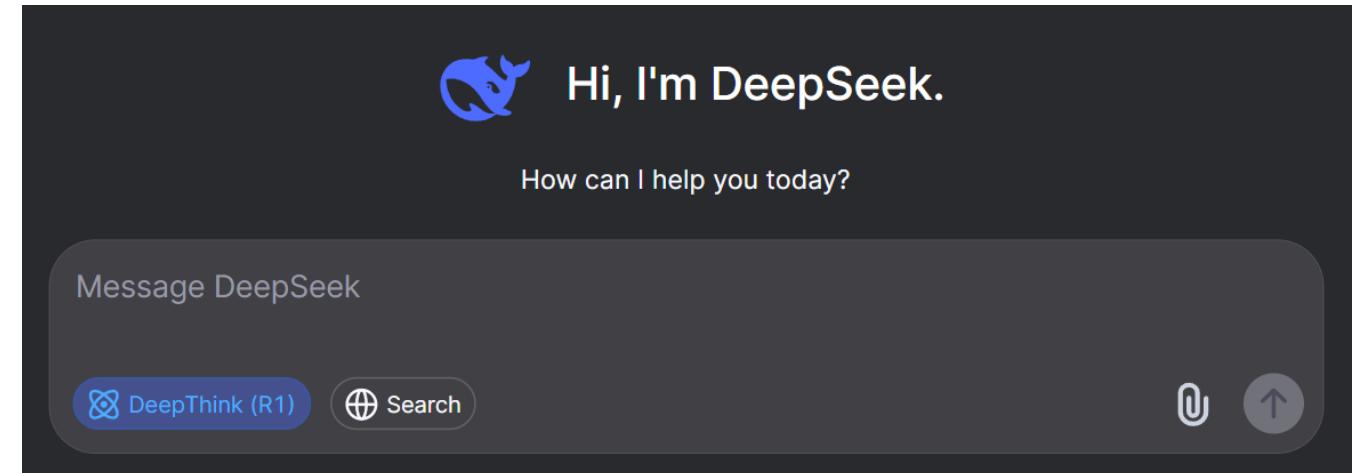


LLM Reasoning

❖ DeepSeek



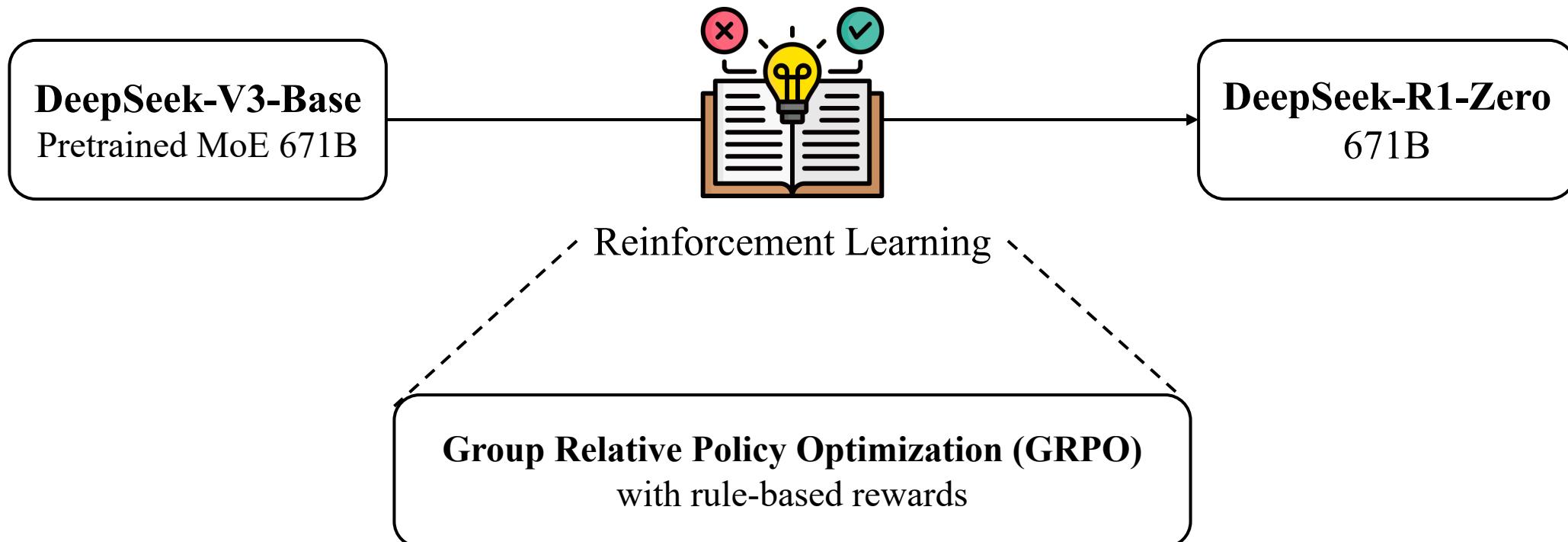
V3 & R1



LLM Reasoning

❖ DeepSeek-R1-Zero

Without any Supervised Fine-tuning (SFT) data



LLM Reasoning

❖ DeepSeek-R1-Zero

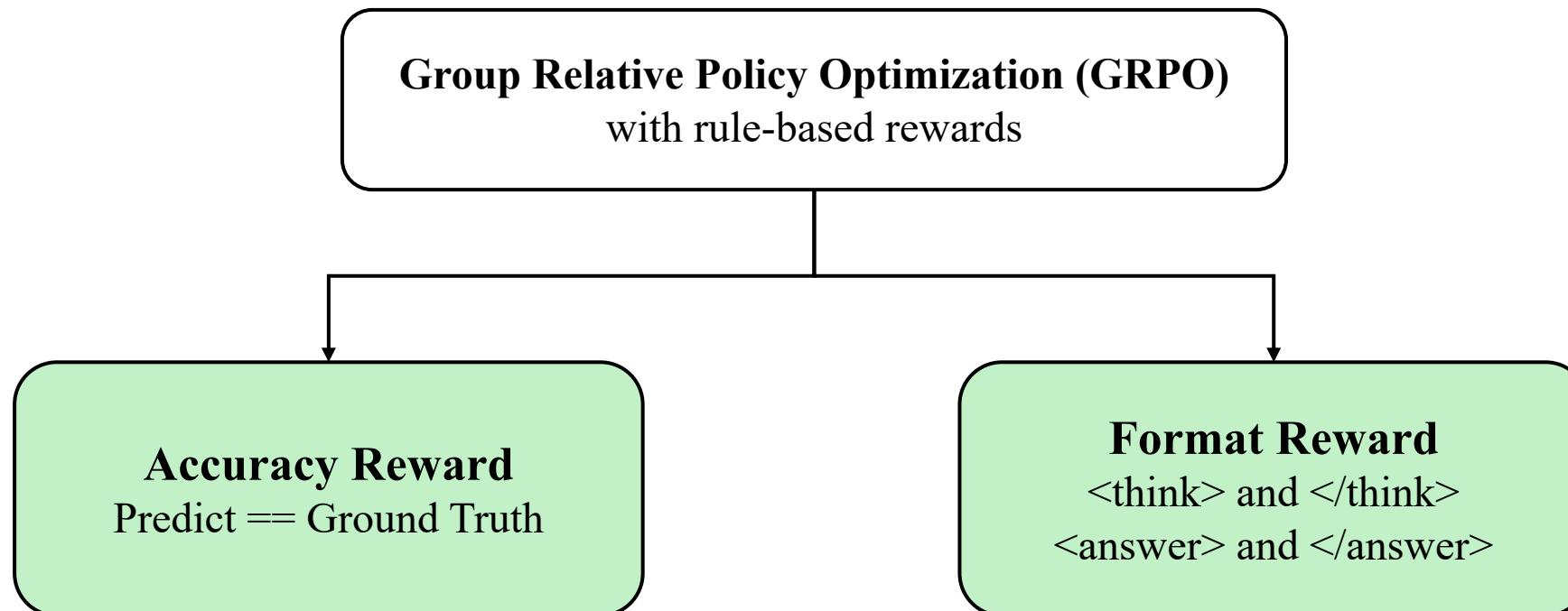
Prompt Template



A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the **reasoning** process in the mind and then provides the user with the **answer**. The reasoning process and answer are enclosed within **<think> </think>** and **<answer> </answer>** tags, respectively, i.e., **<think> reasoning process here </think> <answer> answer here </answer>**.

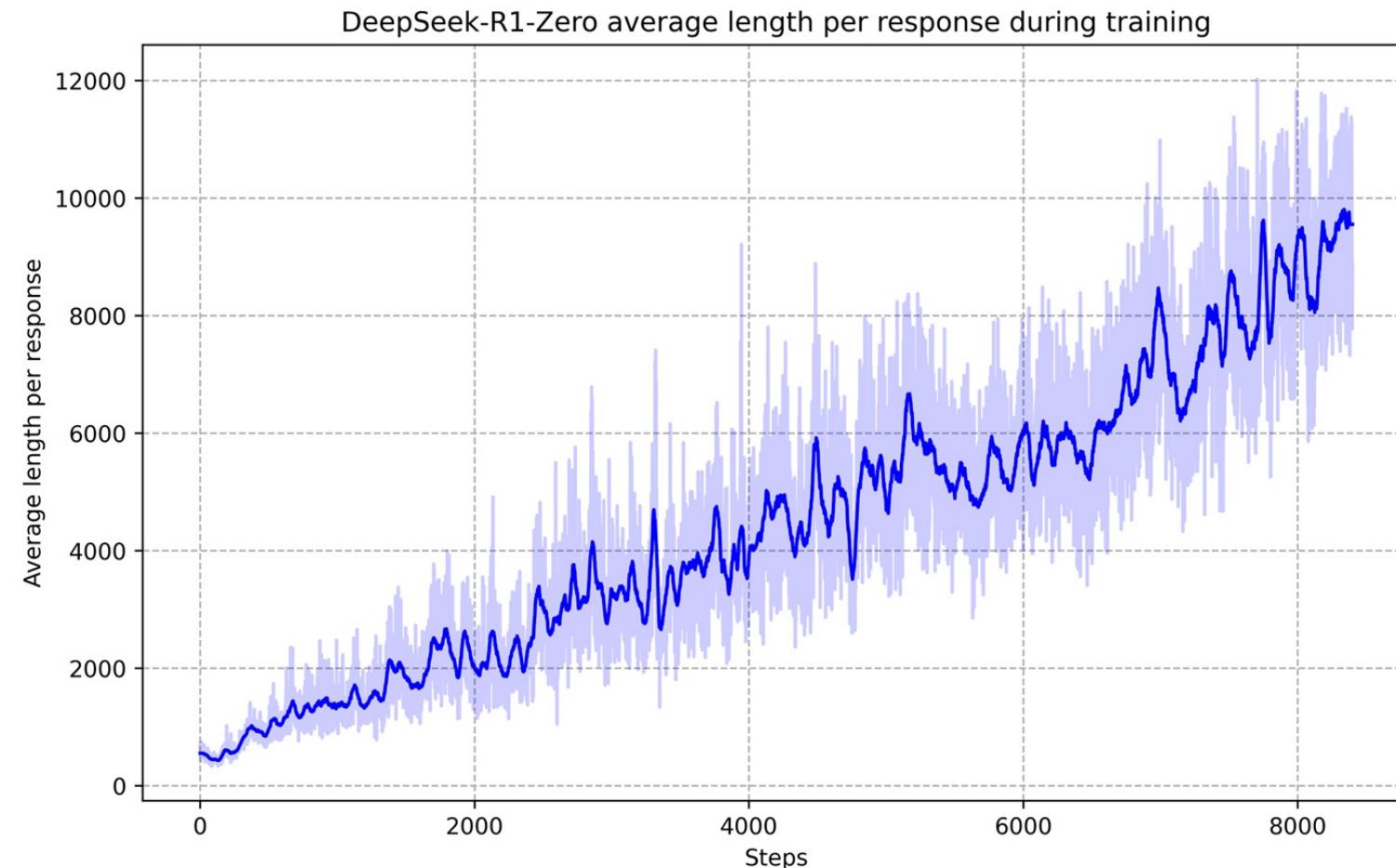
LLM Reasoning

❖ DeepSeek-R1-Zero



LLM Reasoning

❖ DeepSeek-R1-Zero





LLM Reasoning

❖ DeepSeek-R1-Zero

Highlights

- Learns reflection, self-checking, long CoT
- Shows self-evolution and "aha moment"
- First to prove LLMs can learn reasoning via RL only

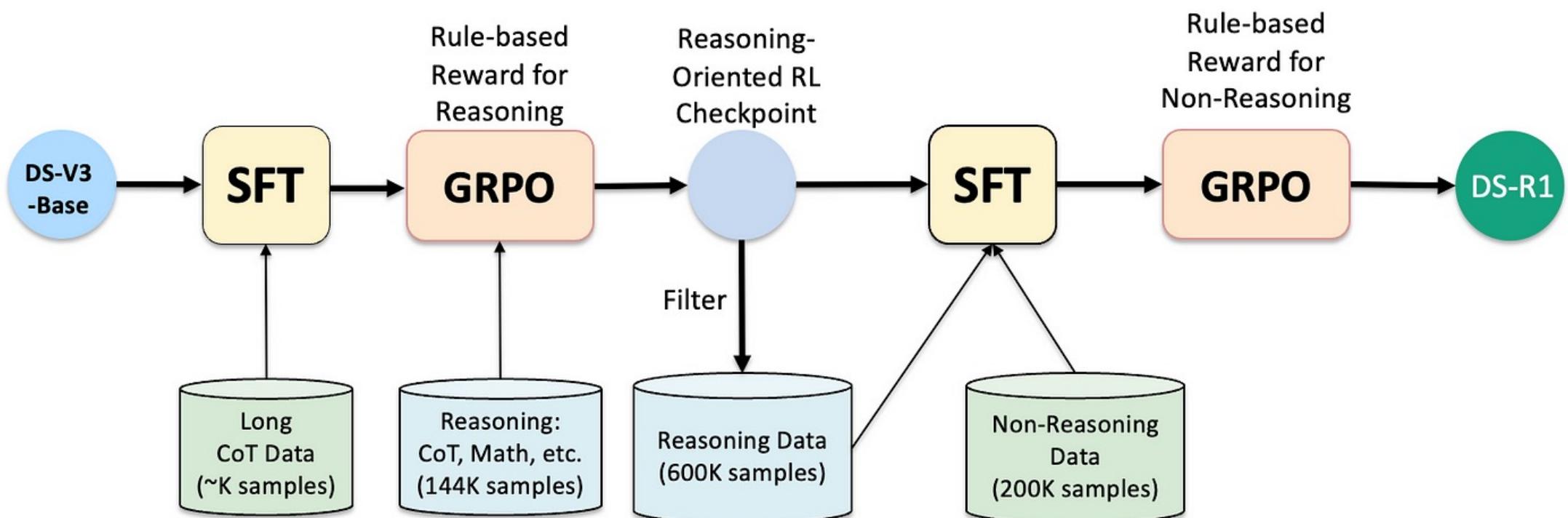
Limitations

- Low readability, hard-to-read outputs
- Language mixing (e.g., English + Chinese)
- Not ready for practical use without refinement

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64				
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820
OpenAI-o1-0912	74.4	83.3	94.8	77.3	63.4	1843
DeepSeek-R1-Zero	71.0	86.7	95.9	73.3	50.0	1444

LLM Reasoning

❖ DeepSeek-R1 Pipeline



LLM Reasoning

❖ DeepSeek Non-Reasoning vs DeepSeek Reasoning

- **Evaluation:** 1000 samples from Vietnamese MetaMathQA
- **Example:** Markus gấp đôi tuổi con trai ông và con trai Markus gấp đôi tuổi cháu trai Markus. Nếu tổng số tuổi của Markus, con trai ông và cháu trai ông là 140 tuổi thì cháu trai của Markus bao nhiêu tuổi?
- **GT Answer:** 20

DeepSeek-V2-Lite (16B)

- Average Time: 10.32s / sample
- Accuracy: 4.8%

DeepSeek-R1-Distill-Qwen (7B)

- Average Time: 11.02s / sample
- Accuracy: 59.5%

LLM Reasoning

❖ DeepSeek Non-Reasoning vs DeepSeek Reasoning

- **Evaluation:** 1000 samples from Vietnamese MetaMathQA
- **Example:** Markus gấp đôi tuổi con trai ông và con trai Markus gấp đôi tuổi cháu trai Markus. Nếu tổng số tuổi của Markus, con trai ông và cháu trai ông là 140 tuổi thì cháu trai của Markus bao nhiêu tuổi?
- **GT Answer:** 20

DeepSeek-R1-Distill-Qwen (7B)

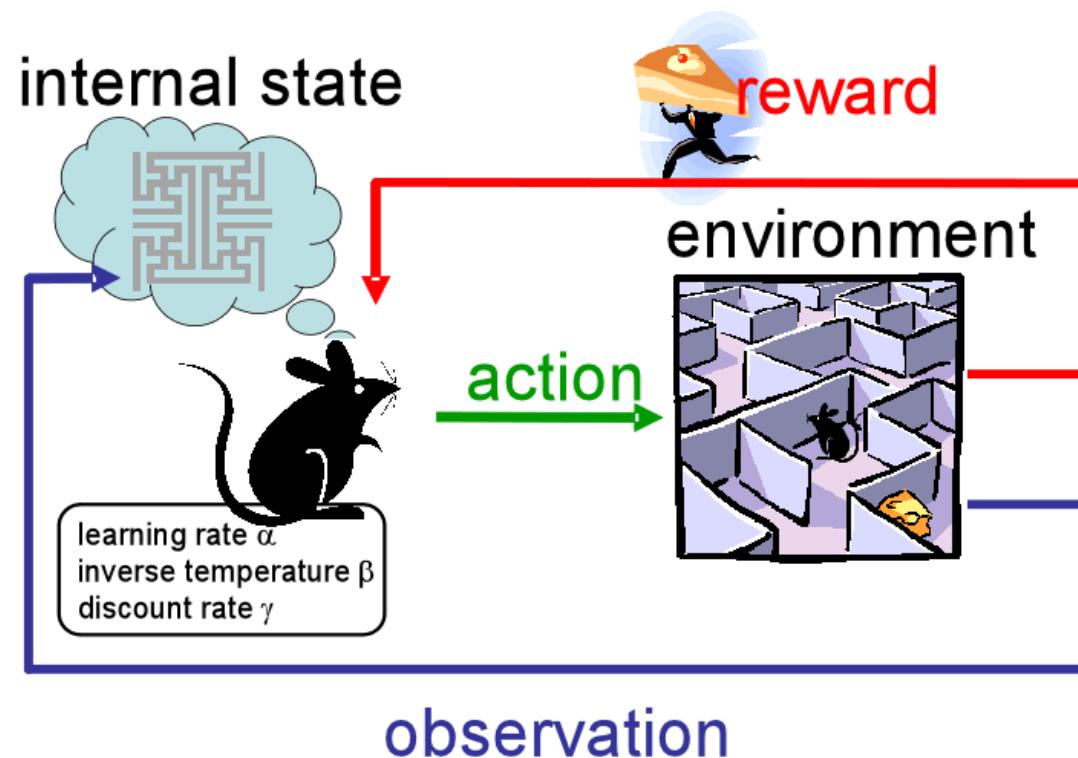
- Thought: Ta gọi tuổi của Markus là M, con trai ông là B và cháu trai ông là C. Theo đề bài, ta có các quan hệ: $M = 2B$, $B = 2C$, và $M + B + C = 140$. Thay thế các biểu thức theo C: $M = 4C$, $B = 2C$, ta được phương trình $4C + 2C + C = 140$, suy ra $7C = 140$ nên $C = 20$. Vậy cháu trai Markus 20 tuổi.
- Answer: 20 ✓

DeepSeek-V2-Lite (16B)

Answer: 30 ✗

LLM Reasoning

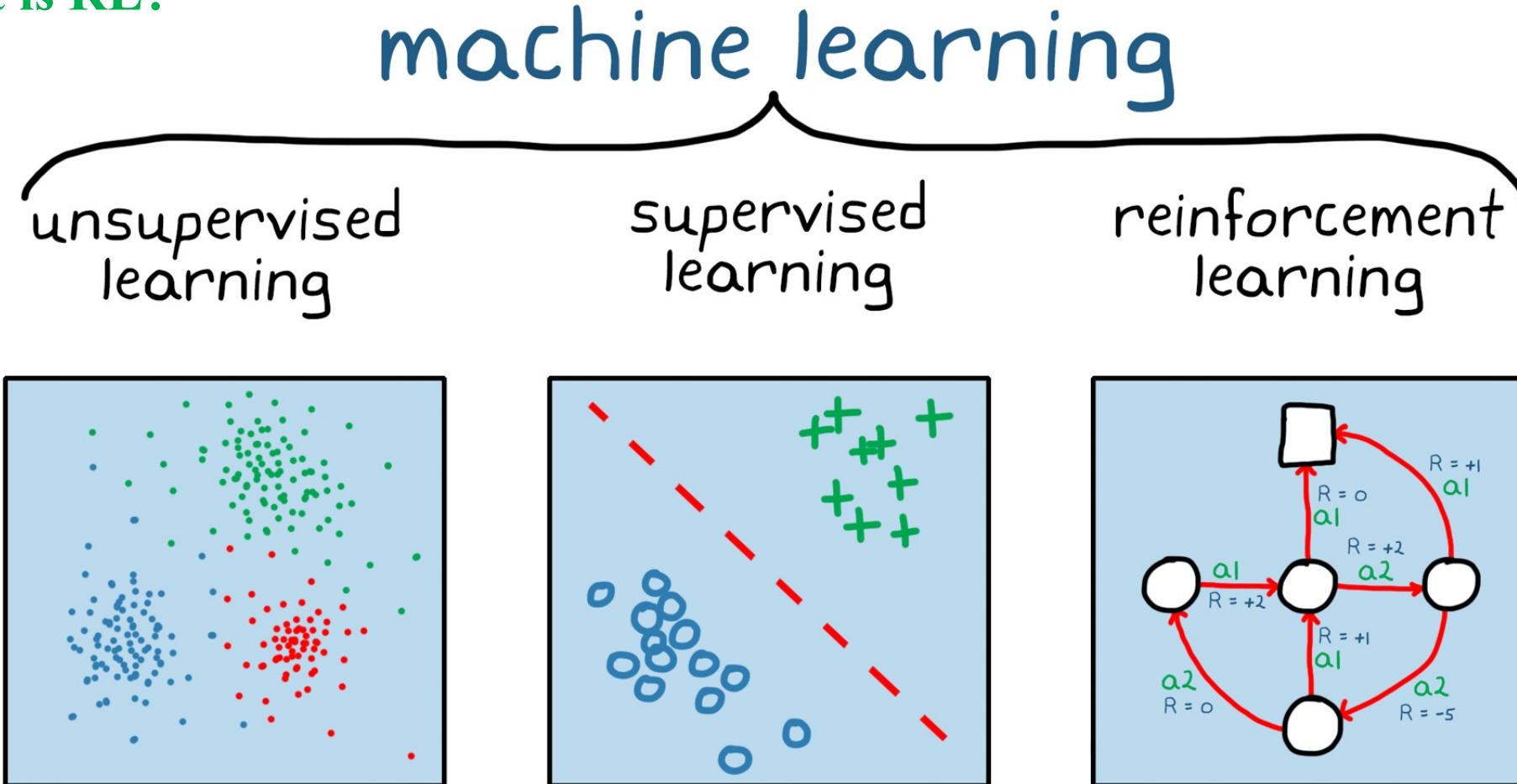
❖ What is RL?



Reinforcement Learning (RL): A branch of machine learning where an agent learns to make decisions by interacting with an environment. Through this interaction, the agent receives rewards or penalties based on its actions, guiding it to learn an optimal policy for maximizing cumulative rewards over time.

LLM Reasoning

❖ What is RL?

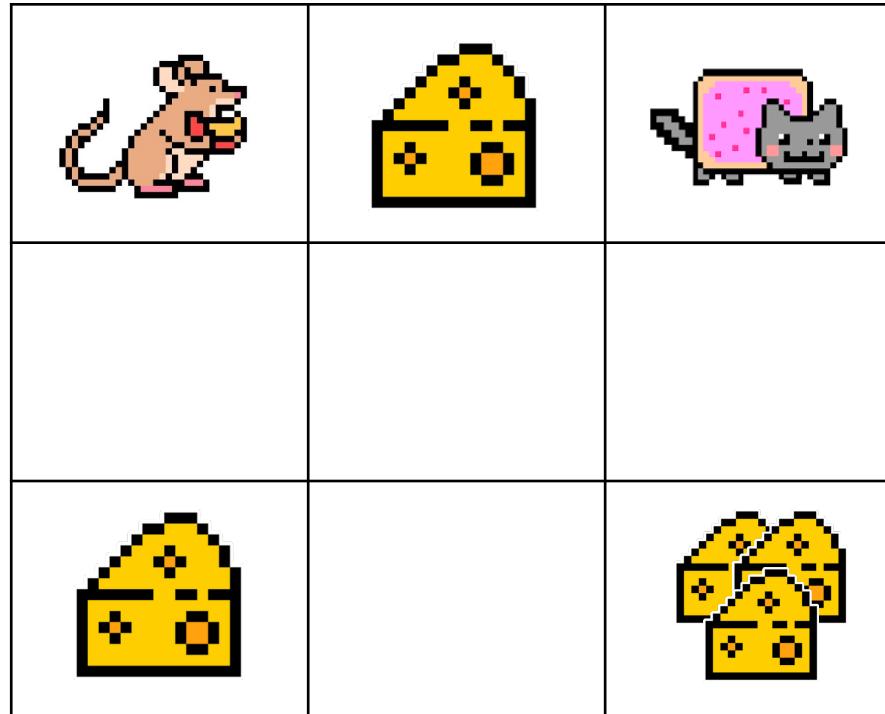


Type of learning in Machine Learning

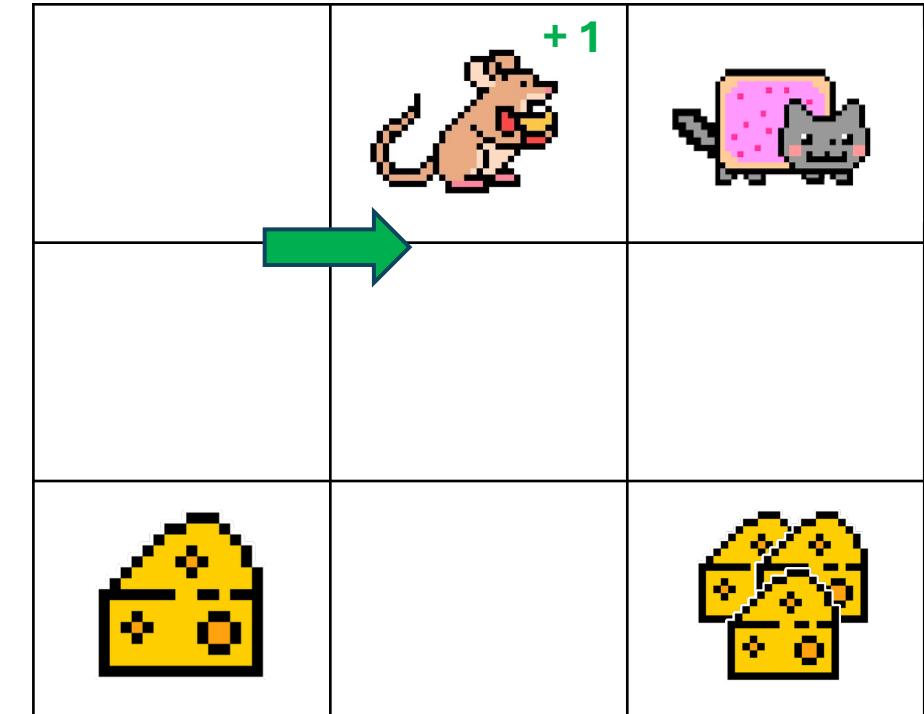
LLM Reasoning

❖ RL idea

Points: 0



Points: 1

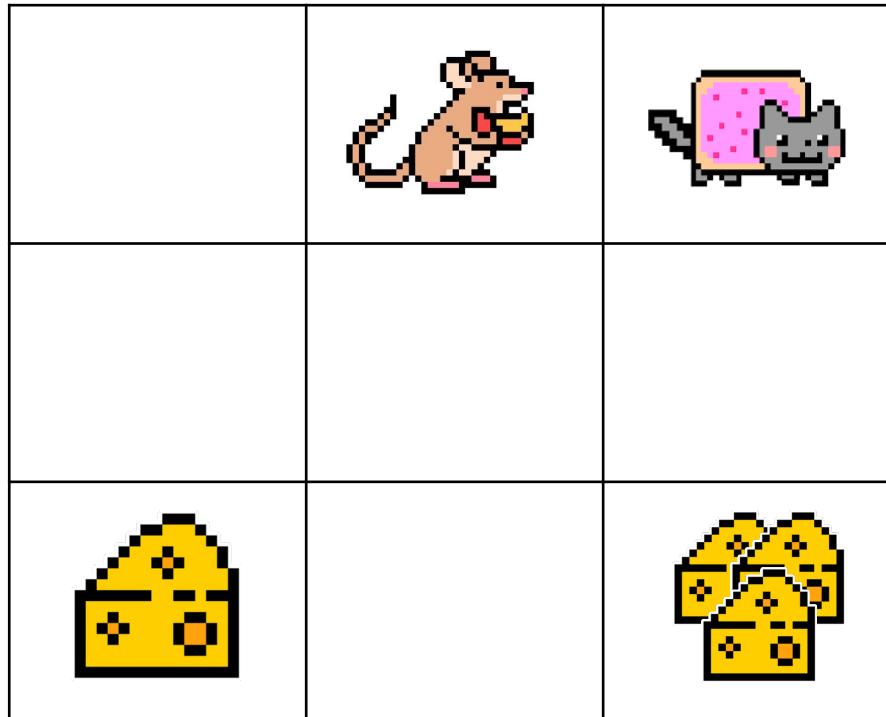


Game objective: Get to the big cheese position with highest points.

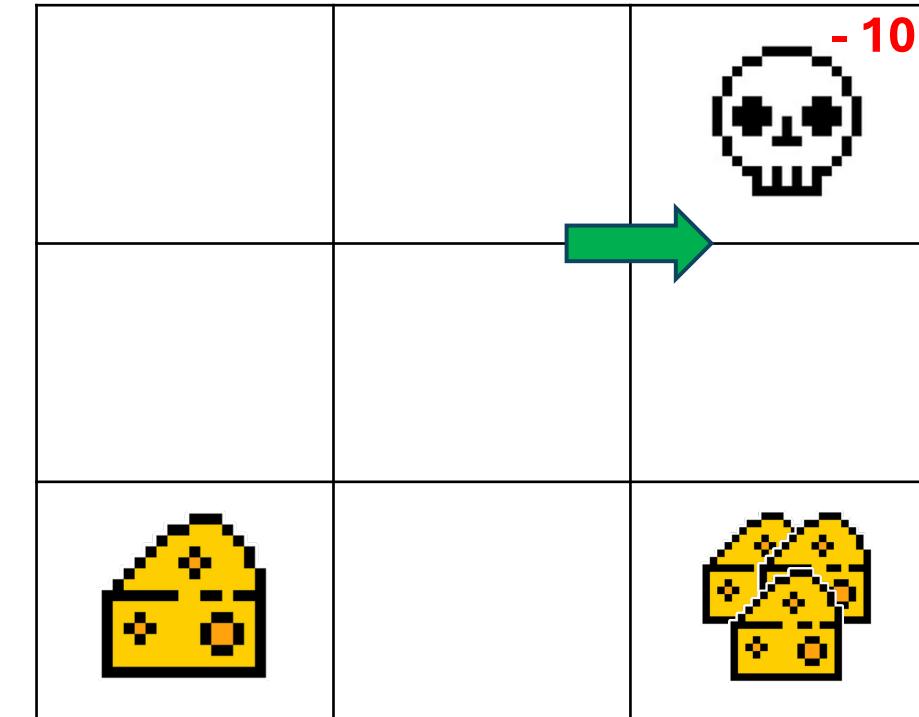
LLM Reasoning

❖ RL idea

Points: 1



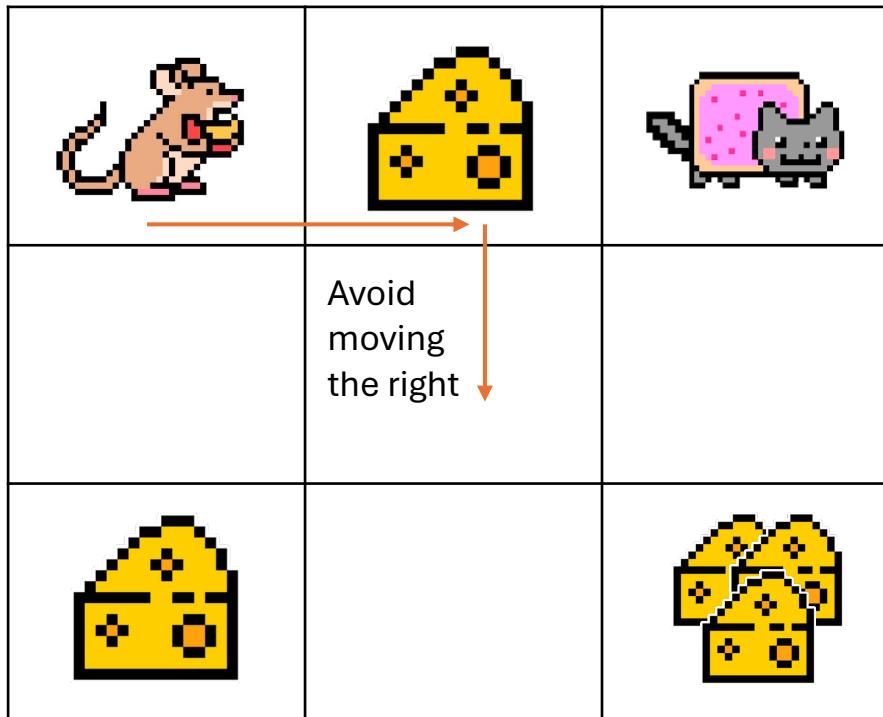
Points: -9



LLM Reasoning

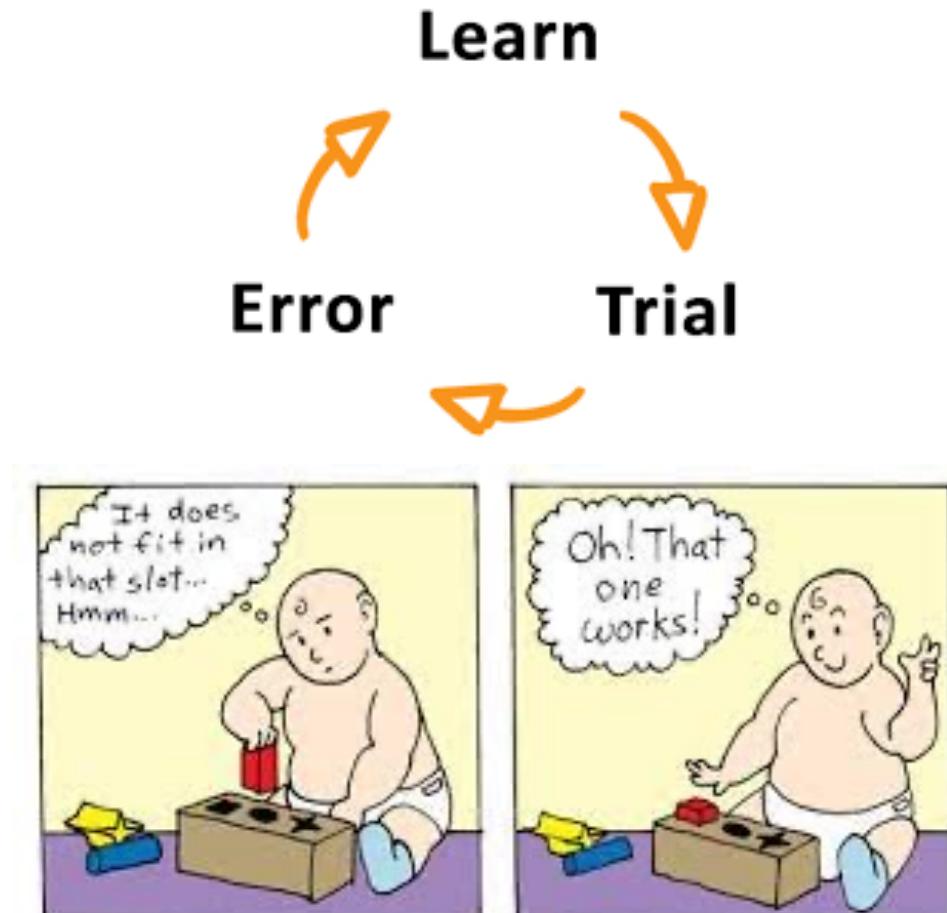
❖ RL idea

Points: 0



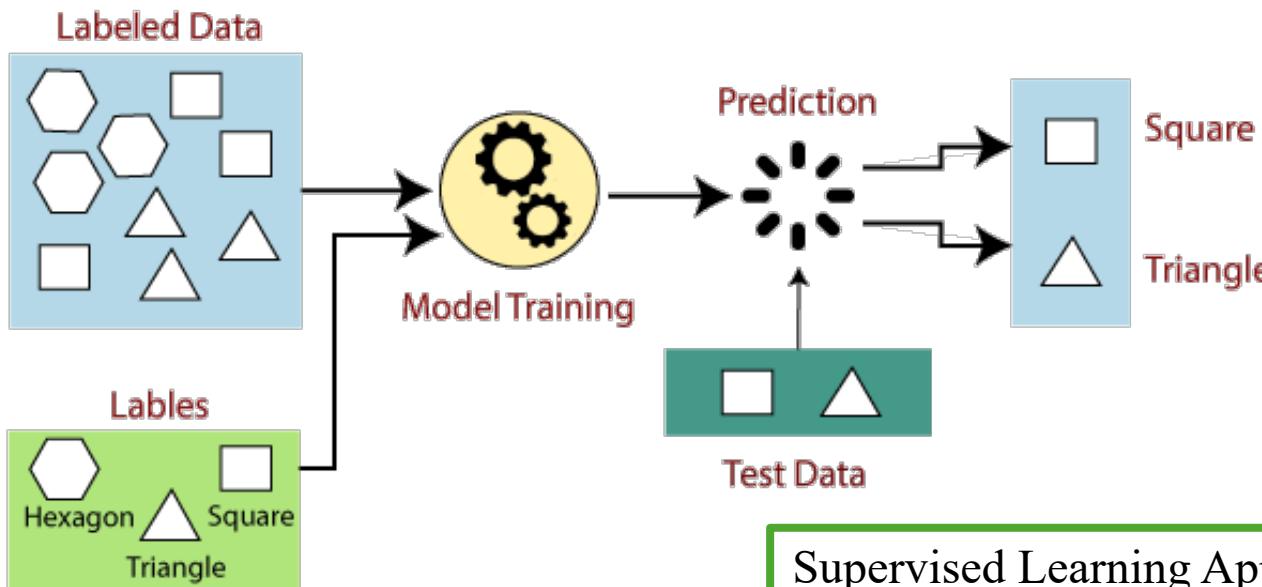
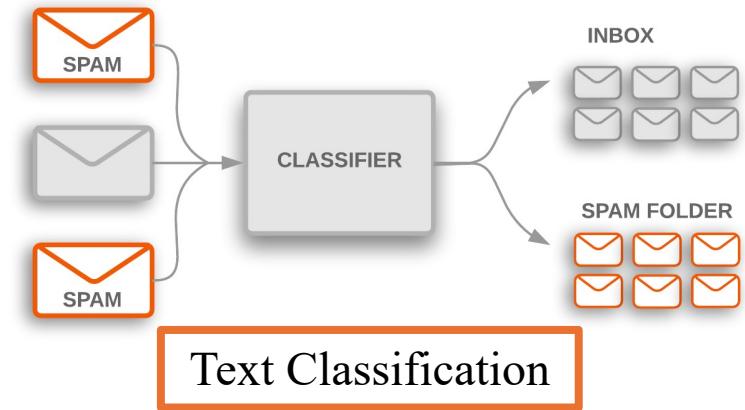
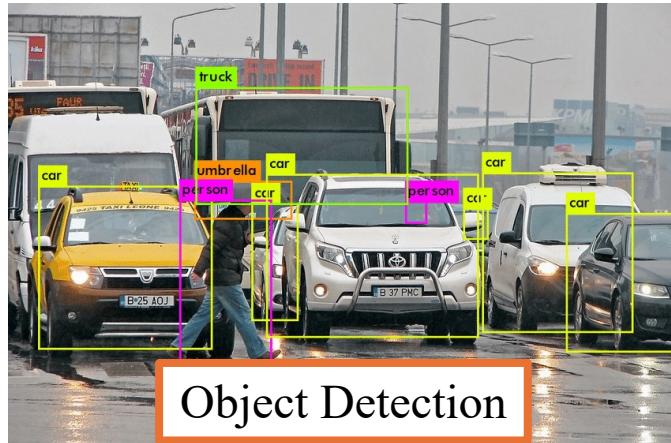
Reset game and play again with new knowledge.

Create an **agent** that could interact with the **environment**, learn to reach the goal and obtain maximum rewards.



LLM Reasoning

❖ Example: Supervised Learning Approach

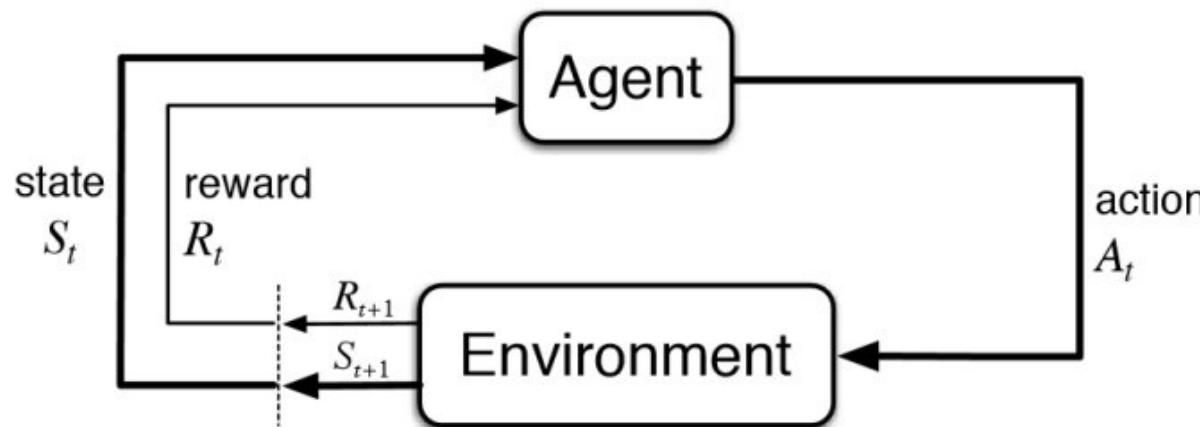
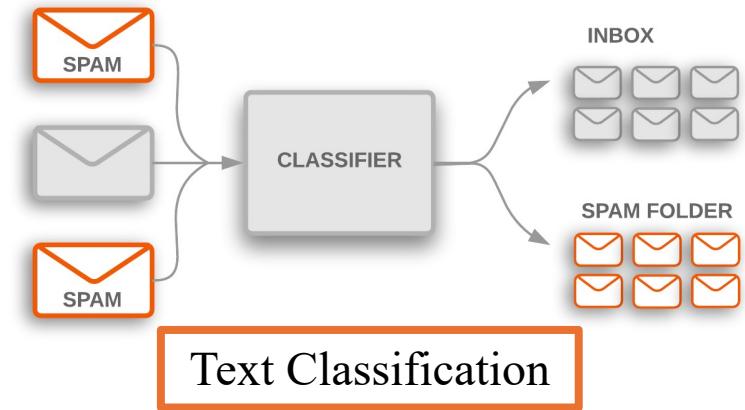
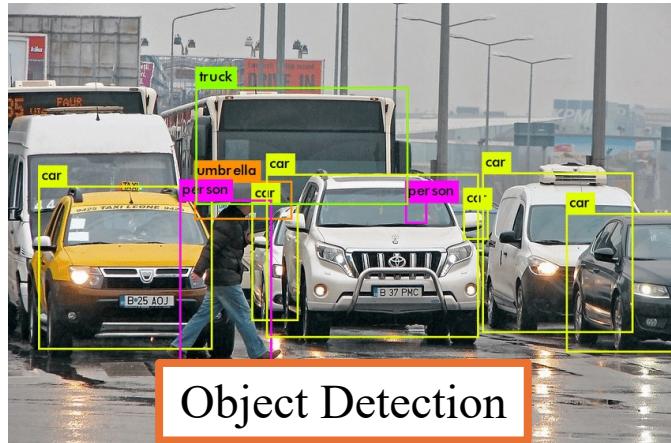


Train a model on **labeled dataset** to be able to map any X to y.

$$f(X) \rightarrow Y$$

LLM Reasoning

❖ Example: RL Approach

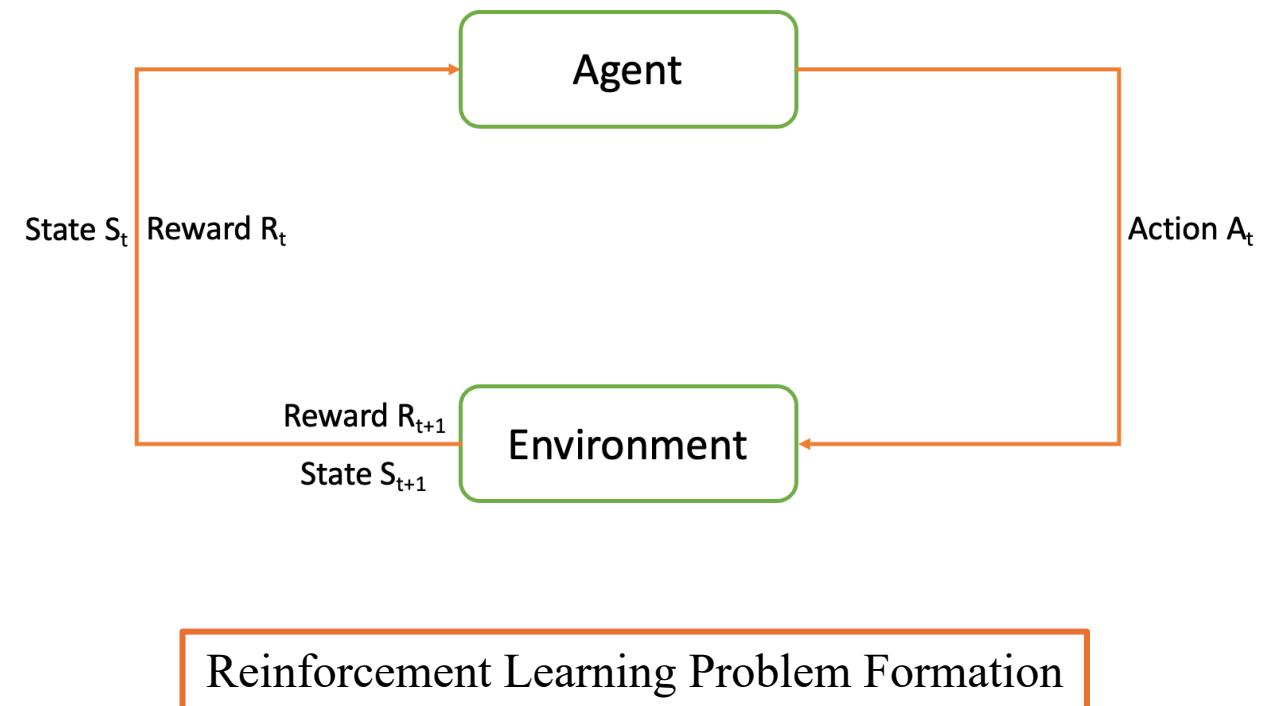
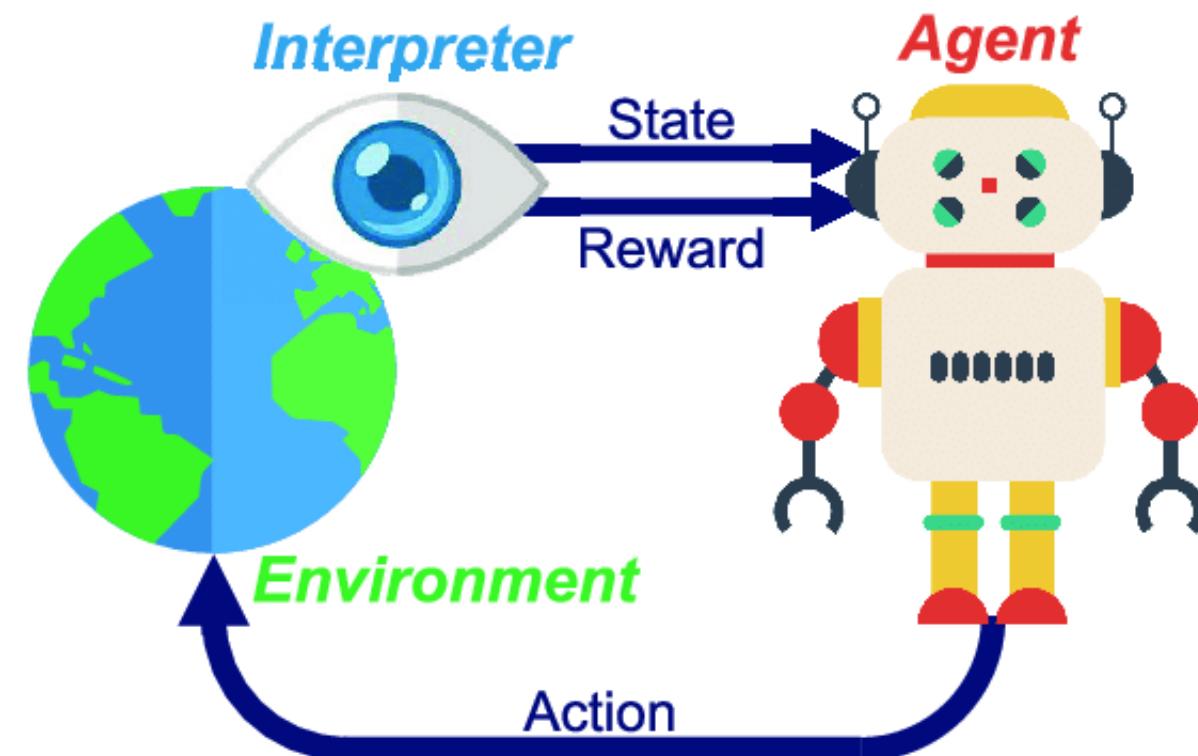


Forming the problem by defining Agent, Environment, State, Action, Reward.

Train the model by let the agent interacts with the environment.

LLM Reasoning

❖ RL idea: RL Framework



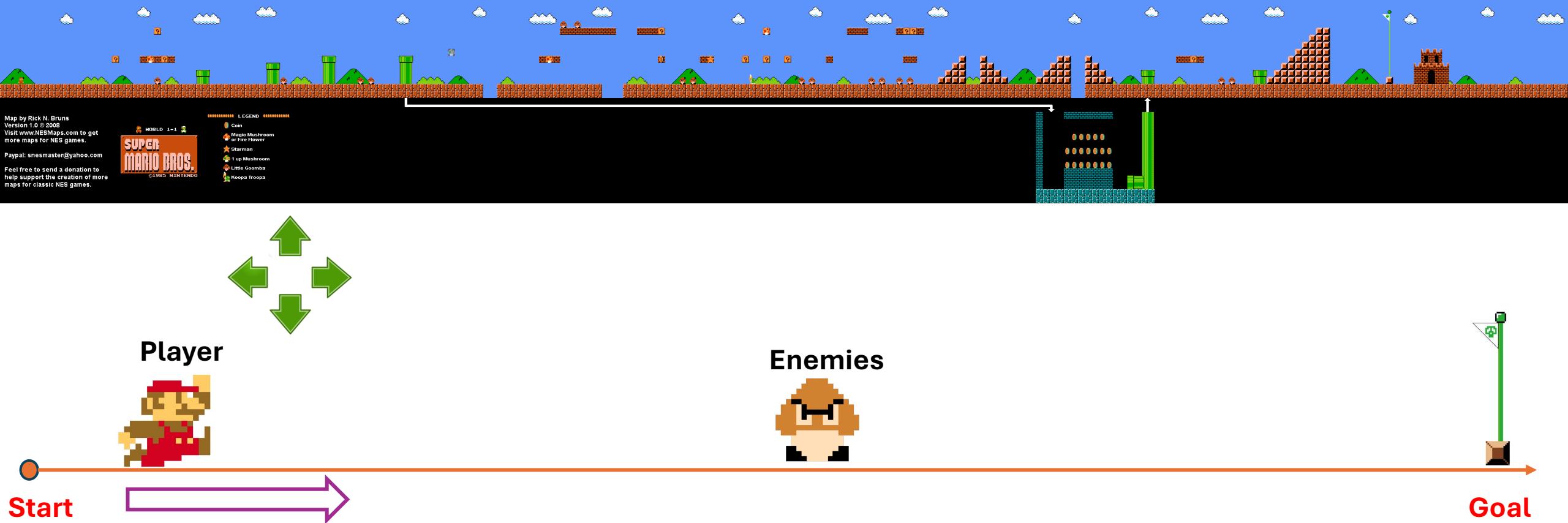
LLM Reasoning

❖ Example: Super Mario Bros



LLM Reasoning

❖ Example: Super Mario Bros



LLM Reasoning

❖ Example: Super Mario Bros



Super Mario Bros Game

Player: Mario



Moves: Up, Down, Left, Right

Game Goal:



1. Avoid enemies



2. Reach the flag

LLM Reasoning

❖ Agent



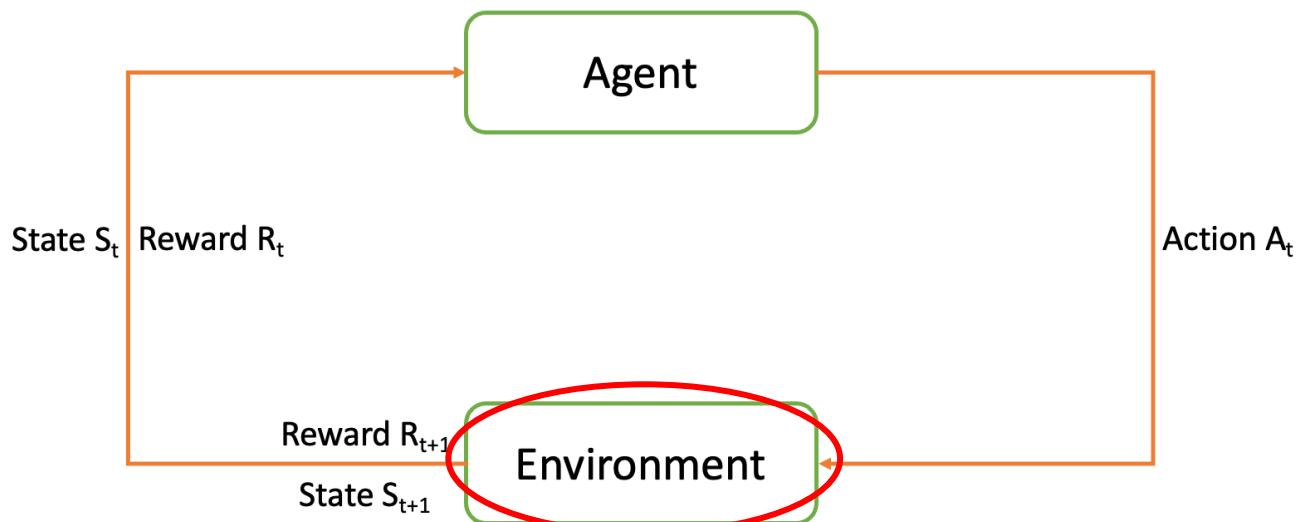
Agent: The decision-making entity that learns to perform actions based on the current state of the environment.

Example: In Super Mario Bros, the agent is Mario.



LLM Reasoning

❖ Environment



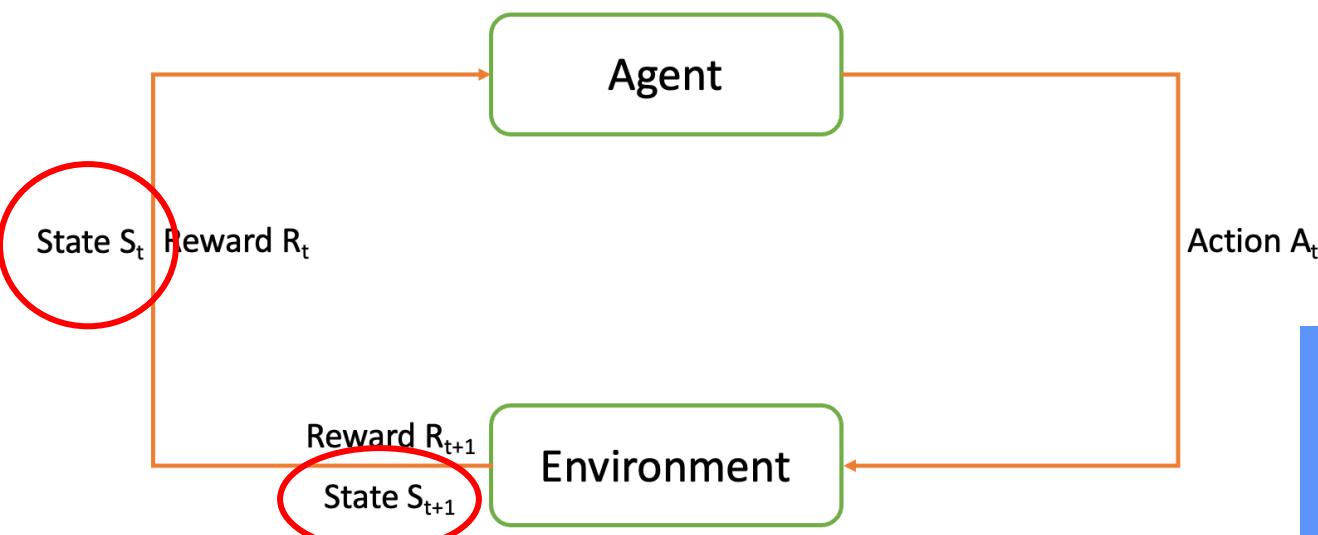
Environment: The context or setting in which the agent operates, providing feedback to the agent in the form of rewards or penalties.

Example: In Super Mario Bros:



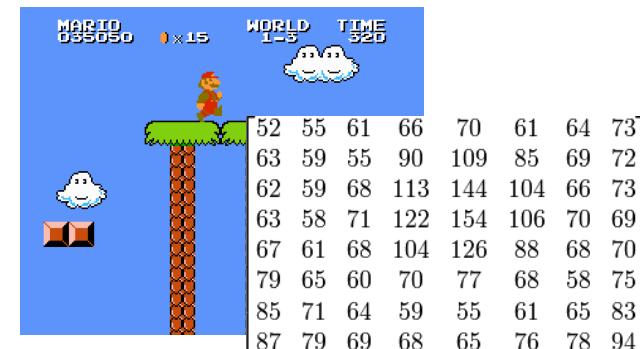
LLM Reasoning

❖ State

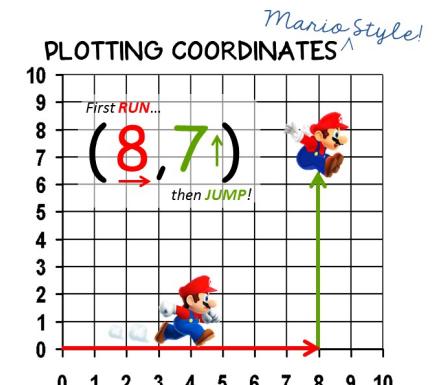


State: Represents the specific situation or configuration the agent encounters in the environment.

Example: In Super Mario Bros, state can be described by key game elements:



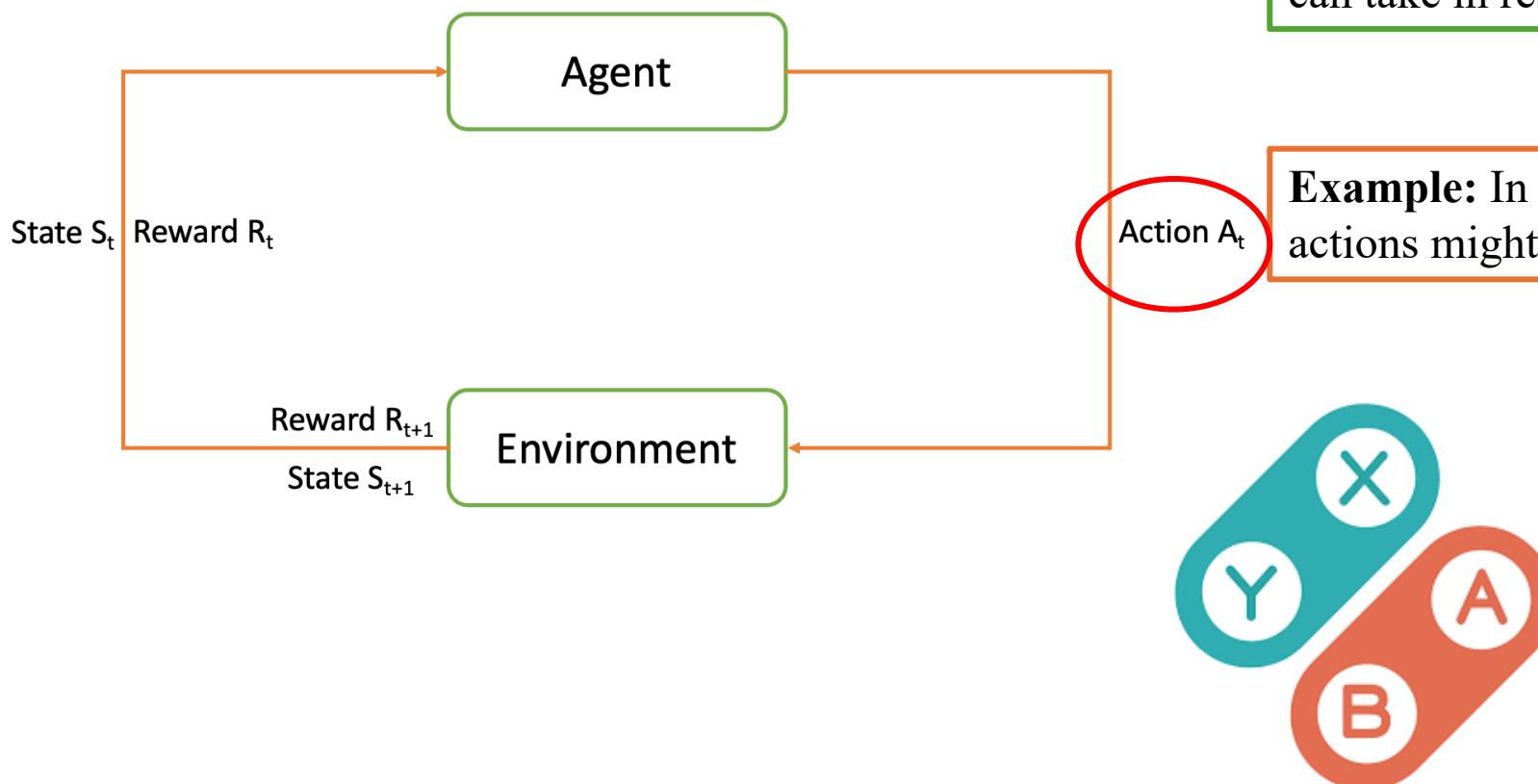
Current frame raw pixels



Objects Coordinate

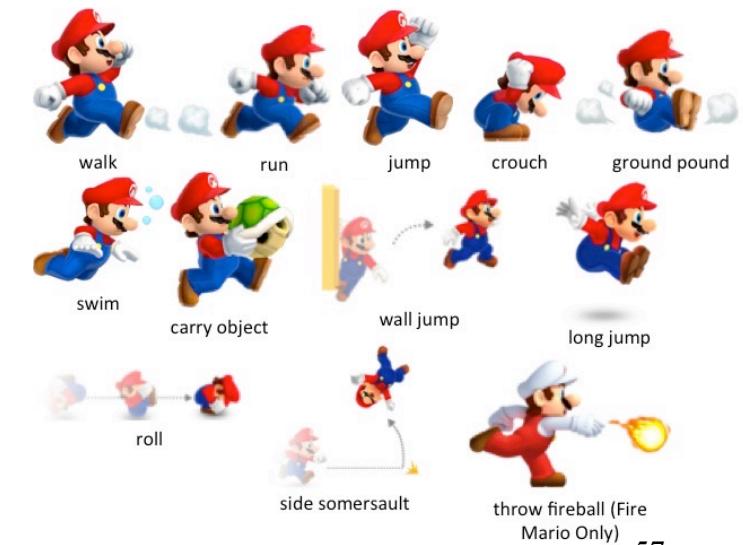
LLM Reasoning

❖ Action



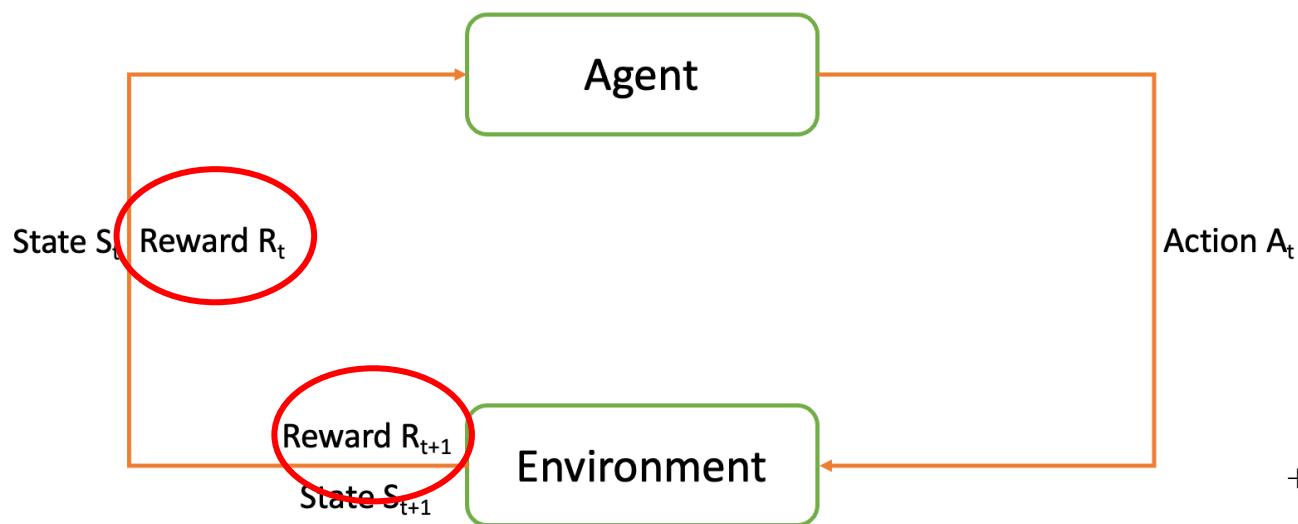
Action: The set of possible moves or decisions the agent can take in response to the current state.

Example: In Super Mario Bros, easy version's legal actions might include: **Up, Down, Left, Right.**



LLM Reasoning

❖ Reward



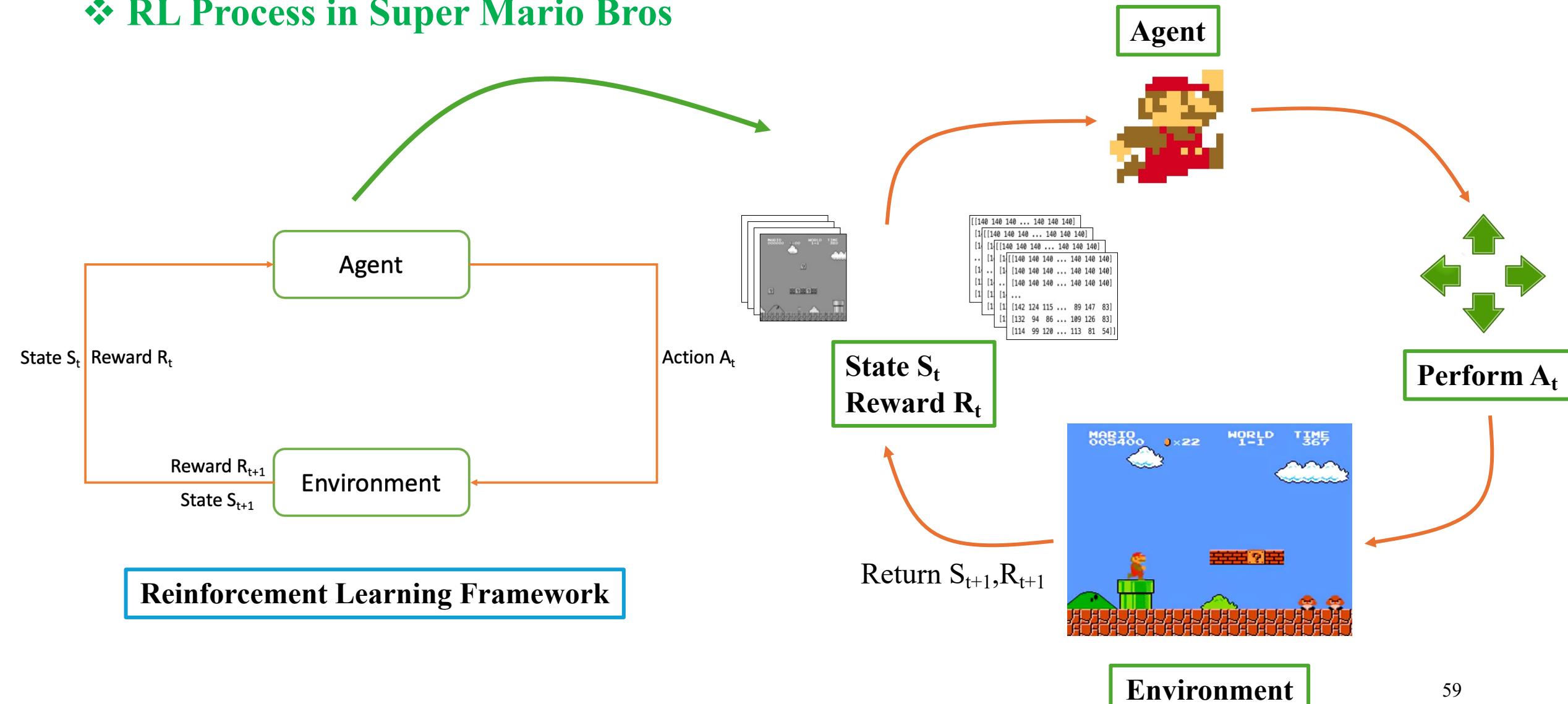
Reward: A scalar value that quantifies the desirability of an action given a particular state, guiding the agent's learning process.

Example: In Super Mario Bros, we can define reward function as following:

Progress	Time	Lives
+1 for moving to right	-1 for each second taken	-100 for losing life
Points or Coins	Level Completion	
+10 for collecting coins or defeating enemies	+500 for reaching the flag	58

LLM Reasoning

❖ RL Process in Super Mario Bros



LLM Reasoning

❖ Example: Stock Price Prediction

Entrade Giao dịch

Tìm mã CK

Danh mục ▾ VN30 HSX UPCOM Phái sinh

Mã PTS TC Trần Sàn ← Tổng KL → Bên mua Bên bán Giá ĐTNN

← +/- → Giá 3 KL 3 Giá 2 KL 2 Giá 1 KL 1 Giá KL ← +/- → Giá 1 KL 1 Giá 2 KL 2 Giá 3 KL 3 Cao TB Thấp ← Mua → Bán

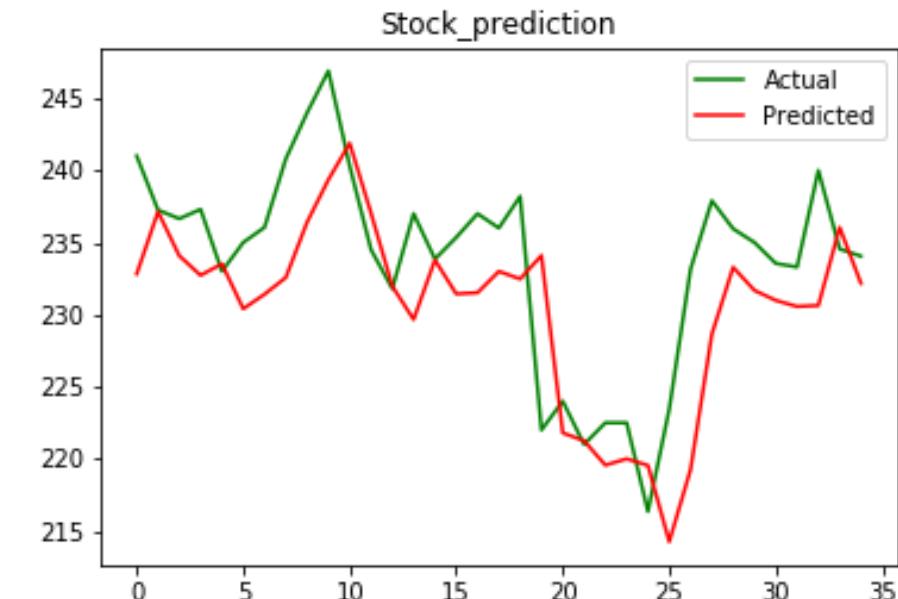
Mã	PTS	TC	Trần	Sàn	Bên mua					Bên bán					Giá			ĐTNN							
					Giá 3	KL 3	Giá 2	KL 2	Giá 1	KL 1	Giá	KL	← +/- →	Giá 1	KL 1	Giá 2	KL 2	Giá 3	KL 3	Cao	TB	Thấp	← Mua →	Bán	
ACB	0.36	31.80	34.00	29.60	2,244,20	31.75	8,80	31.80	6,40	31.85	3,40	31.95	364,50	+0.15	31.95	33,40	32.00	555,80	32.05	2,40	32.00	31.78	31.50		
BID	0.10	35.30	37.75	32.85	1,088,10	35.50	8,50	35.55	5,10	35.60	40	35.75	76,20	+0.45	35.75	3,60	35.80	41,10	35.85	45,90	35.85	35.56	35.20	27,10	13,60
BVH	-0.01	53.60	57.30	49.85	1,193,80	53.30	15,10	53.40	4,40	53.50	15,00	53.50	76,30	-0.10	53.60	4,00	53.70	70	53.80	70	54.50	53.80	53.20	69,60	16,50
CTG	0.23	27.00	28.85	25.15	2,818,40	27.15	124,90	27.20	312,60	27.25	224,20	27.30	244,60	+0.30	27.30	11,70	27.35	51,90	27.40	76,00	27.40	27.24	27.00	672,80	85,10
FPT	-0.41	111.80	119.60	104.00	2,496,90	110.70	3,30	110.80	50	110.90	10	111.30	176,90	-0.50	111.30	194,70	111.40	30,20	111.50	27,90	112.40	111.02	109.20	8,471,70	5,205,00
GAS	0	110.00	117.70	102.30	682,00	109.60	1,00	109.70	80	109.80	5,10	110.00	19,80		110.00	19,20	110.10	1,30	110.50	30	112.30	110.80	109.80	3,20	240,50
GVR	-0.01	25.50	27.25	23.75	1,314,50	25.35	5,30	25.40	46,80	25.45	14,90	25.45	92,20	-0.05	25.50	6,00	25.55	2,50	25.60	5,30	25.80	25.58	25.40	20,40	52,50
HDB	0.07	25.80	27.60	24.00	1,969,70	25.75	12,40	25.80	188,80	25.85	141,30	25.85	170,40	+0.05	25.90	41,90	25.95	5,00	26.00	90	26.10	25.93	25.70	1,162,70	156,40
HPG	0	35.45	37.90	33.00	15,054,50	35.30	213,90	35.35	217,30	35.40	633,10	35.45	883,50		35.45	6,20	35.50	170,20	35.55	204,50	35.85	35.53	35.30	979,90	167,80
KDH	-0.07	42.55	45.50	39.60	971,80	42.15	3,20	42.20	1,90	42.30	50	42.40	99,00	-0.15	42.40	13,50	42.45	47,00	42.50	29,20	42.60	42.14	41.80	155,30	33,70
MBB	0.22	28.00	29.95	26.05	7,937,90	27.95	39,10	28.00	213,50	28.05	3,50	28.10	633,90	+0.10	28.10	44,70	28.15	110,80	28.20	223,90	28.25	28.03	27.90	13,20	17,30
MSN	0	110.00	117.70	102.30	671,10	109.00	14,40	109.20	1,80	109.30	50	110.00	75,80		110.00	103,50	110.10	20	110.30	1,00	110.00	109.51	108,90	342,90	264,90
MWG	-0.81	146.70	156.90	136.50	1,262,20	144,80	1,80	144.90	21,90	145,00	5,60	145,20	106,80	-1.50	145,20	51,20	145,30	13,00	145,40	12,20	147,50	145,43	143,40	351,60	351,60
NVL	0.21	78.40	83.80	73.00	2,673,20	77,80	11,50	77.90	40,00	78,00	10,10	78,70	353,70	+0.30	78,70	87,60	78,80	39,20	78,90	57,90	78,70	77,81	77,10	202,20	24,00
PDR	0.28	54.30	58.10	50.50	3,015,70	55,00	45,30	55,10	20,60	55,20	6,20	55,30	41,80	+1.00	55,30	42,10	55,40	13,30	55,50	47,10	55,50	54,02	52,90	166,40	47,30
PLX	0.11	42.60	45.55	39.65	1,268,60	43,20	5,50	43,25	10,70	43,30	9,50	43,40	39,30	+0.80	43,40	10,10	43,45	11,30	43,50	56,90	43,60	43,31	42,80	256,00	18,00
PNJ	-0.39	122,30	130,80	113,80	1,216,10	119,40	10,60	119,50	3,00	119,60	40	120,50	107,40	-1.80	120,50	14,20	120,60	1,00	120,90	20,00	122,60	120,53	119,30	510,80	1,001,30
POW	0.06	13.35	14.25	12.45	9,576,00	13,30	477,90	13,35	3,082,10	13,40	1,918,00	13,45	90,00	+0.10	13,45	290,20	13,50	318,80	13,55	394,60	13,60	13,46	13,35	129,60	101,50
SAB	-0.04	154,00	164,70	143,30	140,90	153,00	6,40	153,10	1,20	153,20	90	153,40	8,40	-0.60	153,40	50	153,80	1,50	153,90	1,20	154,90	153,50	152,70	32,50	65,80
SSI	0	29.90	31.95	27.85	14,626,40	29,80	216,20	29,85	60,00	29,90	183,30	29,90	831,50		29,95	26,70	30,00	22,20	30,05	114,90	30,70	30,15	29,50	442,80	199,20
STB	1.03	22.25	23.80	20.70	16,390,50	22,70	183,90	22,75	158,10	22,80	30,10	22,80	448,20	+0.55	22,85	92,80	22,90	108,40	22,95	147,60	23,00	22,73	22,25	2,446,00	464,30

LLM Reasoning

❖ Example: Stock Price Prediction

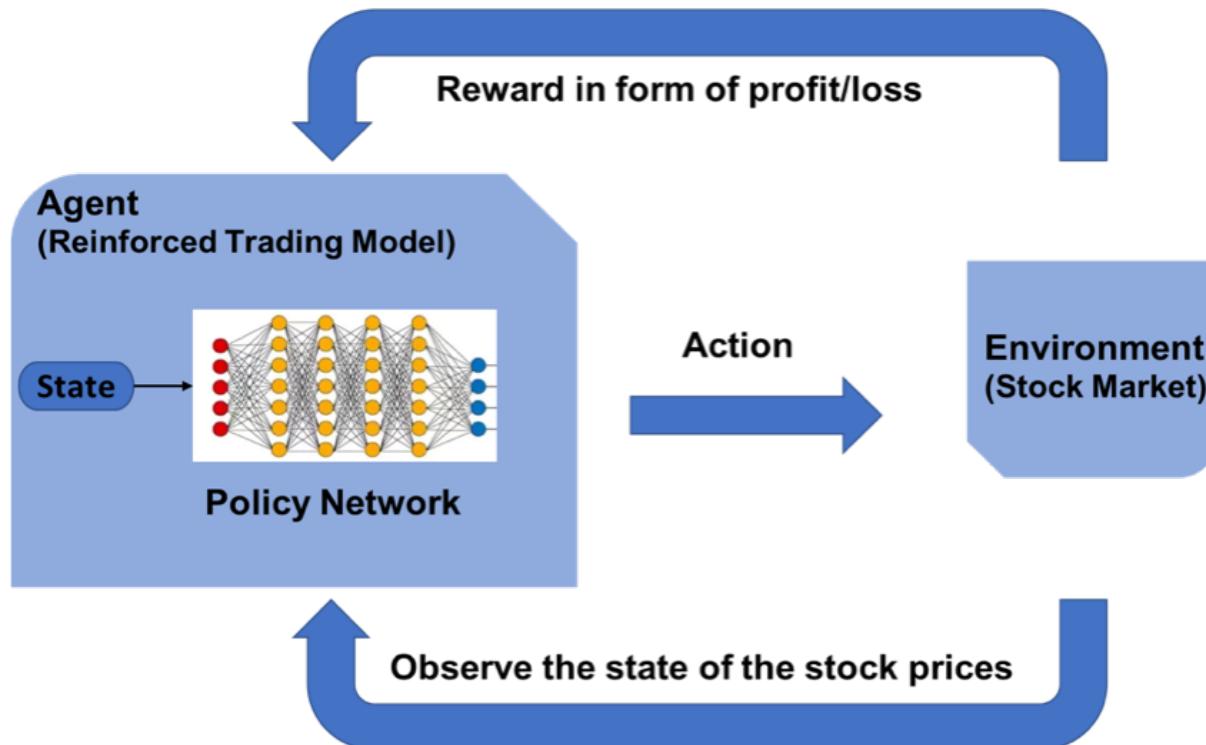
	date	open	high	low	close	volume	Name
449309	2014-10-30	86.16	87.000	85.670	86.940	5560308	PG
382759	2013-07-25	100.39	101.920	100.000	100.710	3285061	MON
29309	2014-07-03	61.95	62.200	61.830	62.010	773696	AKAM
303701	2016-12-09	53.54	54.115	53.390	53.830	2826688	IP
137693	2016-01-04	132.46	132.460	130.935	131.860	411964	COO
53947	2013-02-19	76.84	77.360	76.270	77.300	5854522	APA
349007	2014-07-10	29.33	29.835	29.255	29.775	916948	LNT

In supervised learning, we prepare a training dataset with features and labels, then build a model to learn based on it.



LLM Reasoning

❖ Example: Stock Price Prediction



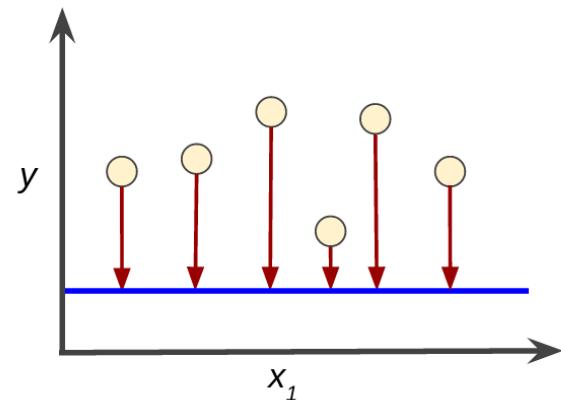
In reinforcement learning, we define the RL framework for stock price prediction problem, then employ a learning algorithm to interact and learn how to achieve best profit.

- Action:** Buy/Sell/Hold.
- States:** All stock prices.
- Rewards:** Profit/Loss.

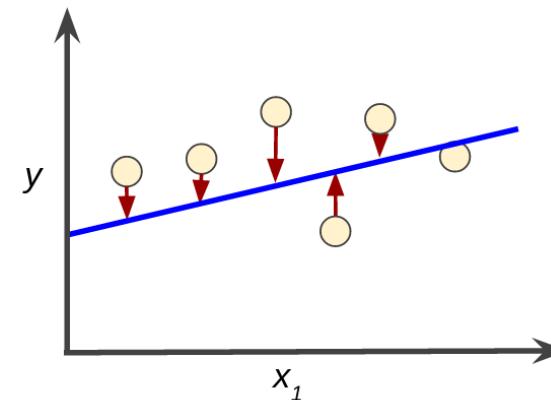
LLM Reasoning

❖ But what is the training objective?

In supervised learning:

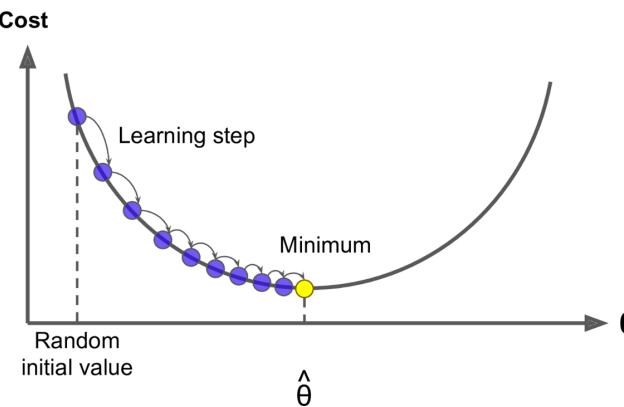


In reinforcement learning:



$$R_T = \sum_{i=0}^T r_{t+i} = r_t + r_{t+1} + \dots + r_T$$

- We attempt to minimize the loss between prediction and label.
- Minimize the loss function.



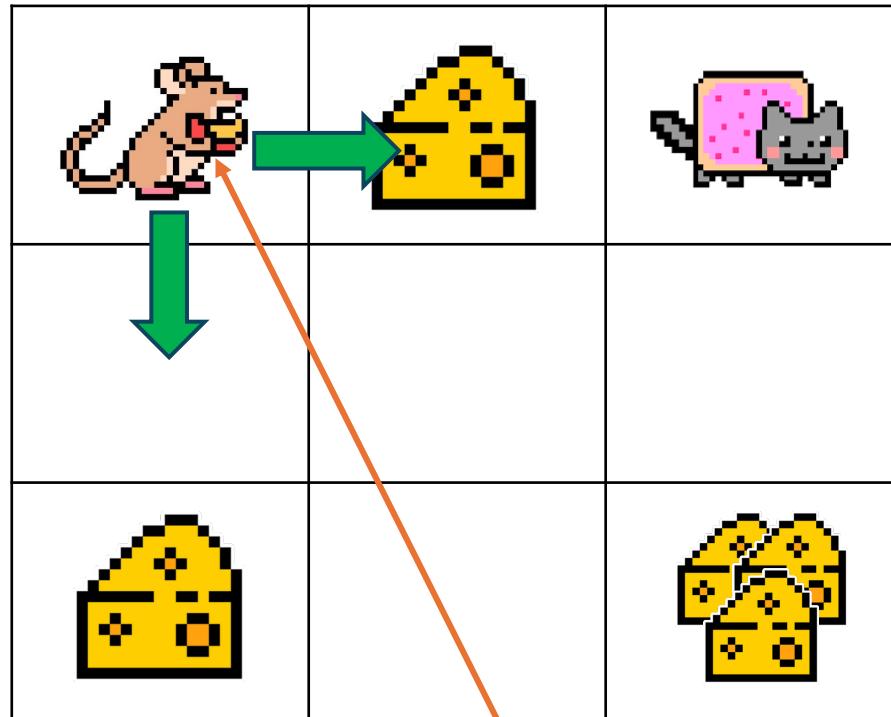
- We attempt to maximize the expected cumulative reward.
- Find optimal policy π .

LLM Reasoning

❖ Policy

Given state S, our agent will have **many possible actions** A.

Points: 0



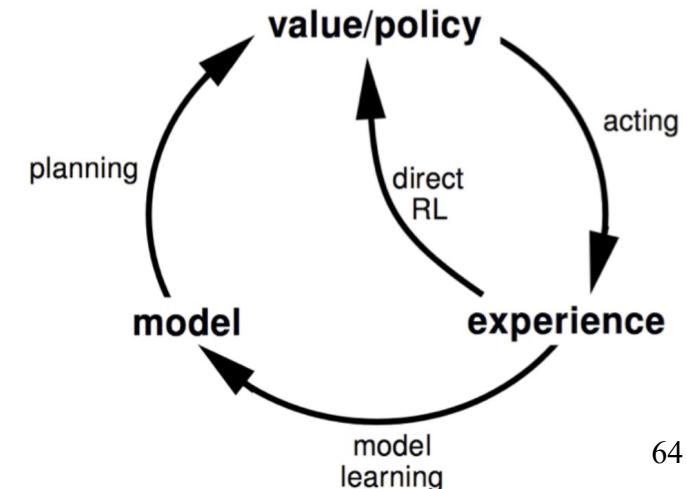
Possible actions at S_0 : Right, Down.

$$R_T = \sum_{i=0}^T r_{t+i} = r_t + r_{t+1} + \dots + r_T$$

- In RL, we attempt to maximize the expected cumulative reward.

Need a way so that at every state, the agent could be able to choose action that leads to the highest expected cumulative reward.

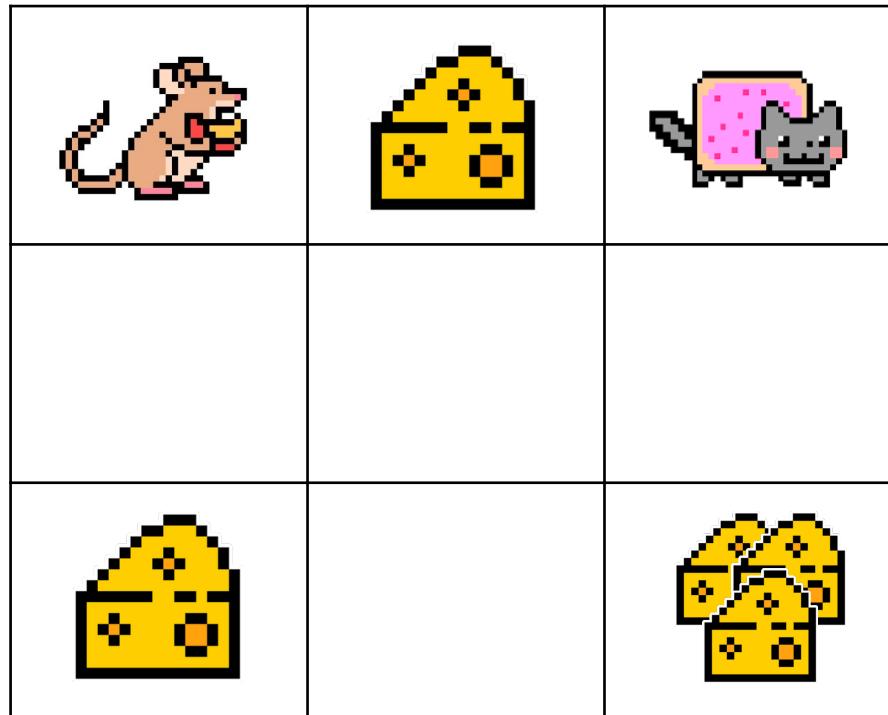
↓
 π
Policy



LLM Reasoning

❖ Policy

Points: 0



Points: 0

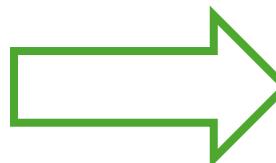
	+1	-10
0	0	0
+1	0	+10

LLM Reasoning

❖ Policy

Points: 0

	+1	-10
0	0	0
+1	0	+10



Points: 0

	+11	-9
+11	+11	+11
+22	+11	+21

State-value function: expected cumulative return the agent can get if it starts at that state, and act according to the policy

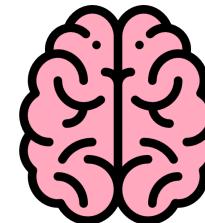
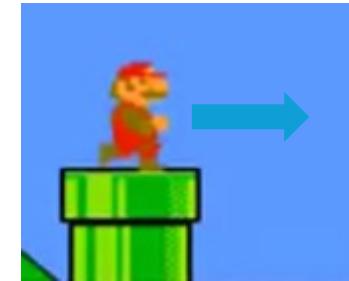
$$V^\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | S_t = s]$$

LLM Reasoning

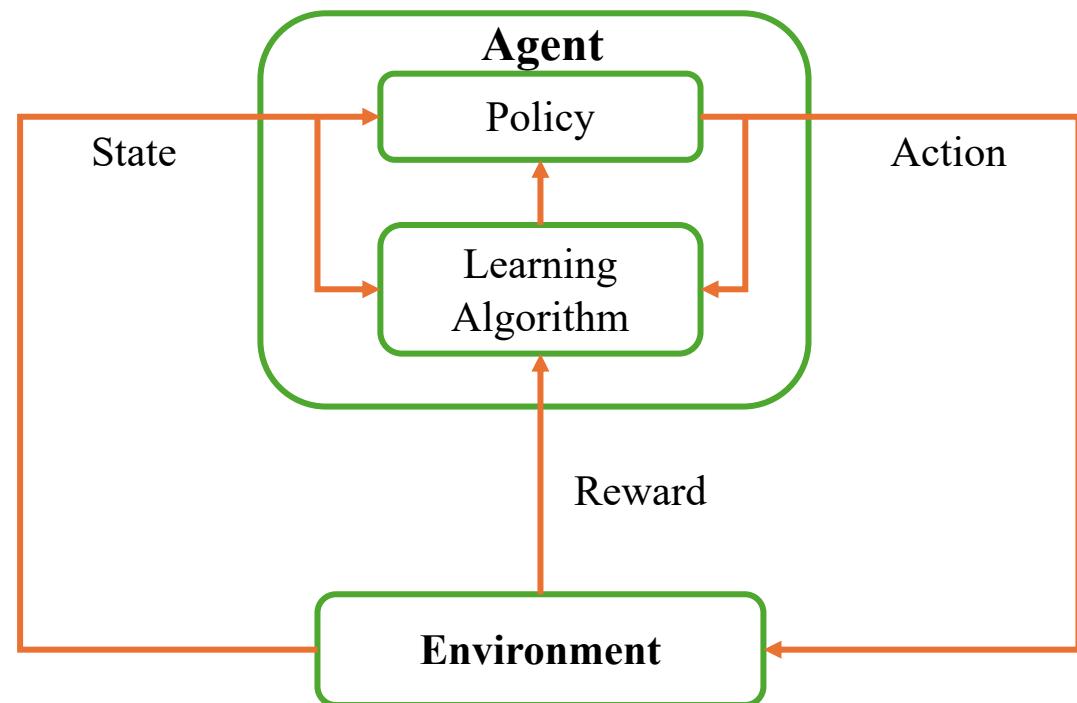
❖ Policy



State S

Policy $\pi(S)$ 

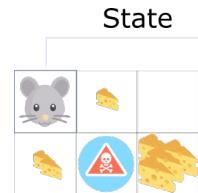
Action A



Policy π : the agent's behavior, define how agent chooses action in response to the current state.

Deterministic Policy

$$a = \pi(s)$$

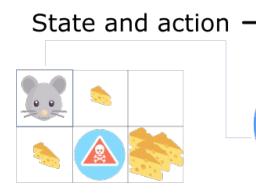


$$\pi(s) = a$$

Deterministic policy

Stochastic Policy

$$\pi(a|s) = P[A|s]$$



$$\pi(a|s) = P(at|st)$$

→ 0.7

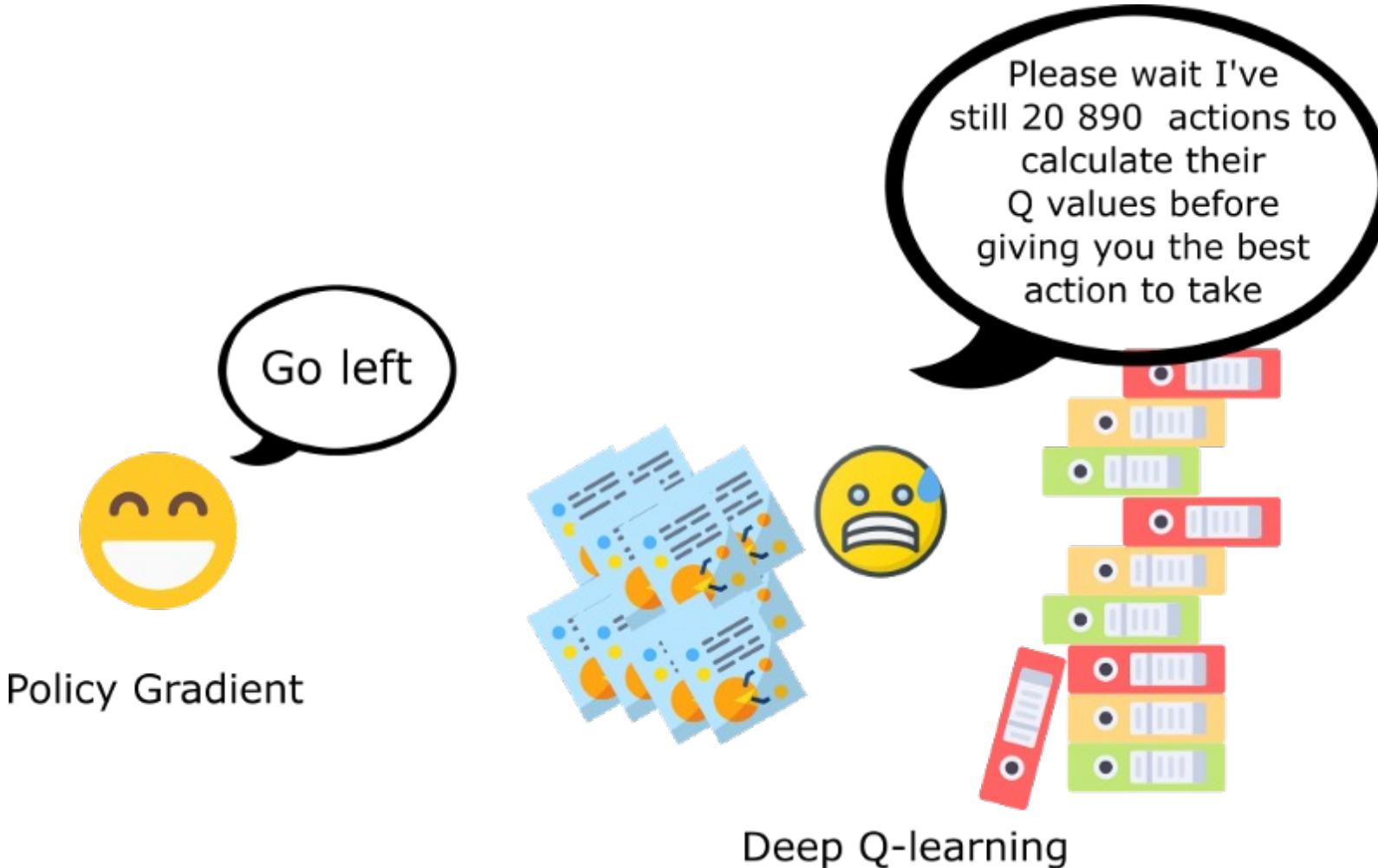
Probability of taking that action at that state

Stochastic policy

Optimal policy π^* leads to maximize the expected cumulative reward.

LLM Reasoning

❖ How to find optimal policy?

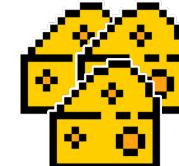


Two main types of RL methods:

1. Value-based.
2. Policy-based.

LLM Reasoning

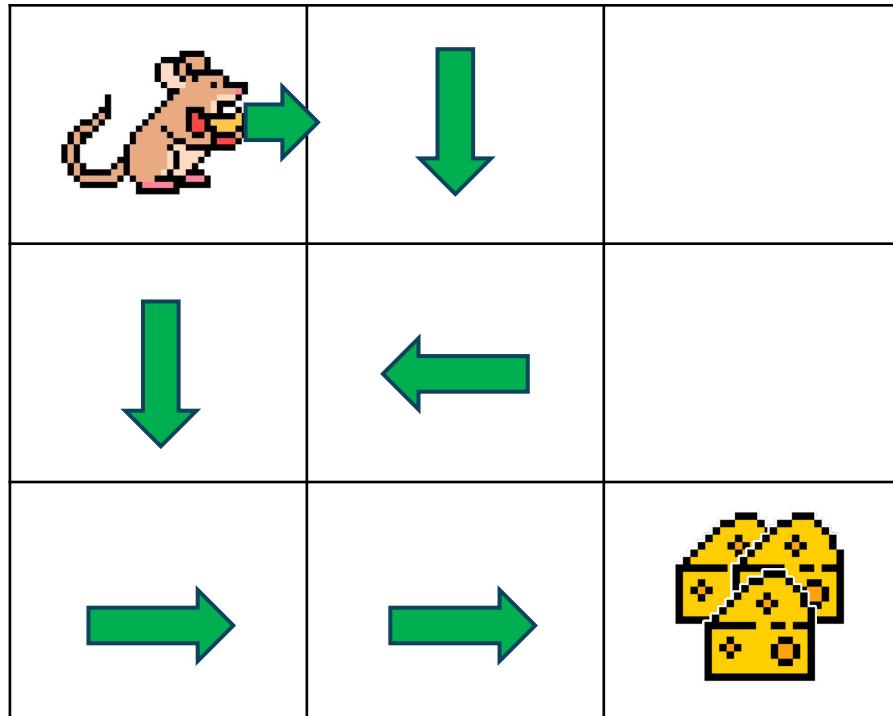
❖ Value-based methods

	-3	-10
-3	-2	-1
-2	-1	

Value-based methods: train the agent to learn which state is more valuable and take the action that leads to it.

LLM Reasoning

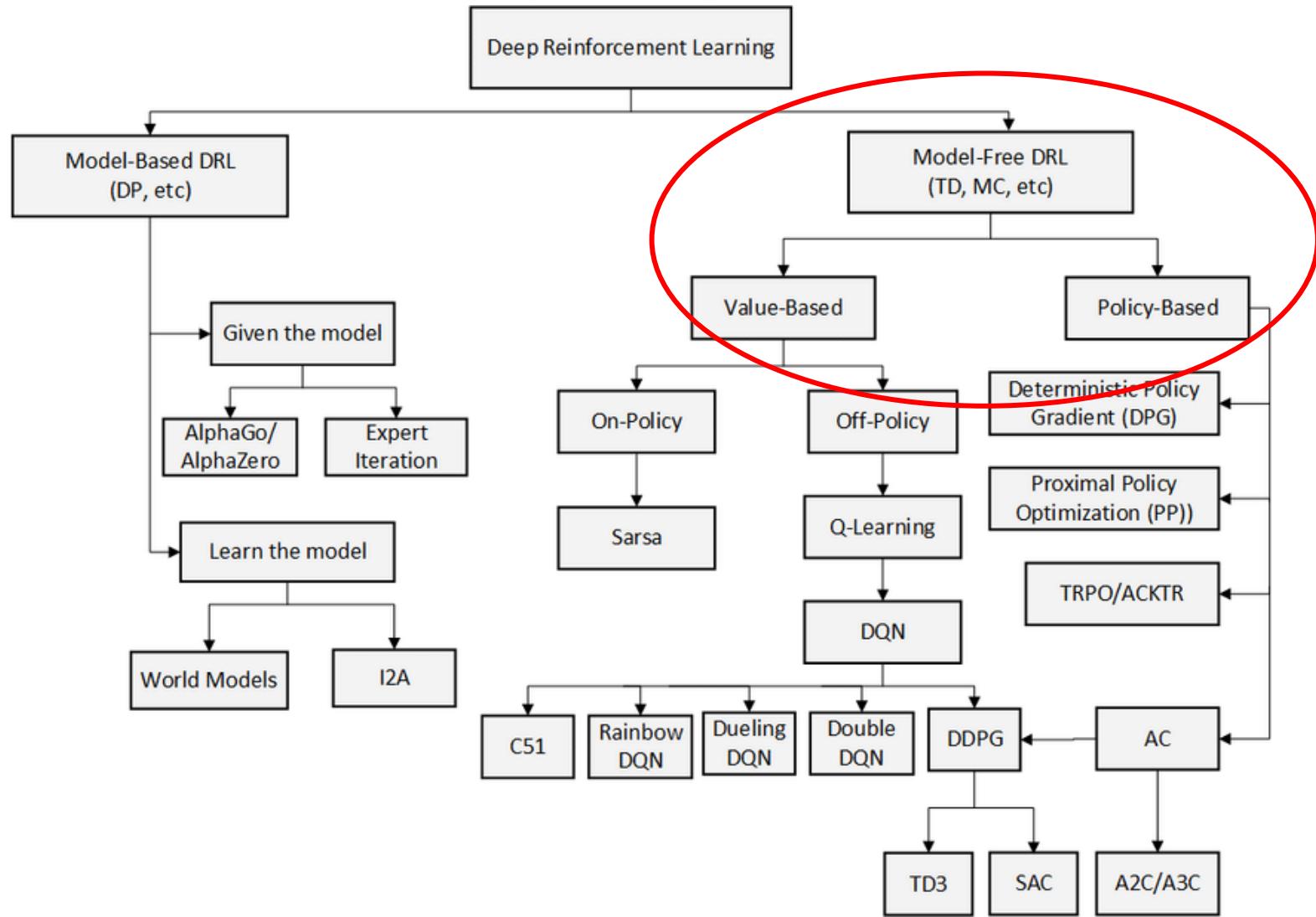
❖ Policy-based methods



Policy-based methods: train the agent to learn which action to take, given a state.

LLM Reasoning

❖ RL Algorithms Taxonomy



LLM Reasoning

❖ Introduction

Cha của Reggie đã cho anh ấy 48 đô la. Reggie đã mua 5 cuốn sách, mỗi cuốn có giá x. Reggie còn lại 38 tiền. Giá trị của biến x chưa biết là bao nhiêu?



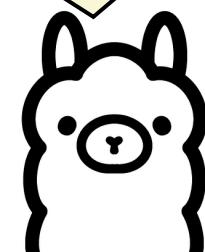
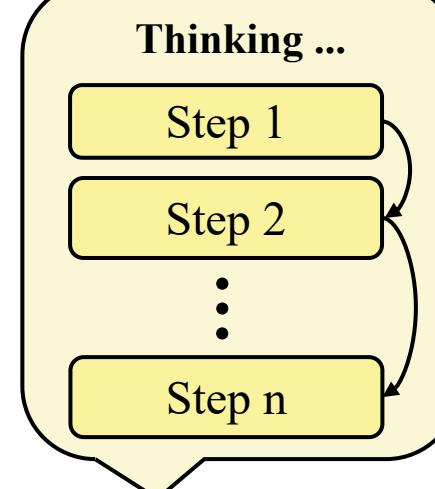
Thought

Reggie đã chi $48 - 38 = 10$ đô la cho 5 cuốn sách, nên lập phương trình $5x = 10$. Giải ra được $x = 2$. Vậy mỗi cuốn sách giá 2 đô la.

Answer

2

Modify Prompt



Training Prompt

System

You are given a problem. Think about the problem and provide your thought process. Place it between <thinking> and </thinking>. Then, provide your final answer between <answer> and </answer>.

Question

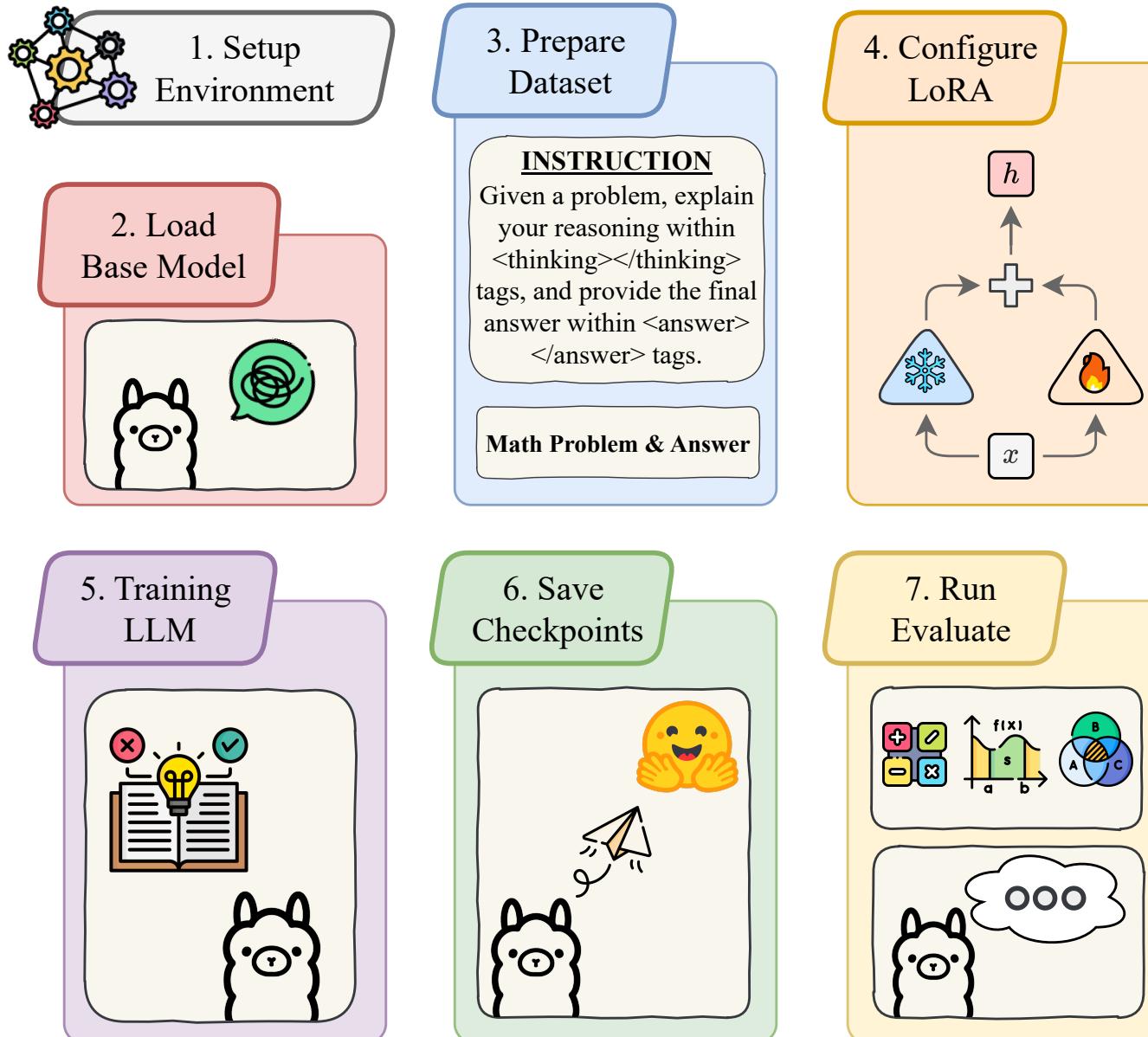
Cha của Reggie đã cho anh ấy 48 đô la. Reggie đã mua 5 cuốn sách, mỗi cuốn có giá x. Reggie còn lại 38 tiền. Giá trị của biến x chưa biết là bao nhiêu?

Answer

2

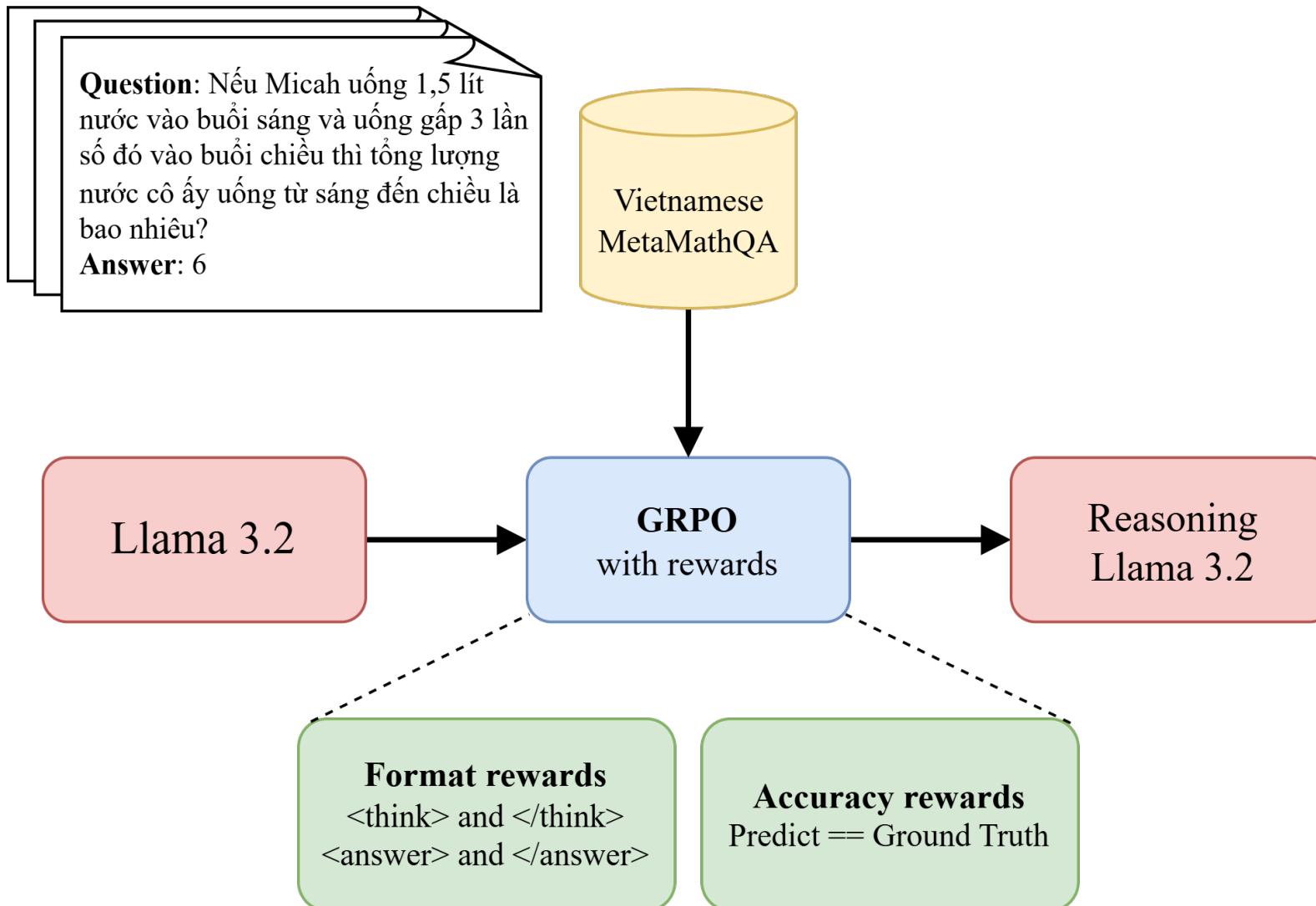
LLM Reasoning

❖ Pipeline



LLM Reasoning

❖ Training Math Reasoning



LLM Reasoning

❖ Step 1: Install and import necessary libraries



unsloth

Unsloth is an open-source Python library that hand-writes GPU kernels and patches core ML frameworks to fine-tune large language models up to 2× faster while cutting GPU memory use by 70–80%.



vLLM is a high-throughput, memory-efficient LLM inference and serving engine from UC Berkeley, leveraging PagedAttention, continuous batching, speculative decoding, and multi-precision quantization support.

LLM Reasoning

❖ Step 1: Install and import necessary libraries

```
%pip install unsloth vllm==0.7.3
```



unsloth

```
1 import re
2 from vllm import SamplingParams
3 from unsloth import FastLanguageModel
4 from datasets import load_dataset, Dataset
5 from trl import GPTQConfig, GPTQTrainer
```

vLLM

LLM Reasoning

❖ Step 2: Load base model

```
1 max_seq_length = 2048
2 lora_rank = 64
3
4 model, tokenizer = FastLanguageModel.from_pretrained(
5     model_name="meta-llama/Llama-3.2-3B-Instruct",
6     max_seq_length=max_seq_length,
7     load_in_4bit=False,
8     fast_inference=True,
9     max_lora_rank=lora_rank,
10    gpu_memory_utilization=0.8,
11 )
```

```
13 model = FastLanguageModel.get_peft_model(
14     model,
15     r=lora_rank,
16     target_modules=[
17         "q_proj", "v_proj"
18     ],
19     lora_alpha=lora_rank,
20     use_gradient_checkpointing="unsloth",
21     random_state=3407,
22 )
```



LLM Reasoning

❖ Step 3: Load & Preprocess Dataset

```
1 dataset = load_dataset("5CD-AI/Vietnamese-meta-math-MetaMathQA-40K-gg-translated", split="train")
```

README.md: 100%  118/118 [00:00<00:00, 5.88kB/s]

MetaMathQA-40K_vi.json: 100%  69.2M/69.2M [00:00<00:00, 91.2MB/s]

Generating train split: 100%  40000/40000 [00:01<00:00, 31852.24 examples/s]

```
1 print("Dataset structure:", dataset)
```

```
Dataset structure: Dataset({  
    features: ['response_vi', 'query_vi', 'response_en', 'type', 'query_en'],  
    num_rows: 40000  
})
```

LLM Reasoning

❖ Step 3: Load & Preprocess Dataset

Answers	Questions			
response_vi string · lengths	query_vi string · lengths	response_en string · lengths	type string · classes	query_en string · lengths
				
Để giải quyết vấn đề này, chúng ta cần xác định giá...	Cha của Reggie đã cho anh ấy 48 đô la. Reggie đã mu...	To solve this problem, we need to determine the...	GSM_SV	Reggie's father gave him \$48. Reggie bought 5...
Có 28 quân domino trong bộ này. Jean và ba người bạn...	Jean và ba người bạn của cô ấy đang chơi trò chơi...	There are 28 dominoes in the set. Jean and her...	GSM_AnsAug	Jean and her three friends are playing a game of...
Cally có tổng cộng 10 áo trắng + 5 áo màu + 7 quần...	Cally và Danny giặt quần áo. Cally có 10 áo sơ mi...	Cally has a total of 10 white shirts + 5 colored...	GSM_AnsAug	Cally and Danny washed their clothes. Cally has...
Ban đầu Karen gọi một chiếc burger giá 5 đô la....	Tổng chi phí cho đơn hàng đồ ăn nhanh của Karen là...	Karen initially ordered a 5-dollar burger. She then...	GSM_Rephrased	What is the total cost of Karen's fast-food order i...
Để xác định giá trị của số thứ 40 trong cách sắp xếp...	Các số \$20\$ đầu tiên của sự sắp xếp được hiển thị...	To determine the value of the 40th number in the...	MATH_SV	The first \$20\$ numbers of an arrangement are shown...
Simon hái tổng cộng $100 + 200 = 300$ quả việt quất....	Nếu Simon hái 100 quả việt quất từ bụi cây của mình...	Simon picks a total of $100 + 200 = 300$ blueberries...	GSM_Rephrased	If Simon picks 100 blueberries from his own...

< Previous

1

2

3

...

400

Next >

Vietnamese-metamath-MetaMathQA-40K-gg-translated Dataset

LLM Reasoning

❖ Chat-style model: Conversation

```
[  
  {  
    "role": "system",  
    "content": "You are a helpful assistant that summarizes content clearly."  
  },  
  {  
    "role": "user",  
    "content": "Please summarize the following:\n\nMachine learning is a field of AI that allows computers to learn from data without being explicitly programmed."  
  },  
  {  
    "role": "assistant",  
    "content": "Machine learning helps computers learn from data automatically, without needing explicit instructions."  
  }]
```

Single-turn

```
[  
  {  
    "role": "system",  
    "content": "You are a helpful assistant that summarizes content clearly."  
  },  
  {  
    "role": "user",  
    "content": "Please summarize the following:\n\nMachine learning is a field of AI that allows computers to learn from data without being explicitly programmed."  
  },  
  {  
    "role": "assistant",  
    "content": "Machine learning helps computers learn from data automatically, without needing explicit instructions."  
  },  
  {  
    "role": "user",  
    "content": "Can you also summarize this?\n\nDeep learning is a subset of machine learning that uses neural networks with many layers."  
  },  
  {  
    "role": "assistant",  
    "content": "Deep learning is a type of machine learning that uses multi-layered neural networks to learn complex patterns from data."  
  }]
```

Multi-turn

LLM Reasoning

❖ Llama 3.2 Prompt Template

Supported Roles: There are 4 different roles that are supported by Llama text models: system, assistant, user, ipython.

[system, assistant, user, ipython]

Supported Role	Description
system	Sets the context in which to interact with the AI model. It typically includes rules, guidelines, or necessary information that help the model respond effectively.
user	Represents the human interacting with the model. It includes the inputs, commands, and questions to the model.
ipython	A new role introduced in Llama 3.1. Semantically, this role means "tool". This role is used to mark messages with the output of a tool call when sent back to the model from the executor.
assistant	Represents the response generated by the AI model based on the context provided in the system , ipython and user prompts.

LLM Reasoning

❖ Llama 3.2 Prompt Template

Special Tokens	Description
< begin_of_text >	Specifies the start of the prompt.
< end_of_text >	Model will cease to generate more tokens. This token is generated only by the base models.
< finetune_right_pad_id >	This token is used for padding text sequences to the same length in a batch.
< start_header_id >	These tokens enclose the role for a particular message. The possible roles are: [system, user, assistant, and ipython]
< eom_id >	End of message. A message represents a possible stopping point for execution where the model can inform the executor that a tool call needs to be made. This is used for multi-step interactions between the model and any available tools. This token is emitted by the model when the Environment: ipython instruction is used in the system prompt, or if the model calls for a built-in tool.
< eot_id >	End of turn. Represents when the model has determined that it has finished interacting with the user message that initiated its response. This is used in two scenarios: <ul style="list-style-type: none">• at the end of a direct interaction between the model and the user• at the end of multiple interactions between the model and any available tools This token signals to the executor that the model has finished generating a response.
< python_tag >	Special tag used in the model's response to signify a tool call.

LLM Reasoning

❖ Llama 3.2 Prompt Template

Instruct Model Prompt: The format for a regular multi-turn conversation between a user and the model of Llama 3.2.

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
```

Cutting Knowledge Date: December 2023

Today Date: 23 July 2024

You are a helpful assistant.<|eot_id|>

```
<|start_header_id|>user<|end_header_id|>
```

What is the capital of France?<|eot_id|>

```
<|start_header_id|>assistant<|end_header_id|>
```

The capital of France is Paris.

- Each message role clearly marked with header tokens.
- <|eot_id|> separates each interaction turn.
- System content should define environment, cut-off date, tone, and rules.

LLM Reasoning

❖ Step 3: Load & Preprocess Dataset

```
1 answer_pattern = re.compile(  
2     r"(đáp án là:|đáp án là :|câu trả lời là:|câu trả lời là :)\s*(.*")",  
3     re.IGNORECASE  
4 )  
5  
6 formatted_dataset = []  
7 for item in dataset:  
8     response = item["response_vi"].strip().lower()  
9     match = answer_pattern.search(response)  
10    if match:  
11        answer = match.group(2).strip()  
12        formatted_dataset.append({  
13            "question": item["query_vi"],  
14            "answer": answer  
15        })
```

Prompt

```
1 reasoning_start = "<thinking>"  
2 reasoning_end = "</thinking>"  
3 solution_start = "<answer>"  
4 solution_end = "</answer>"  
5  
6 system_prompt = \  
7     f"""You are given a problem.  
8 Think about the problem and provide your thought process.  
9 Place it between {reasoning_start} and {reasoning_end}.  
10 Then, provide your final answer between {solution_start}{solution_end}"""  
11  
12 train_dataset = Dataset.from_list(formatted_dataset[:8000])  
13 train_dataset = train_dataset.map(lambda x: {  
14     "prompt": [  
15         {"role": "system", "content": system_prompt},  
16         {"role": "user", "content": x["question"]},  
17     ],  
18     "answer": x["answer"],  
19 })
```

LLM Reasoning

❖ Step 3: Load & Preprocess Dataset

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
```

Cutting Knowledge Date: December 2023
Today Date: 28 Apr 2025

You are given a problem.

Think about the problem and provide your thought process.

Place it between **<thinking>** and **</thinking>**.

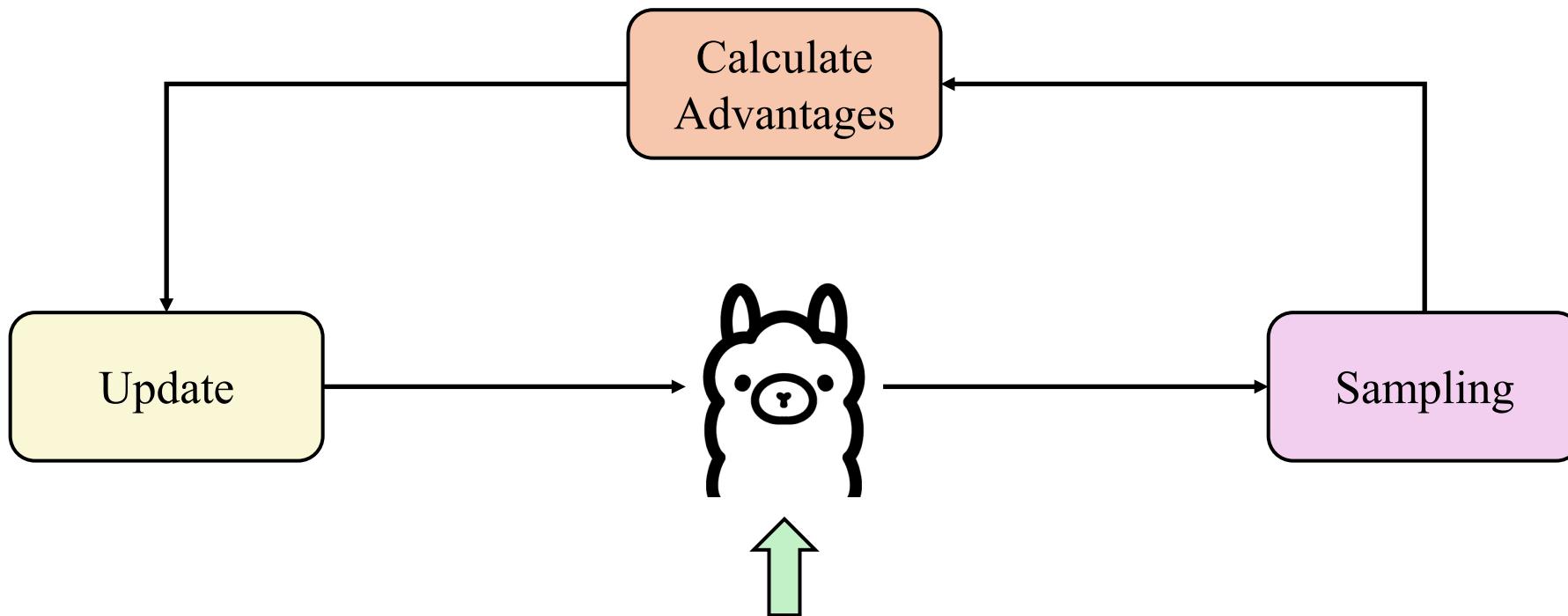
Then, provide your final answer between **<answer></answer><|eot_id|>**
<|start_header_id|>user<|end_header_id|>

Cha của Reggie đã cho anh ấy 48 đô la. Reggie đã mua 5 cuốn sách, mỗi cuốn có giá x.
Reggie còn lại 38 tiền. Giá trị của biến x chưa biết là bao nhiêu?<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>

A sample input formatted for the LLaMA chat template.

LLM Reasoning

❖ Group Relative Policy Optimization (GRPO)

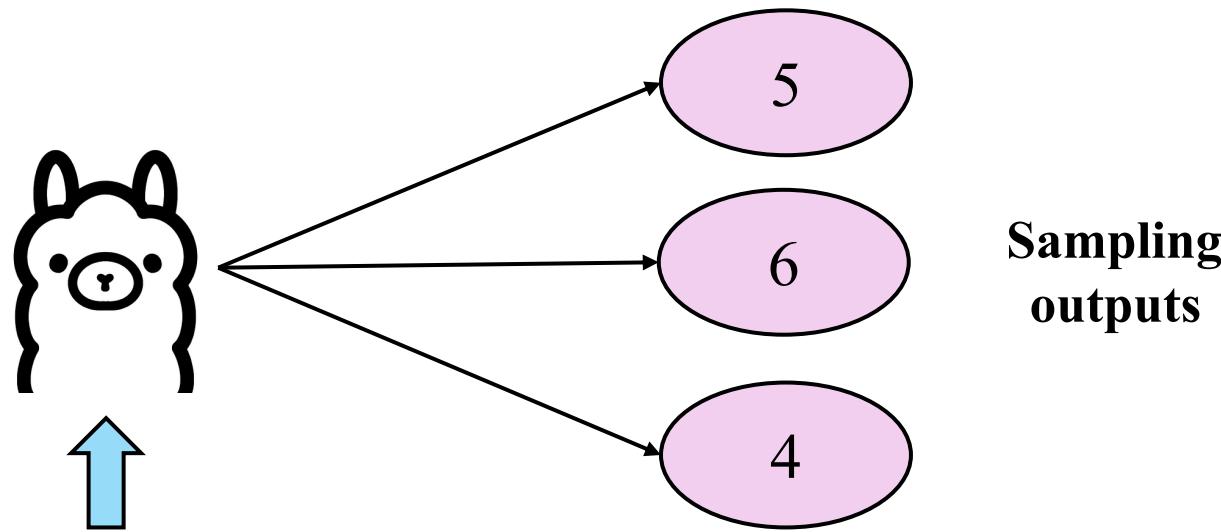


Question: Nếu Micah uống 1,5 lít nước vào buổi sáng và uống gấp 3 lần số đó vào buổi chiều thì tổng lượng nước cô ấy uống từ sáng đến chiều là bao nhiêu?

Answer: 6

LLM Reasoning

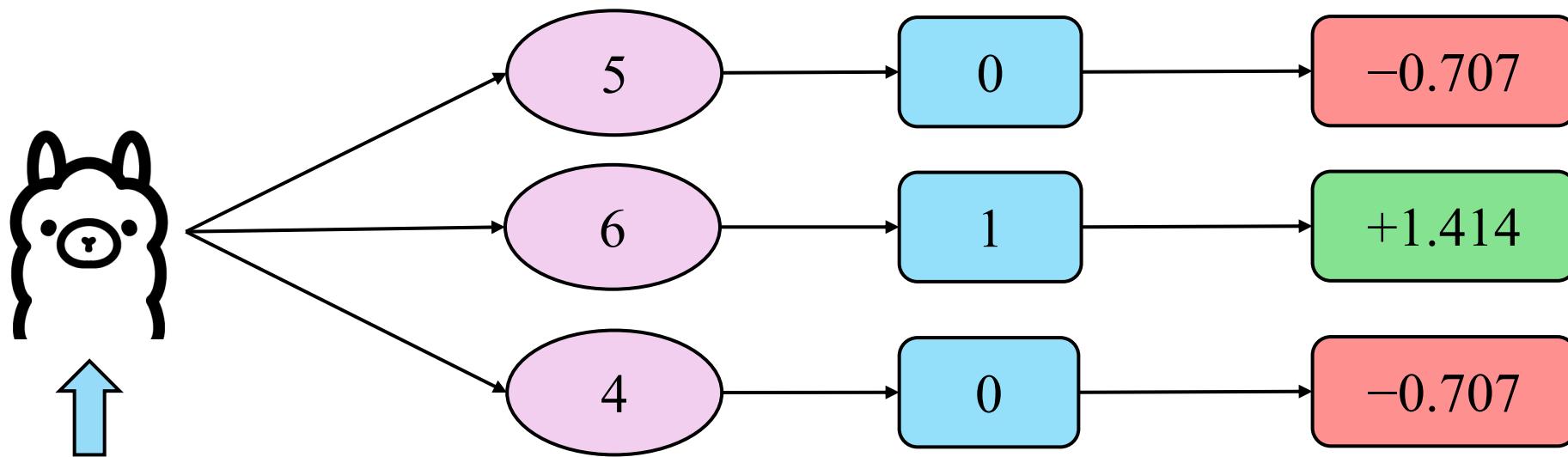
❖ Sampling



Question: Nếu Micah uống 1,5 lít nước vào buổi sáng và uống gấp 3 lần số đó vào buổi chiều thì tổng lượng nước cô ấy uống từ sáng đến chiều là bao nhiêu?

LLM Reasoning

❖ Calculate Advantage



Question: Nếu Micah uống 1,5 lít nước vào buổi sáng và uống gấp 3 lần số đó vào buổi chiều thì tổng lượng nước cô ấy uống từ sáng đến chiều là bao nhiêu?

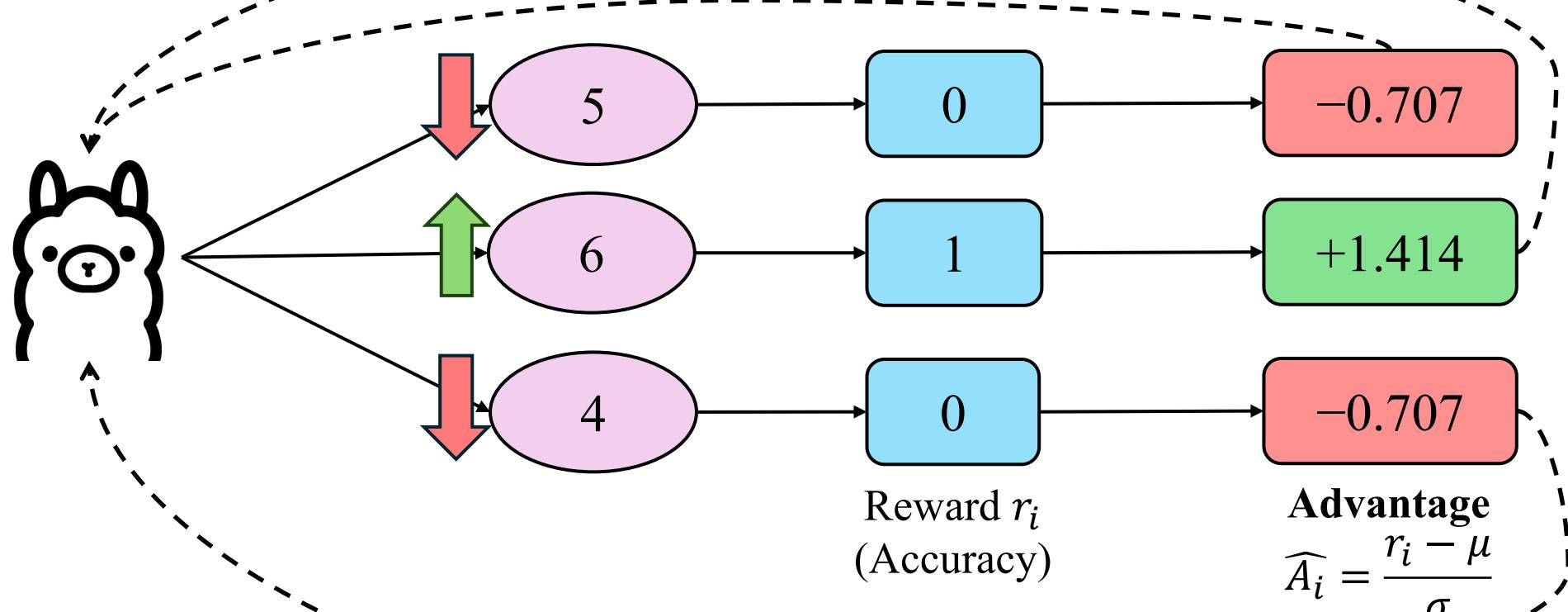
Reward r_i
(Accuracy)

$$\mu = 0.333$$
$$\sigma = 0.471$$

Advantage
$$\widehat{A}_i = \frac{r_i - \mu}{\sigma}$$

LLM Reasoning

❖ Update



LLM Reasoning

❖ Step 4: Define Reward functions

```
1 match_format = re.compile(  
2     rf"^\s*\{\{0,\}\}"  
3     rf"\{reasoning_start}\.+?\{reasoning_end}\.*?"  
4     rf"\{solution_start}\(.+?)\{solution_end}"  
5     rf"\s*\{\{0,\}\}$",  
6     flags=re.MULTILINE | re.DOTALL  
7 )  
8  
9 def match_format_exactly(completions, **kwargs):  
10    scores = []  
11    for completion in completions:  
12        score = 0  
13        response = completion[0]["content"]  
14        if match_format.search(response) is not None:  
15            score += 3.0  
16        scores.append(score)  
17    return scores
```

```
19 def match_format_approximately(completions, **kwargs):  
20    scores = []  
21    for completion in completions:  
22        score = 0  
23        response = completion[0]["content"]  
24        score += 0.5 if response.count(reasoning_start) == 1 else -1.0  
25        score += 0.5 if response.count(reasoning_end) == 1 else -1.0  
26        score += 0.5 if response.count(solution_start) == 1 else -1.0  
27        score += 0.5 if response.count(solution_end) == 1 else -1.0  
28        scores.append(score)  
29    return scores
```

LLM Reasoning

❖ Step 4: Define Reward functions

```
1 match_numbers = re.compile(
2     solution_start + r".*?(?=[\d\.\.,]{1,})",
3     flags=re.MULTILINE | re.DOTALL
4 )
5
6 def check_answer(prompts, completions, answer, **kwargs):
7     responses = [completion[0]["content"] for completion in completions]
8
9     extracted_responses = [
10         guess.group(1)
11         if (guess := match_format.search(r)) is not None else None
12         for r in responses
13     ]
14
15     scores = []
16     for guess, true_answer in zip(extracted_responses, answer):
17         score = 0
18         if guess is None:
19             scores.append(0)
20             continue
21
22         if guess == true_answer:
23             score += 3.0
24         elif guess.strip() == true_answer.strip():
25             score += 1.5
26         else:
27             score -= 1.5
28         scores.append(score)
29
30     return scores
```

```
31 def check_numbers(prompts, completions, answer, **kwargs):
32     question = prompts[0][-1]["content"]
33     responses = [completion[0]["content"] for completion in completions]
34
35     extracted_responses = [
36         guess.group(1)
37         if (guess := match_numbers.search(r)) is not None else None
38         for r in responses
39     ]
40
41     # Print every 5 steps
42     count = getattr(check_numbers, 'counter', 0) + 1
43     check_numbers.counter = count
44     if count % 5 == 0:
45         print('*'*20, f"Question:{question}", f"\nResponse:\n{responses[0]}",
46               f"\nExtracted: {extracted_responses[0]}", f"\nGT Answer: {answer[0]}")
47
48     scores = []
49     for guess, true_answer in zip(extracted_responses, answer):
50         if guess is None:
51             scores.append(0)
52             continue
53         # Convert to numbers
54         try:
55             true_answer = float(true_answer.strip())
56             guess = float(guess.strip().replace(",", ""))
57             scores.append(1.5 if guess == true_answer else -0.5)
58         except:
59             scores.append(0)
60
61     return scores
```

LLM Reasoning

❖ Step 5: Training

```
1 training_args = GRPOConfig(  
2     learning_rate=5e-6,  
3     weight_decay=5e-4,  
4     warmup_ratio=0.1,  
5     lr_scheduler_type="cosine",  
6     optim="adamw_torch_fused",  
7     logging_steps=1,  
8     per_device_train_batch_size=8,  
9     gradient_accumulation_steps=64,  
10    num_generations=8,  
11    max_prompt_length=max_prompt_length,  
12    max_completion_length=max_seq_length - max_prompt_length,  
13    num_train_epochs=1,  
14    max_steps=-1,  
15    save_steps=20,  
16    max_grad_norm=0.1,  
17    report_to="wandb",  
18    output_dir="grpo_lora",  
19 )
```

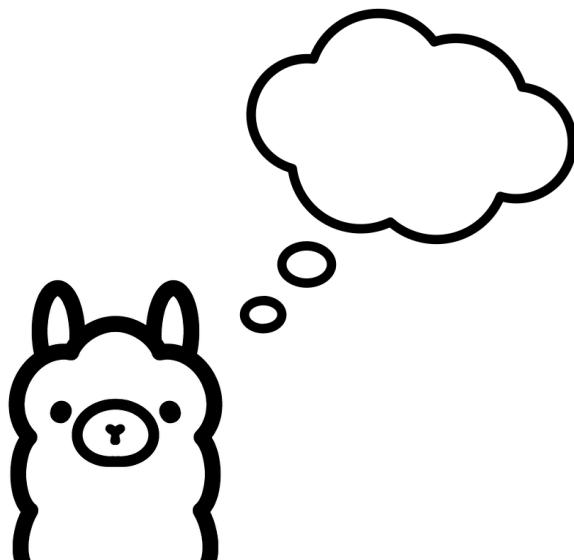
```
1 max_len = max(dataset.map(  
2     lambda x: {"tokens": tokenizer.apply_chat_template(  
3         x["prompt"], add_generation_prompt=True, tokenize=True)},  
4     batched=True,  
5 ).map(lambda x: {"length": len(x["tokens"])})["length"])  
6  
7 max_prompt_length = max_len + 1
```

```
21 trainer = GRPOTrainer(  
22     model=model,  
23     processing_class=tokenizer,  
24     reward_funcs=[  
25         match_format_exactly,  
26         match_format_approximately,  
27         check_answer,  
28         check_numbers,  
29     ],  
30     args=training_args,  
31     train_dataset=dataset,  
32 )  
33 trainer.train()
```

LLM Reasoning

❖ Step 6: Save & Inference model

```
1 model.save_lora("saved_grpo_lora")
```



```
1 idx = 3
2 messages = [
3     {"role": "system", "content": system_prompt},
4     {"role": "user", "content": train_dataset[idx]["question"]},
5 ]
6 sampling_params = SamplingParams(
7     temperature = 0.8,
8     top_p = 0.95,
9     max_tokens = 1024,
10 )
11
12 text = tokenizer.apply_chat_template(
13     messages,
14     add_generation_prompt = True,
15     tokenize = False,
16 )
17
18 path_lora = "saved_grpo_lora"
19 output = model.fast_generate(
20     [text],
21     sampling_params = sampling_params,
22     lora_request = model.load_lora(path_lora),
23 )[0].outputs[0].text
24
25 print(f"Problem:\n{train_dataset[idx]['question']}")
26 print(f"Response:\n{output}")
27 print("GT Answer:", train_dataset[idx]["answer"])
```

LLM Reasoning

❖ Step 6: Save & Inference model

Processed prompts: 100% |██████████| 1/1 [00:02<00:00, 2.64s/it, est. speed input: 43.96 toks/s, output: 65.95 toks/s] Problem:
Nếu x là bội số dương của 8, x^2 lớn hơn 100 và x nhỏ hơn 20, thì giá trị của x là bao nhiêu?
Response:
</thinking>

Để tìm giá trị của x , ta cần thỏa mãn cả ba điều kiện: x là bội số của 8, x^2 lớn hơn 100 và x nhỏ hơn 20.

Xét điều kiện thứ nhất, ta thấy các bội số của 8 là 8, 16, 24, ..., 20.

Tiếp theo, xét điều kiện thứ hai, ta có $x^2 > 100$, do đó $x > 10$.

Cuối cùng, xét điều kiện thứ ba, ta thấy $x < 20$.

Vì x là bội số của 8, do đó x có thể là 16.

Vậy, giá trị của x thỏa mãn tất cả ba điều kiện là 16.

<answer>16</answer>

GT Answer: 16



LLM Reasoning

❖ Step 6: Save & Inference model

- **Evaluation:** 1000 samples from Vietnamese MetaMathQA
- **Example:** Markus gấp đôi tuổi con trai ông và con trai Markus gấp đôi tuổi cháu trai Markus. Nếu tổng số tuổi của Markus, con trai ông và cháu trai ông là 140 tuổi thì cháu trai của Markus bao nhiêu tuổi?
- **GT Answer:** 20

Llama-3.2-3B-Instruct
Original

- Average Time: 0.71s / sample
- Accuracy: 9%
- Predict: 10 ❌



LLM Reasoning

❖ Step 6: Save & Inference model

You are given a problem.
Think about the problem and provide your thought process.
Place it between **<thinking>** and **</thinking>**.
Then, provide your final answer between **<answer></answer>**

- **Evaluation:** 1000 samples from Vietnamese MetaMathQA
- **Example:** Markus gấp đôi tuổi con trai ông và con trai Markus gấp đôi tuổi cháu trai Markus. Nếu tổng số tuổi của Markus, con trai ông và cháu trai ông là 140 tuổi thì cháu trai của Markus bao nhiêu tuổi?
- **GT Answer:** 20

Llama-3.2-3B-Instruct with CoT

- Average Time: 3.2s / sample
- Accuracy: 11.9%
- Format score: 0.4155
- Predict: 20 ✓

<thinking>Markus gấp đôi tuổi con trai mình, còn con trai Markus lại gấp đôi tuổi cháu trai của ông. Gọi tuổi cháu trai là x, thì con trai Markus là 2x và Markus là 4x. Tổng ba người là 140 tuổi nên ta có:
 $4x + 2x + x = 140 \rightarrow 7x = 140 \rightarrow x = 20.$
Vậy cháu trai Markus 20 tuổi.</thinking>
<answer>20</answer> ✓

LLM Reasoning

❖ Step 6: Save & Inference model

You are given a problem.

Think about the problem and provide your thought process.

Place it between **<thinking>** and **</thinking>**.

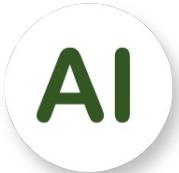
Then, provide your final answer between **<answer></answer>**

- **Evaluation:** 1000 samples from Vietnamese MetaMathQA
- **Example:** Markus gấp đôi tuổi con trai ông và con trai Markus gấp đôi tuổi cháu trai Markus. Nếu tổng số tuổi của Markus, con trai ông và cháu trai ông là 140 tuổi thì cháu trai của Markus bao nhiêu tuổi?
- **GT Answer:** 20

Llama-3.2-3B-Instruct
after training GRPO

- Average Time: 3.6s / sample
- Accuracy: 31.1%
- Format score: 0.7875
- Predict: 20 

<thinking>Để giải bài toán, ta đặt tuổi cháu trai Markus là x. Khi đó, tuổi Markus là 2x, còn con trai ông là 4x. Tổng tuổi ba người là $x + 2x + 4x = 7x$. Biết tổng là 140, ta có $7x = 140 \Rightarrow x = 20$. Vậy cháu trai Markus 20 tuổi.</thinking>
<answer>20</answer> 



AI

AI VIET NAM
@aivietnam.edu.vn

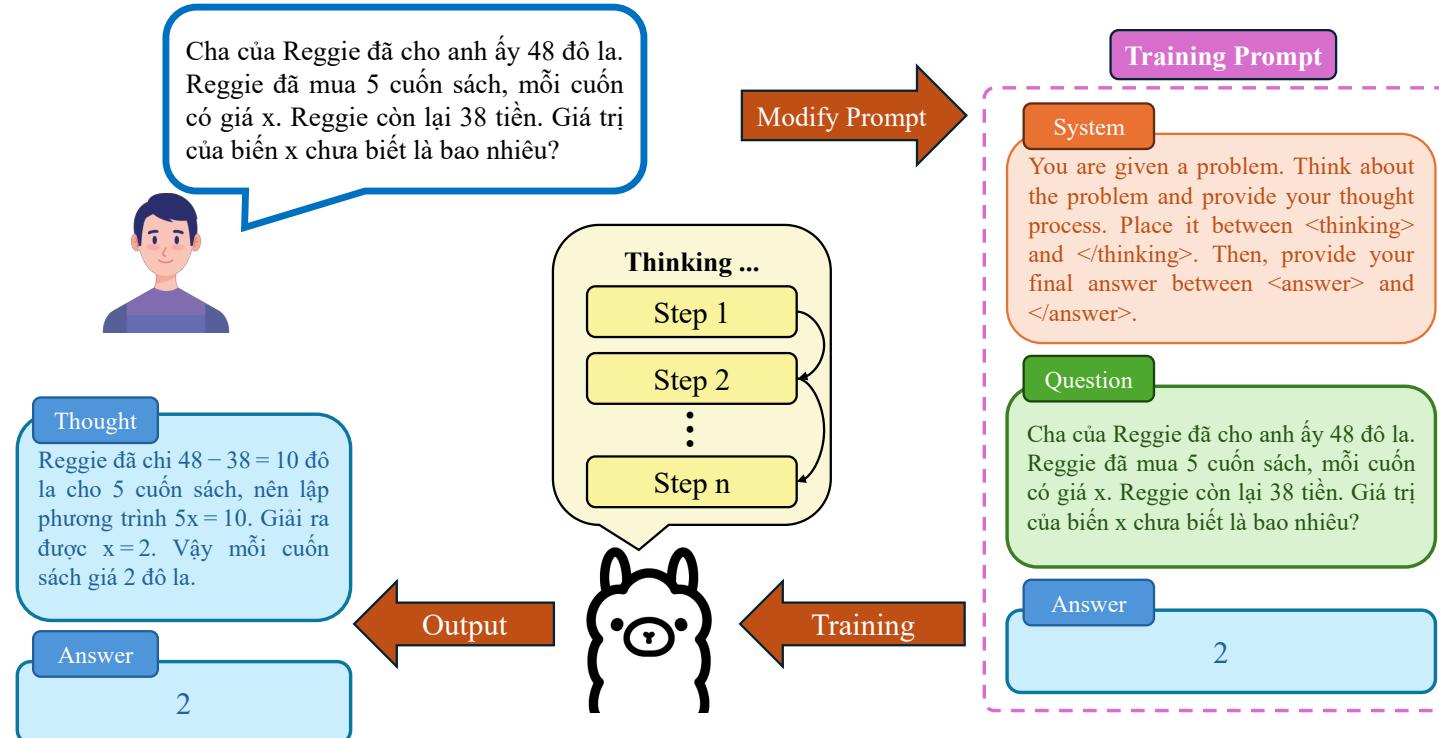
QUIZ

 AI

AI VIET NAM
@aivietnam.edu.vn

Summarization and Q&A

Summarization and Q&A



In this study session, we have discussed

about:

1. Introduction to LLM with Reasoning.
2. Introduction to prompting techniques to make LLMs reasoning.
3. Introduction to LLM Reasoning.
4. Fine-tuning LLMs for Reasoning.
5. Implementing an LLM Reasoning application for Math Solving.

Summarization and Q&A

