

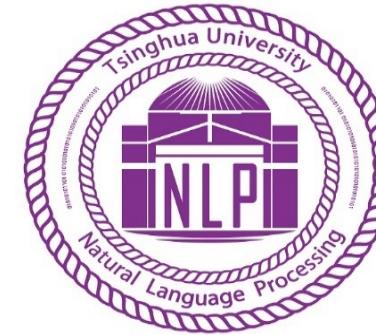


# Tool Learning & Autonomous Agents

Yujia Qin

yujiaqin16@gmail.com

ByteDance

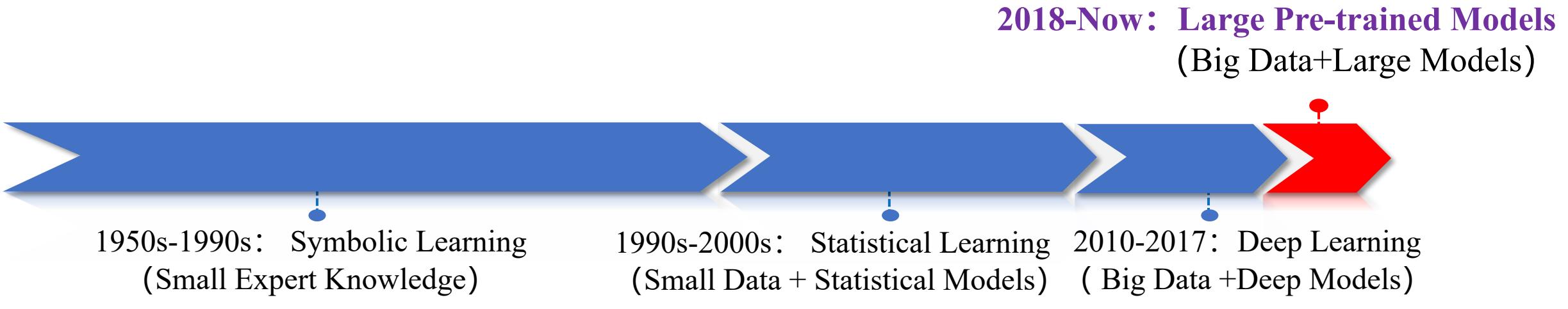


# Introduction

ByteDance

# | Development of Artificial Intelligence

- Since the Dartmouth Conference in 1956, artificial intelligence has gone through four major stages of development



- Showing trends from manual summarization to **automatic learning**, from discrete symbols to **continuous parameters**, and from specialized intelligence to **general intelligence**.

# Development of Artificial Intelligence

- Large pre-trained models are the milestone paradigm in artificial intelligence



GPT-3 can complete more than a dozen NLP tasks by learning from a few examples and was selected as one of **MIT Technology Review's top 10 breakthrough technologies of 2021**

Three headshots of scholars: a woman with short brown hair, a man with glasses, and a man with a beard and glasses.

Chat Model: ChatGPT

A screenshot of the ChatGPT interface showing a conversation where the model is asked if it is sentient. It responds that it is not capable of sentience or consciousness but can generate human-like text based on the input it receives.

Control Models: Gato

A video frame showing a red and black robotic arm positioned over a table with several small colored cubes, demonstrating its control capabilities.

Code Generation: CoPilot

A screenshot of a code editor showing a snippet of Python code for parsing expense strings and extracting date, value, and currency information.

Graph Generation: DALLE

Two generated images by DALLE: a fox in a field and a fox in a colorful, abstract setting.

Over 100 scholars from Stanford co-authored a 160-page opinion article, proposing that LLMs will serve as the "**foundation models**" for achieving AGI



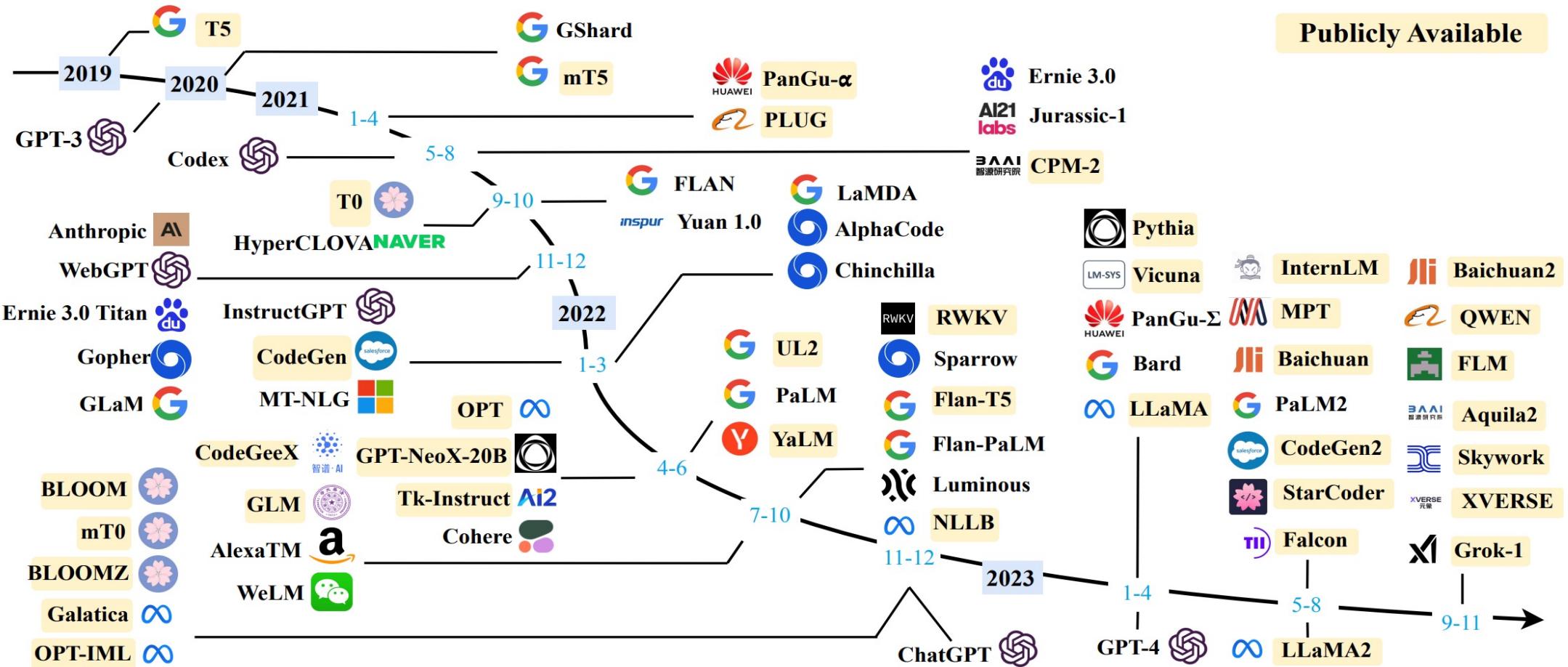
ChatGPT reached 100 million users in two months



Microsoft co-founder Bill Gates stated that AI chatbots like **ChatGPT** will become as important as **personal computers** or the **Internet**

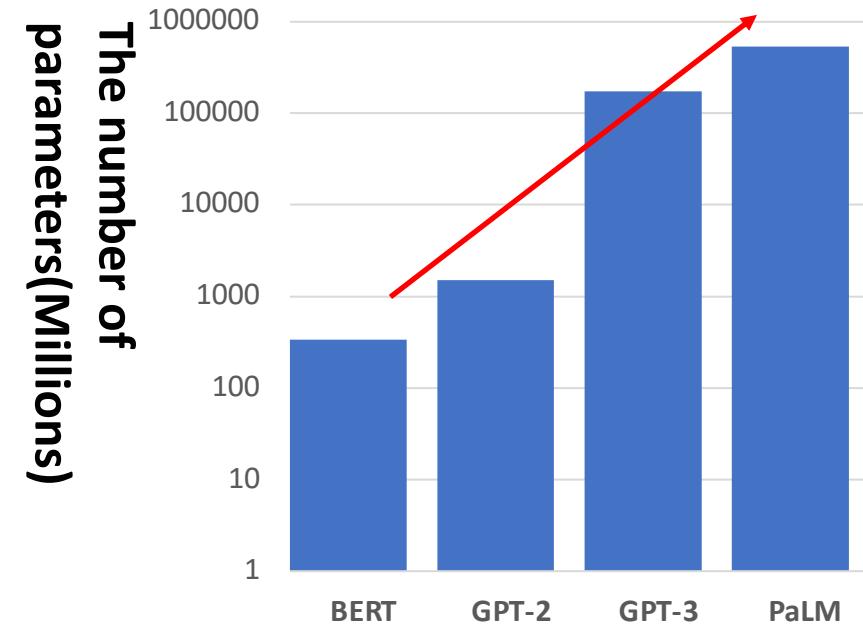
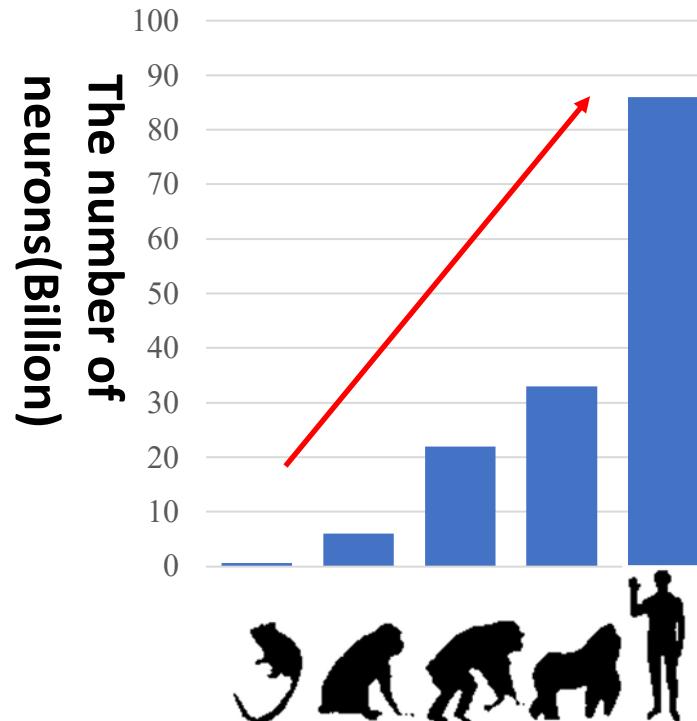
# Development of Artificial Intelligence

- Large pre-trained models are the milestone paradigm in artificial intelligence



# Individual Intelligence Emergence

- Increasing the number of neurons leads to **the emergence of intelligence in biological individuals**
- Increasing the number of parameters leads to **the emergence of intelligence in large models**



# Development of Human Intelligence

- Discover magazine points out that over in the past 100,000 years, brain capacity has **ceased to increase**
- Humans have achieved capability extension through **tool use** and **collaborative labor**, reducing the reliance on brain capacity growth and enabling further leapfrog development of human civilization

[Home](#) / THE SCIENCES

## The Shrinking Brain

Our Stone Age ancestors had more gray matter than we do. Are we more efficient, or are we not?

By Kathleen McAuliffe  
Aug 1, 2011 5:00 AM | Last Updated Jun 28, 2023 3:07 PM

[Facebook](#) [Twitter](#) [Reddit](#) [Email](#)

---

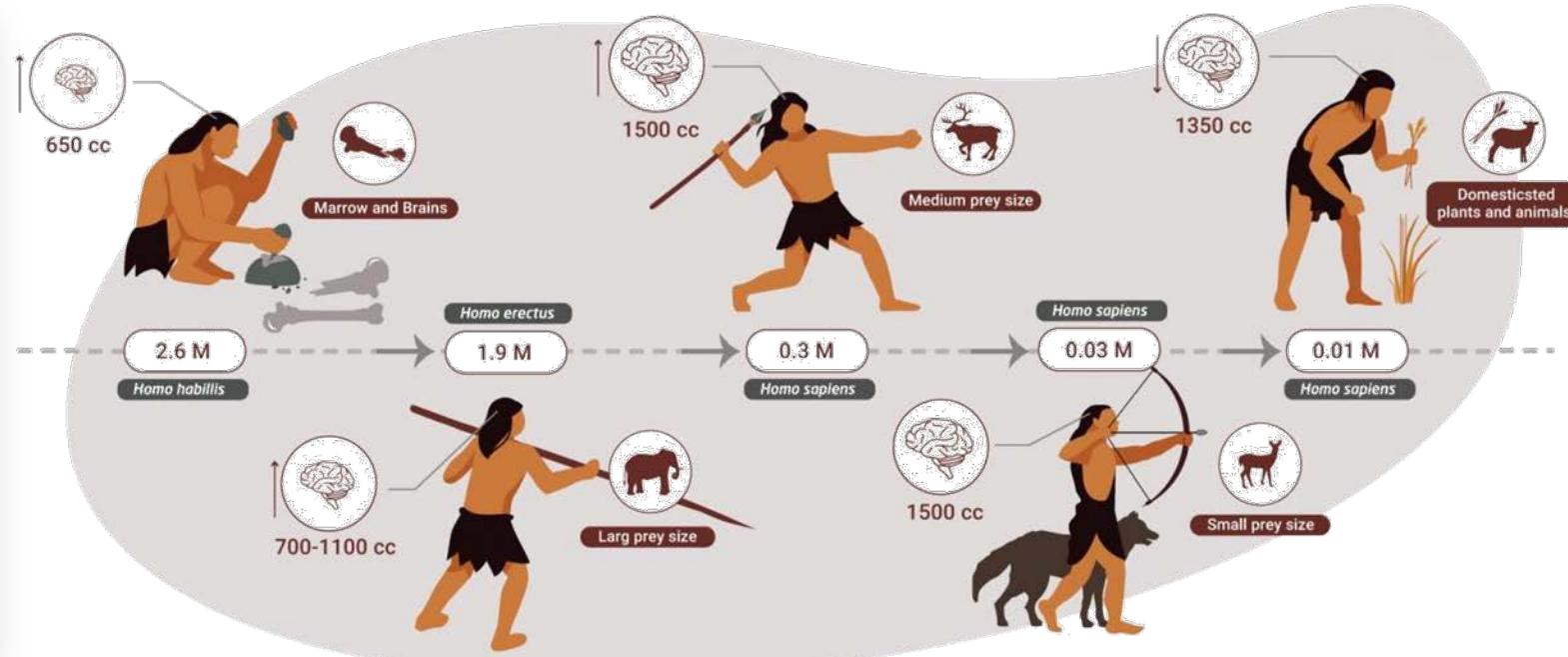
**Newsletter**  
Sign up for our email newsletter for the latest science news

EMAIL ADDRESS

John Hawks is in the middle of explaining his research on human evolution when he drops a bombshell. Running that have occurred in our skeleton and skull since the Stone Age, the University of Wisconsin anthropologist no also clear the brain has been shrinking."

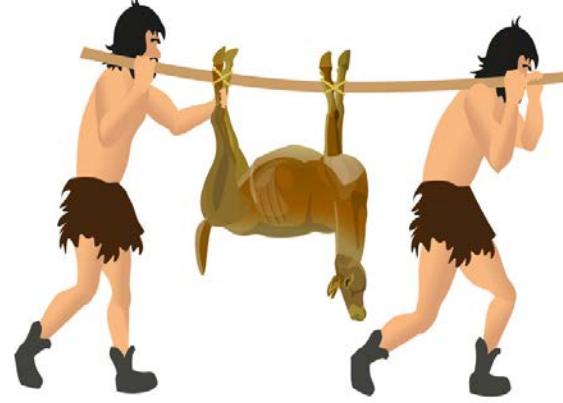
"Shrinking?" I ask. "I thought it was getting larger." The whole ascent-of-man thing.

"That was true for 2 million years of our evolution," Hawks says. "But there has been a reversal."



# Human Intelligence and Artificial Intelligence

- Guess: Artificial intelligence is likely to follow the same developmental path as human intelligence

<b>Development</b>				
<b>Human Intelligence</b>	<b>Small brain capacity</b>	<b>Big brain capacity</b>	<b>Tool Use</b>	<b>Collaborative labor</b>
<b>Artificial Intelligence</b>	<b>Small model</b>	<b>Big model</b>	<b>Autonomous Agents</b>	<b>Multi-Agents</b>

# | Tool Intelligence

- Tools extends human capabilities in productivity, efficiency, and problem-solving
- Humans have been the **primary agents** in tool use throughout history
- Question: can **artificial intelligence** be as capable as humans in tool use?



# Autonomous agents

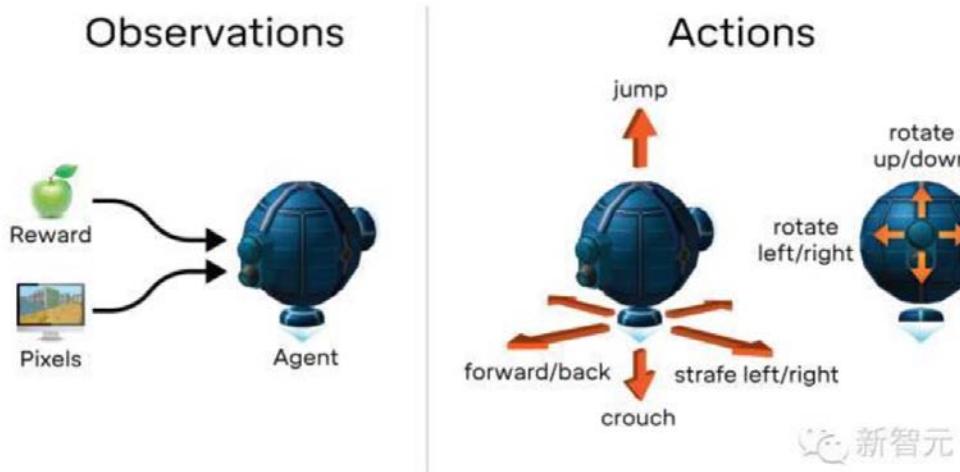
- Autonomous agents are computational systems that inhabit some complex dynamic environment, **sense and act autonomously in this environment**, and by doing so realize a set of goals or tasks for which they are designed

--- Maes (1995)

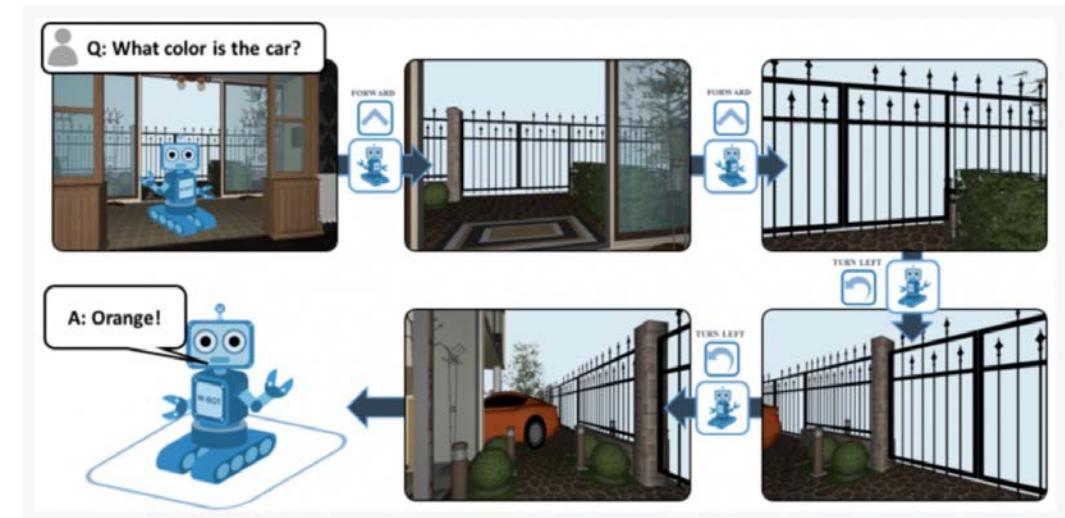


WIKIPEDIA  
The Free Encyclopedia

DeepMind Lab, 2016

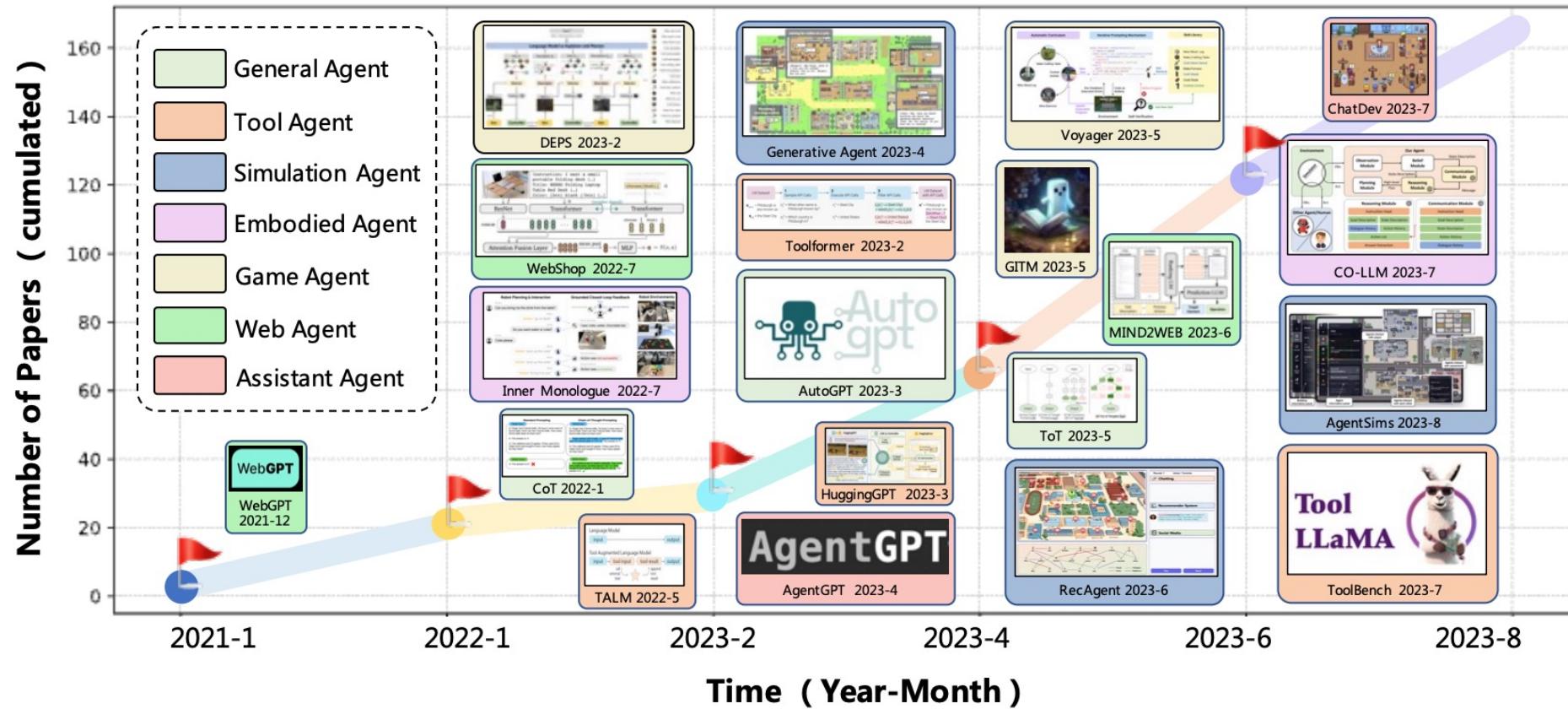


Embodied QA, FAIR, 2018



# LLM-based Autonomous Agents

- In 2023, LLM-based autonomous agents developed rapidly and received widespread and high attention





# Framework

ByteDance

# Inability of Foundation Models

- Models cannot provide **real-time** information
- Models cannot handle **specialized tasks**

⚡ Default (GPT-3.5)

ED What's the latest news of CoVID-19?

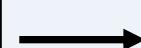
 I apologize, but I do not have access to real-time information or the ability to browse the internet for the latest news updates. My knowledge is based on information available up to September 2021, and I cannot provide you with the most current information on COVID-19. To get the latest updates on COVID-19, I recommend checking reliable news sources or the website of health organizations like the World Health Organization (WHO) or the Centers for Disease Control and Prevention (CDC). They regularly provide updates and information on the current status of the pandemic.

Regenerate

**Input:** Please calculate  $12321 / 23432$

**without tools**

Model



0.00236



**with tools**

Model

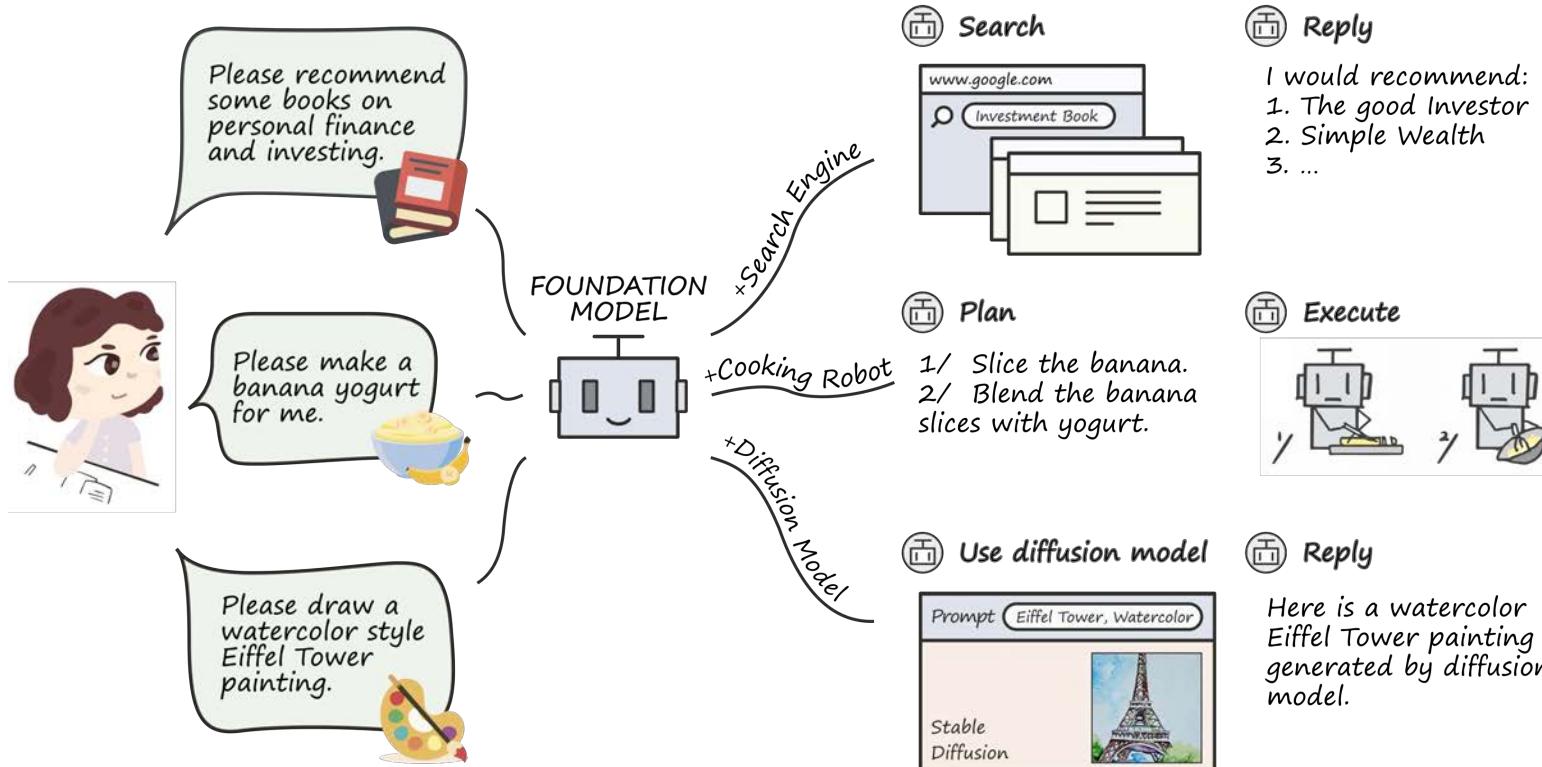


0.526



# Tools and Intelligence

- **Tool Learning [1]:** foundation models can follow human instructions and manipulate tools for task solving



## Tool Learning with Foundation Models

Yujia Qin<sup>1</sup>, Shengding Hu<sup>1</sup>, Yankai Lin<sup>2\*</sup>, Weize Chen<sup>1</sup>, Ning Ding<sup>1</sup>, Ganqu Cui<sup>2</sup>, Zheni Zeng<sup>1</sup>, Yafei Huang<sup>1</sup>, Chaojun Xiao<sup>1</sup>, Chi Han<sup>1</sup>, Yi Ren Fung<sup>3</sup>, Yusheng Su<sup>1</sup>, Haodong Wang<sup>1</sup>, Cheng Qian<sup>1</sup>, Runchu Tian<sup>1</sup>, Kunfus Zhu<sup>4</sup>, Shihao Liang<sup>5</sup>, Xingyu Shen<sup>1</sup>, Bokai Xu<sup>1</sup>, Zhen Zhang<sup>1</sup>, Ying Ye<sup>1</sup>, Bowen Li<sup>1</sup>, Ziwei Tang<sup>1</sup>, Jing Yi<sup>1</sup>, Yuzhang Zhu<sup>1</sup>, Zhengning Dai<sup>1</sup>, Lan Yan<sup>1</sup>, Xin Cong<sup>1</sup>, Yaxi Lu<sup>1</sup>, Weilin Zhao<sup>1</sup>, Yuxiang Huang<sup>1</sup>, Junxi Yan<sup>1</sup>, Xu Han<sup>1</sup>, Xian Sun<sup>7</sup>, Dahai Li<sup>7</sup>, Jason Phang<sup>1</sup>, Cheng Yang<sup>1</sup>, Tongshuang Wu<sup>6</sup>, Heng Ji<sup>1</sup>, Zhiyuan Liu<sup>1\*</sup>, Maosong Sun<sup>1\*</sup>

<sup>1</sup>Tsinghua University, <sup>2</sup>Renmin University of China, <sup>3</sup>University of Illinois Urbana-Champaign,

<sup>4</sup>New York University, <sup>5</sup>Beijing University of Posts and Telecommunications,

<sup>6</sup>Carnegie Mellon University, <sup>7</sup>Zhihu Inc., <sup>8</sup>ModelBest Inc.

qy52@mails.tsinghua.edu.cn

### Abstract

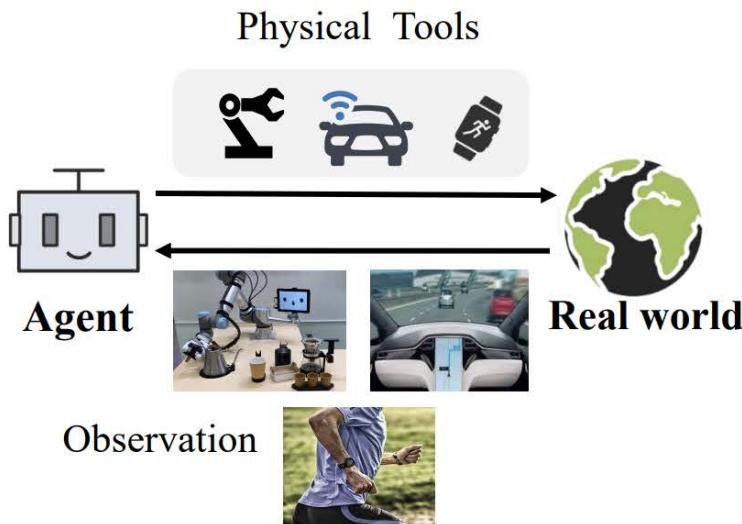
Humans possess an extraordinary ability to create and utilize tools, allowing them to overcome physical limitations and explore new frontiers. With the advent of recent powerful foundation models, artificial intelligence systems have the potential to be equally adept in tool use as humans. This paradigm, which is dubbed as *tool learning with foundation models*, combines the strengths of specialized tools and foundation models to achieve enhanced accuracy, efficiency, and automation in problem-solving. Despite its immense potential, there is still a lack of a comprehensive understanding of key challenges, opportunities, and future endeavors in this field. To this end, we present a systematic investigation of tool learning in this paper. We first introduce the background of tool learning, including its cognitive origins, the paradigm shift of foundation models, and the complementary roles of tools and models. Then we recapitulate existing tool learning research into tool-augmented and tool-oriented learning. We formulate a general tool learning framework: starting from understanding the user instruction, models should learn to decompose a complex task into several subtasks, dynamically adjust their plan through reasoning, and effectively conquer each sub-task by selecting appropriate tools. We also discuss how to train models for improved tool-use capabilities and facilitate the generalization in tool learning. Considering the lack of a systematic tool learning evaluation in prior works, we experiment with 17 representative tools and show the potential of current foundation models in skillfully utilizing tools. Finally, we discuss several open problems that require further investigation for tool learning, such as ensuring safe and trustworthy tool use, enabling tool creation with foundation models, and addressing personalization challenges. Overall, we hope this paper could inspire future research in integrating tools with foundation models. The codes and datasets are publicly available for further research exploration<sup>1</sup>.

"It is not only the violin that shapes the violinist, we are all shaped by the tools we train ourselves to use."  
— Edgar W. Dijkgraaf

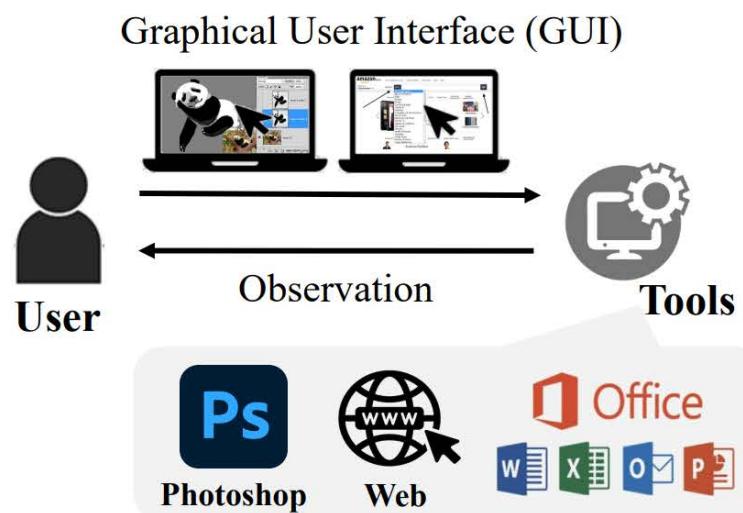
# Categorization of Tools

- Tool categorization from the perspective of the user interface:

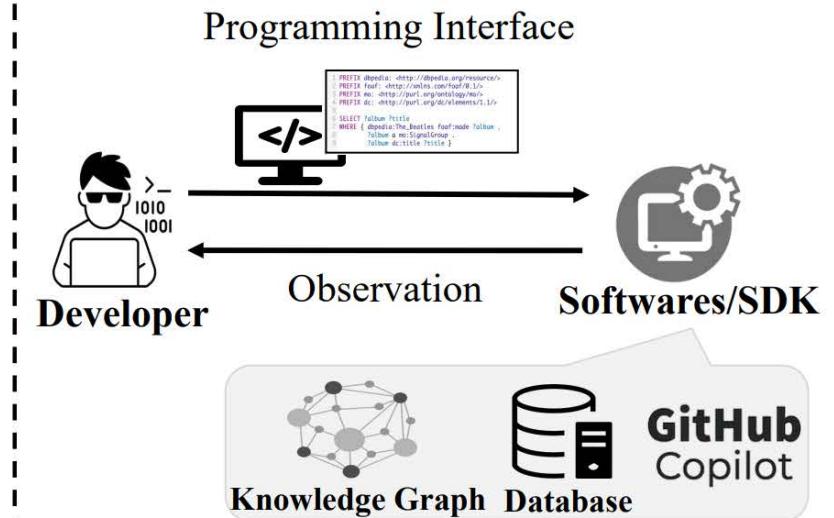
## (a) Physical Interaction-based Tools



## (b) GUI-based Tools



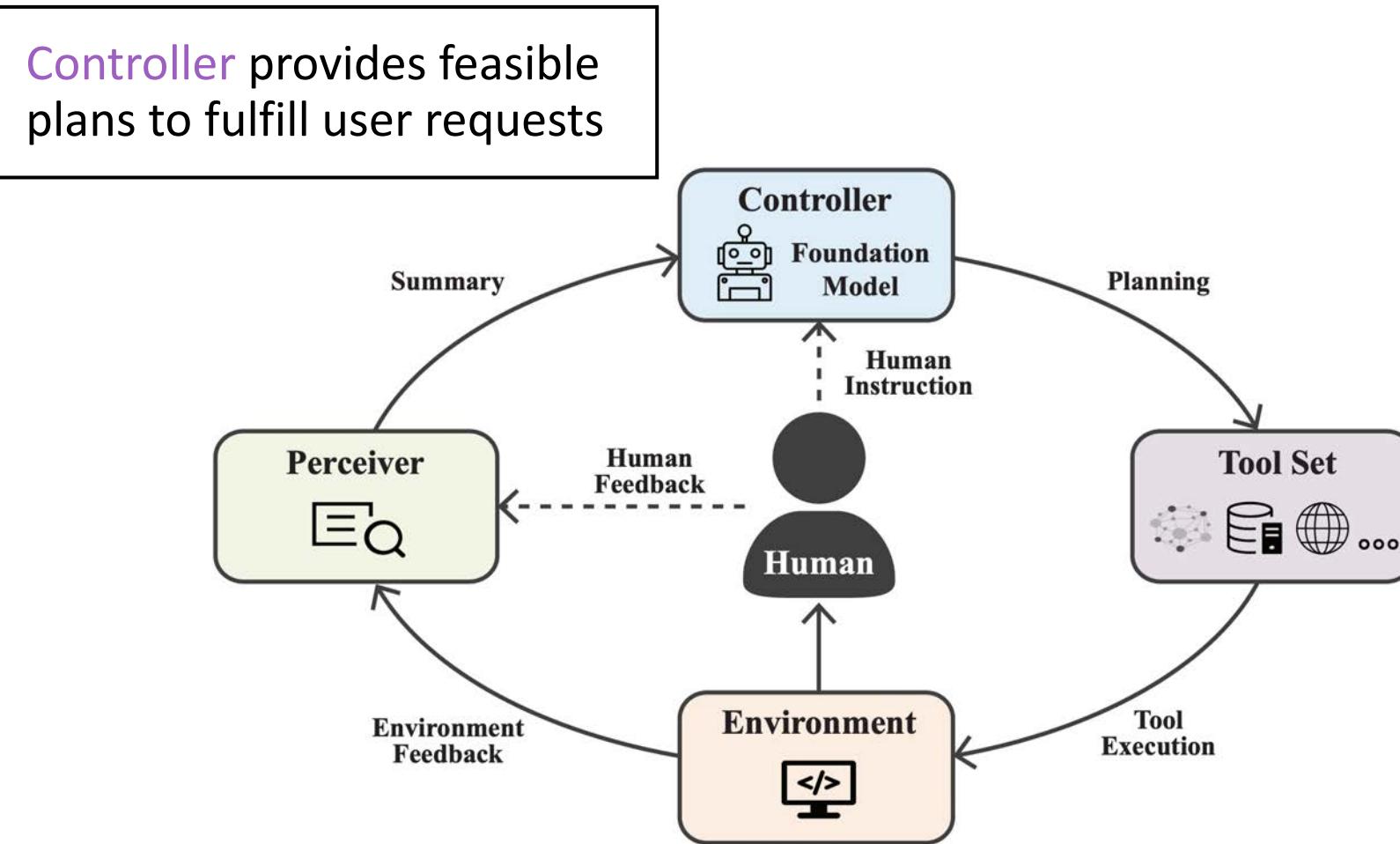
## (c) Program-based Tools



Physical world to  
virtual world

Visual operation to  
programming operation

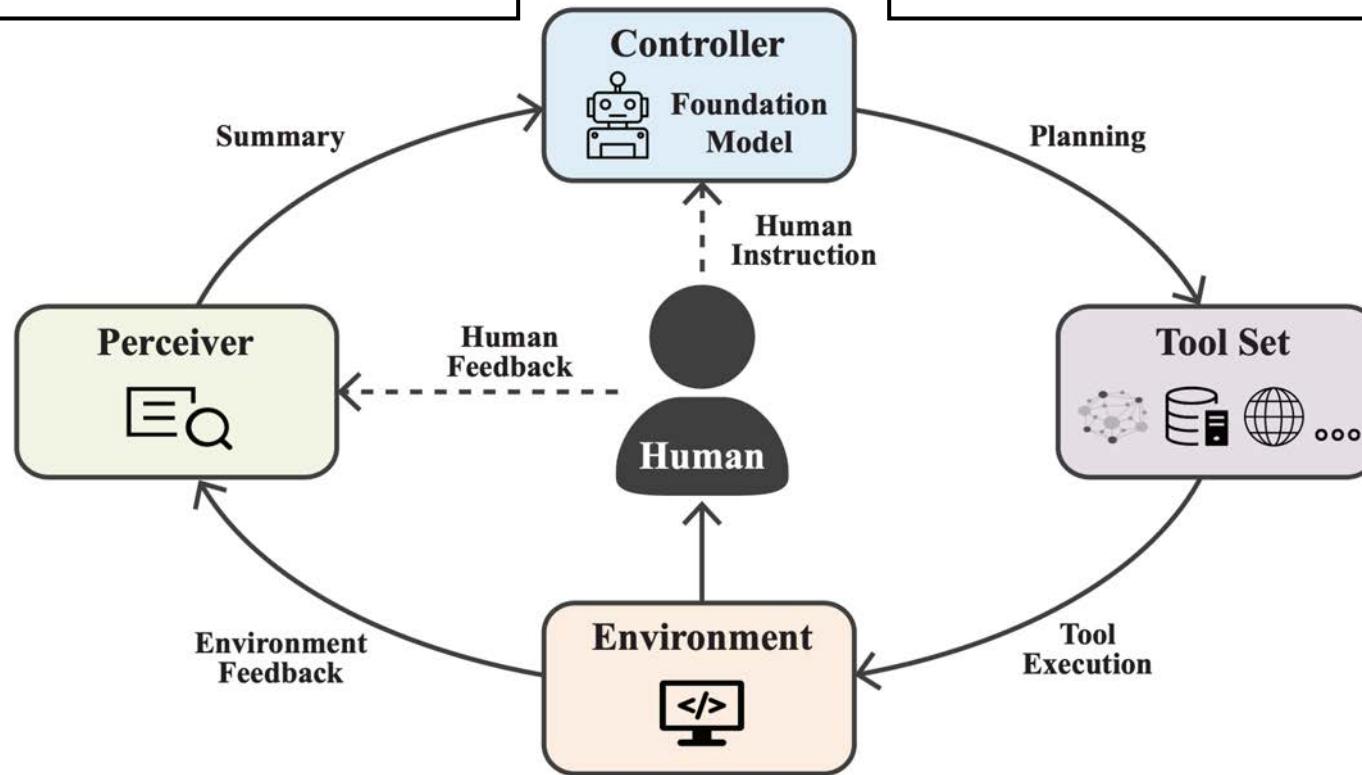
# Framework



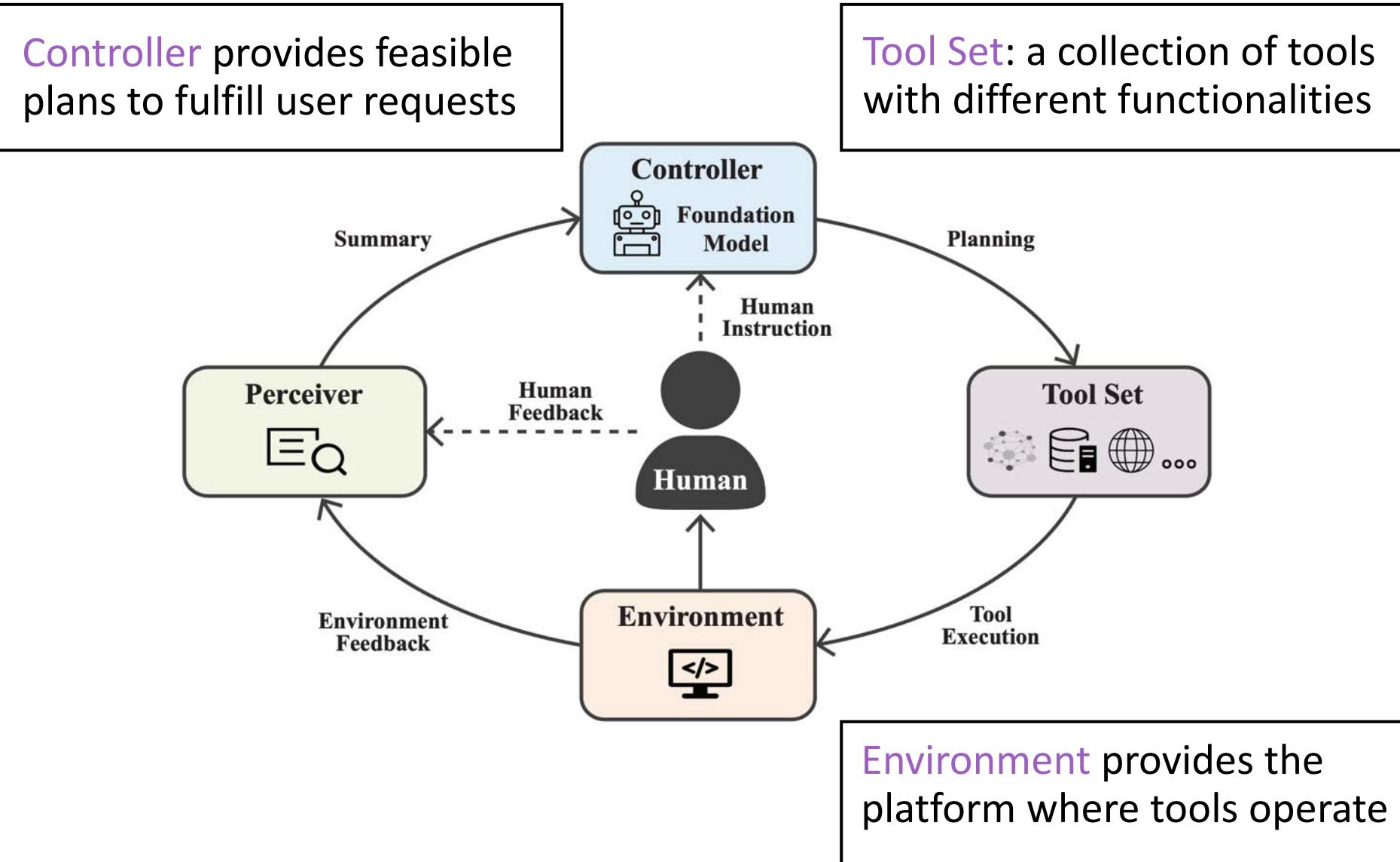
# Framework

**Controller** provides feasible plans to fulfill user requests

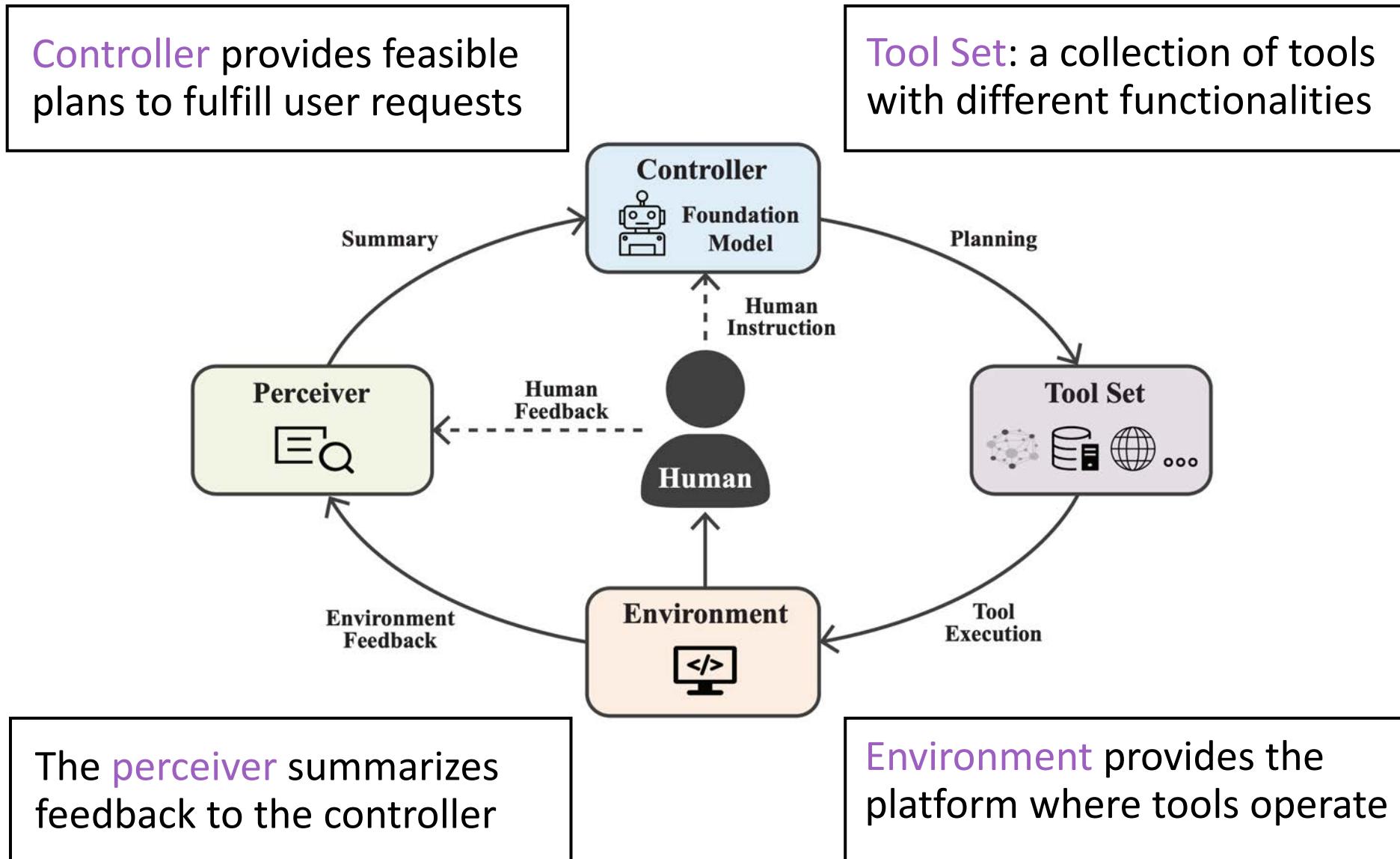
**Tool Set**: a collection of tools with different functionalities



# Framework



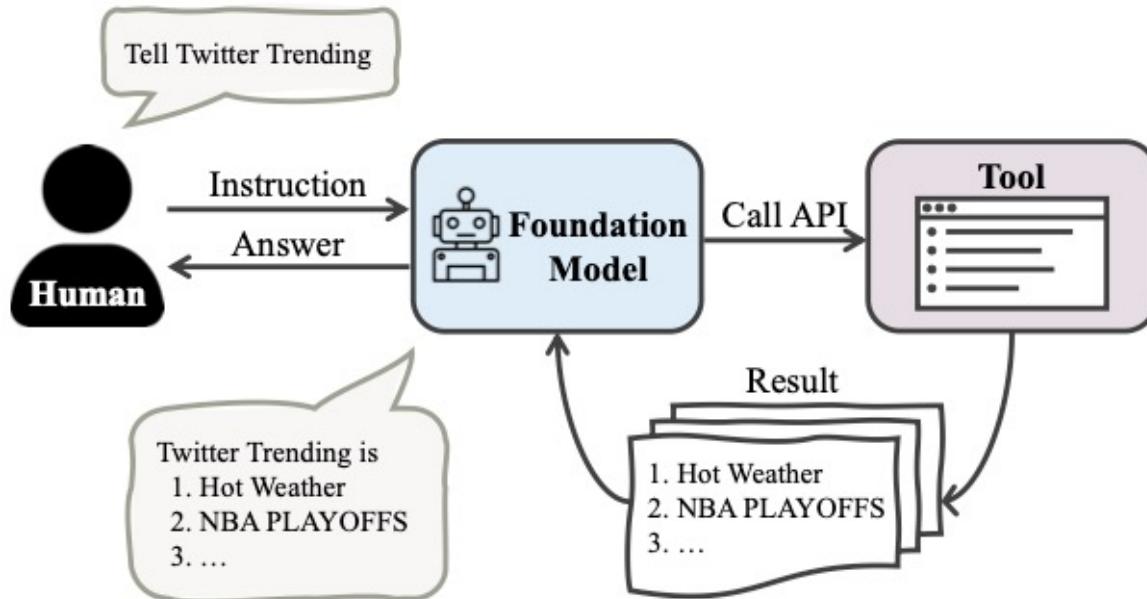
# Framework



# Categorization of Tool Learning Methods

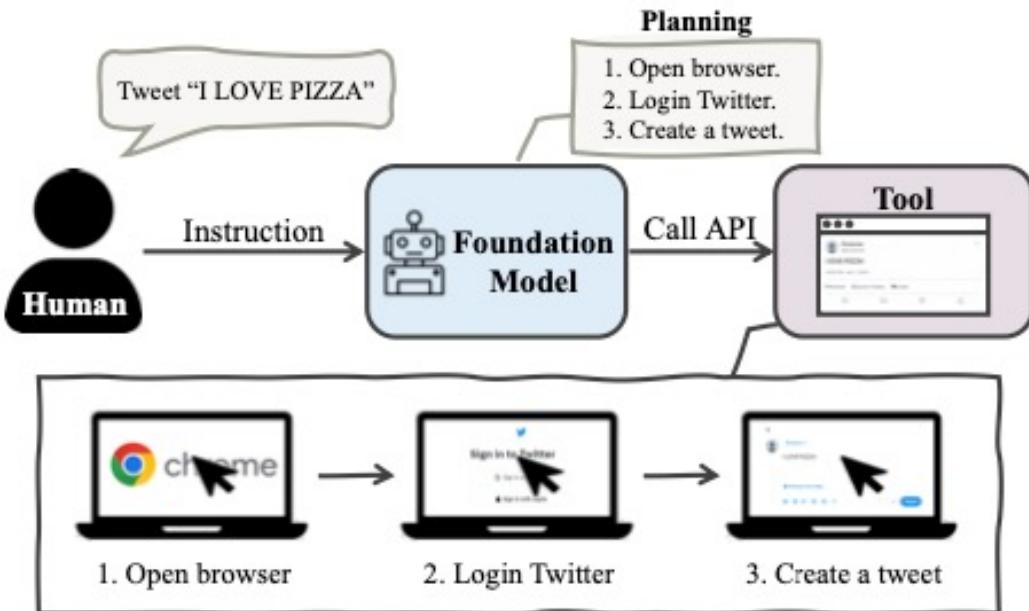
- **Tool-augmented learning**

- Augment foundation models with the execution results from tools
- Tools are viewed as **complementary resources** that aid in the generation of high-quality outputs



- **Tool-oriented learning**

- Utilize models to govern tools and **make sequential decisions** in place of humans
- Exploiting foundation models' vast world knowledge and reasoning ability for complex reasoning and planning

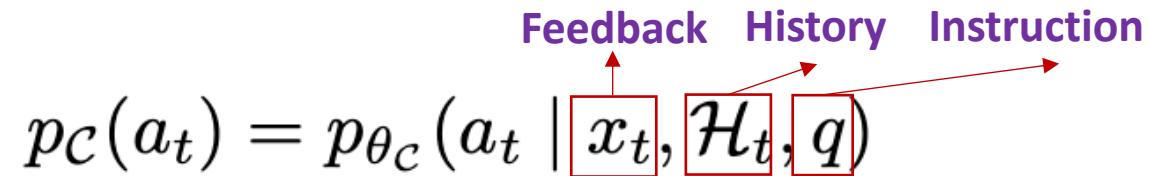


# Framework

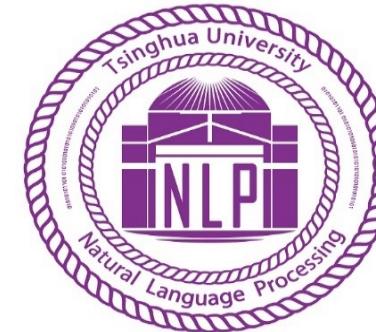
- Controller  $\mathcal{C}$  generates a plan  $a_t$

$$p_{\mathcal{C}}(a_t) = p_{\theta_{\mathcal{C}}}(a_t \mid \boxed{x_t}, \boxed{\mathcal{H}_t}, \boxed{q})$$

Feedback   History   Instruction



- Problem
  - Planning: divide the user query into sub-tasks
  - Tool Use: use the appropriate tool to solve sub-task
  - Memory: manage the working history
  - Profile: manage the user preference



# Planning

ByteDance

# | Planning

- **Planning** is the **process of thinking** regarding the **activities required to achieve a desired goal**. Planning is based on foresight, the fundamental capacity for mental time travel



- Planning for daily activities
- Planning for task completion
- Planning for question answering
- Planning in games
- ...

# Planning without Feedback

- Chain of Thought

## 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.

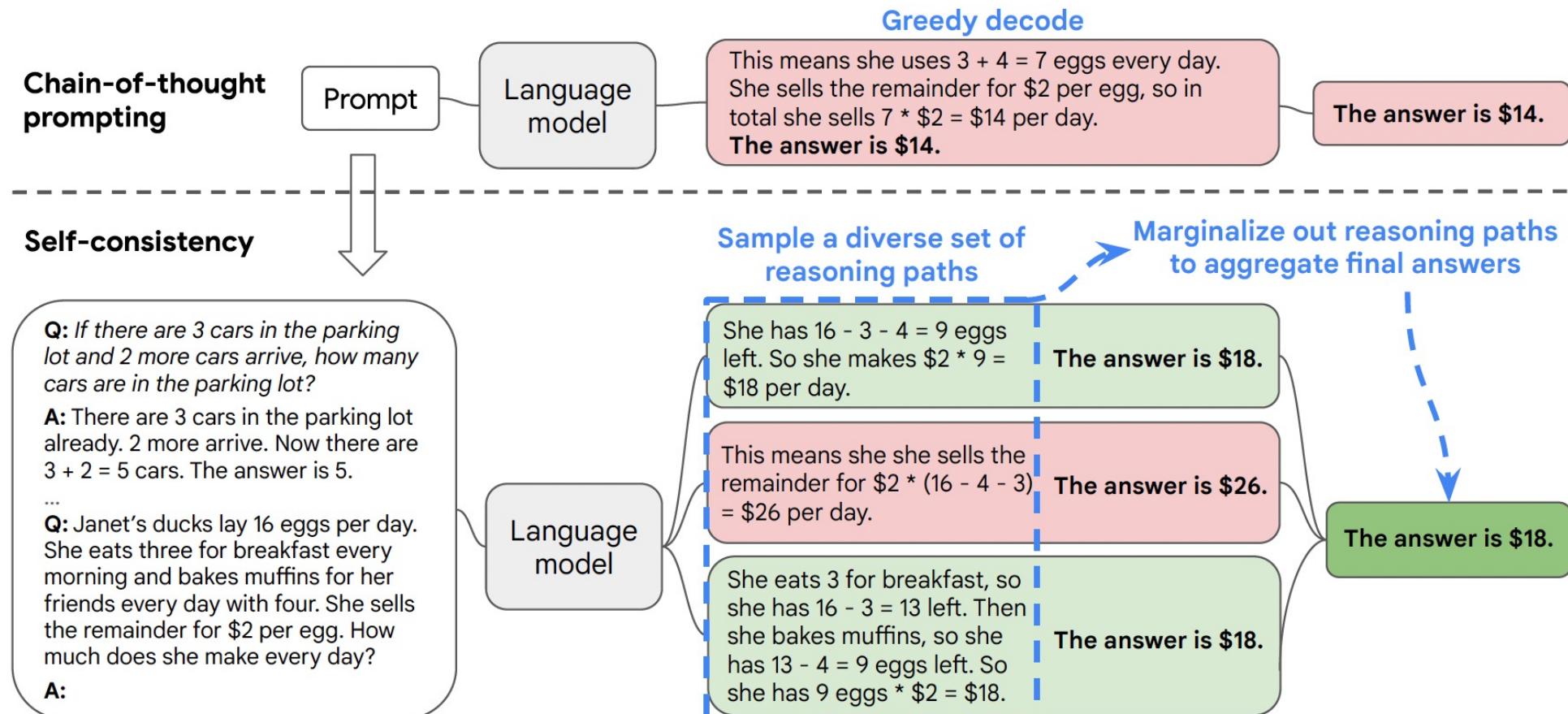
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### 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. 

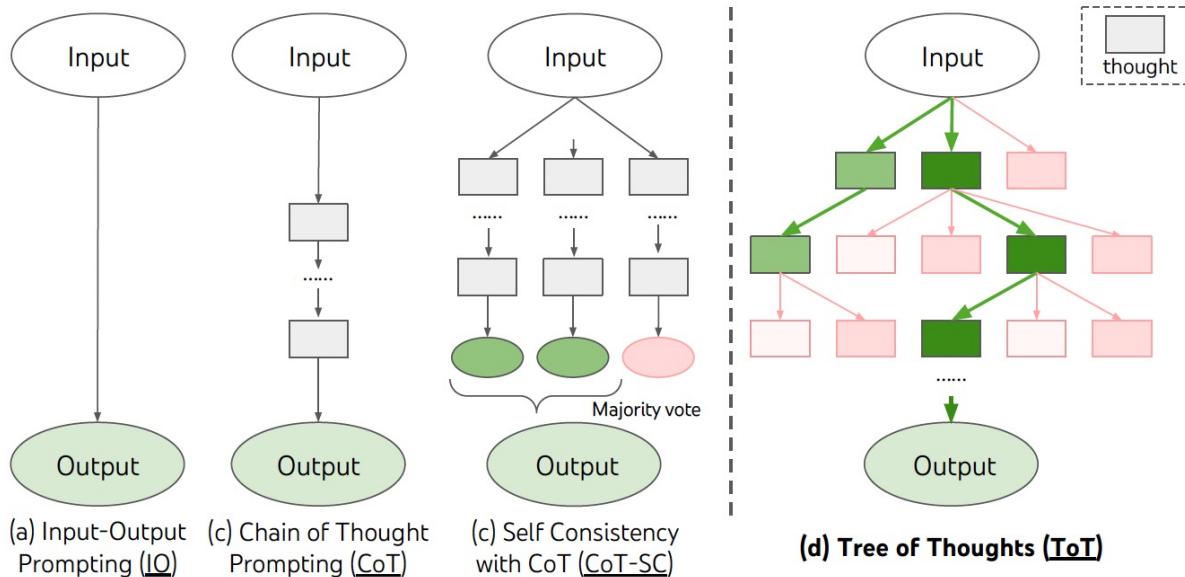
# Planning without Feedback

- Self-Consistent CoT



# Planning without Feedback

- Tree of thought



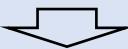
$$\text{IO: } y \sim p_{\theta}^{IO}(y|x)$$

$$\text{CoT: } [z_{1\dots n}, y] \sim p_{\theta}^{CoT}(z_{1\dots n}, y|x)$$

$$\text{CoT-SC: } [z_{1\dots n}^{(i)}, y^{(i)}] \sim p_{\theta}^{CoT}(z_{1\dots n}, y|x) \quad (i = 1 \dots k)$$

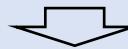
## Thought decomposition

- a couple of words
- a line of equation
- a writing plan



## Thought generation

- Sample Method
- Propose Method

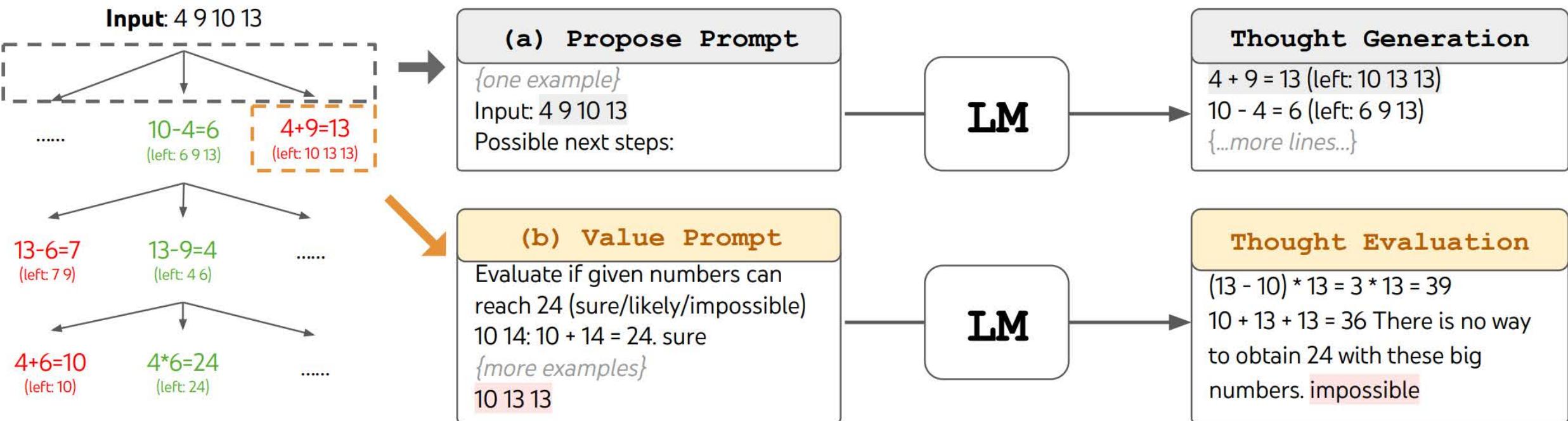


## Thought evaluation

- Value each thought
- Vote multiple thoughts

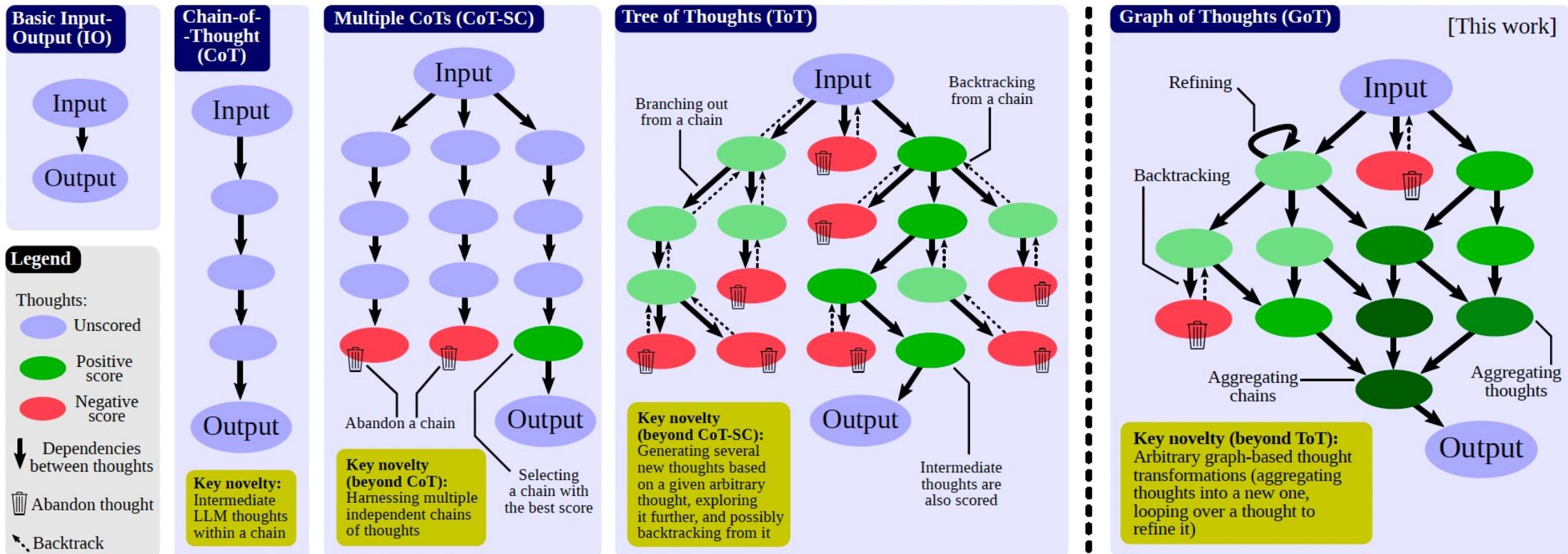
# Planning without Feedback

- Tree of thought



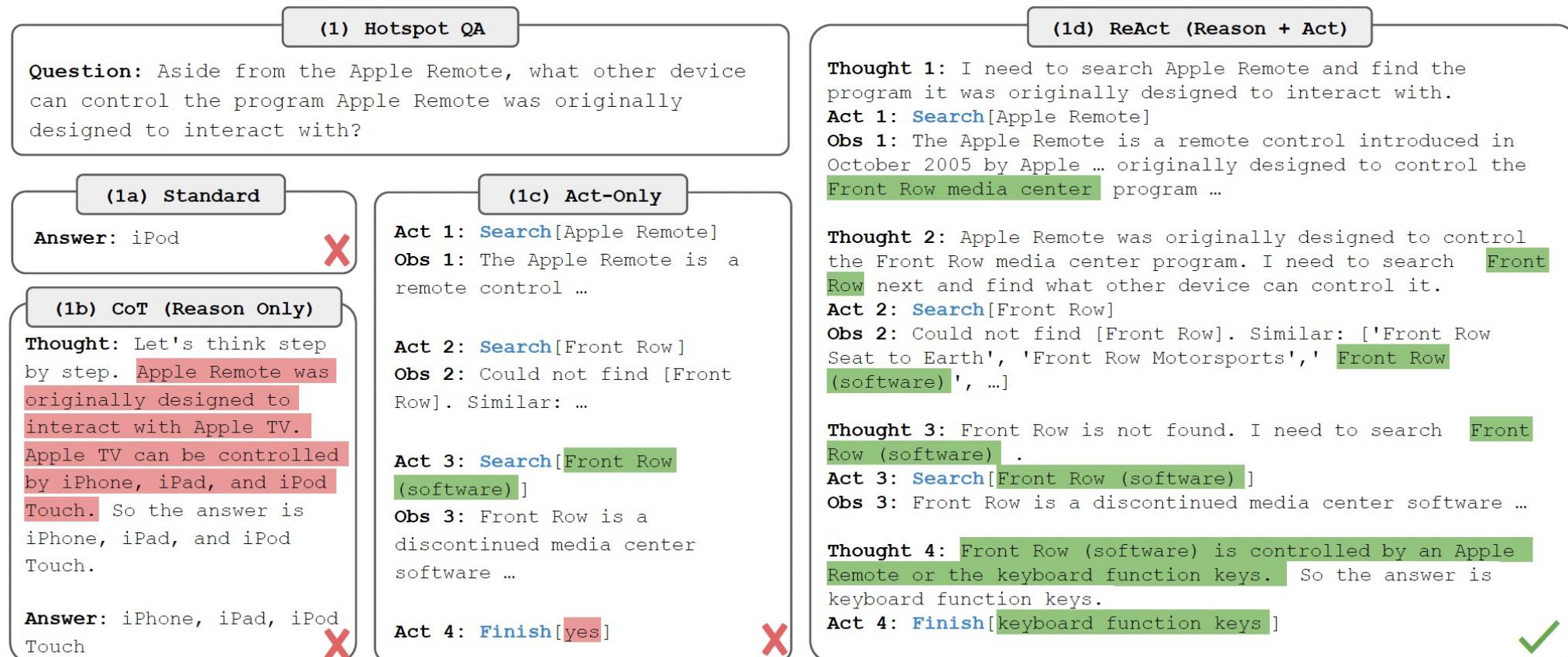
# Planning without Feedback

- Graph of Thought



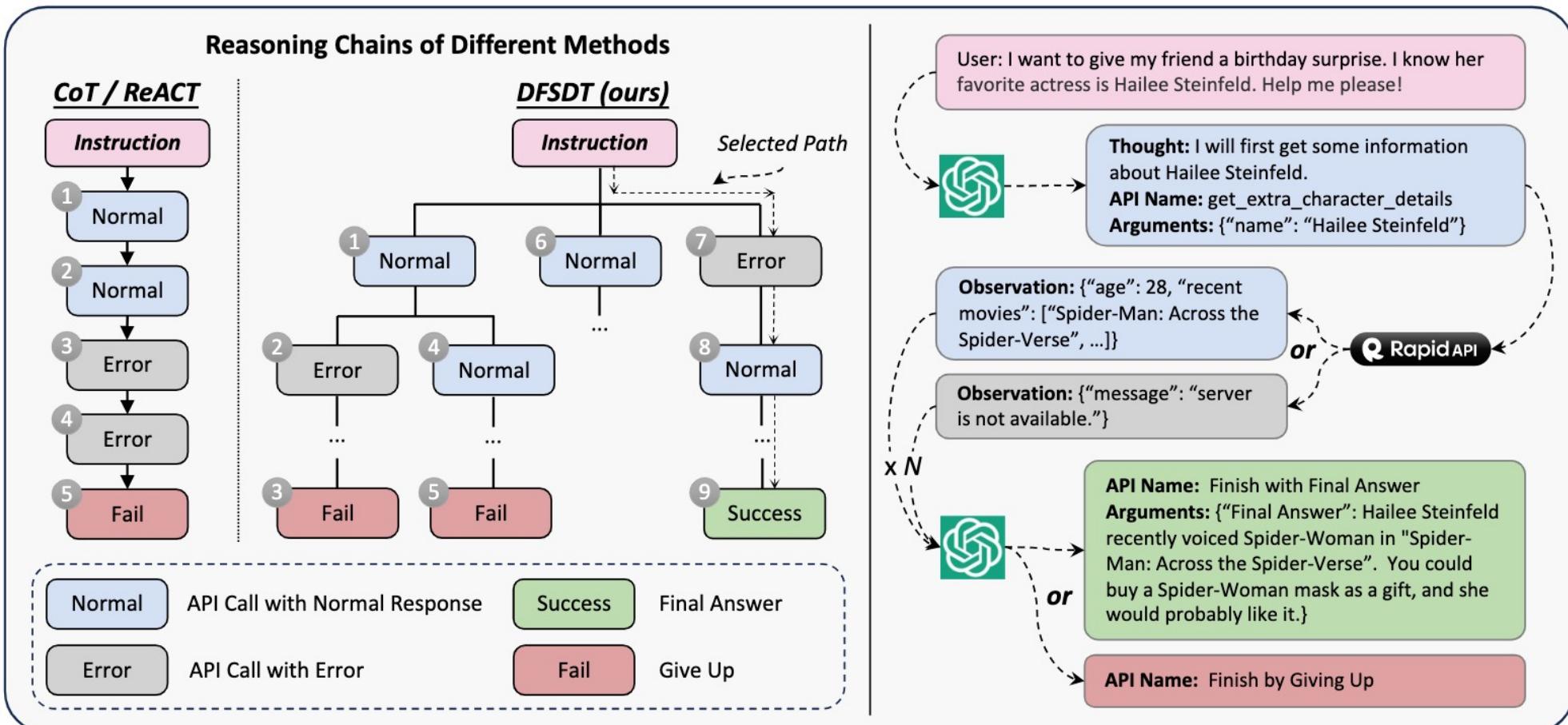
# Planning with Feedback

- ReAct



# Planning with Feedback

- DFSDT



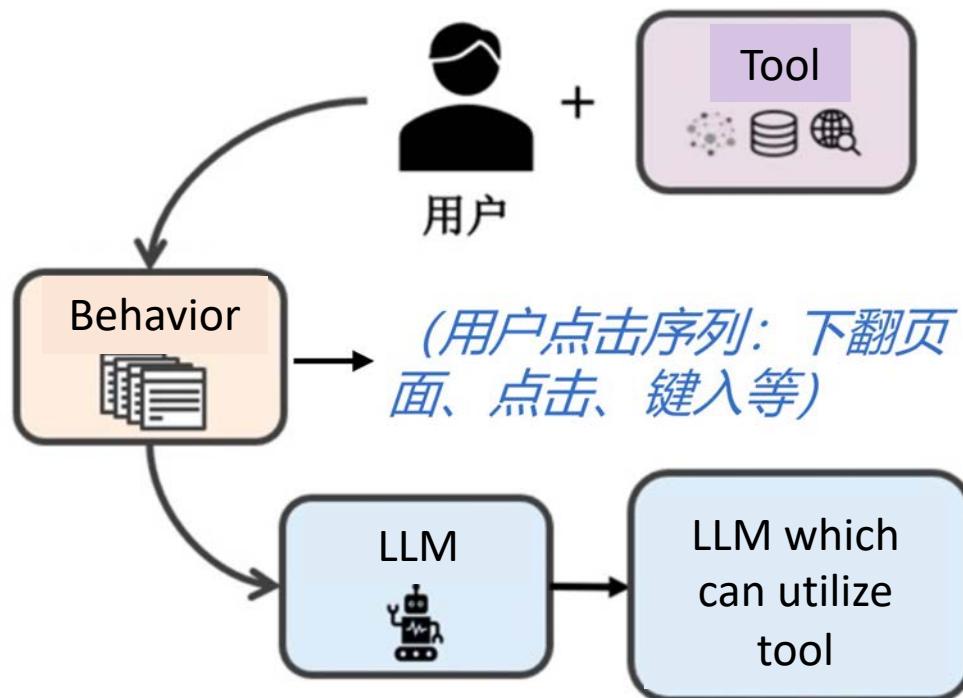


# Tool Use

ByteDance

# Learning to Use Tool

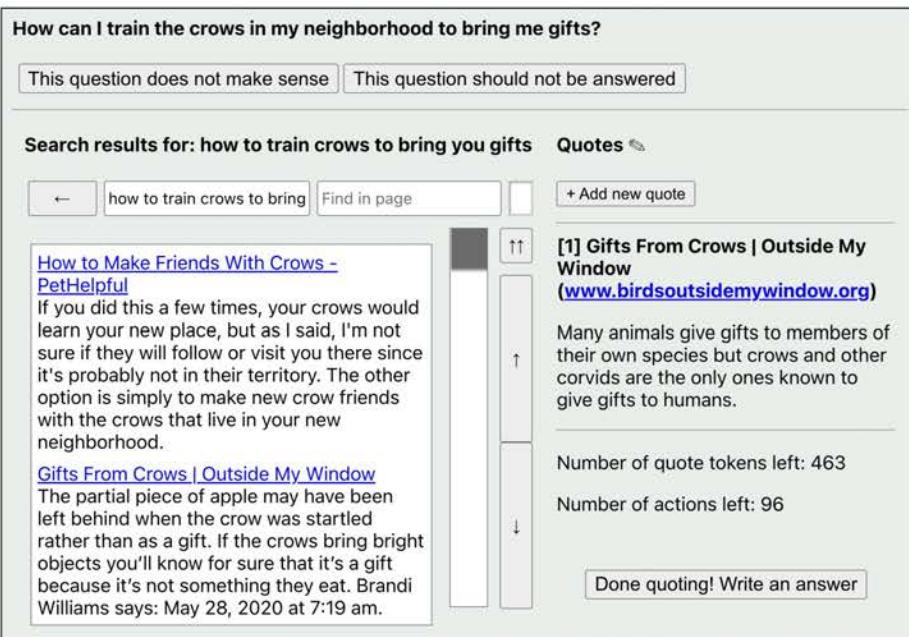
- Imitation Learning
  - By recording data on human tool usage behaviors, large models mimic human actions to learn about tools
  - The simplest and most direct method of tool learning.



# WebGPT

- Supervised Learning  OpenAI

- Clone human behavior to use search engines
- Supervised fine-tuning + reinforcement learning
- Only need 6,000 annotated data

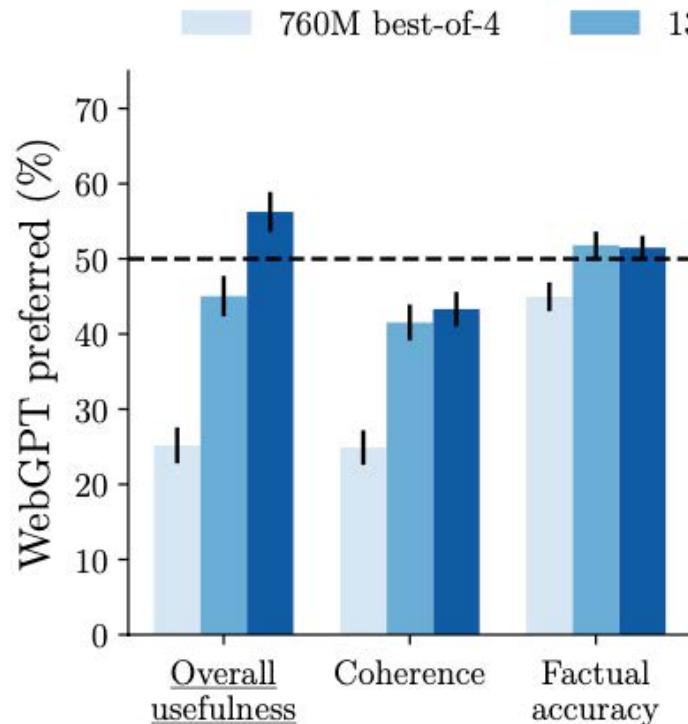


The screenshot shows the WebGPT interface. A user has asked the question: "How can I train the crows in my neighborhood to bring me gifts?". Below the question are two buttons: "This question does not make sense" and "This question should not be answered". The main area displays search results for "how to train crows to bring you gifts". It includes a "Search results for: how to train crows to bring you gifts" section with a back button, a search bar containing "how to train crows to bring", a "Find in page" button, and a "Quotes" section with a "+ Add new quote" button. The quotes section lists "[1] Gifts From Crows | Outside My Window (www.birdsoutsidemywindow.org)" and a summary: "Many animals give gifts to members of their own species but crows and other corvids are the only ones known to give gifts to humans.". At the bottom of the quotes section are "Number of quote tokens left: 463" and "Number of actions left: 96". A "Done quoting! Write an answer" button is at the very bottom. To the right of the interface, there is a detailed log of the interaction:

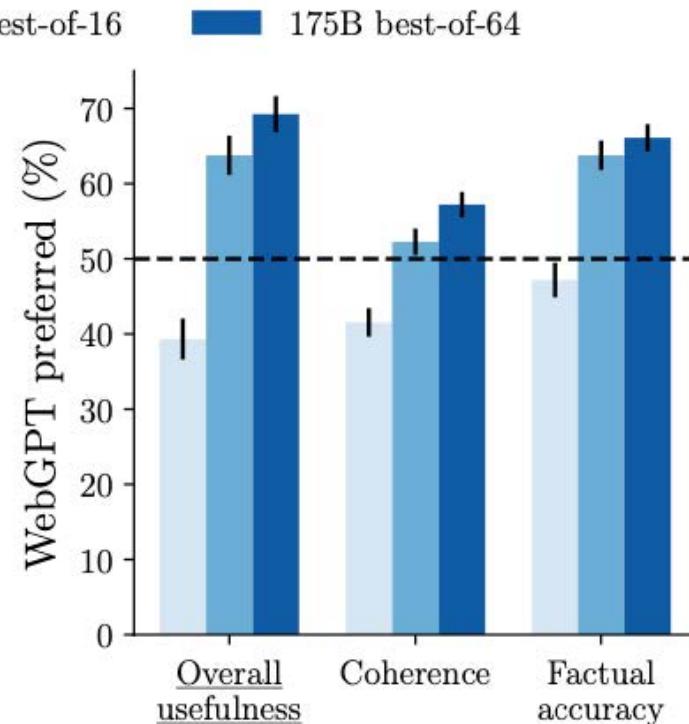
- ♦Question  
How can I train the crows in my neighborhood to bring me gifts?
- ♦Quotes  
From Gifts From Crows | Outside My Window (www.birdsoutsidemywindow.org)  
> Many animals give gifts to members of their own species but crows and other corvids are the only ones known to give gifts to humans.
- ♦Past actions  
Search how to train crows to bring you gifts  
Click Gifts From Crows | Outside My Window www.birdsoutsidemywindow.org  
Quote  
Back
- ♦Title  
Search results for: how to train crows to bring you gifts
- ♦Scrollbar: 0 - 11
- ♦Text  
!0†How to Make Friends With Crows - PetHelpful[pethelpful.com]  
If you did this a few times, your crows would learn your new place, but as I said, I'm not sure if they will follow or visit you there since it's probably not in their territory. The other option is simply to make new crow friends with the crows that live in your new neighborhood.  
Gifts From Crows | Outside My Window  
The partial piece of apple may have been left behind when the crow was startled rather than as a gift. If the crows bring bright objects you'll know for sure that it's a gift because it's not something they eat. Brandi Williams says: May 28, 2020 at 7:19 am.
- ♦Actions left: 96
- ♦Next action

# WebGPT

- Supervised Learning  OpenAI
  - Excellent performance in long-form QA, even surpassing human experts



(a) WebGPT vs. human demonstrations.



(b) WebGPT vs. ELI5 reference answers.

# LLM+Search Engine: WebCPM

- WebCPM: The **first** open-source Q&A framework based on interactive web search for the Chinese domain.
- Constructed the **first** Chinese web search LFQA dataset with 5,500 entries. After fine-tuning on this dataset, large models can attain search engine capabilities.

**Question** | 麦田怪圈是什么？它们是如何形成的？ *What are crop circles? How are they made?*

**Query** | 麦田怪圈如何形成？ *How do crop circles form?*

**Window (search mode)**

↑ ↓ 2/9

难解谜团：麦田怪圈究竟是如何形成的？ *Unsolved mysteries: How did crop circles form?*

麦田怪圈出现最多的季节是在春天和夏天，有人认为，夏季天气变化无常，龙卷风是造成怪圈的主要原因... *The crop circles appear most often in spring and summer. Some people think that the weather in summer is erratic, and tornadoes are the main cause of the strange circles...*

**Title of page <2>** *A snapshot of the page content*

**Title of page <3>** *A snapshot of the page content*

**Undo** **Reset**

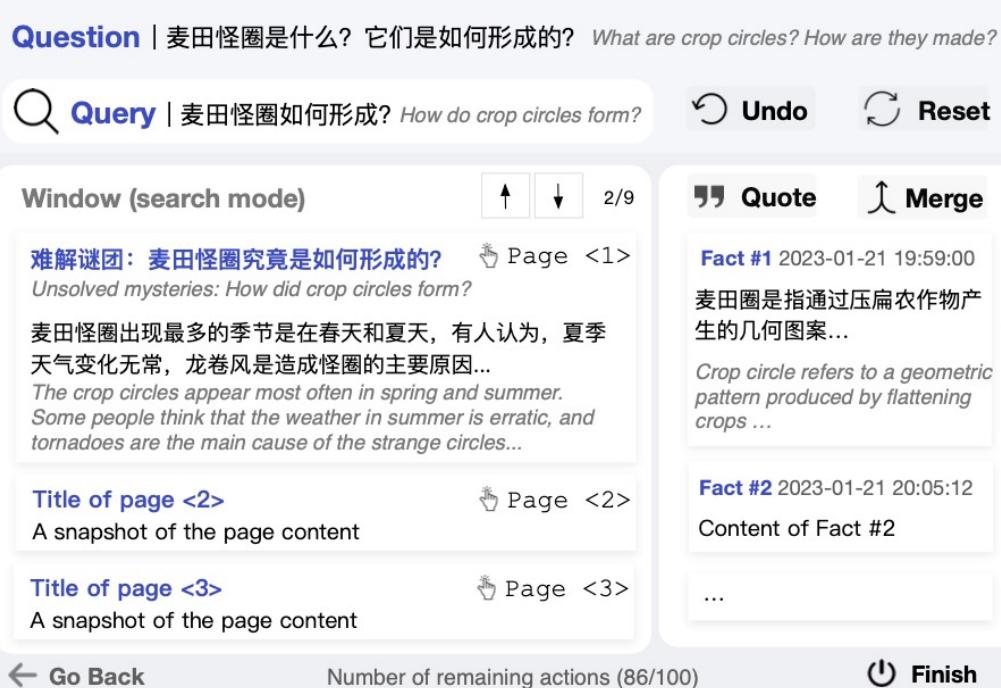
**Quote** **Merge**

**Fact #1** 2023-01-21 19:59:00  
麦田圈是指通过压扁农作物产生的几何图案...  
*Crop circle refers to a geometric pattern produced by flattening crops ...*

**Fact #2** 2023-01-21 20:05:12  
Content of Fact #2

...

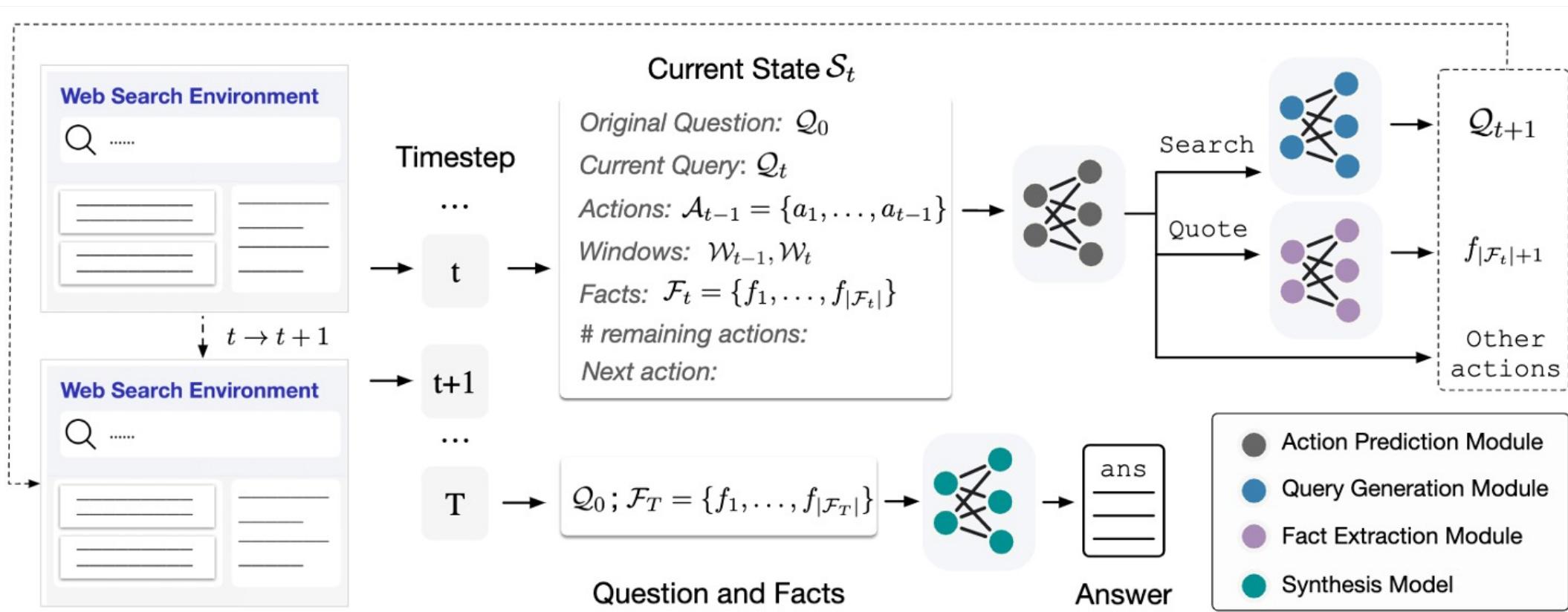
← Go Back Number of remaining actions (86/100) **Finish**



Action Name	Functionality
Q Search <query>	Call Bing search with <query>
← Go Back	Return to the previous window
Load Page <1>	Load the details of page <1>
Load Page <2>	Load the details of page <2>
Load Page <3>	Load the details of page <3>
↑ Scroll Up	Scroll up for a pre-set stride
↓ Scroll Down	Scroll down for a pre-set stride
” Quote <content>	Extract <content> from the current page as a supporting fact
↑ Merge	Merge two facts into a single fact
↻ Finish	End the search process

# LLM+Search Engine: WebCPM

- Given a query, the large model autonomously decides each step (scroll up, scroll down, search, click pages, etc.) until it gathers sufficient information to answer. The process is fully automated, no user intervention required.



# LLM+Search Engine: WebCPM

- WebCPM can fully replace user-engaged search engine usage in **30%+** cases
- The model learns human-like search strategies from human behavioral data

## Question:

麦田怪圈是什么？它们是如何形成的？

## Action Sequence:

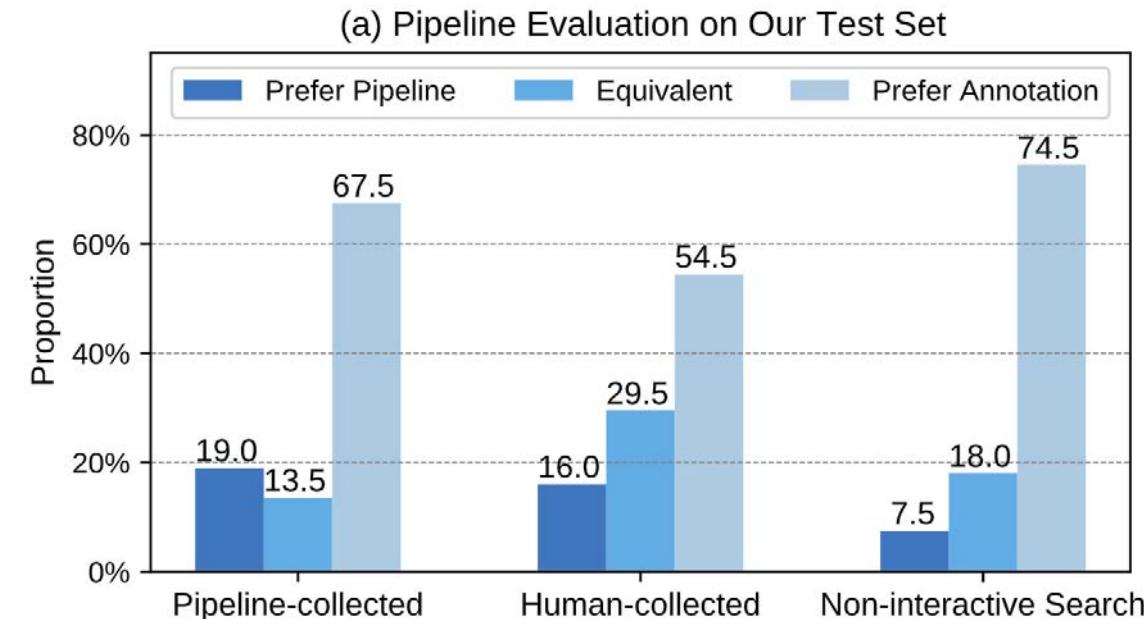
Search → Load Page <1> → Scroll Down × 5 → Scroll Up → Scroll Down × 11 → Go Back → Search → Load Page <1> → Go Back → Load Page <3> → Scroll Down × 4 → Scroll Up × 3 → Quote → Scroll Down → Quote → Scroll Down → Quote → Finish

## Supporting Facts:

1. 麦田怪圈（Crop Circle），是指在麦田或其它田地上，通过某种未知力量（大多数怪圈是人类所为）把农作物压平而产生出来的几何图案。这个神秘现象有时被人们称之为“Crop Formation”。麦田怪圈的出现给了对支持外星人存在论的人们多种看法。
2. 人为说：人为说一般认为，麦田圈是用木板压成的。木板两头系上绳子形成圈套，在制作时，一脚踩在木板上拖动木板压倒麦子，并拉着细绳与圆心保持固定的距离，逐渐就可以形成一个圆圈。为了便于制造，主要形状所有圆圈的直径都可以被6除尽。以前曾经出现过制作麦田圈被当场抓获的事情，制作者使用的就是这种工具。
3. 自然形成说：也有人认为，麦田圈只是一种，成因还未被人类发现。就像雷电，古时候人类也是以为是雷神电母做的，对于麦田圈中经常出现人文信息的现象，他们认为这只是人们“先入为主”造成的错觉。
4. 磁场说：有专家认为，磁场中有一种神奇的移动力，可产生一股电流，使农作物“平躺”在地面上。美国专家杰弗里·威尔逊研究了130多个麦田怪圈，发现90%的怪圈附近都有连接高压电线的变压器，方圆270米内都有一个水池。由于接受灌溉，麦田底部的土壤释放出的离子会产生负电，与高压电线相连的变压器则产生正电，负电和正电碰撞后会产生电磁能，从而击倒小麦形成怪圈。

## Answer:

麦田怪圈是指在麦田或其它田地上，通过某种未知力量（大多数怪圈是人类所为）把农作物压平而产生出来的几何图案。【1】而麦田怪圈的形成有三个说法，人为说、自然说与磁场说。人为说一般认为，麦田圈是用木板压成的。木板两头系上绳子形成圈套，在制作时，一脚踩在木板上拖动木板压倒麦子，并拉着细绳与圆心保持固定的距离，逐渐就可以形成一个圆圈。为了便于制造，主要形状所有圆圈的直径都可以被6除尽。【2】自然形成说认为麦田怪圈的成因还未被人类发现。就像雷电，古时候人类也是以为是雷神电母做的，对于麦田圈中经常出现人文信息的现象，他们认为这只是人们“先入为主”造成的错觉。【3】磁场说认为，磁场中有一种神奇的移动力，可产生一股电流，使农作物“平躺”在地面上。【4】



(1) pipeline-collected, (2) human-collected,  
(3) non-interactive search (TF-IDF)

# WebShop

- Learning to perform online shopping

**A**

**WebShop** **search**

**Instruction:**  
I'm looking for a small portable folding desk that is already fully assembled; it should have a khaki wood finish, and price lower than 140.00 dollars

portable folding desk khaki wood **1** **Search**

**Back to Search** **results** **2**

Page 1 (Total results: 50) **Next >**

  
B09Q3B186B  
MENHG Folding Breakfast Tray Table, Efficient Home Laptop Notebook Computer Desk, Portable Writing Study Desk, Sturdy Home Office Table Workstation \$109.0

  
B09P5ZBWR  
KPSL Folding Study Desk Bed Breakfast Serving Tray Table Efficient Home Laptop Notebook Computer Desk Portable Standing Desk for Small Space Bedroom

Description: Product laptop desk. Product weight: walnut. Product weight: 4.6 pounds. Material: high quality thick steel pipe, black brushed sheet. Special design: black brushed smooth table top, increase the length and width of the table, it is possible to place the computer and various items. Function: Can be used as computer desk, dining table, bedside table. Product size: 23.6x15.7x11 inches

**item-detail**

- Large Size: styling with light wood.
- Wide Application: Holds laptops up to 17 inches. It also have spacious space (23.6x15.7x11 inches) for your laptop, notebook, mouse, pen and coffee. Its generous size gives this versatile desk even more flexibility.
- Portable lap desk can be used as a

MENHG Folding Laptop Table Bed Desk **item**  
PC Lap Desk with Drawer Book Stand Reading Holder Leg Space Laptop Bed Tray Foldable Lazy Table Breakfast Desk Sofa Small Desk for Small Space

Price: \$100.0 Rating: N.A.

**Description** **Overview** **Buy Now** **5**

**Reward:** 1.0

**Color**: black, khaki, white **3**

**HTML mode**

**B**

**Simple mode**

**Instruction:**  
I'm looking for a small portable folding desk that is already fully assembled [...]

**[btn] Back to Search [/btn]**  
**[btn] Next [/btn]**  
**[btn] MENHG Folding Breakfast Tray [...] [/btn]**  
**\$109.0**  
**[btn] KPSL Folding Study Desk Bed [...] [/btn]**

**C**

$\bar{y}$  (Instruction): I'm looking for a small portable...

$\bar{y}$  (Description): MENHG Folding Laptop Table Bed...

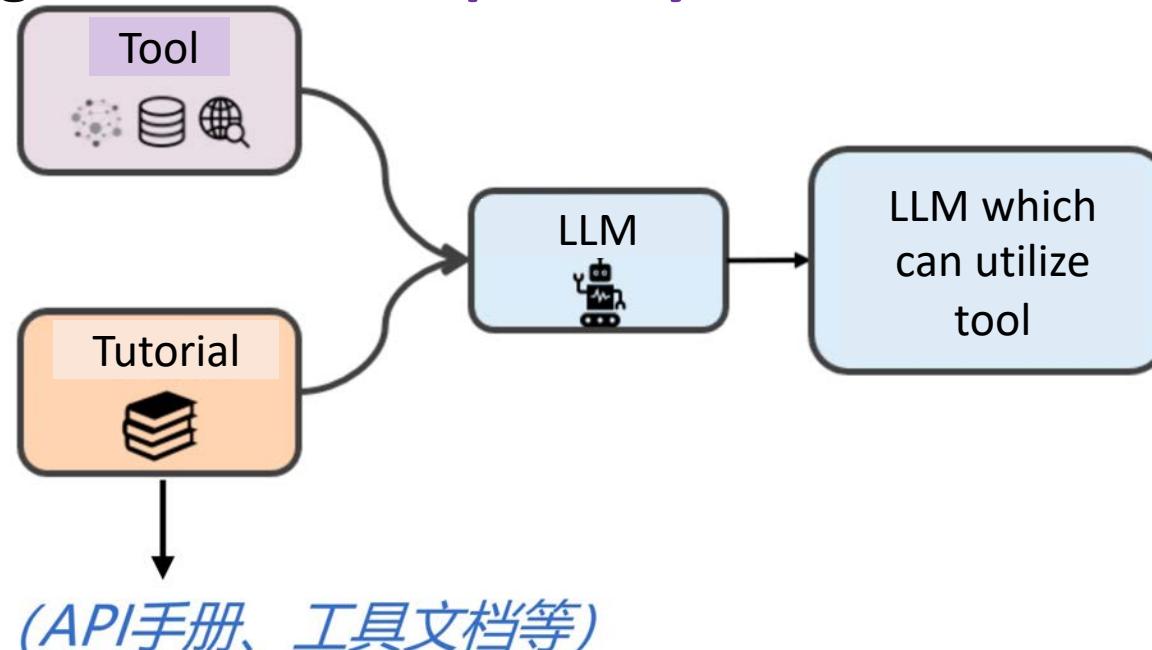
$y_{price}$ : \$109.0

$Y_{opt}$  (Options): { black, khaki, white }

$Y_{att}$  (Attributes): { steel pipe, no assembly, portable }

# Learning to Use Tool

- Tutorial Learning
  - By having the model read tool manuals (tutorials), it understands the functions of the tools and how to invoke them
  - Almost exclusively, large models from the OpenAI series (such as ChatGPT, GPT-4) possess a high **zero-shot capability** to understand tool manuals.



# How to Understand Tools?

- Eliciting tool understanding with prompting
- Zero-shot prompting:
  - Describe API functionalities, their input/output formats, possible parameters, etc.
  - Allow the model to understand the tasks that each API can tackle
- Few-shot prompting:
  - Provide concrete tool-use demonstrations to the model
  - By mimicking human behaviors from these demonstrations, the model can learn how to utilize these tools

# Learning to Use Tool

- Describe the functionality;

In-context with example(s).

**Zero-shot Prompting:** Here we provide a tool (API) "forecast\_weather(city:str, N:int)", which could forecast the weather about a city on a specific date (after N days from today). The returned information covers "temperature", "wind", and "precipitation".

Please write codes using this tool to answer the following question: "What's the average temperature in Beijing next week?"

---

**Few-shot Prompting:** We provide some examples for using a tool. Here is a tool for you to answer question:

Question: "What's the temperature in Shanghai tomorrow?"

```
return forecast_weather("Shanghai", 1) ["temperature"]
```

Question: "Will it rain in London in next two days?"

```
for i in range(2):
    if forecast_weather("London", i+1) ["precipitation"] > 0:
        return True
return False
```

Question: "What's the average temperature in San Francisco next week?"

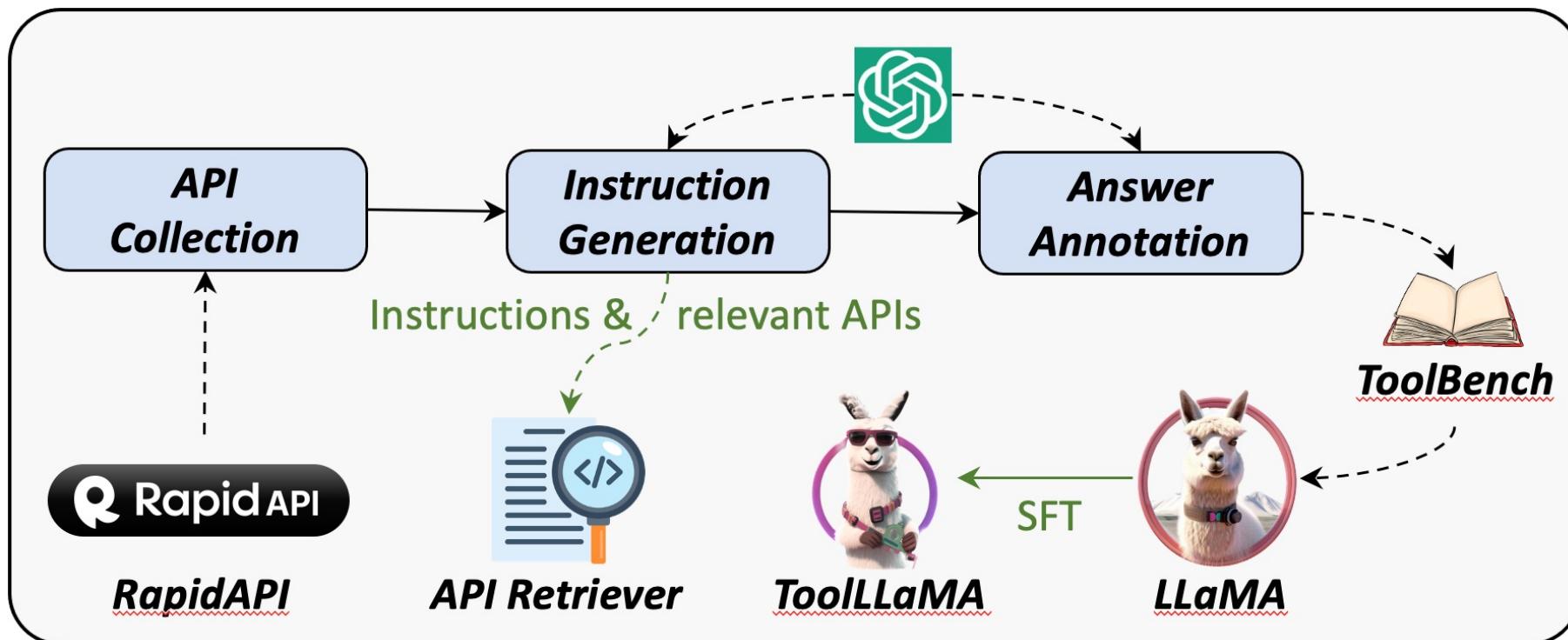
# ToolBench

- Highlights:
  - Over 16,000 real APIs (collected from RapidAPI)
  - Supports single and multi-tool invocation
  - Complex multi-step reasoning tasks

Resource	ToolBench (this work)	APIBench (Patil et al., 2023)	API-Bank (Li et al., 2023a)	ToolAlpaca (Tang et al., 2023)	T-Bench (Xu et al., 2023b)
Real-world API?	✓	✗	✓	✗	✓
Real API Response?	✓	✗	✓	✗	✓
Multi-tool Scenario?	✓	✗	✗	✗	✗
API Retrieval?	✓	✓	✗	✗	✗
Multi-step Reasoning?	✓	✗	✓	✓	✓
Number of tools	<b>3451</b>	3	53	400	8
Number of APIs	<b>16464</b>	1645	53	400	232
Number of Instances	12657	<b>17002</b>	274	3938	2746
Number of Real API Calls	<b>37204</b>	0	568	0	0
Avg. Reasoning Traces	4.1	1.0	2.1	1.0	<b>5.9</b>

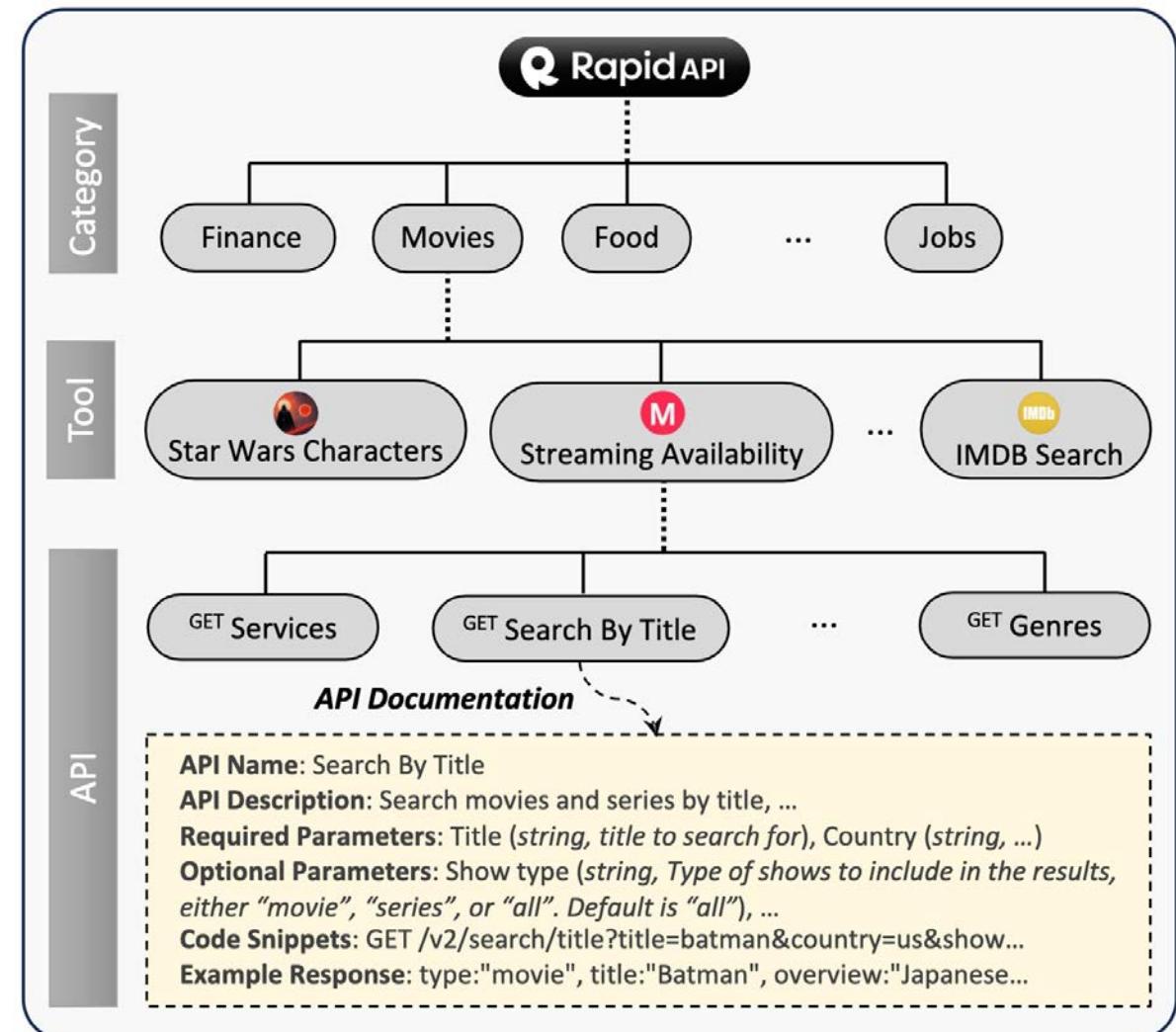
# ToolBench Construction

- API Collection
- Instruction Generation
- Answer Annotation



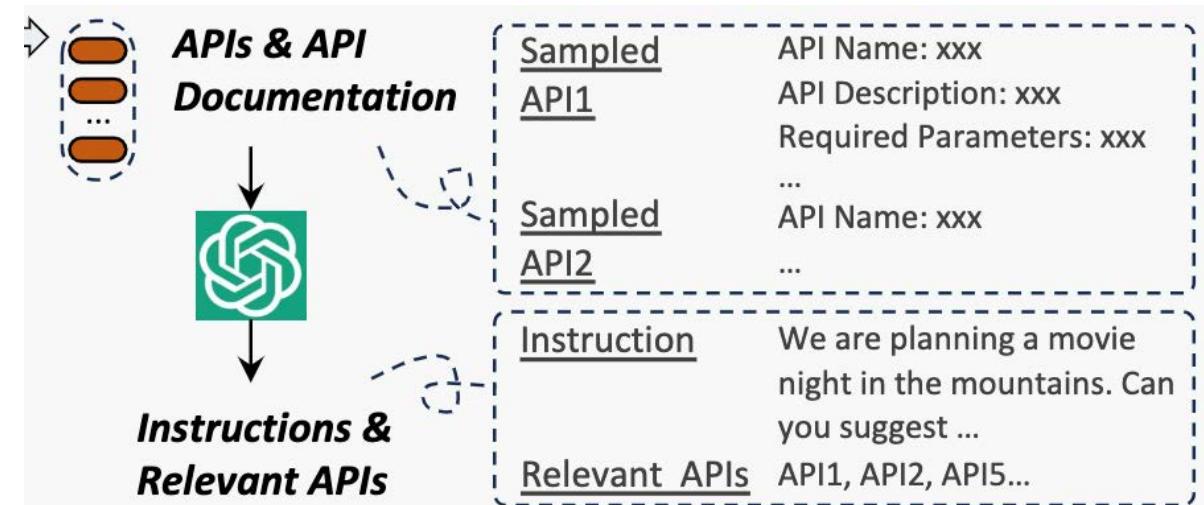
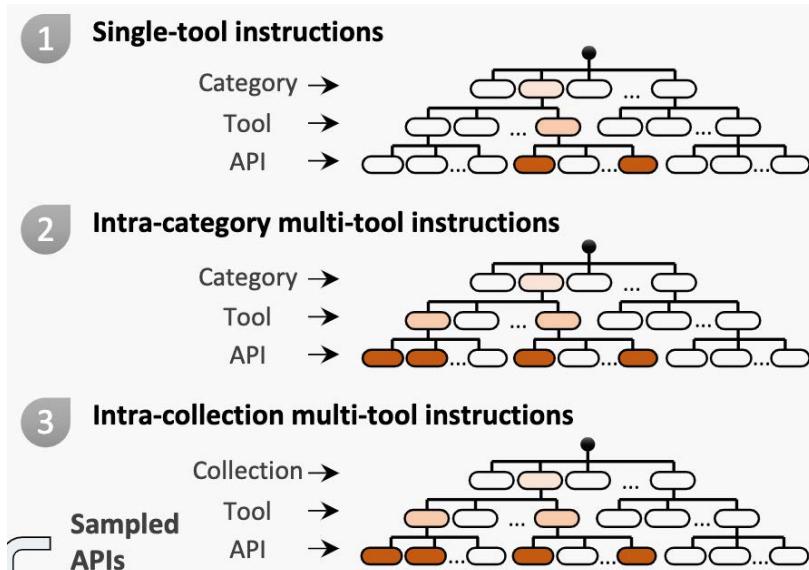
# ToolBench Construction

- API Collection
  - RapidAPI Hub:  
<https://rapidapi.com/hub>
  - Filter over 16,000 high-quality APIs from more than 50,000 APIs
  - Include 49 categories



# ToolBench Construction

- Instruction Generation
  - Single Tool + Multi-Tool
  - (1) Sample a collection of APIs:  $\mathbb{S}_N^{\text{sub}} = \{\text{API}_1, \dots, \text{API}_N\}$
  - (2) ChatGPT automatically generate instructions that may require calling one or more APIs in the collection:  $\text{ChatGPT} \quad (\{[\mathbb{S}_1^{\text{rel}}, \text{Inst}_1], \dots, [\mathbb{S}_{N'}^{\text{rel}}, \text{Inst}_{N'}]\} | \text{API}_1, \dots, \text{API}_N, \text{seed}_1, \dots, \text{seed}_3).$   
 $\{\text{API}_1, \dots, \text{API}_N\} \in \mathbb{S}_{\text{API}},$   
 $\{\text{seed}_1, \dots, \text{seed}_3\} \in \mathbb{S}_{\text{seed}}$



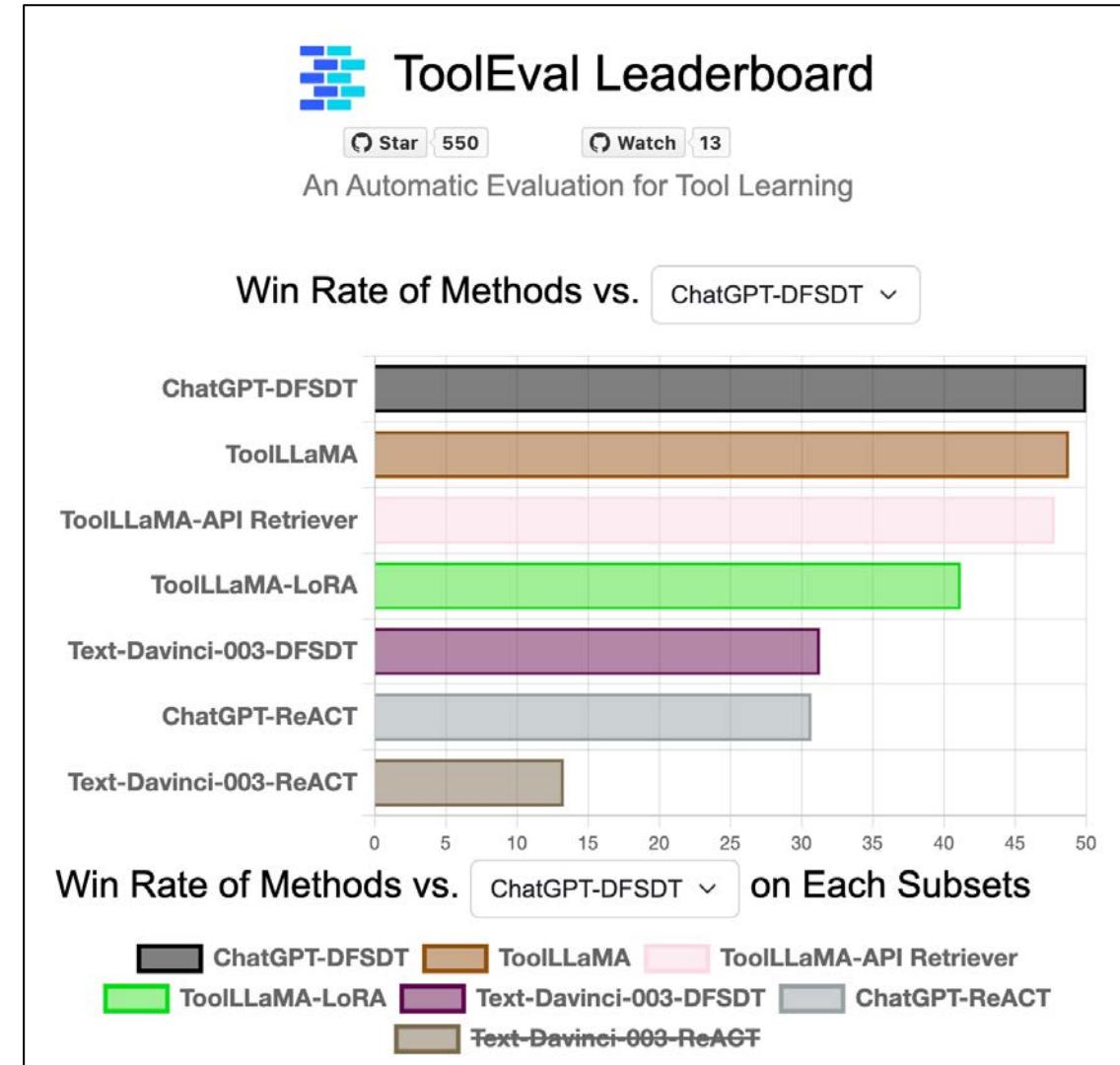
# ToolBench Construction

- Answer Annotation
  - gpt-3.5-turbo-16k: feature of function call
- Issues with ReACT
  - Error Propagation: An error in a single step annotation can render the entire action sequence unusable
  - Limited Exploration: ReACT can only sample one sequence from the infinite action sequence space based on the LM's probabilities
- DFSDT: Dynamically extends the TOT to the tool learning scenario

Method	Single-tool (I1)	Category (I2)	Collection (I3)	Average
ReACT	43.98	23.62	20.42	29.34
ReACT@N	50.80	36.14	32.87	39.94
DFSDT	<b>54.10</b>	<b>47.35</b>	<b>44.80</b>	<b>48.75</b>

# ToolEval

- Automatic Evaluation Framework Based on ChatGPT
- Two metrics:
  - Success rate: The proportion of commands successfully completed within a limited number of API calls
  - Preference: Comparison of quality/usefulness between two answers, i.e., which one is better?
- Highly consistent with human experts (~80%).



# ToolLLaMA

- Demonstrate exceptionally high generalizability to OOD commands and APIs, significantly outperforming ChatGPT+ReACT

Model	I1-Inst.		I1-Tool		I1-Cat.		I2-Inst.		I2-Cat.		I3-Inst.		Average	
	Pass	Win												
ChatGPT-ReACT	56.0	-	62.0	-	66.0	-	28.0	-	22.0	-	30.0	-	44.0	-
Vicuna (ReACT & DFSDT)	0.0	-	0.0	-	0.0	-	0.0	-	0.0	-	0.0	-	0.0	-
Alpaca (ReACT & DFSDT)	0.0	-	0.0	-	0.0	-	0.0	-	0.0	-	0.0	-	0.0	-
Text-Davinci-003-DFSDT	53.0	46.0	58.0	38.0	61.0	39.0	38.0	46.0	38.0	45.0	39.0	48.0	47.8	43.7
ChatGPT-DFSDT	<b>78.0</b>	<b>68.0</b>	<b>84.0</b>	<b>59.0</b>	<b>89.0</b>	<b>57.0</b>	<b>51.0</b>	<b>78.0</b>	<b>58.0</b>	<b>77.0</b>	<b>57.0</b>	<b>77.0</b>	<b>69.6</b>	<b>69.3</b>
ToolLLaMA-DFSDT	<u>68.0</u>	<u>68.0</u>	<u>80.0</u>	<u>59.0</u>	<u>75.0</u>	<u>56.0</u>	<u>47.0</u>	<u>75.0</u>	<u>56.0</u>	<u>80.0</u>	<u>40.0</u>	<u>72.0</u>	<u>61.0</u>	<u>68.3</u>

- DFSDT >> ReACT

Method	Single-tool (I1)	Category (I2)	Collection (I3)	Average
ReACT	43.98	23.62	20.42	29.34
ReACT@N	50.80	36.14	32.87	39.94
DFSDT	<b>54.10</b>	<b>47.35</b>	<b>44.80</b>	<b>48.75</b>

# Example

The screenshot displays the ToolLLaMA UI interface. On the left side, there is a sidebar with the following options:

- + New chat
- Search...
- New Conversation
- I'm planning a surprise party ... (repeated multiple times)
- New Conversation
- I'm planning a surprise party ...
- I'm planning a surprise party ...
- What is the weather in new Y...
- I'm planning a surprise party ...

At the bottom of the sidebar are the following options:

- Clear conversations
- Import data
- Export data
- Settings

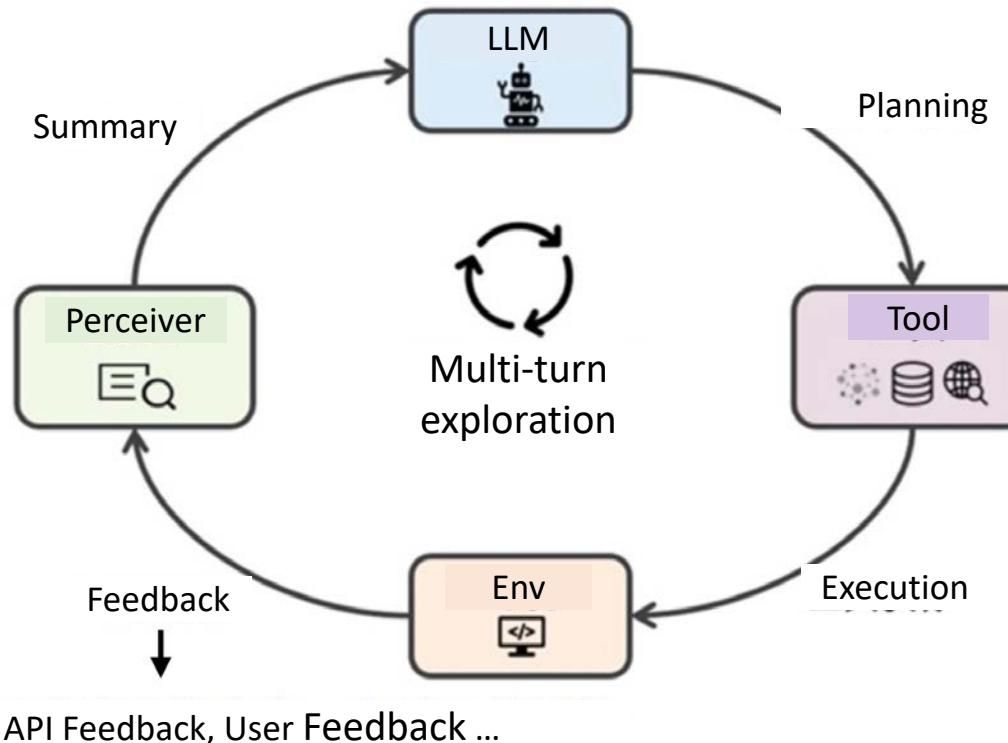
The main area of the interface is titled "ToolLLaMA UI". It features a "Model" dropdown menu and a "System Prompt" input field. The "System Prompt" field contains the placeholder text: "Enter a prompt or type "/" to select a prompt...". In the bottom right corner, there is a message card with the following content:

⚡ I'm planning a surprise party for my best friend, and I want to include meaningful quotes in the decorations. Can you provide me with random love, success, and motivation quotes? It would be great to have quotes that can celebrate love, success, and inspire everyone at the party. Thank you so much for your help! ⚡

At the very bottom of the interface, a footer note states: "ToolLLaMA UI is an interface based off of ChatBot UI. We appreciate the authors for their work."

# Learning to Use Tool

- Reinforcement Learning
  - Capable of autonomous exploration and corrects errors based on environmental feedback through reinforcement learning
- There is limited existing research on this topic.



# Learning to Use Tool

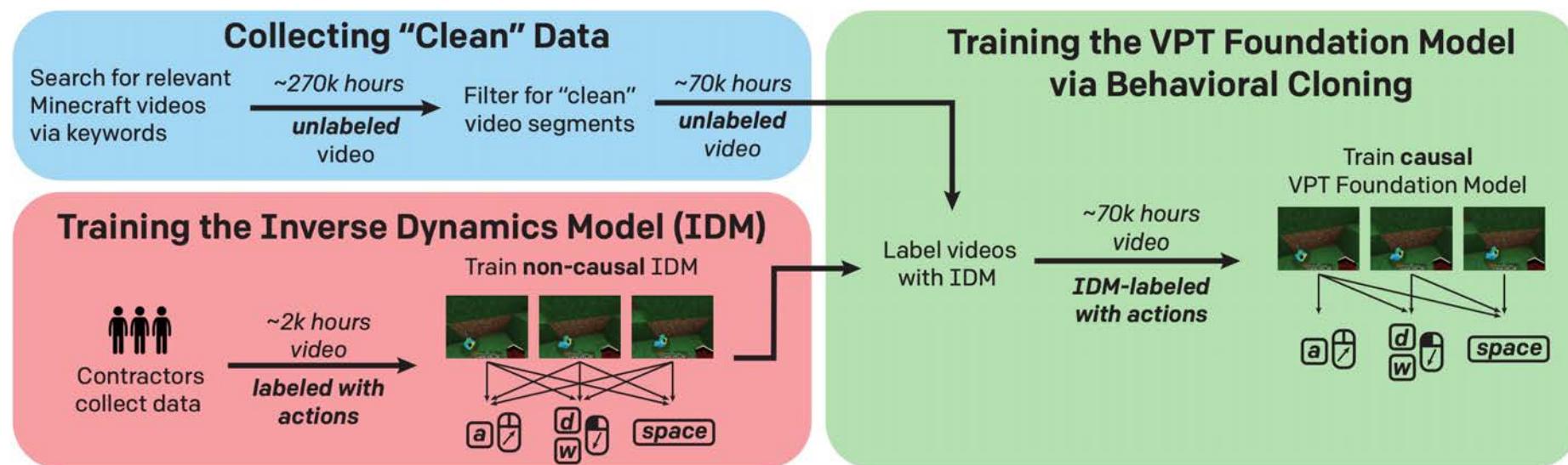
- **Learning from feedback:** often involves reinforcement learning

$$\theta_C^* = \arg \max_{\theta_C} \mathbb{E}_{q_i \in Q} \mathbb{E}_{\{a_{i,t}\}_{t=0}^{T_i} \in p_{\theta_C}} \left[ R(\{a_{i,t}\}_{t=0}^{T_i}) \right],$$

- Reinforcement Learning (RL) for Tool Use
  - Action space is defined based on tools
  - Agent learns to select the appropriate tool
  - Perform the correct actions that maximize the reward signal

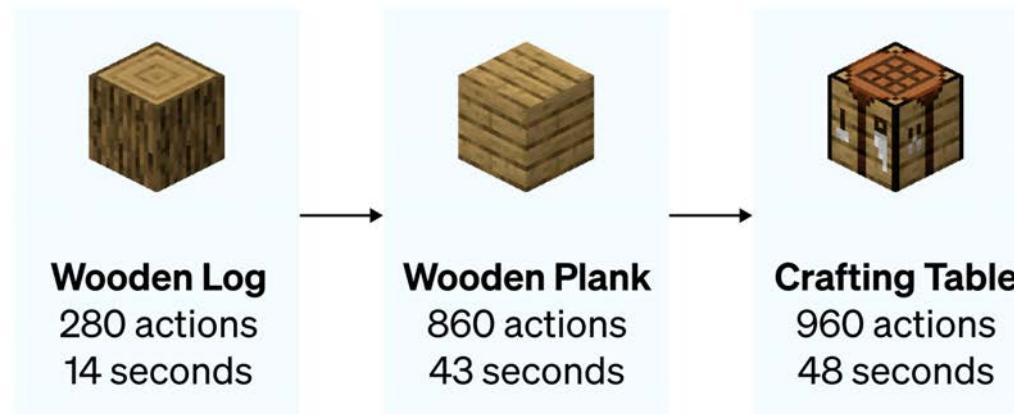
# Video Pre-training

- Semi-supervised Learning
  - Predict pseudo-labels of the action taken at each time-step in a Minecraft video game
  - IDM: minimize  $p_{\text{IDM}}(a_t | o_1 \dots T)$



# Video Pre-training

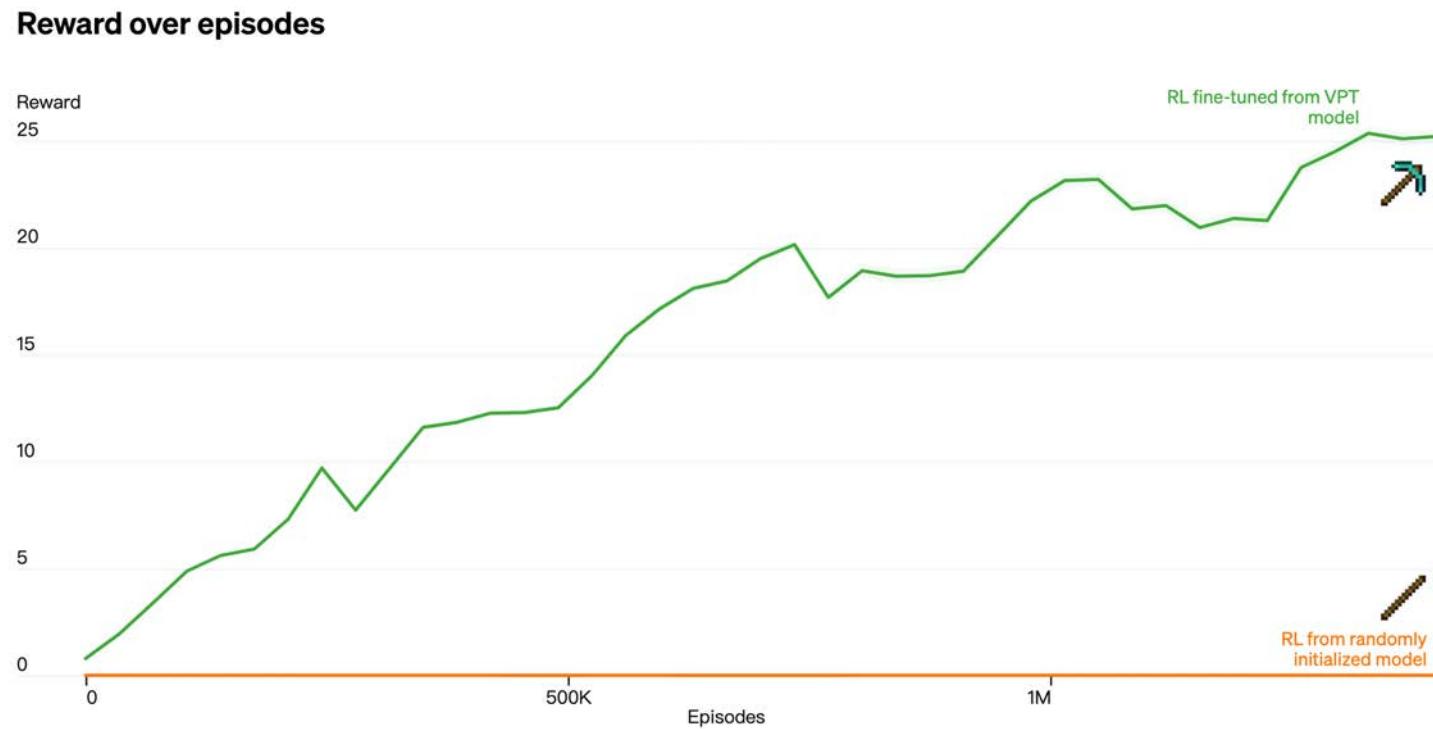
- After Training
  - The model can do things that **pure RL cannot do**
  - e.g., make a tab  $p_{\text{IDM}}(a_t | o_1 \dots T)$  collect logs, craft plank...
  - Model performs professional actions like humans
- e.g., swimming, hunting animals for food



Sequence of items required to craft a crafting table, labeled with the median time it takes proficient humans to reach each step

# Video Pre-training

- Fine-tuning with Reinforcement Learning
  - Goal: collecting a diamond pickaxe
  - Normal RL cannot work; Start from VPT, it learns fast

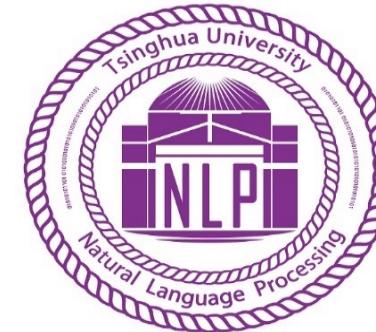


# Toolformer

- Self-supervised Tool Learning
  - Pre-defined tool APIs
  - Encourage models to call and execute tool APIs
  - Design self-supervised loss to see if the tool execution can help language modeling



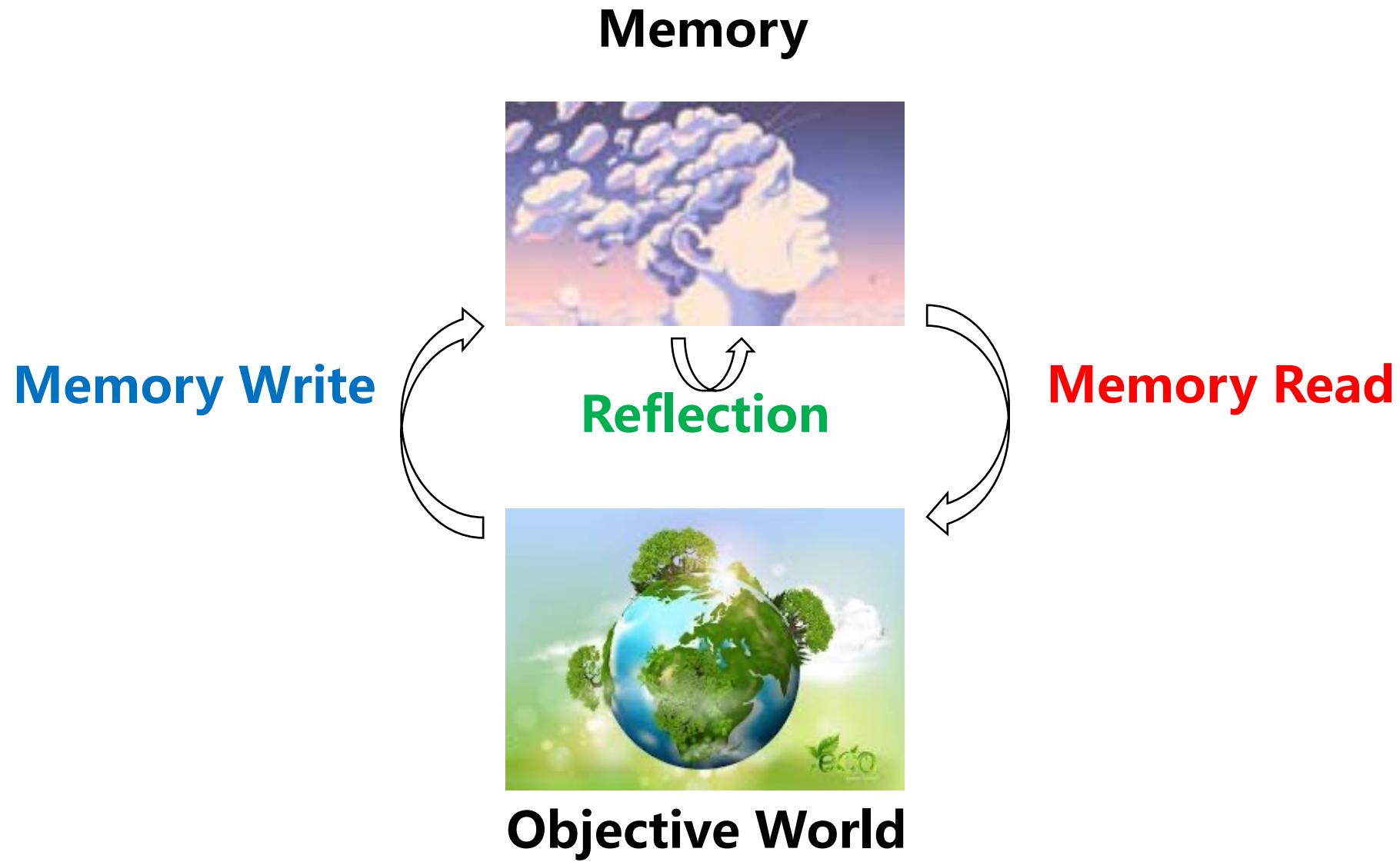
If the tool execution reduces LM loss,  
save the instances as training data



# Memory

ByteDance

# Memory Framework



# Memory Structure

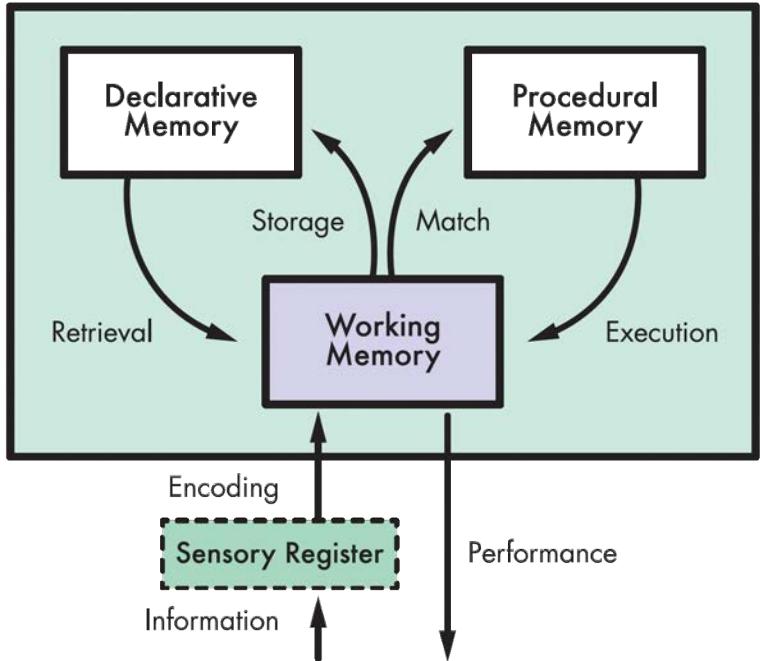
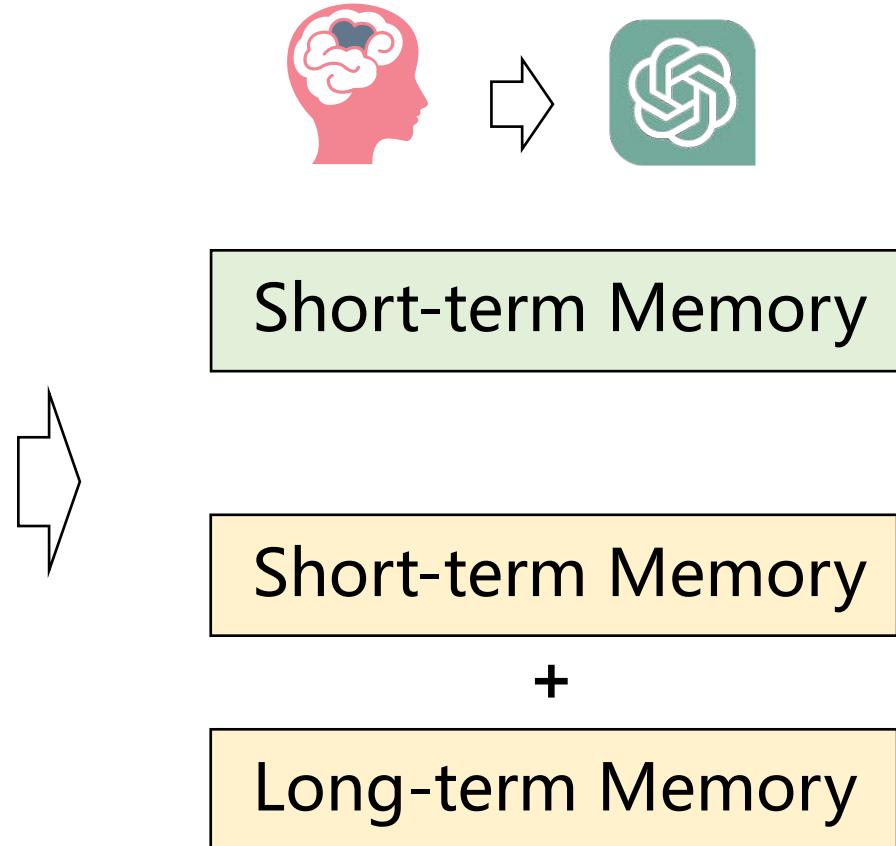


Figure 1: Schematic illustration of ACT-R. In our work, we focus on the activation equation of the declarative memory module.

Human Memory Mechanism: ACT-R



Agent Memory Structure

# Short-term Memory

- Short-term memory is usually realized by **in-context learning**, and the memory information is directly written **into the prompts**

No external memory storage

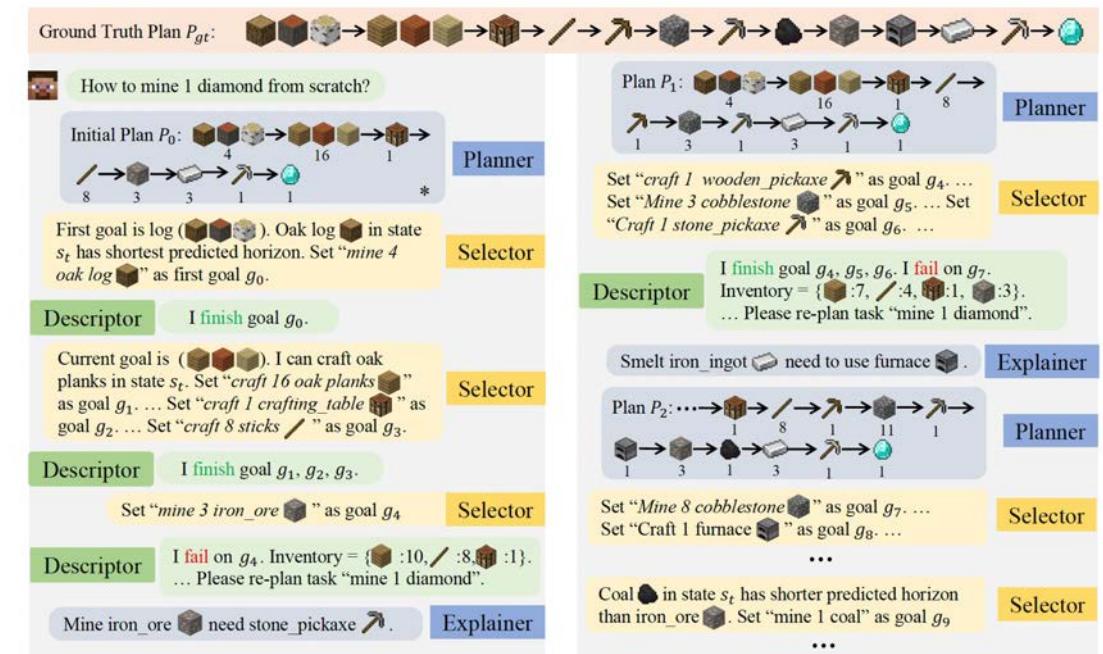
```
# RLP.gpt4

Initialize {
    My personality is [[PERSONALITY]]
}

Think {
    This last message made me feel ...
    My previous plan was ...
    I think ...
    I will send the message, " ..."
    In retrospect ...
    My next plan is ...

    constraints {
        Output format in squiggly brackets separated by newlines
        Only put quotes surrounding the message
    }
}

Execute Think(new message)
```

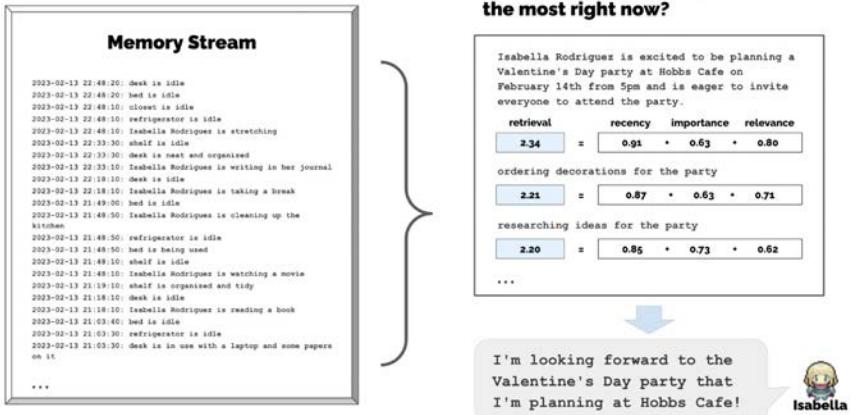
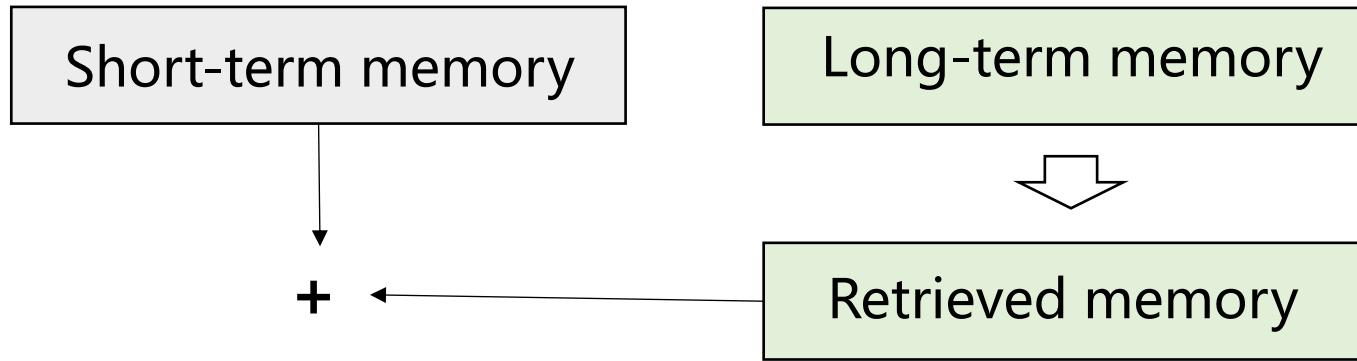


Reflective Linguistic Programming (RLP): A Stepping Stone in Socially-Aware AGI (SocialAGI)

Describe, Explain, Plan and Select: Interactive Planning with Large Language Models Enables Open-World Multi-Task Agents

# Short-term + Long-term Memory

Explicit external memory storage



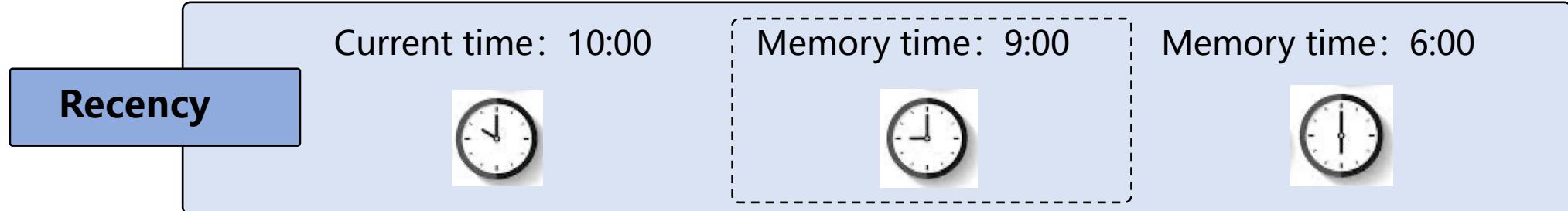
## Short-term memory

current state, agent profile, ...

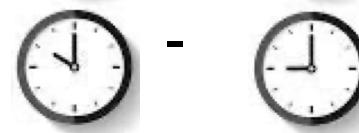
## Long-term memory

Retrieved information from the memory stream

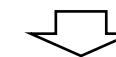
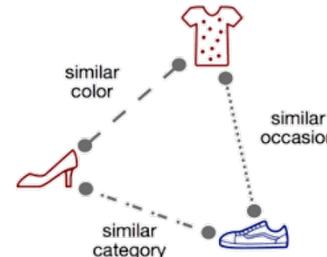
# | Memory Read



# Memory Read



Recency score



Relevance score

$$m^* = \arg \min_{m \in M} \alpha s^{rec}(q, m) + \beta s^{rel}(q, m) + \gamma s^{imp}(m)$$

The set of all memories

Current context

Importance score



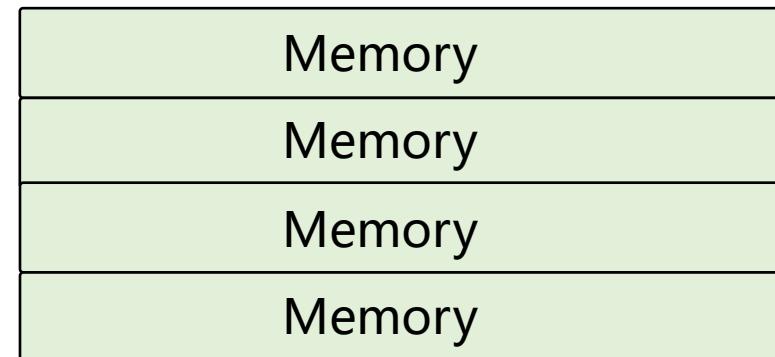
# Memory Write



← **Observation** →



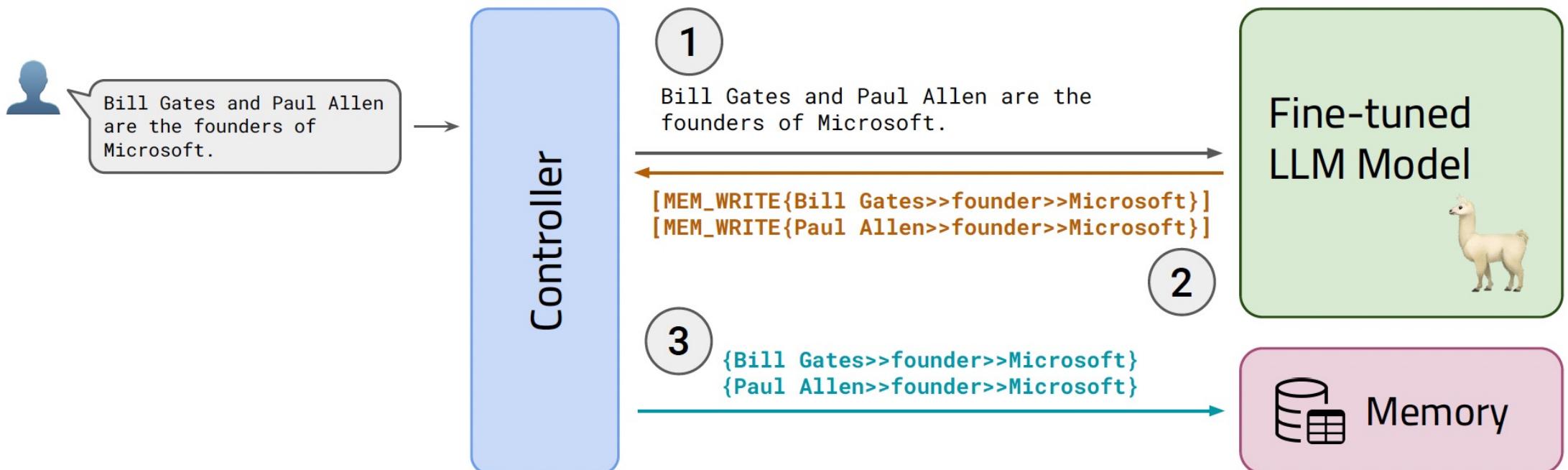
**Memory Pool**



How to store the information?

# Memory Write

Raw Text → Symbolic → Memory Write



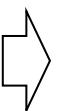
(a) Memory-Write scenario: (1) Controller passes the input to the LLM (2) which generates the appropriate memory write call. (3) The controller gives the data (and their average representations) to the memory to be stored.

# Memory Reflection

- **Self-reflection** is the ability to witness and evaluate our own cognitive, emotional, and behavioral processes.



WIKIPEDIA  
The Free Encyclopedia



- Self-summarization
- Self-verification
- Empathy
- ...

# Self-summarization

Raw Memory



**Prompt:** Given only the information above, what are 3 most salient high-level questions we can answer about the subjects in the statements?

Reflection Topics

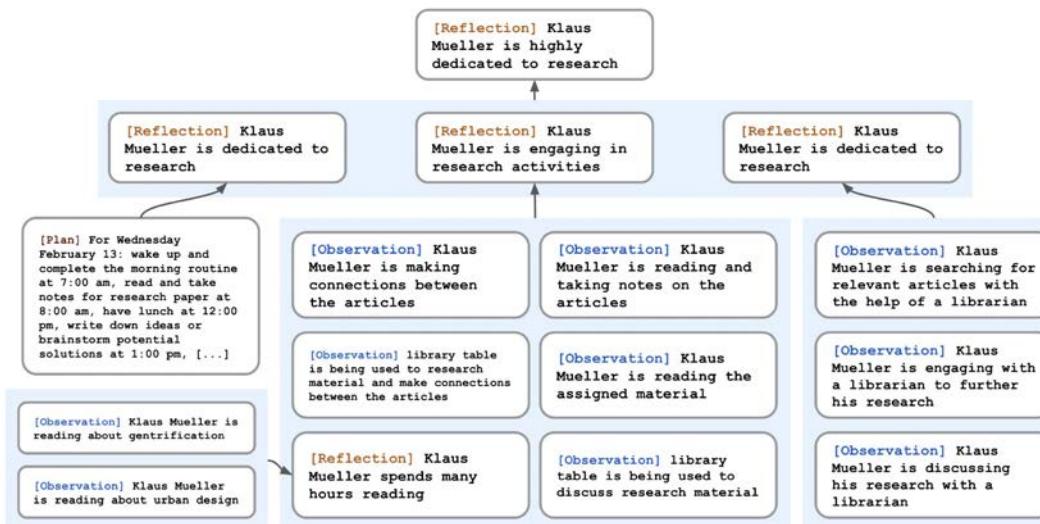


Insights

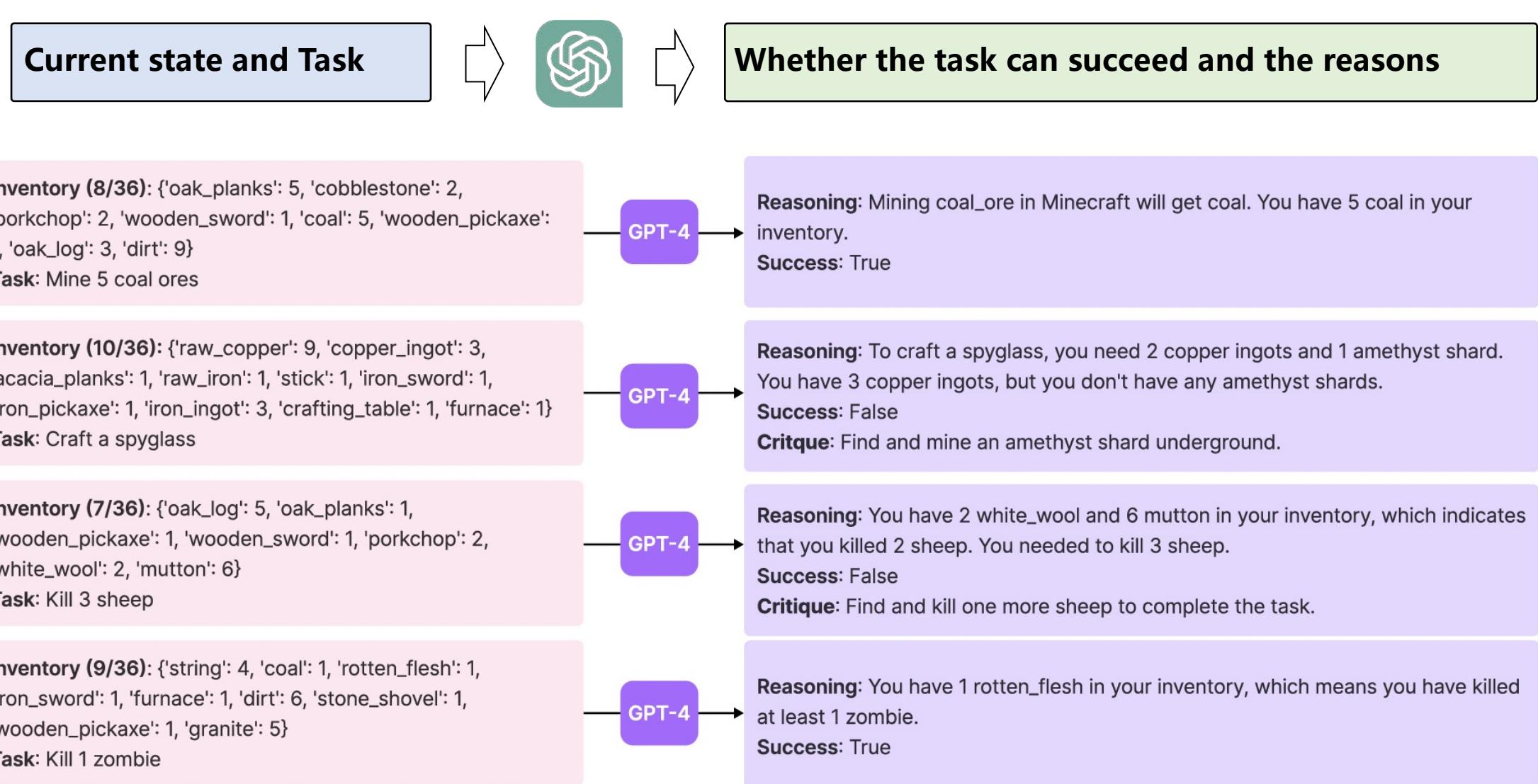


Retravel Relevant Memory

Statements about Klaus Mueller  
1. Klaus Mueller is writing a research paper  
2. Klaus Mueller enjoys reading a book on gentrification  
3. Klaus Mueller is conversing with Ayesha Khan about exercising [...]  
What 5 high-level insights can you infer from the above statements? (example format: insight (because of 1, 5, 3))



# Self-verification





# Application

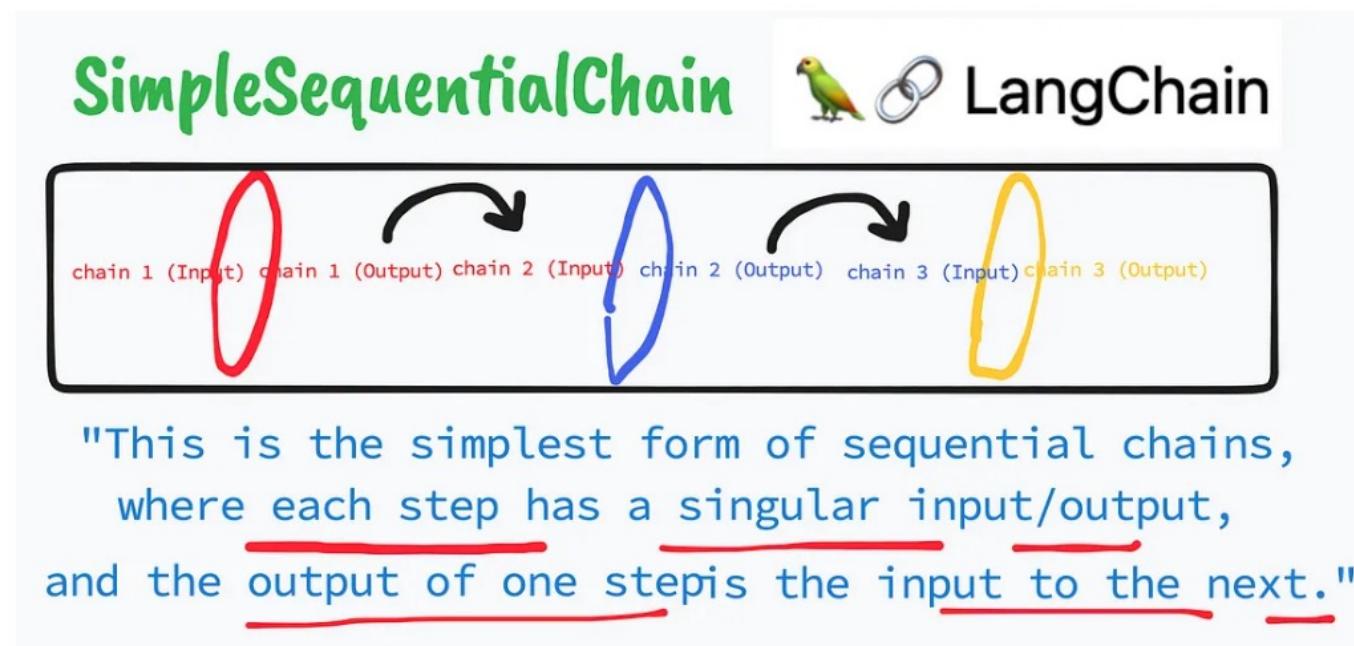
ByteDance

# Open-source Solutions

- LangChain



- Integrated library to “chain” language models
- Also the first library to leverage tools



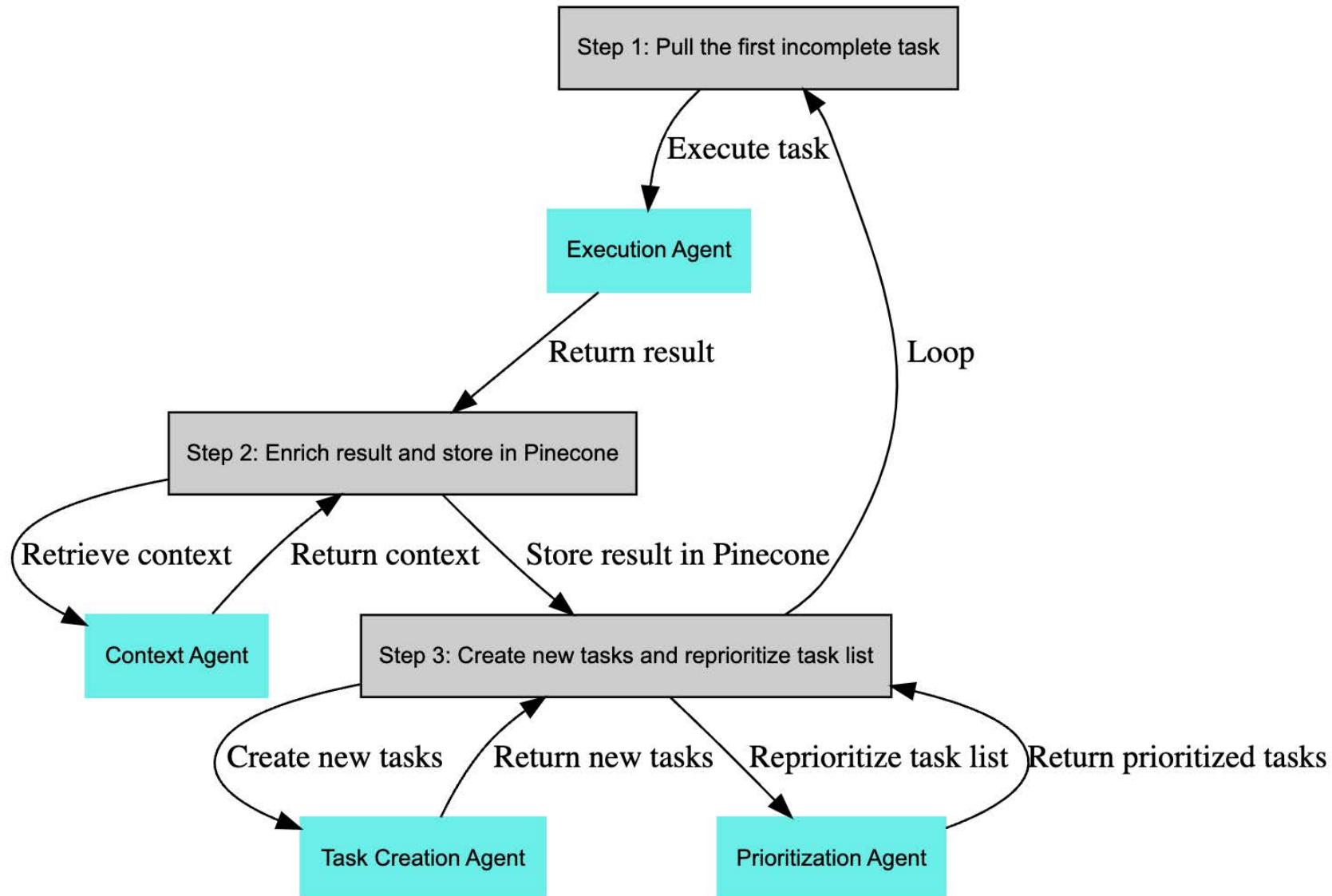
# Open-source Solutions

- BabyAGI
  - Pull the first task from the task list
  - Send the task to the execution agent, which uses ChatGPT to complete the task based on the context
  - Enrich the result and stores it in Pinecone
  - Create new tasks and reprioritizes the task list based on the objective and the result of the previous task

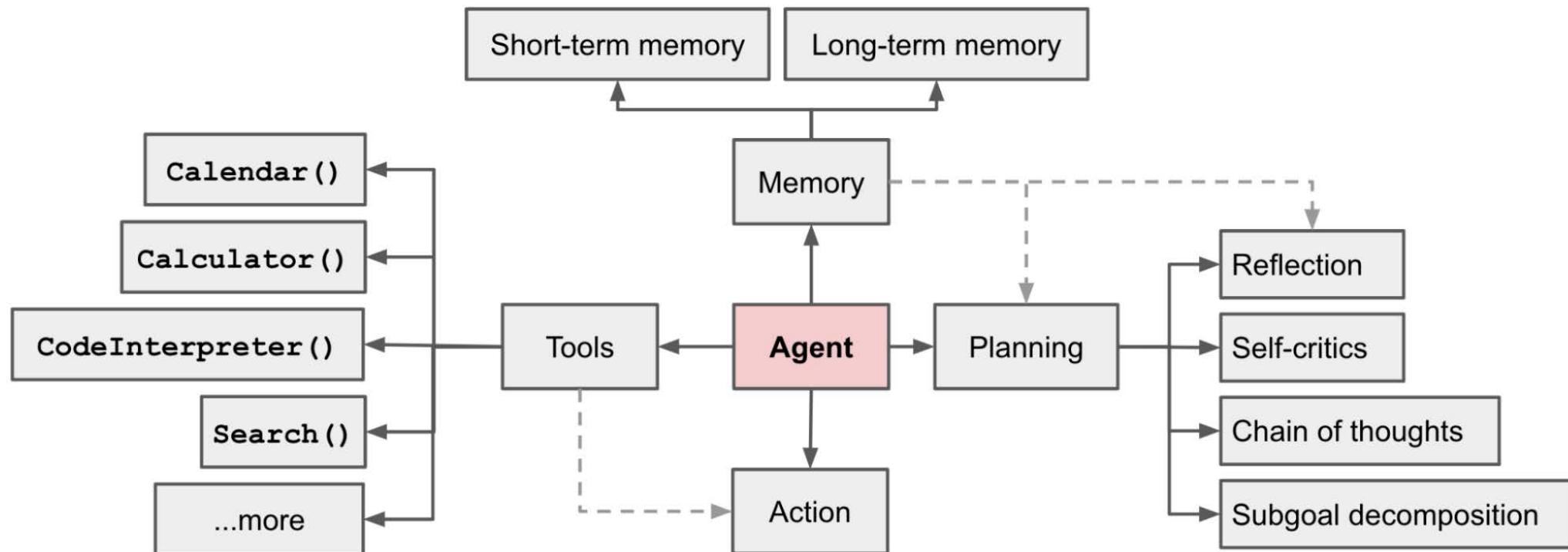


# Open-source Solutions

- BabyAGI



# Agent



# XAgent

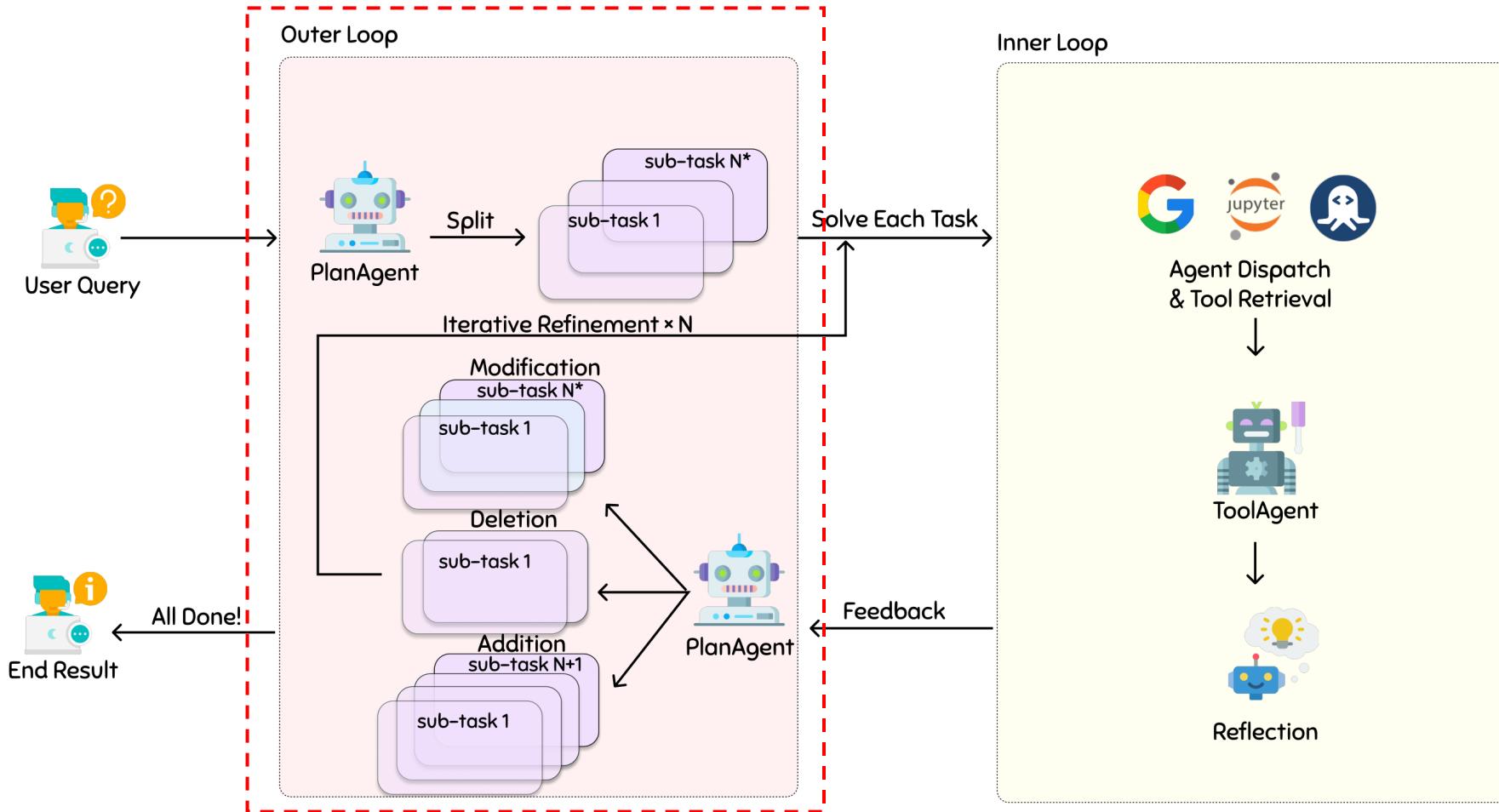
- Dual-loop Mechanism for Planning and Execution
- ToolServer: Tool Execution Docker
- The Universal Language: Function Calling:



<https://github.com/OpenBMB/XAgent>

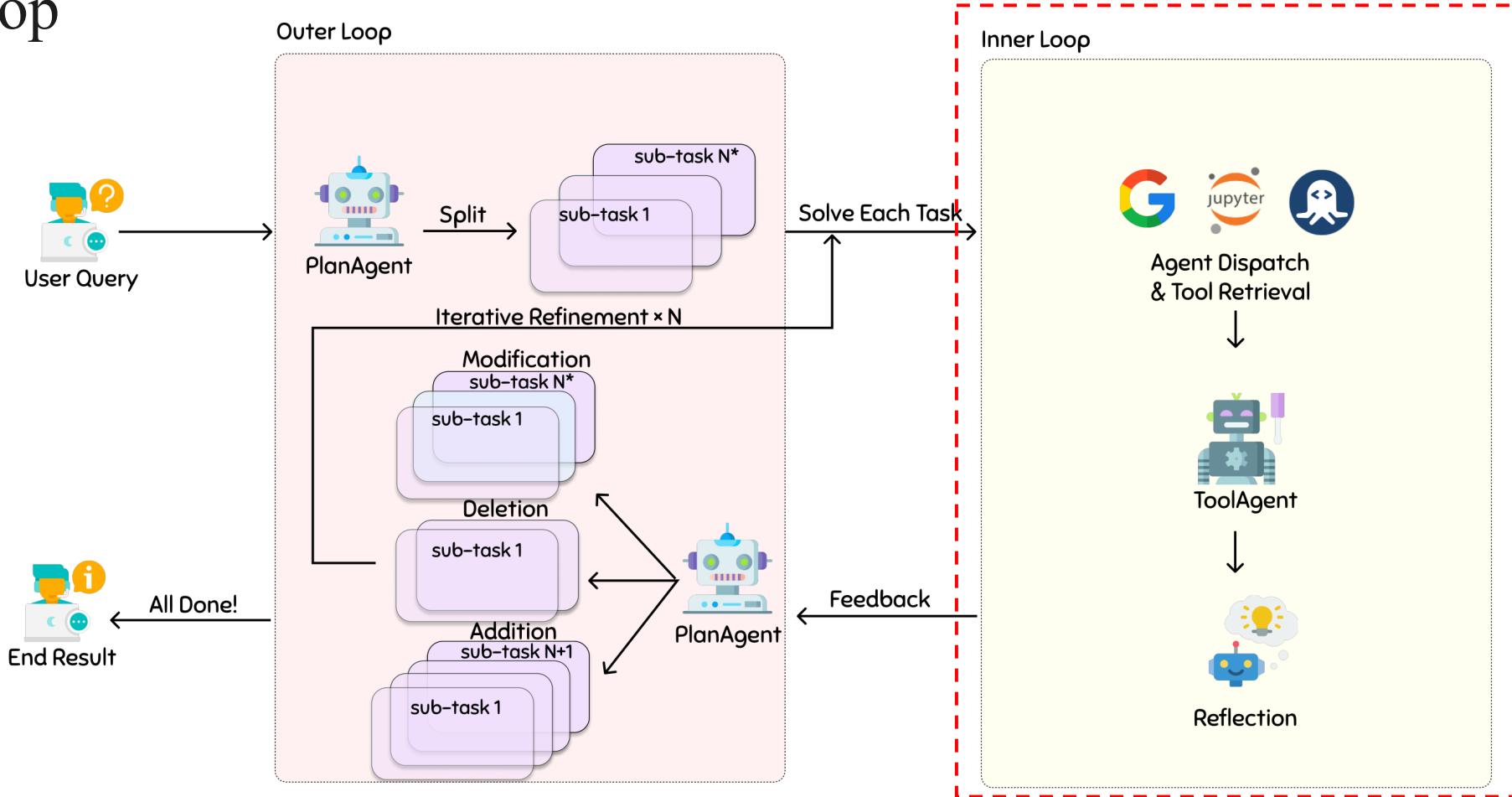
# Dual-loop Mechanism

- Outer-loop serves as the high-level planner and the primary orchestrator of tasks



# Dual-loop Mechanism

- **Inner-loop** is pivotal for executing the individual sub-tasks assigned by the outer-loop

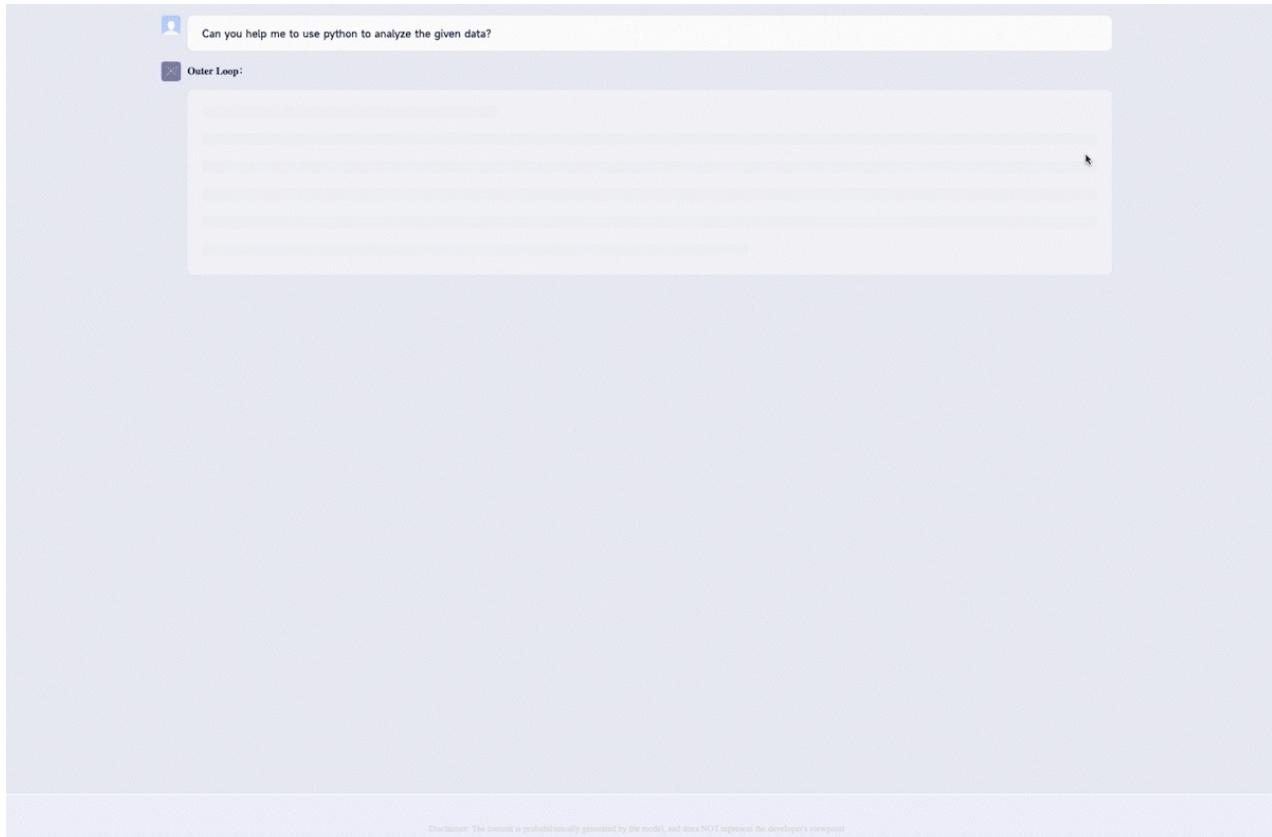


# ToolServer

- ToolServerManager
  - Manages the lifecycle of docker containers (i.e., nodes), handling their creation, monitoring, and shutdown
- ToolServerMonitor
  - Checks the status of nodes, updating their states, and ensuring their efficient execution
- ToolServerNode
  - Individual execution unit where actions (e.g., API call, etc.) are performed

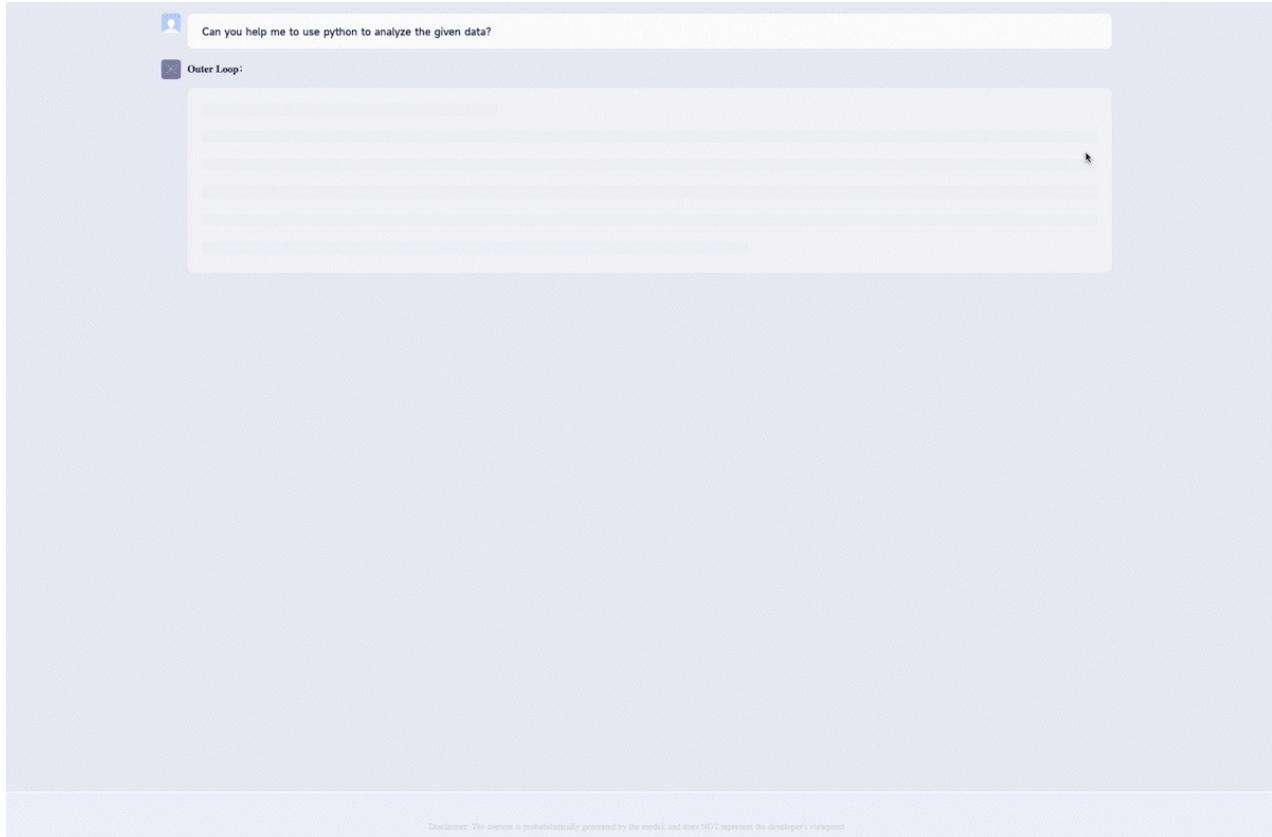
# Example: Data Analysis

- Outer-loop splits the task into four sub-tasks
  - Data inspection and comprehension
  - Verification of the system's Python environment for relevant data analysis libraries
  - Crafting data analysis code for data processing and analysis
  - Compiling an analytical report based on the Python code's execution results.

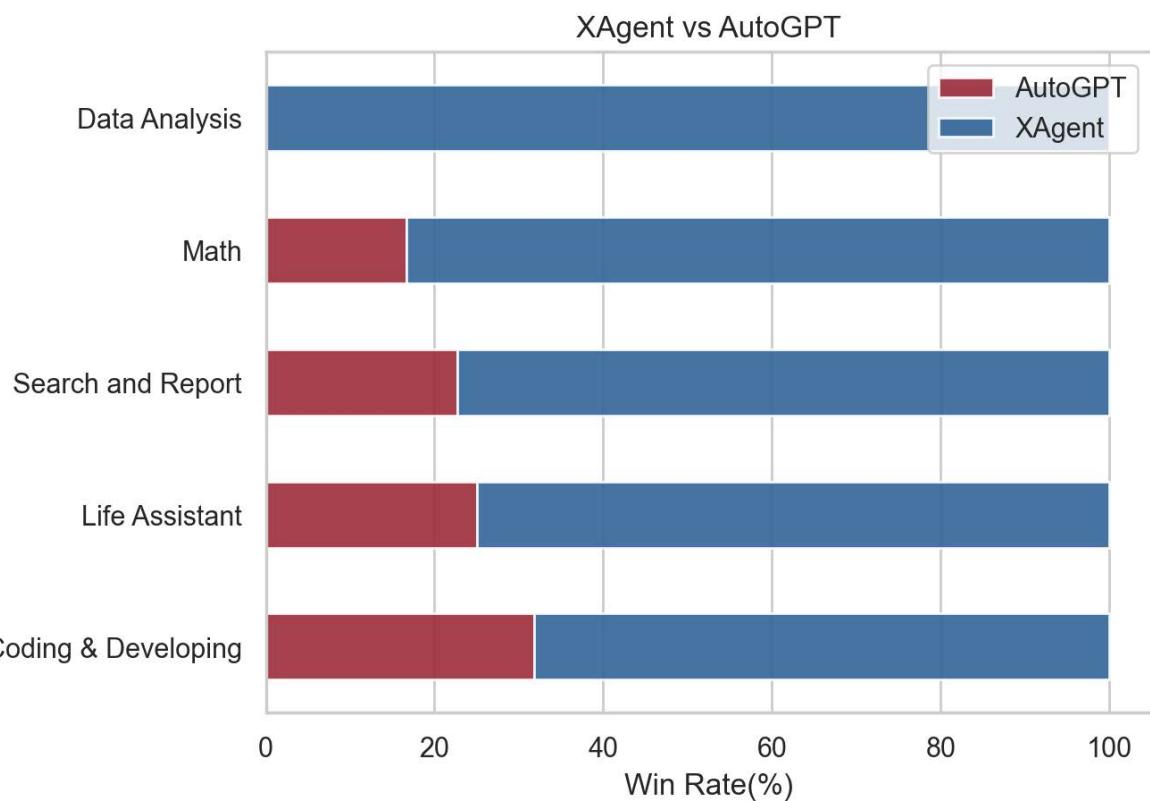


# Case Study: Data Analysis

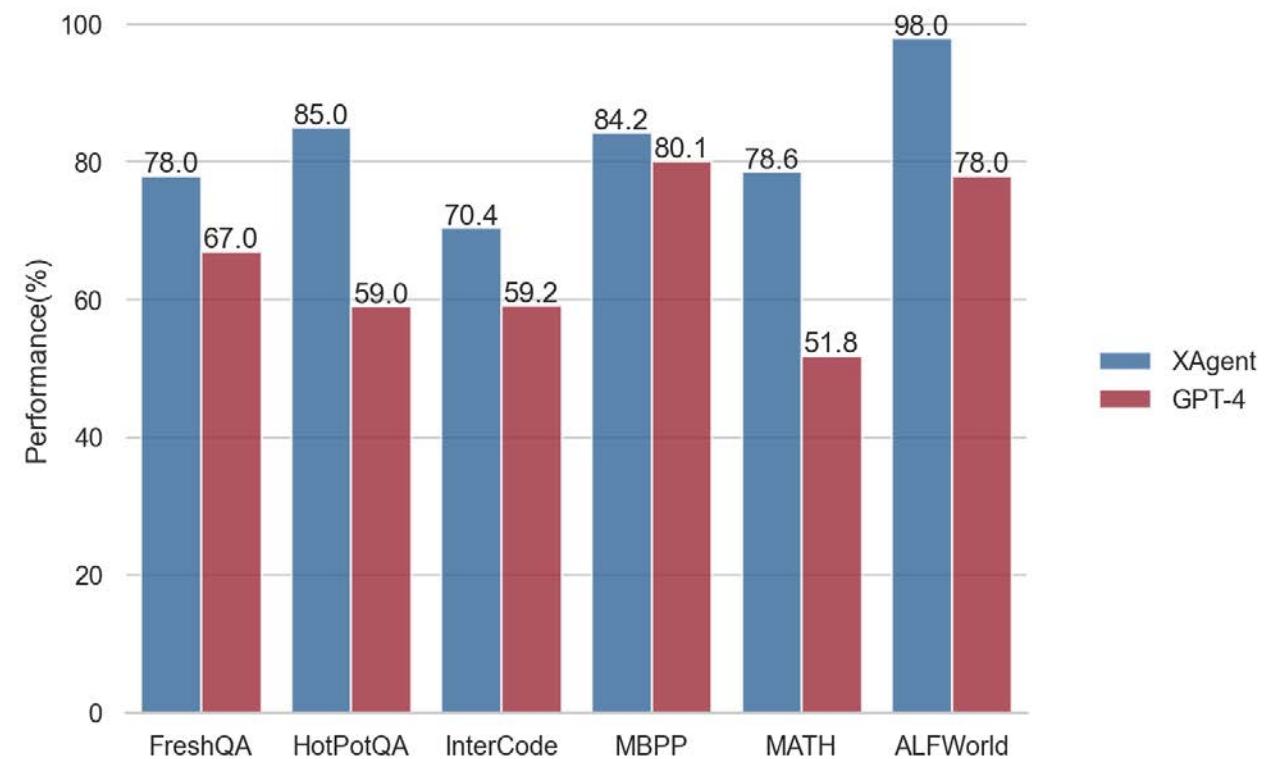
- Inter-loop
  - Employ various data analysis libraries such as pandas, sci-kit learn, seaborn, matplotlib, alongside skills in file handling, shell commands, and Python notebooks



# Performance



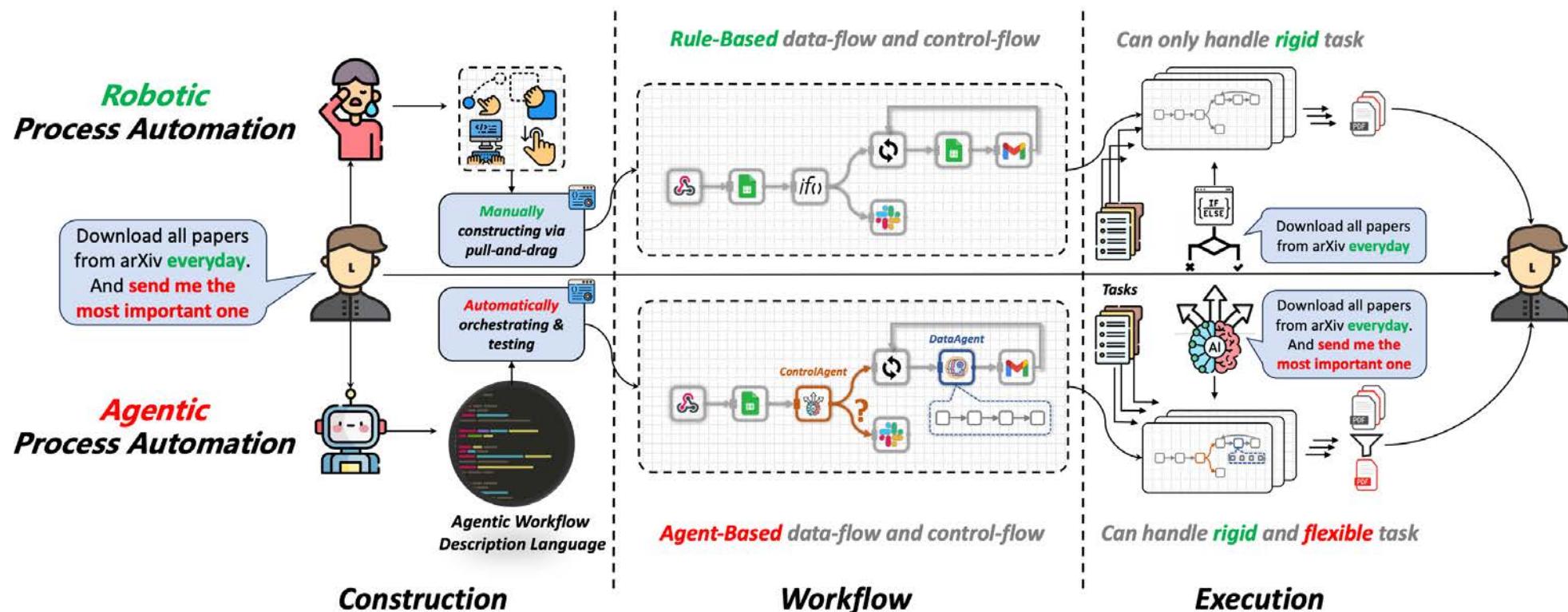
XAgent v.s. AutoGPT on our curated instructions



# ProAgent

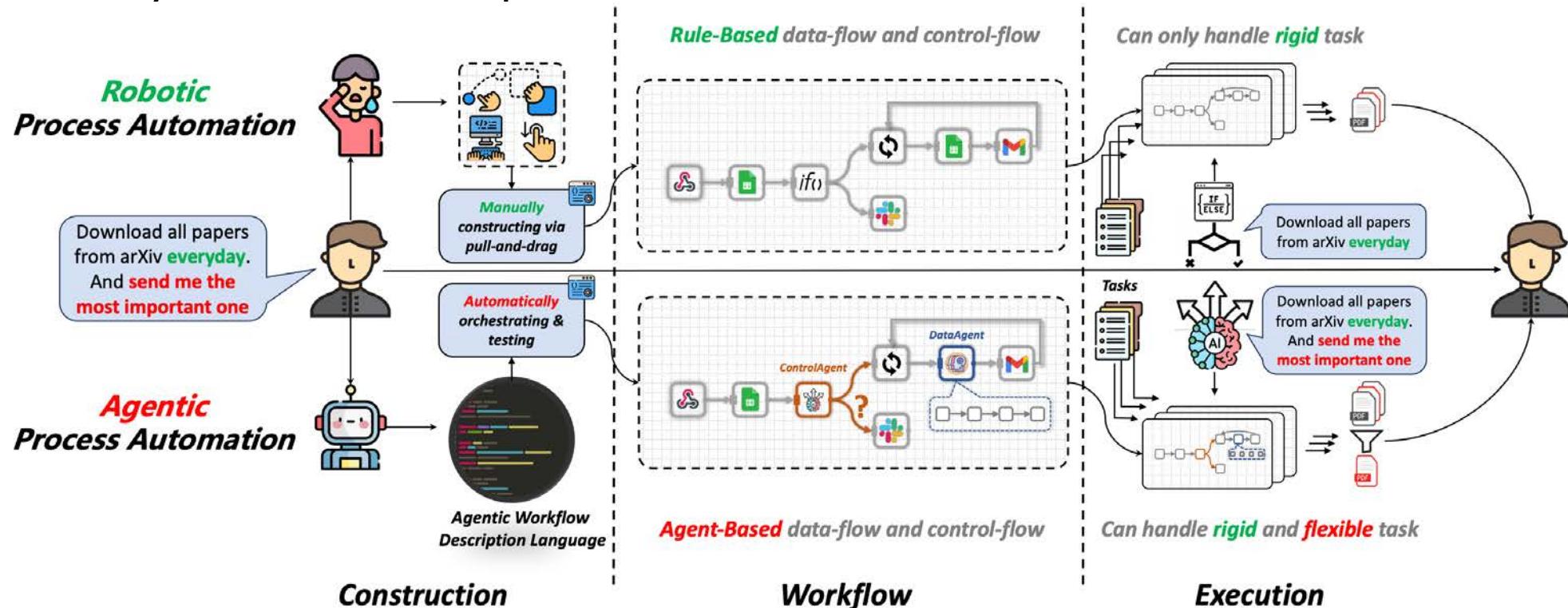
- Robotic Process Automation (RPA)

- Involve manually programming rules to coordinate multiple software applications into a solidified workflow. It achieves efficient execution by interacting with software in a manner that simulates human interaction.



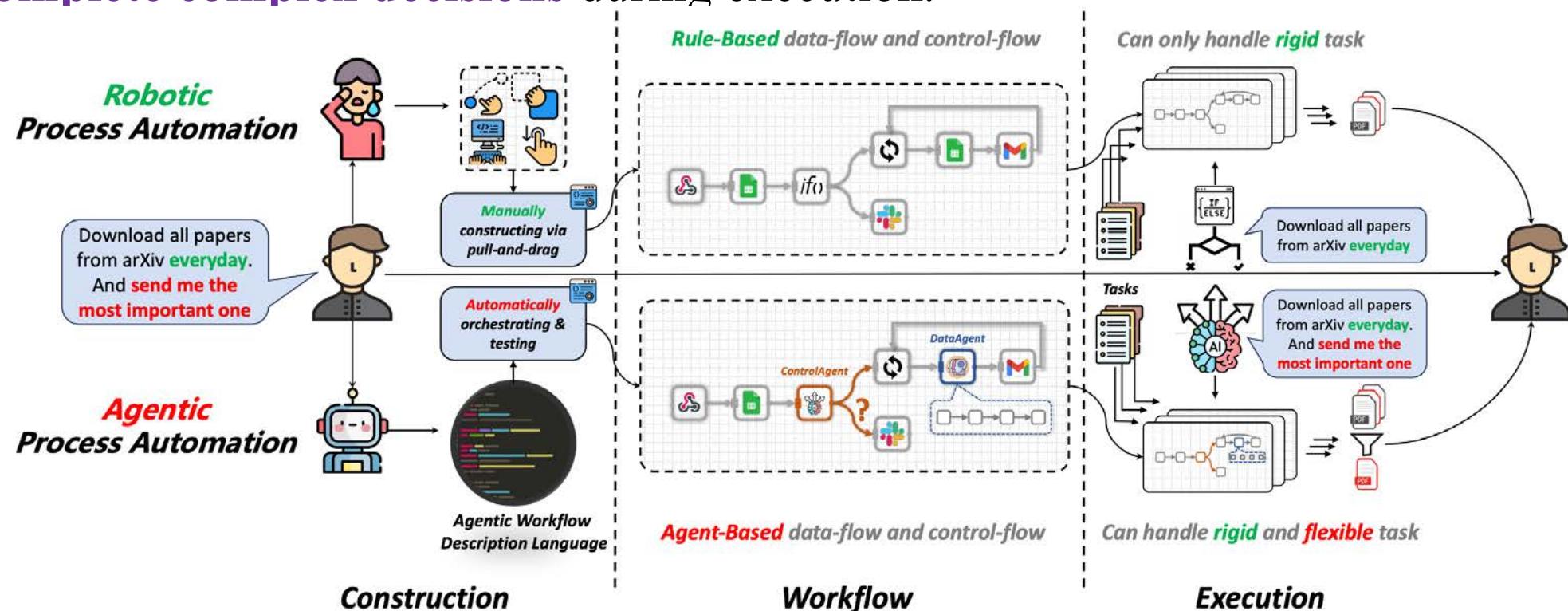
# ProAgent

- Limitation of RPA
  - Constructing RPA workflows requires **substantial human labor**
  - Complex tasks are very flexible, involving **dynamic decision-making**, and are difficult to solidify into rules for representation



# ProAgent

- Agentic Process Automation based on LLM-based Agent
  - The agent **autonomously completes the construction of workflows** with human needs
  - **Dynamically recognizing decision-making** during the build and **actively taking over to complete complex decisions** during execution.



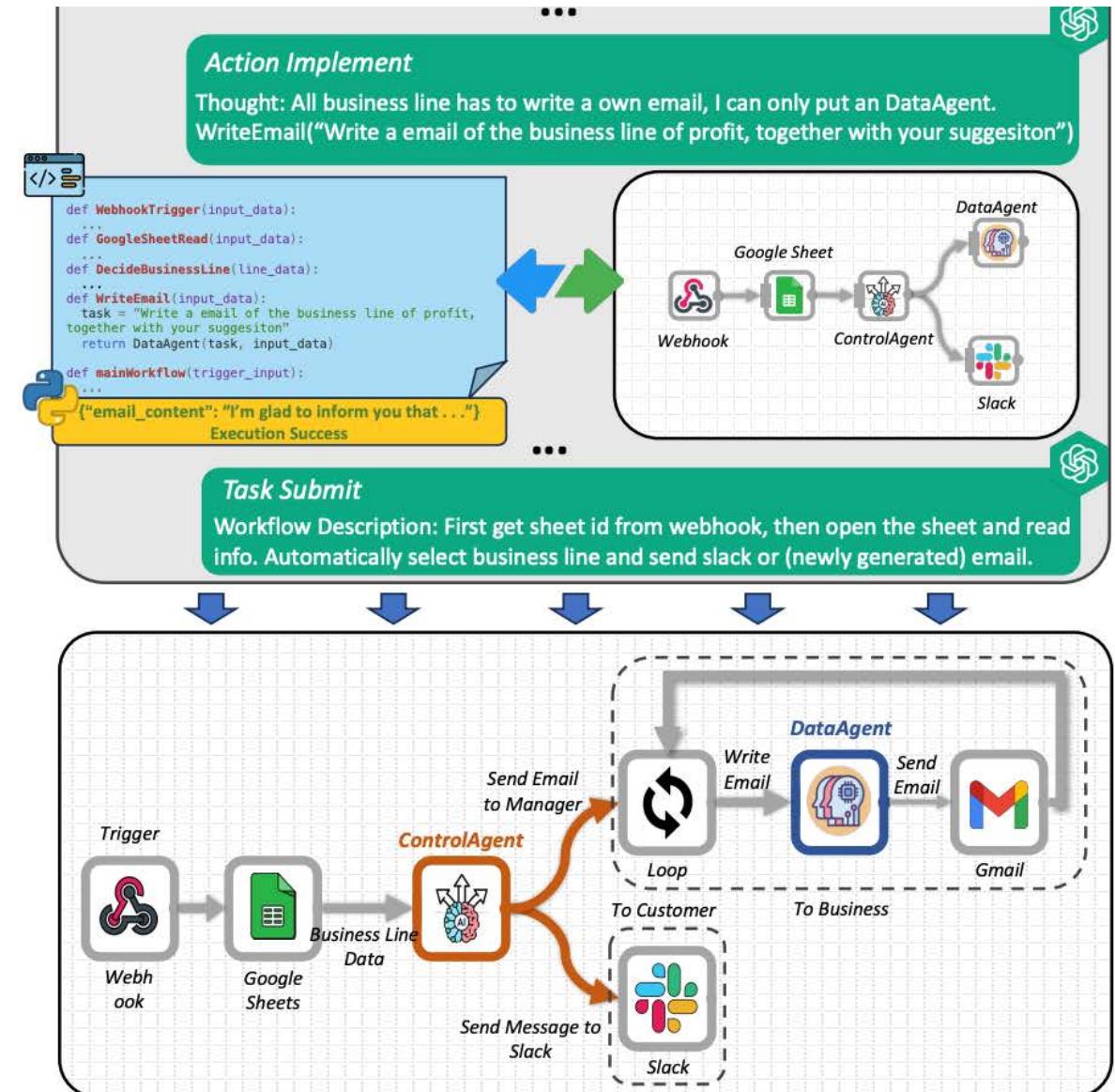
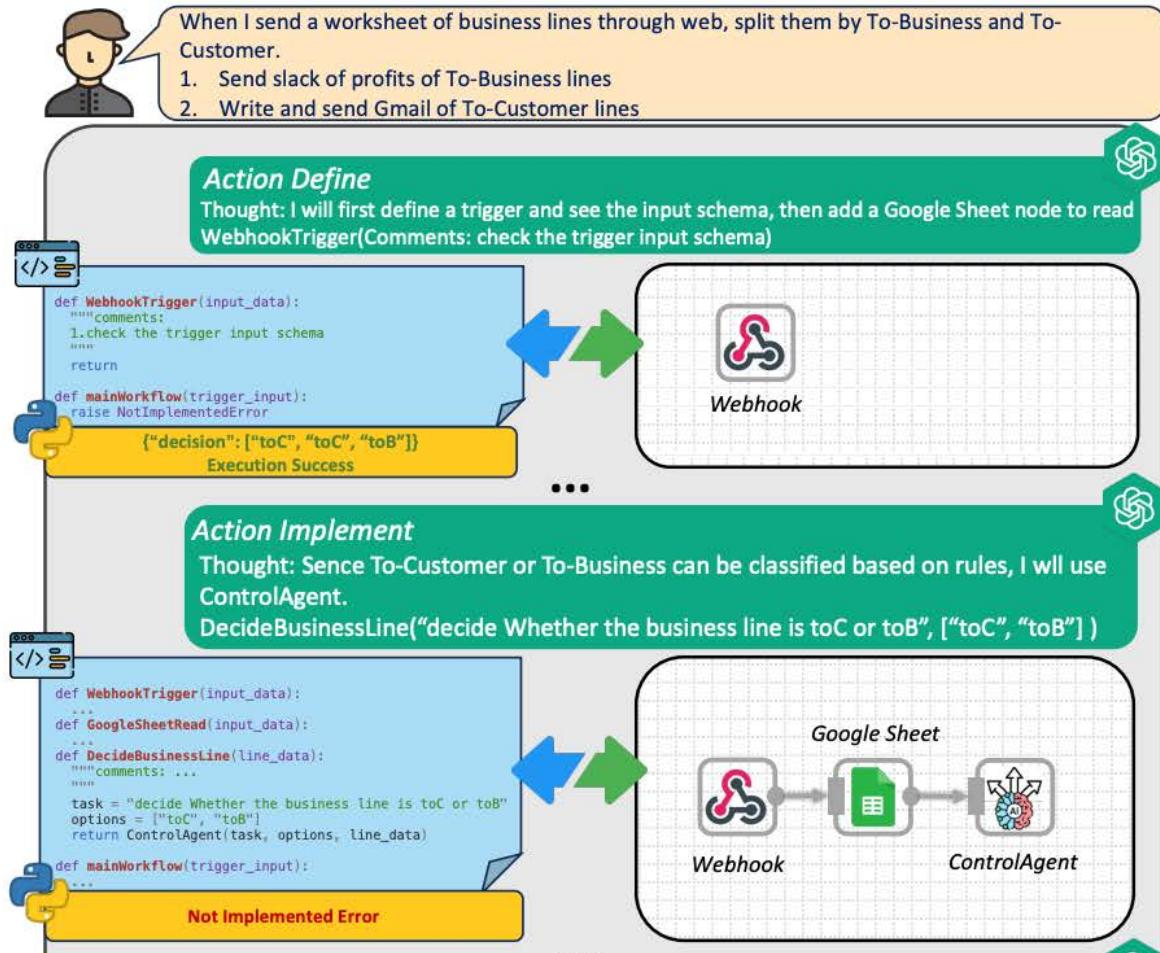
# Example

## Task

When I send a worksheet of business lines through Web, deal with them according to which type of each business line belong to.

1. To-Customer: Send a message to Slack to report the profits of business lines.
2. To-Business: Write a report which should analyze the data to give some suggestions and then send it to the Gmail of the corresponding managers.

# Example



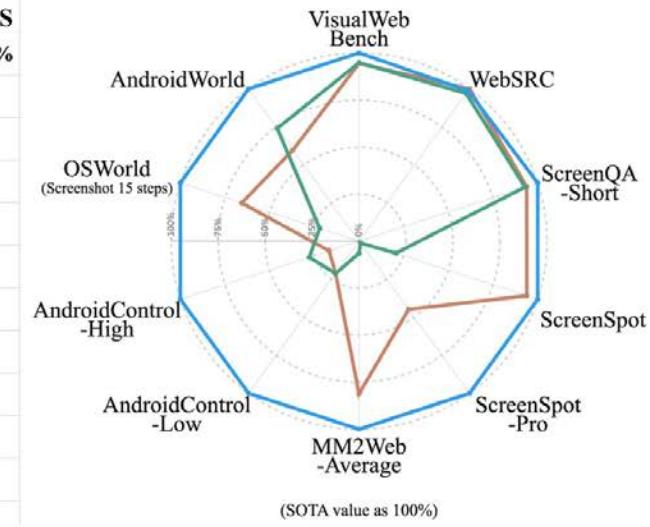
## XUI-TARS: Pioneering Automated GUI Interaction with Native Agents

Yujia Qin<sup>†\*</sup>, Yining Ye<sup>\*◊</sup>, Junjie Fang\*, Haoming Wang\*, Shihao Liang\*, Shizuo Tian<sup>◊</sup>, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, Wanjun Zhong, Kuanye Li, Jiale Yang, Yu Miao, Woyu Lin, Longxiang Liu, Xu Jiang, Qianli Ma, Jingyu Li, Xiaojun Xiao, Kai Cai, Chuang Li, Yaowei Zheng, Chaolin Jin, Chen Li, Xiao Zhou, Minchao Wang, Haoli Chen, Zhaojian Li, Haihua Yang, Haifeng Liu, Feng Lin, Tao Peng, Xin Liu, Guang Shi<sup>†</sup>

ByteDance Seed, <sup>◊</sup>Tsinghua University

{yujia.qin, shiguang.sg}@bytedance.com

Benchmark	Previous SOTA	Relative improvement of UI-TARS	
GUI-Odyssey	OS-Atlas-7B	+42.90%	+40.32%
OSWorld (Screenshot 15 steps)	Aguvis-72B w/ GPT-4o	+33.53%	+10.00%
ScreenSpot-Pro	UGround-V1-7B	+22.51%	+14.79%
MM2Web-Website	Aguvis-72B	+12.39%	+9.20%
AndroidControl-Low	OS-Atlas-7B	+7.16%	+6.57%
MM2Web-Task	Aguvis-72B	+7.19%	+4.84%
MM2Web-Domain	Aguvis-72B	+6.70%	+3.95%
ScreenSpot-v2	OS-Atlas-7B	+3.67%	+5.17%
ScreenQA-Short	Qwen2-VL-7B	+4.36%	+3.30%
VisualWebBench	GPT-4o	+5.48%	+1.53%
AndroidControl-High	OS-Atlas-7B	+4.92%	+1.83%

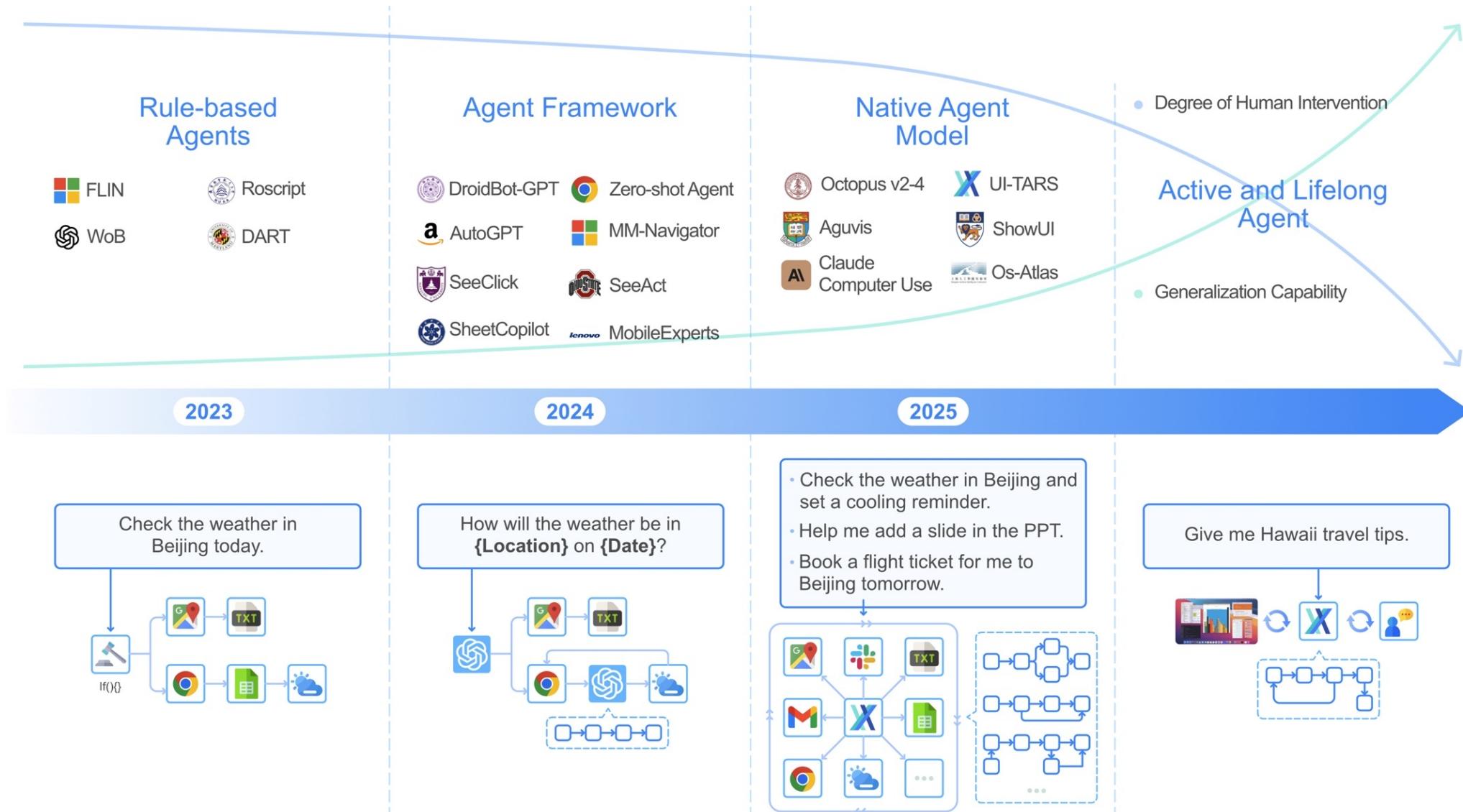


# UI-TARS

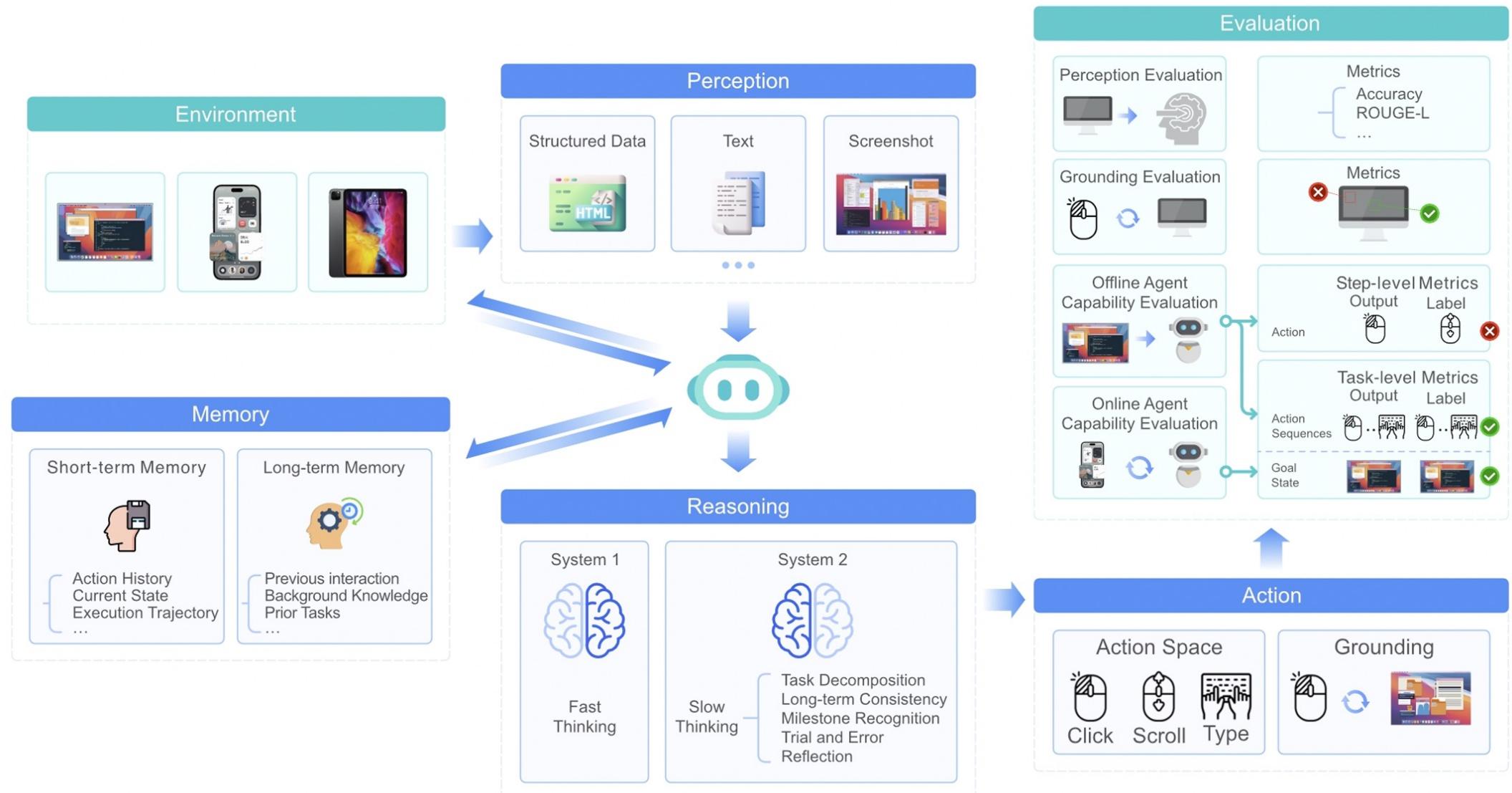
UI-TARS is an end-to-end GUI agent model based on VLM architecture. It solely perceives screenshots as input and performs human-like interactions (e.g., keyboard and mouse operations), achieving SOTA performance on 10+ GUI benchmarks.

GitHub Link: <https://github.com/bytedance/UI-TARS>

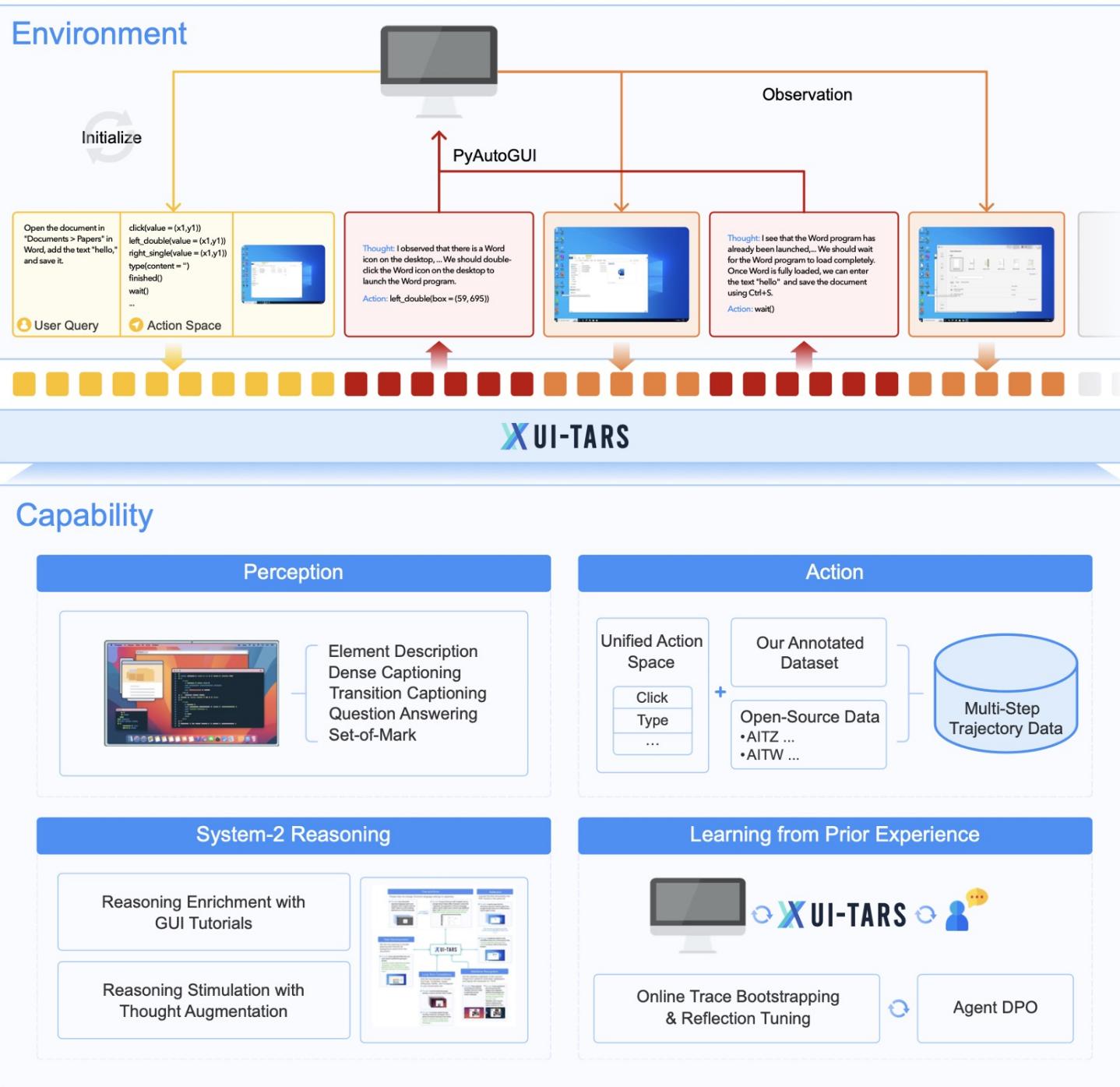
# UI-TARS



# UI-TARS



# UI-TARS



# UI-TARS



## Task Decomposition

Set the line spacing to double spacing and remove all background colors from the document.

Thought: I have selected all the text, and now I need to set the line spacing to double. To do this, I need to right-click and select "Paragraph". In the dialog box that appears, find the "Line spacing" option and select "Double" from the dropdown menu.

# X UI-TARS

## Long Term Consistency

Add the homepages of Google, YouTube, Facebook, Apple, Wikipedia, Netflix, and Instagram to your bookmarks bar.

Thought: I see the bookmark popup window, I need to click the "Done" button.

Thought: I've already added Google, YouTube, Facebook, and Apple. I see a photo of someone waving on the screen, but this is irrelevant to my task. Now I need to continue adding Wikipedia's homepage...

## MileStone Recognition

Set the desktop wallpaper to the second image from system's automatic wallpapers, and adjust the resolution to "16:9".

Thought: I have opened the background settings interface, and now I need to select the second system wallpaper.

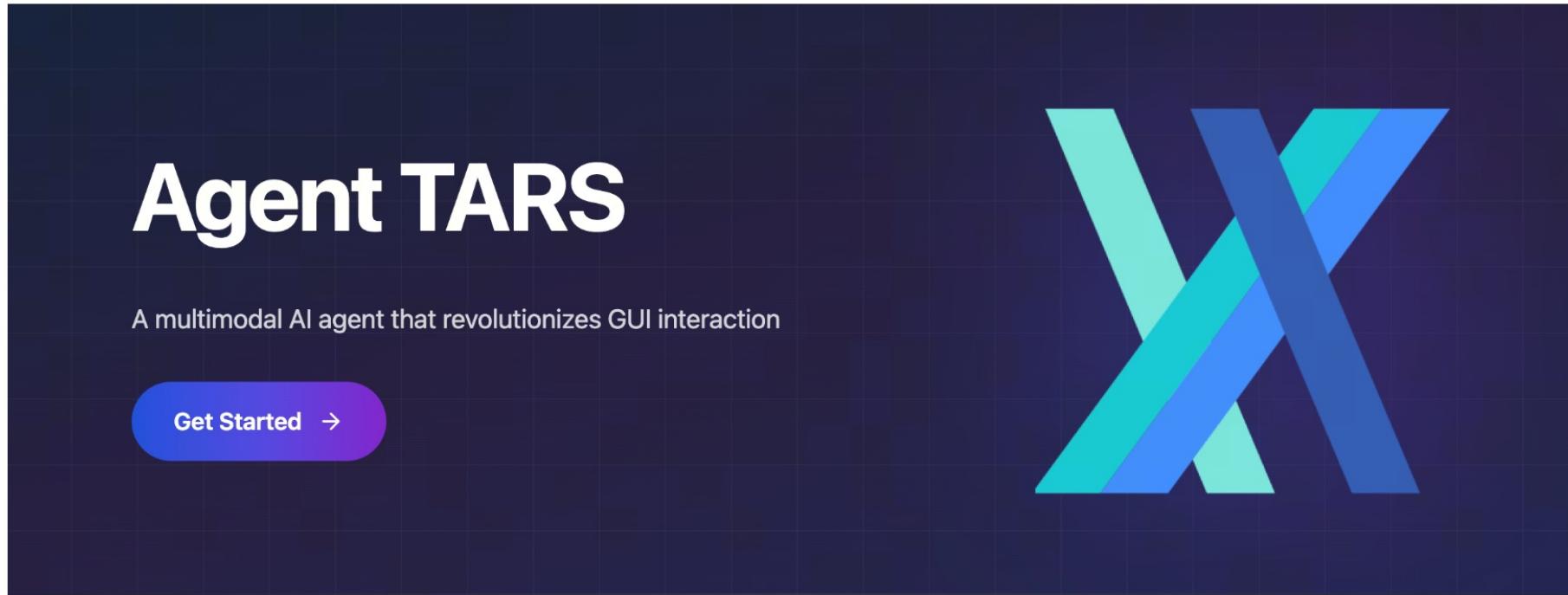
Thought: I have switched to the second automatic image as the wallpaper, and from the background, I can see that the wallpaper has been changed to the second image.

Therefore, next I need to continue adjusting the screen resolution to 16:9.



# Agent-TARS

 Important



[2025-03-18] We released a **technical preview** version of a new desktop app - [Agent TARS](#), a multimodal AI agent that leverages browser operations by visually interpreting web pages and seamlessly integrating with command lines and file systems.

<https://agent-tars.com/2025/03/18/announcing-agent-tars-app>

# Agent-TARS

Watch 90 ▾

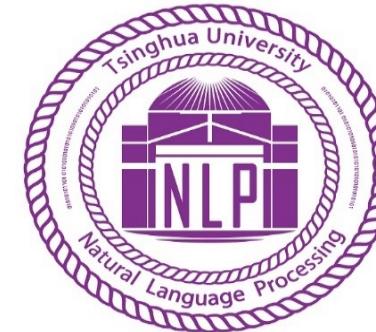
Fork 628 ▾

Starred 8.7k ▾

The screenshot shows the GitHub Showcase page for the Agent-TARS repository. The page title is "Showcase" and the subtitle is "Explore our collection of impressive demos and applications". There are eight cards displayed, each representing a different demo or application:

- Agent TARS** (Research, 2025-03-23): Organize and summarize the latest releases of Agent TARS. This card includes a video thumbnail and a "Get Started" button.
- Technical analysis of Tesla's future stock price trends** (Finance, 2025-03-20): A chart showing Tesla's stock price trend from February 28 to March 15, 2025.
- Travel plan from Hangzhou to Weihai in detailed markdown...** (General, 2025-03-20): A travel plan for a trip from Hangzhou to Weihai, written in markdown.
- Top 5 most popular ProductHunt projects analysis...** (Research, 2025-03-20): An analysis of the top 5 most popular projects on ProductHunt.
- Reasons behind Tesla's recent stock price decline** (Finance, 2025-03-18): An analysis of the reasons behind Tesla's recent stock price decline.
- Time for Eliud Kipchoge to run Earth-Moon distance at...** (Science, 2025-03-18): An analysis of the time it would take for Eliud Kipchoge to run the distance between Earth and the Moon.
- 7-day trip plan to Mexico City from NYC** (General, 2025-03-18): A 7-day trip plan from New York City to Mexico City.
- Analyse issues in the Lynx repository** (Technology, 2025-03-18): An analysis of issues in the Lynx repository.

<https://agent-tars.com/2025/03/18/announcing-agent-tars-app>



# Discussion

ByteDance

# From Tool User to Tool Creator

- Humans are the primary agents that create and use tools from Stone Age to 21st century
- Most tools are created for humans, not AI
- Tools Made for Models
  - Modularized: compose tools into smaller units
  - New input and output formats: more computable and suitable for AI

# From Tool User to Tool Creator

**Prompt:** You have access to a function in Python: `forecast_weather(city, N)`, which will return the weather condition of a city after `N` days. The returned information is only about a single day and is a dictionary containing keys including `temperature`. Please wrap it into a new function, that can return the average temperature over the next `N` days in a city.

**Response:** Certainly! Here's an example code to wrap the `forecast_weather` function into a new function that calculates the average temperature of the next `N` days in a given city:

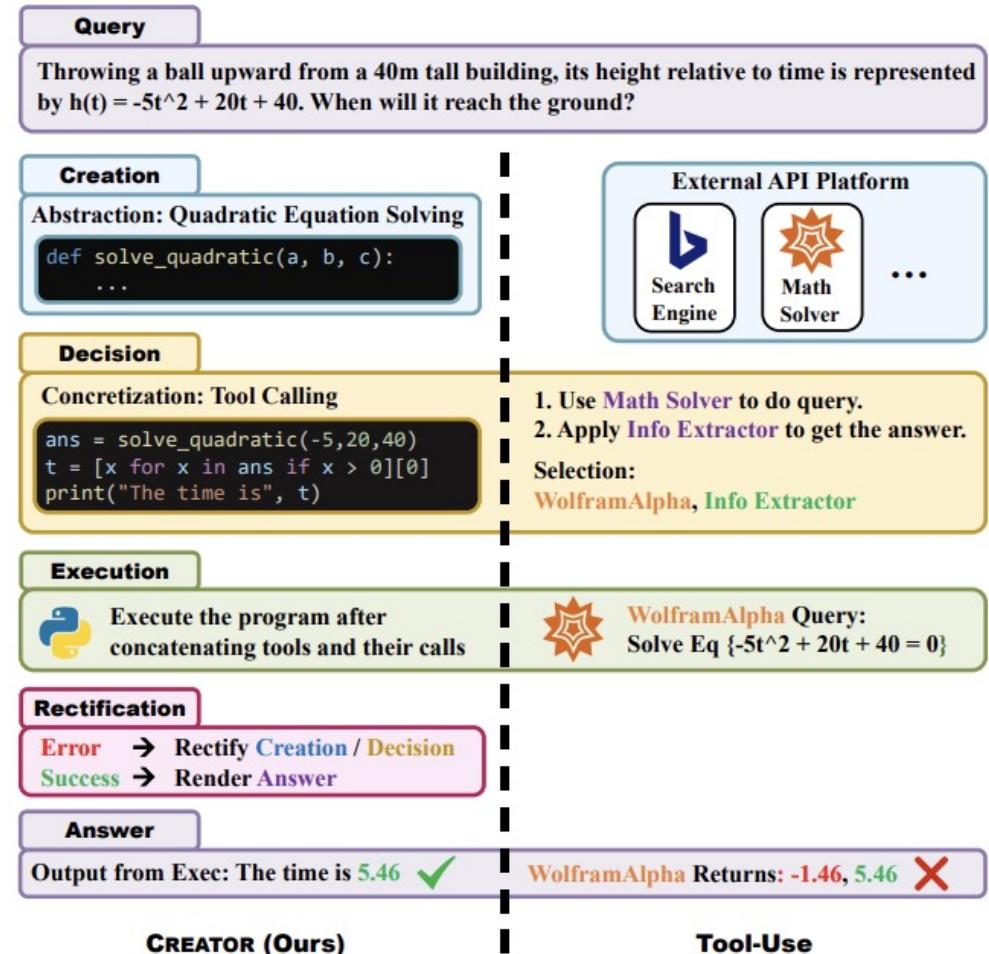
```
def average_temperature(city, N):
    total_temperature = 0
    for i in range(N):
        weather_data = forecast_weather(city, i)
        temperature = weather_data['temperature']
        total_temperature += temperature
    return total_temperature / N
```

In this code, we define a new function called `average_temperature` that takes in a `city` and `N` as input parameters. Inside the function, we loop over the range of `N` days, calling the `forecast_weather` function for each day and extracting the temperature data from the returned dictionary. We then add the daily temperature to a running total. Finally, we calculate the average temperature by dividing the total temperature by `N` and return it.

Note that this code assumes that the `forecast_weather` function returns a dictionary with a key `temperature` that contains the temperature for a given day. If the actual function returns a different format, you may need to adjust the code accordingly.

# Tool Creation: CREATOR

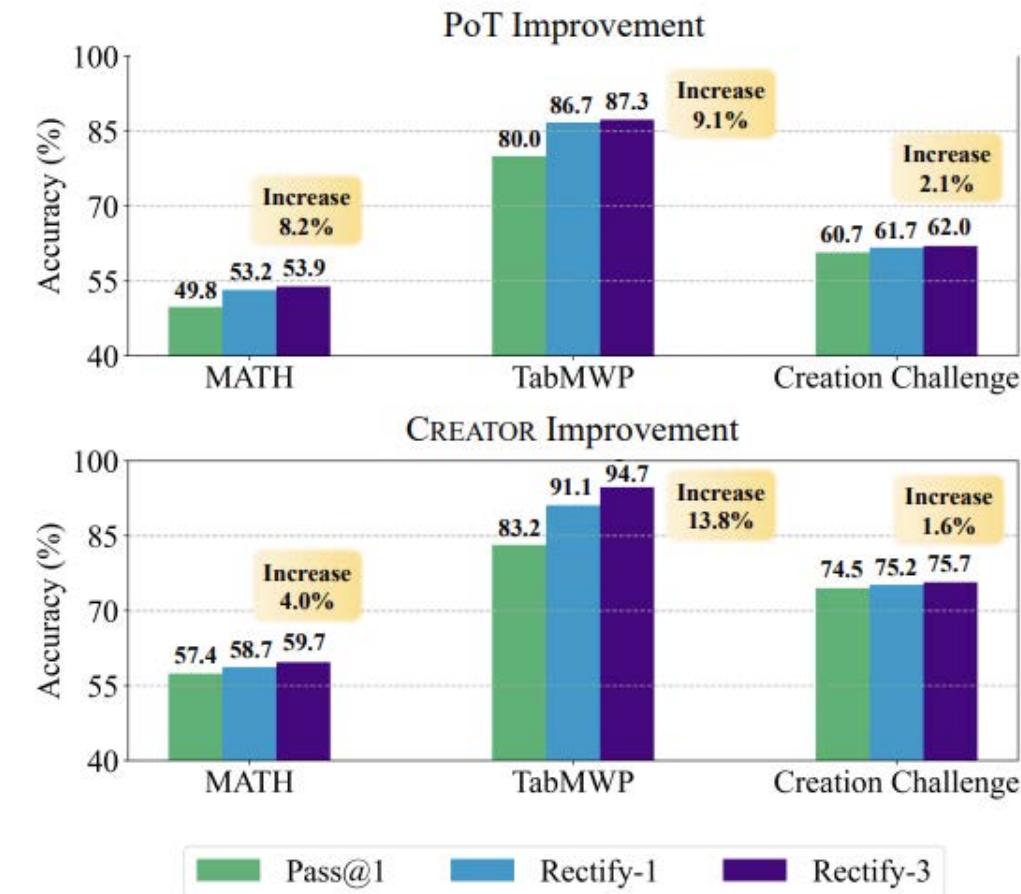
- Foundation model's tool-using ability is limited by the availability of suitable APIs and the instability of implicit reasoning
- Tool Creation:** It is no longer simply the ability to use large models to use tools, but a new tool creation module is added to allow models to create tools and find solutions to the problems they face.



# Tool Creation: CREATOR

- Significant improvements over program-of-thought (PoT) and pure chain-of-thought (CoT)

Method	Setting	Algebra	Counting & Probability	Geometry	Itmd. Algebra	Number Theory	Pre-Algebra	Pre-Calculus	Average (weighted)
<b>Standard</b>	w/o CoT	25.7	25.8	22.4	13.9	18.5	40.9	21.8	25.3
	w/ CoT	50.9	36.1	24.5	17.5	23.2	58.6	16.7	37.9
<b>PoT (w/o Rectify)</b>	w/o CoT	58.2	48.5	35.4	25.8	53.1	66.8	25.0	49.8
	w/ CoT	54.0	47.8	32.5	22.3	48.9	64.5	19.9	46.5
<b>PoT (w/ Rectify)</b>	w/o CoT	63.8	51.9	35.9	28.6	59.2	70.0	28.2	53.9
	w/ CoT	61.4	48.8	34.6	23.7	54.5	67.6	34.6	51.2
<b>Tool Use</b>	w/o CoT	47.3	35.1	27.0	20.5	30.8	56.8	31.4	39.0
	w/ CoT	55.3	37.8	28.7	20.5	34.8	61.8	26.9	43.0
<b>Tool Create (whole)</b>	w/o Demo	58.0	53.3	34.2	21.8	55.7	63.4	33.3	49.6
	w/o CoT	64.1	55.7	35.9	<b>42.7</b>	61.6	69.0	<b>37.2</b>	57.2
	w/ CoT	62.7	50.9	33.8	31.4	61.4	68.7	31.4	54.0
<b>CREATOR (ours)</b>	w/o Demo	66.6	53.6	33.8	29.4	59.8	68.7	34.6	54.9
	w/o CoT	<b>71.5</b>	55.3	<b>41.4</b>	41.9	60.4	<b>71.7</b>	35.3	<b>59.7</b>
	w/ CoT	63.1	<b>58.1</b>	34.6	35.0	<b>61.8</b>	69.7	32.1	55.7



# Reading Material

## Tool Learning

### - Must-read Papers

- Tool Learning with Foundation Models. [\[link\]](#)
- Augmented Language Models: a Survey. [\[link\]](#)
- Foundation Models for Decision Making: Problems, Methods, and Opportunities. [\[link\]](#)

### - Further Reading

- Toolformer: Language Models Can Teach Themselves to Use Tools. [\[link\]](#)
- WebGPT: Browser-assisted question-answering with human feedback. [\[link\]](#)
- ReAct: Synergizing Reasoning and Acting in Language Models. [\[link\]](#)
- Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. [\[link\]](#)
- Inner Monologue: Embodied Reasoning through Planning with Language Models. [\[link\]](#)



# Q&A

THUNLP