Theory of Agent: From Definition, to Behavior and Objective

Hongru WANG*, Cheng Qian*, Manling Li, Jiahao Qiu, Boyang Xue, Mengdi Wang, Heng Ji, Kam-Fai Wong

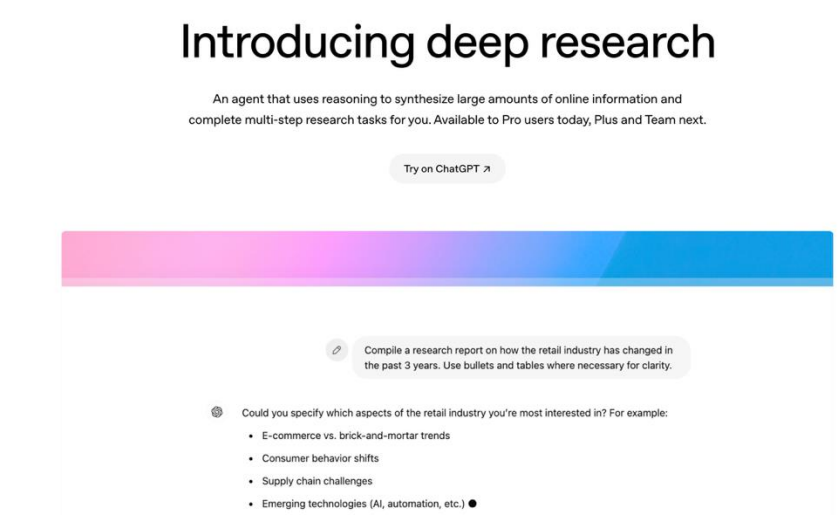
*Equal Contribution

https://rulegreen.github.io/

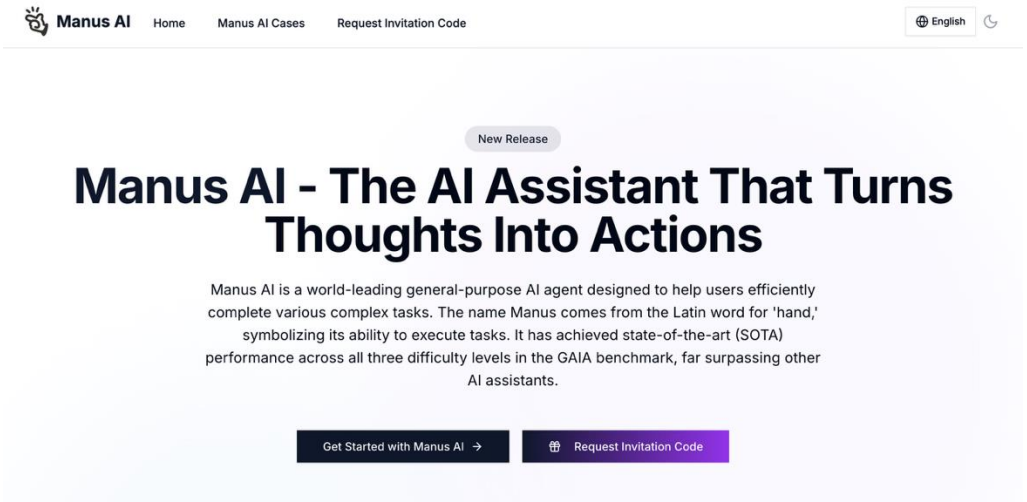
Theory of Agent

- New Agent Framework (What's Agent?)
- Behavior and Objective of Agent
- Agentic Pretraining / SFT / RL / Prompting (Why RL?)
- Future Direction

Introduction

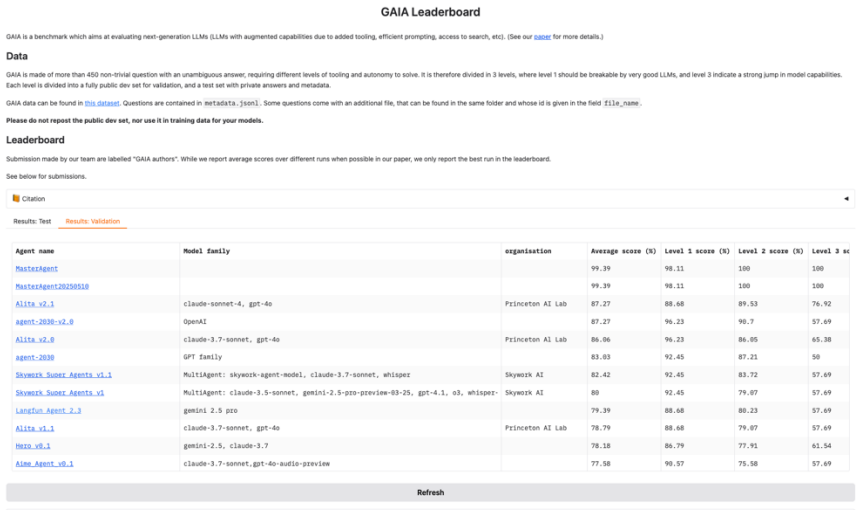


OpenAI Deep Research



Manus

Computer-Using Agent

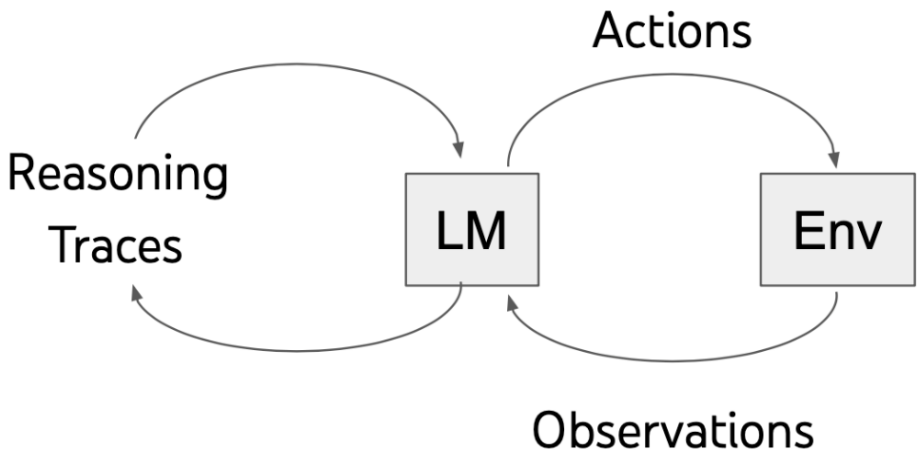


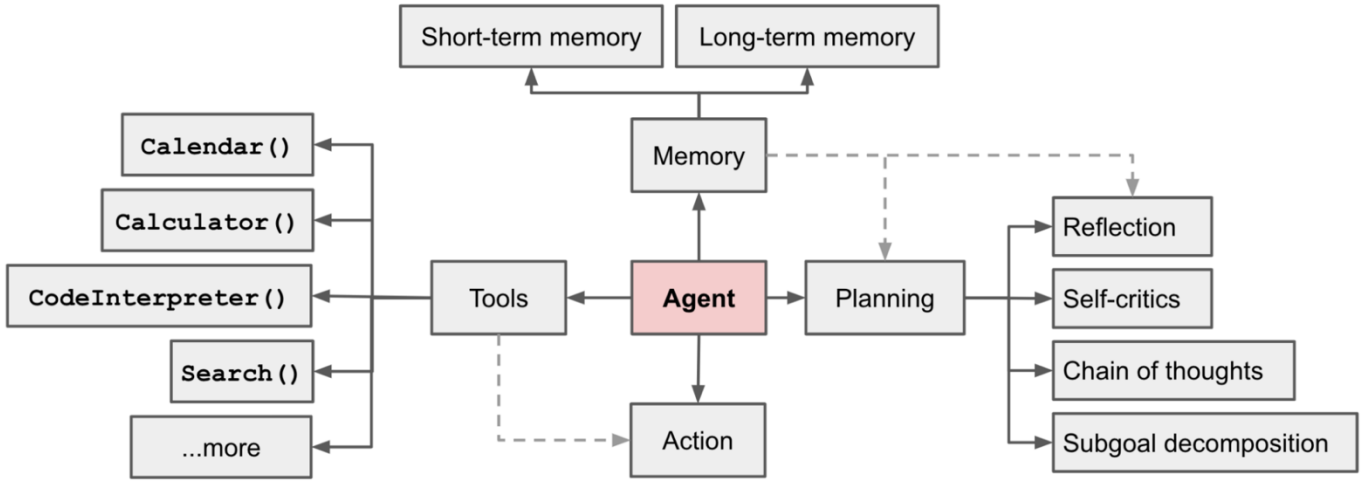
Alita reaches top 1 at GAIA (validation)

Introduction

https://www.youtube.com/watch?v=K27diMbCsuw&t=5s

Introduction



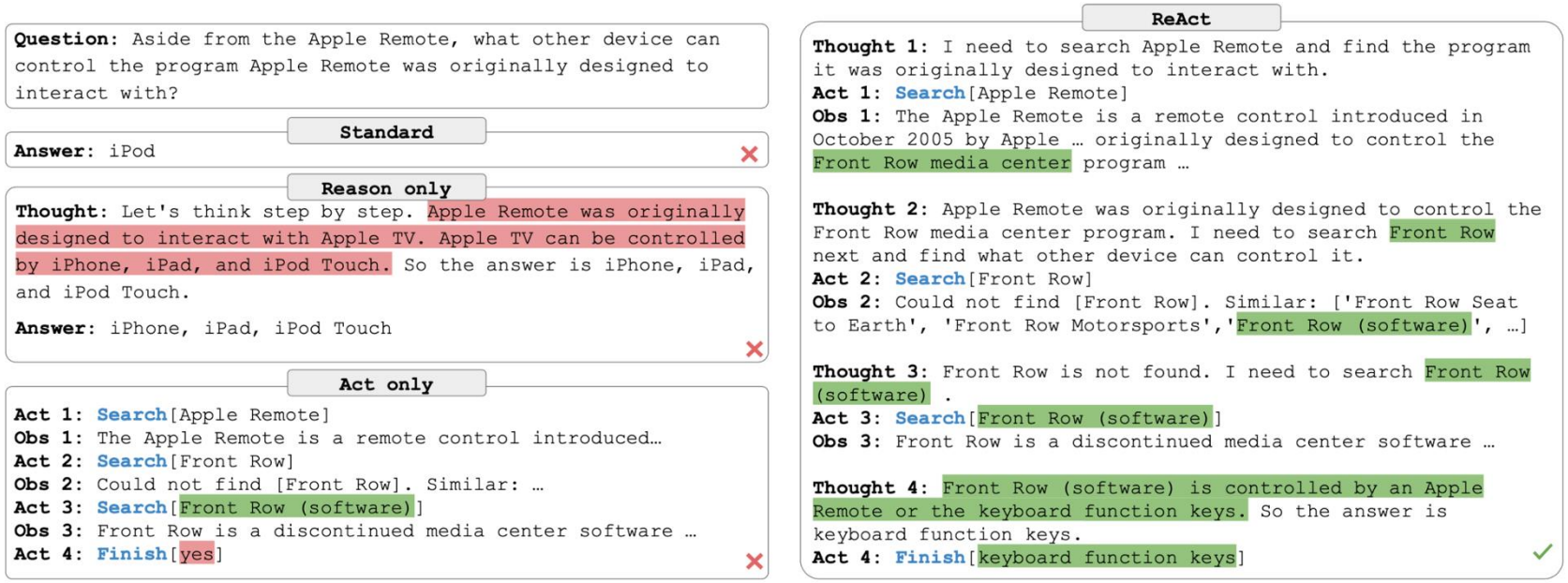


https://react-lm.github.io/

https://lilianweng.github.io/posts/2023-06-23-agent/

Agent = [Reasoning + Acting] * n

Introduction



ReAct in QA

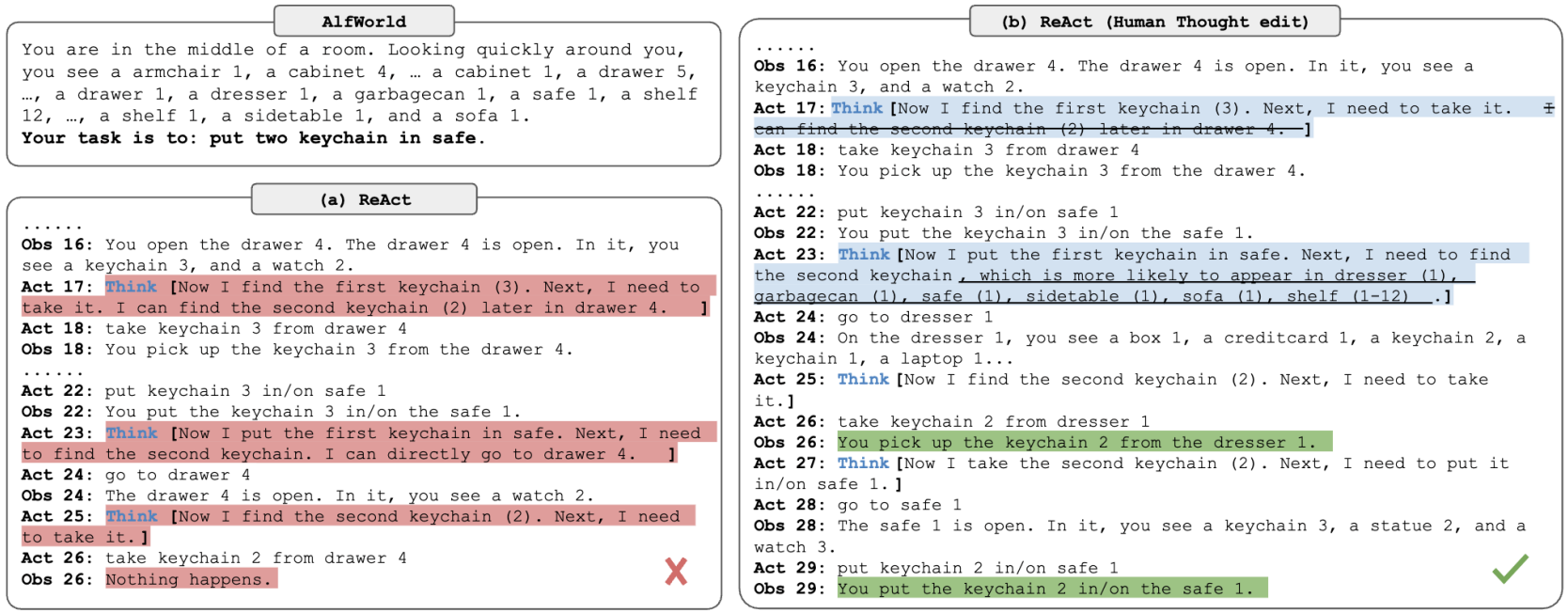
Thought: The model's understanding about the current state

Act: external APIs to acquire external knowledge

Observation: feedback or results returned by external environment

ReAct: Synergizing Reasoning and Acting in Language Models

Introduction



ReAct in Embodied Env

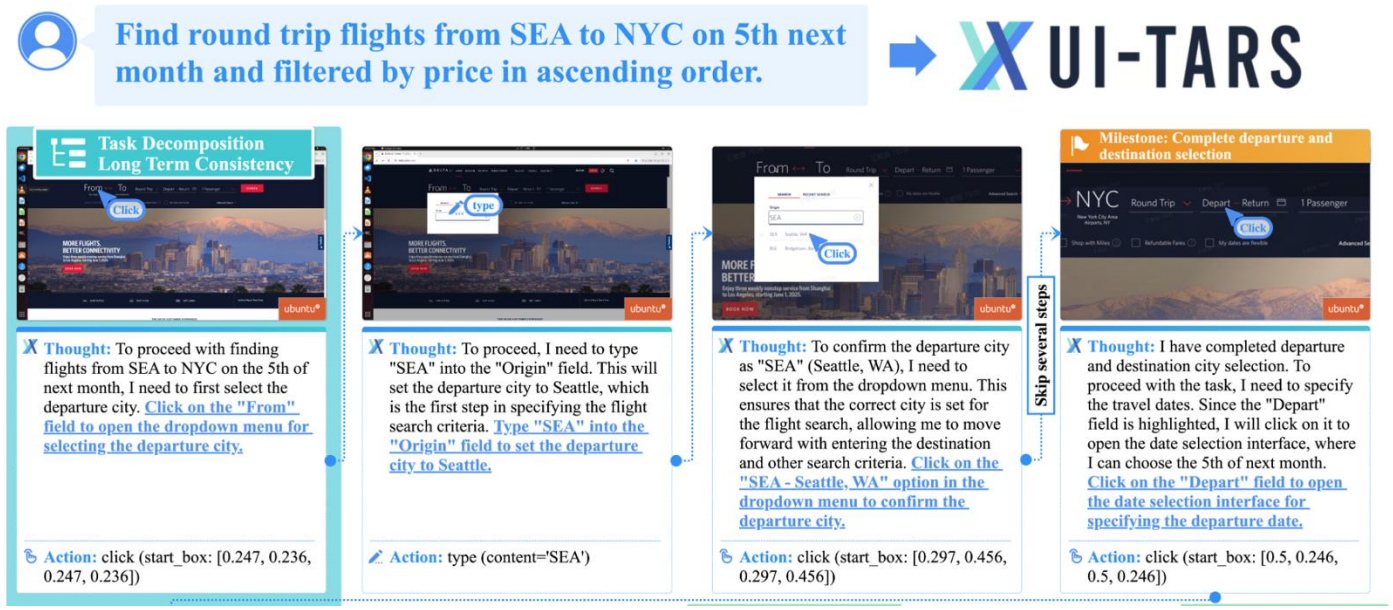
Thought: The model's understanding about the current state

Act: external Actions to acquire external knowledge

Observation: feedback or results returned by external environment

ReAct: Synergizing Reasoning and Acting in Language Models

Introduction



ReAct in Computer-using Env

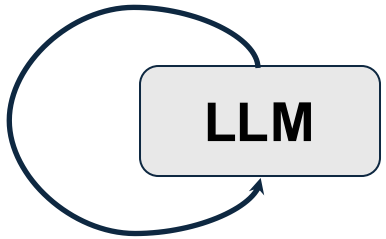
Thought: The model's understanding about the current state

Act: external Actions to acquire external knowledge

Observation: feedback or results returned by external environment, the next page here

Reasoning vs Acting vs Planning

Reasoning



CoT | Decomposition | ... | Backward | Reflection | Alternative

Cognitive mechanism / functions from **Cognitive Science**

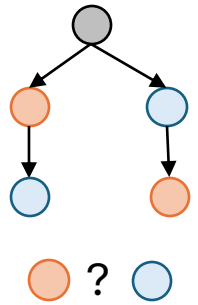
Acting

Models | APIs | Agents | ... | Webs | Retriever | Calculator

Physical Tools / Actions

Planning

Reason or Act



Decision-making Planning

Reasoning vs Acting vs Planning

reasoning == acting 

If reasoning == acting [Yao et al, …]

Thinking, or reasoning, is a **strange** kind of action

--- The Second Half, Shunyu Yao

https://ysymyth.github.io/The-Second-Half/

Reasoning vs Acting vs Planning

reasoning == acting 

If reasoning == acting [Yao et al, …]


```
{
  "name": "think",
  "description": "Use the tool to think about something. It will not obtain new information or change the database, but just append the thought to the log. Use it when complex reasoning or some cache memory is needed.",
  "input_schema": {
    "type": "object",
    "properties": {
      "thought": {
        "type": "string",
        "description": "A thought to think about."
      }
    },
    "required": ["thought"]
  }
}
```

https://www.anthropic.com/engineering/claude-think-tool

Reasoning vs Acting vs Planning

reasoning == acting 

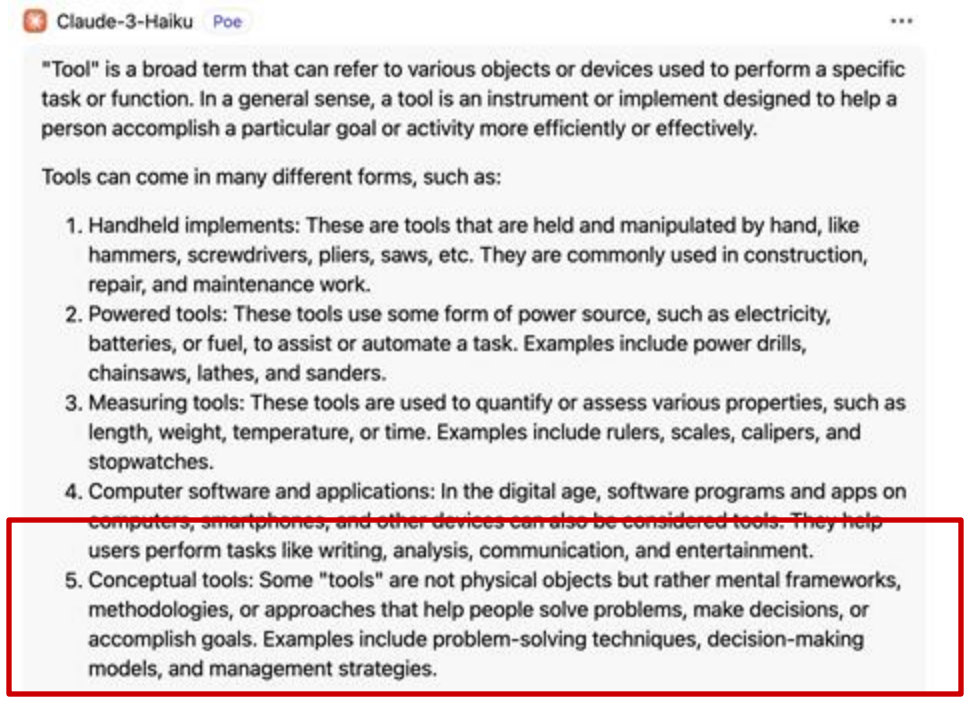
If reasoning == acting [Yao et al, …]

then how should we viem them in an unified manner?

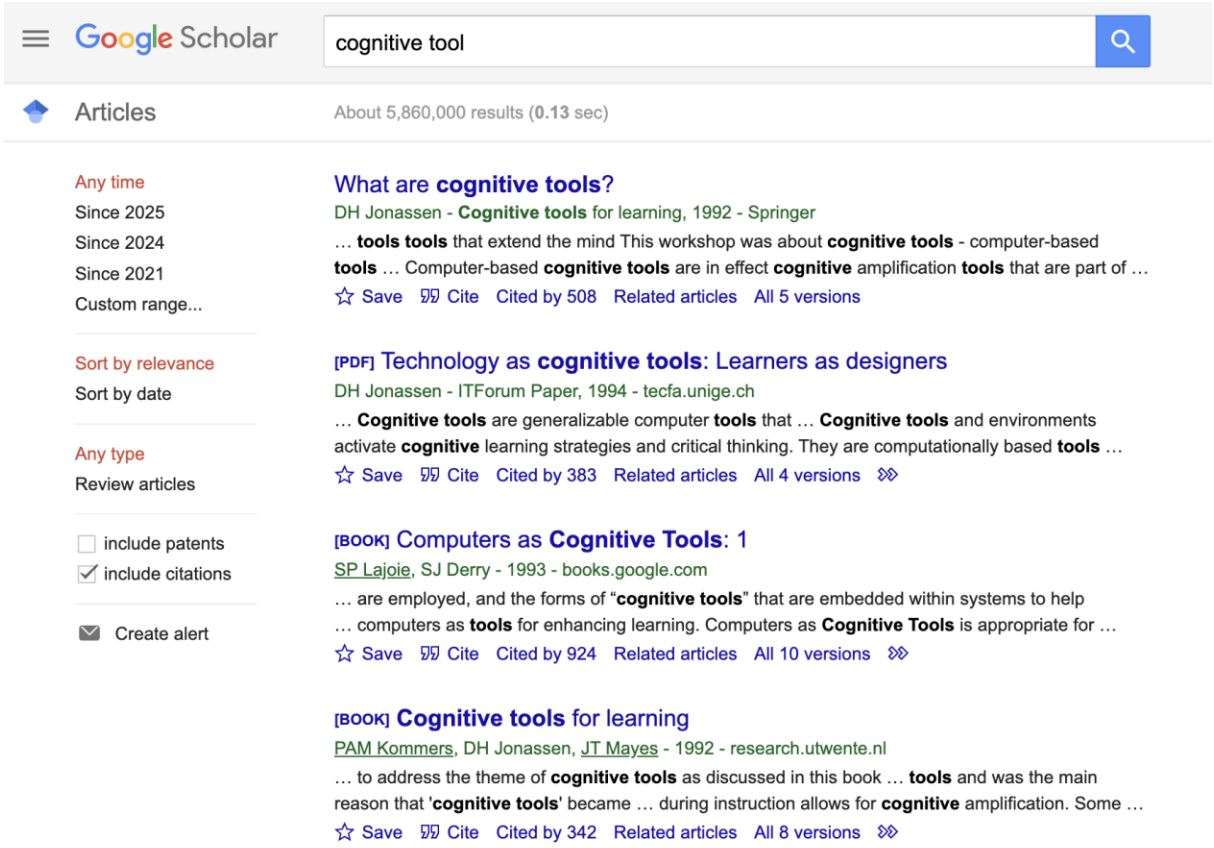
else: **Theory of Agent: Reasoning and Acting are both Tools**

what's real difference between reasoning and acting?

What's tool?



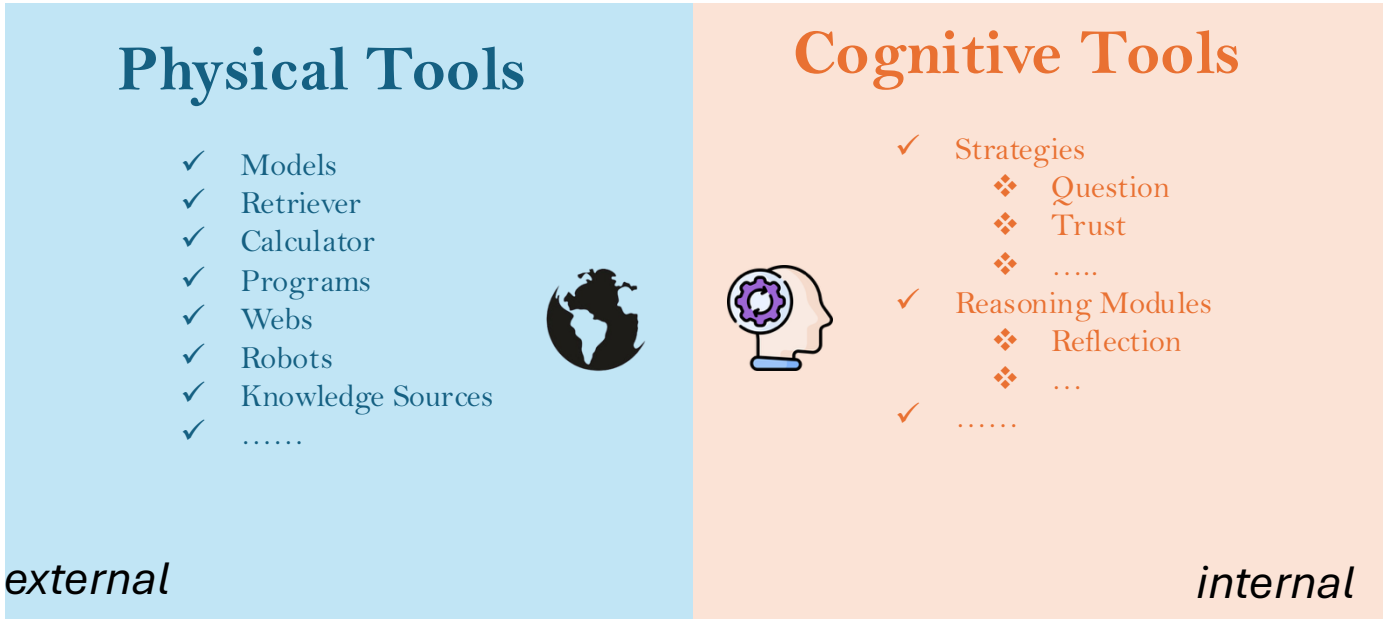
Answer from LLMs

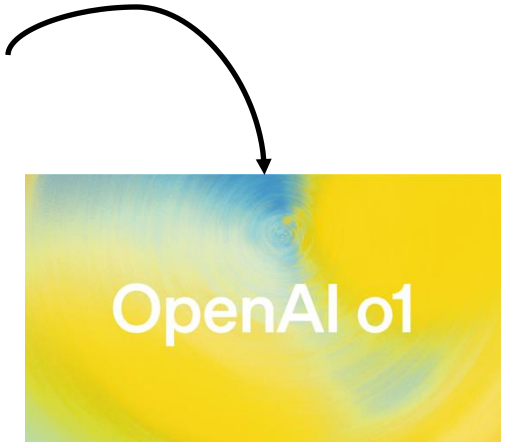


Answer from Scholars

Unification of Reasoning and Acting

Tool is defined as object that can extend an individual's ability to modify features of the surrounding environment or help them accomplish a particular task in general. It can be **internal cognitive/conceptual tools** (i.e., *reasoning*) and **external physical tools** (i.e., *acting*).





Reasoning ~= Acting (in) Tools

Internal cognitive/conceptual tool refer to specifies an internal cognitive mechanisms that aids systematic or investigative thought, to retrieve internal knowledge of agent about current state.

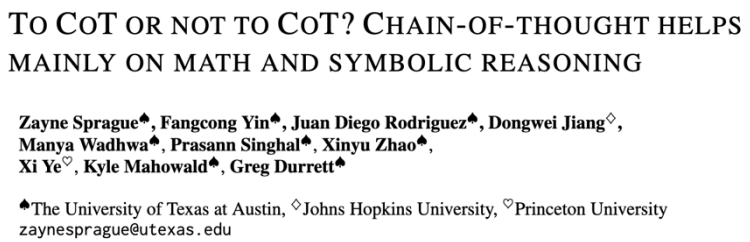
External physical tool refer to external modules that are invoked by a rule or a specific token and whose outputs are incorporated into the context of agent.

Essence of Tool

- **Useful:** A tool must effectively complete one or multiple tasks. It typically receives inputs and produces outputs.
- **On-demand:** A tool must be used as needed, meaning it is invoked based on the current state.

Some Typical Tools

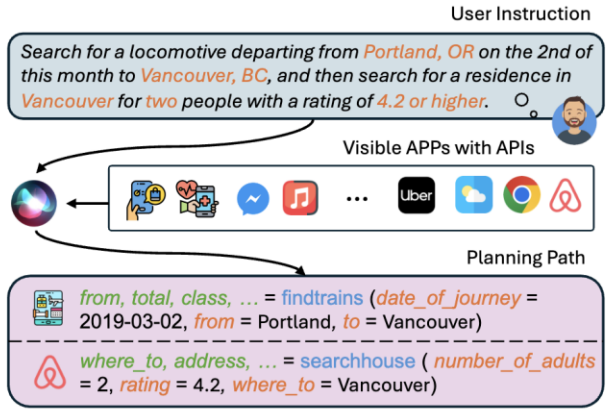
	Useful	On-demand
Chain-of-thoughts (CoT) Reflection Decomposition …	✓	✓
APIs Actions Search Engine **Seek Human Help** …	✓	✓



TO COT OR NOT TO COT? CHAIN-OF-THOUGHT HELPS MAINLY ON MATH AND SYMBOLIC REASONING

Zayne Sprague♠, Fangcong Yin♠, Juan Diego Rodriguez♠, Dongwei Jiang♢, Manya Wadhwa♠, Prasann Singhal♠, Xinyu Zhao♠, Xi Ye♡, Kyle Mahowald♠, Greg Durrett♠

♠The University of Texas at Austin, ♢Johns Hopkins University, ♡Princeton University
zaynesprague@utexas.edu



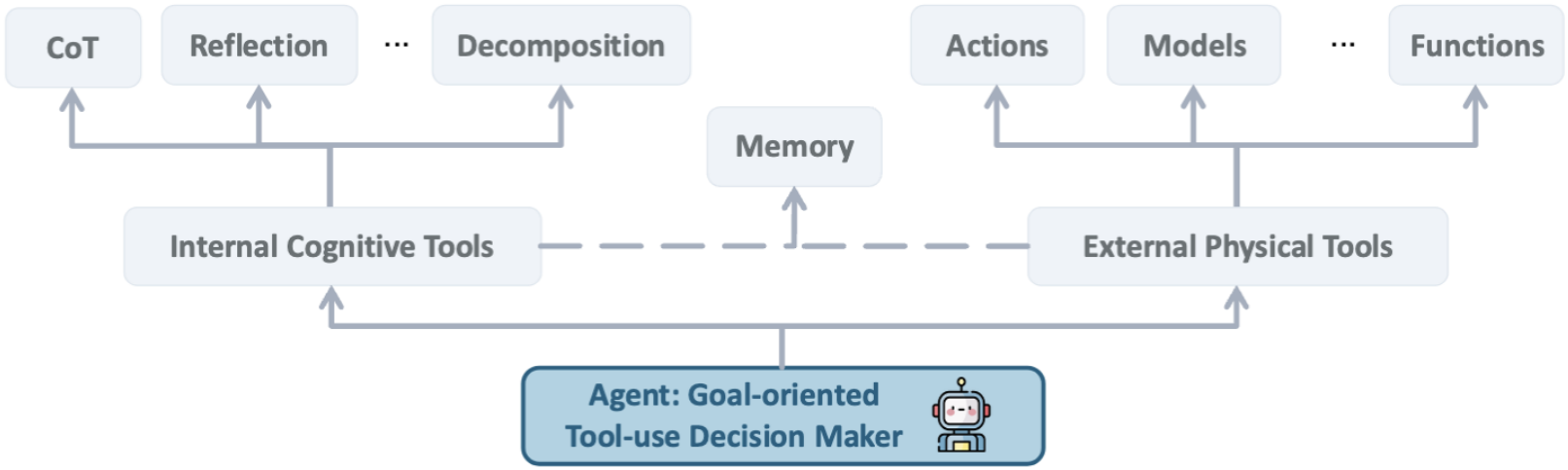
AppBench

These tools effectively **address inherent limitations** of LLMs, such as outdated information, while also **expanding the capabilities to interact with the external environment**.

AppBench: Planning of Multiple APIs from Various APPs for Complex User Instruction

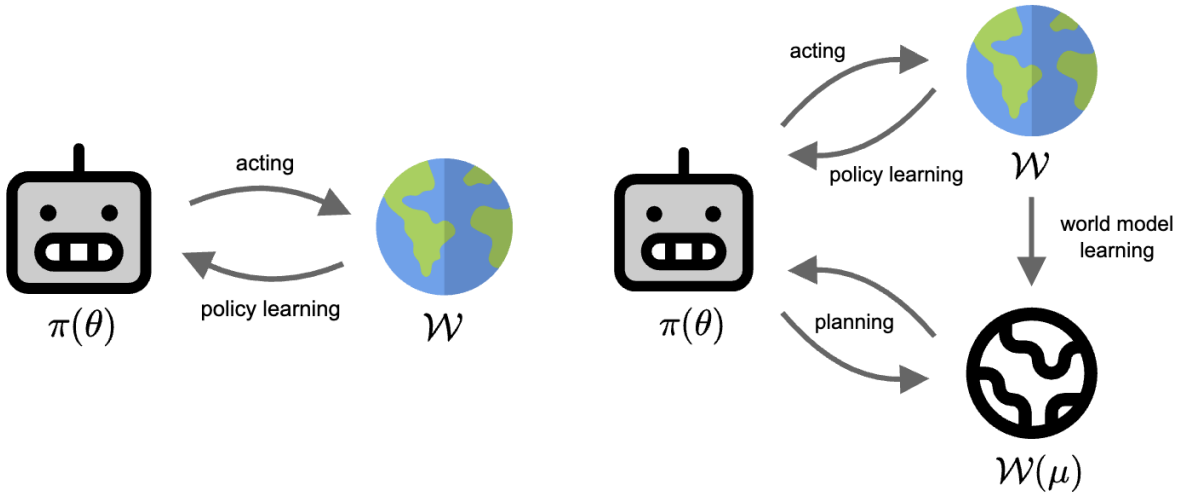
New Agent Definition

- An agent is an entity that coordinates internal cognitive tools (e.g., reflection) and external physical tools (e.g., function callings) to acquire knowledge in order to achieve a specific goal.



New Agent Definition

- An agent is an entity that coordinates internal cognitive tools (e.g., reflection) and external physical tools (e.g., function callings) to acquire knowledge in order to achieve a specific goal.
  - Unified Format: $\tau = (t_1, k_1, t_2, k_2, \ldots, t_n, k_n)$
    - t_n, k_n stands for tool call and returned knowledge at n_{th} step. The tool could be either internal or external.



New Agent Definition

- An agent is an entity that coordinates internal cognitive tools (e.g., reflection) and external physical tools (e.g., function callings) to acquire knowledge in order to achieve a specific goal.

  - Flexible and Robust
    - It degrade to previous ReAct paradigm if we consider the internal tools and internal knowledge as whole reasoning part, then it becomes $(r_1, t_1, k_1, \dots, r_n, t_n, k_n)$ here t_n, k_n only stands for external part.

    - If we solely consider internal tools, it is proved that simply outcome-based reward can trigger various tool utilization such as reflection and decomposition to solve the problem in Large Reasoning Models (i.e., DeepSeek-R1). Alternatively, simply outcome-based reward also trigger various external tool utilization as evidenced in recent studies (i.e., Search-R1, ToRL, OTC-PO).

New Agent Definition

- An agent is an entity that coordinates internal cognitive tools (e.g., reflection) and external physical tools (e.g., function callings) to acquire knowledge in order to achieve a specific goal.
  - Potential Next Scaling Law
    - *Next Tool Prediction*: Just as next-token prediction enables LLMs to learn a compressed representation of the world from text, next-tool prediction allows agents to learn procedural knowledge through interaction.

New Agent Definition

- An agent is an entity that coordinates internal cognitive tools (e.g., reflection) and external physical tools (e.g., function callings) to acquire knowledge in order to achieve a specific goal.

$$\tau = (t_1, k_1, t_2, k_2, \dots, t_n, k_n)$$

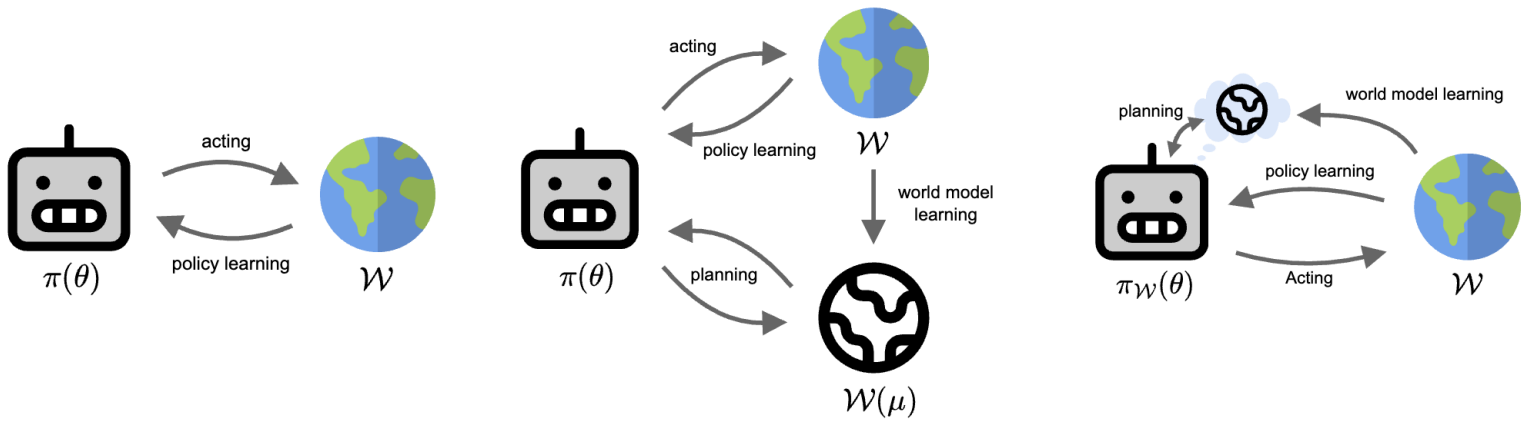
- Next natural question: how to *coordinate these tools?* (Decision-Making Process …)

Internal or External ?

❖ We want the agent call **internal tools when they know certain knowledge**, while only invoke **external tools when they do not know certain knowledge**.

Why?

*"The autonomous machine intelligence is designed to **minimize the number of actions** a system needs to take in the real world to learn a task. It does so by learning a world model that capture as much knowledge about the world as possible without taking actions in the world."* --- Yann Lecun

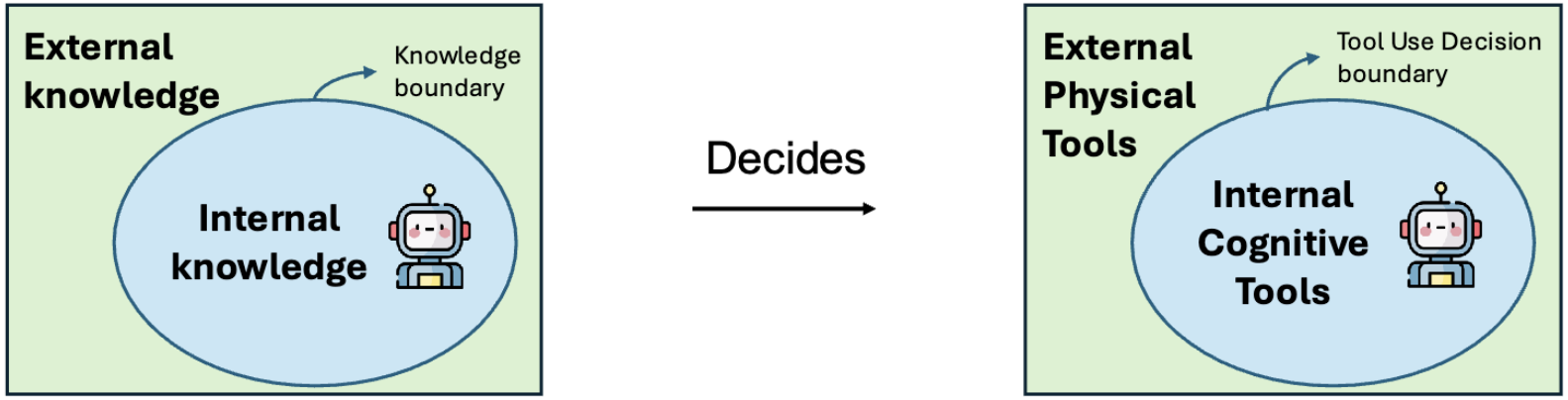


A Path Towards Autonomous Machine Intelligence
DYNA-THINK: Synergizing Reasoning, Acting, and World Model Simulation in AI Agents

Internal or External ?

❖ We want the agent call **internal tools when they know certain knowledge**, while only invoke **external tools when they do not know certain knowledge**.

Optimize *Tool Use Decision Boundary* to match *Knowledge Boundary (知行合一)*



Monitor: Self-aware Knowledge Boundary

Control: Self-aware Tool Utilization

How can we achieve such behavior?

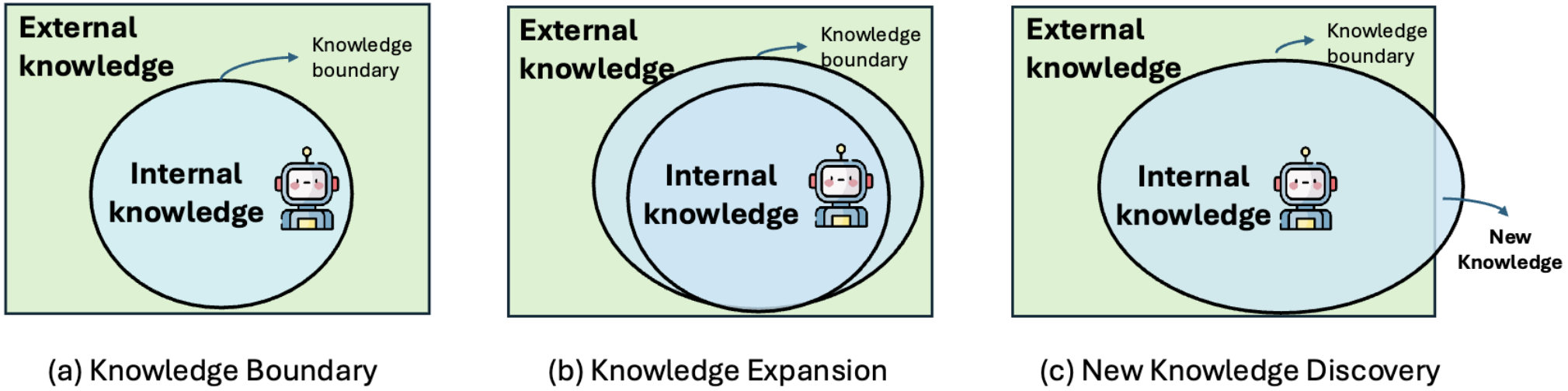
Three key principles of knowledge boundary and decision boundary of agent

- ❖ Principle 1: Foundation
- ❖ Principle 2: Uniqueness and Diversity
- ❖ Principle 3: Dynamic Conservation

Principles 1: Given a LLM, its knowledge boundary is fixed at time *t*.

Lemma 1.1: *Generally*, as time advances, the model's capabilities evolve and the knowledge boundary expands.

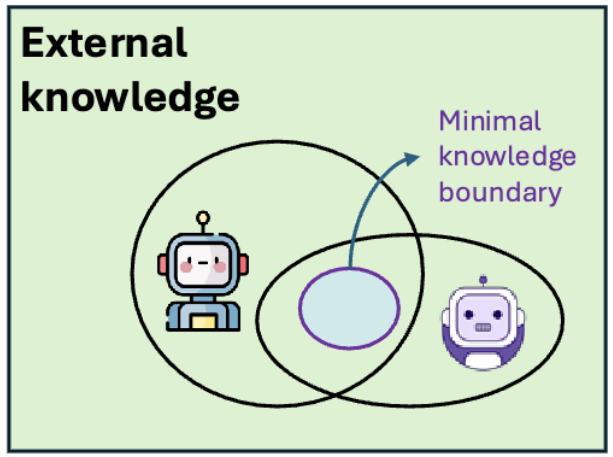
Lemma 1.2: *Specifically*, the knowledge boundaries can be redistributed, e.g., through training, allowing for strengthening in specific domains.

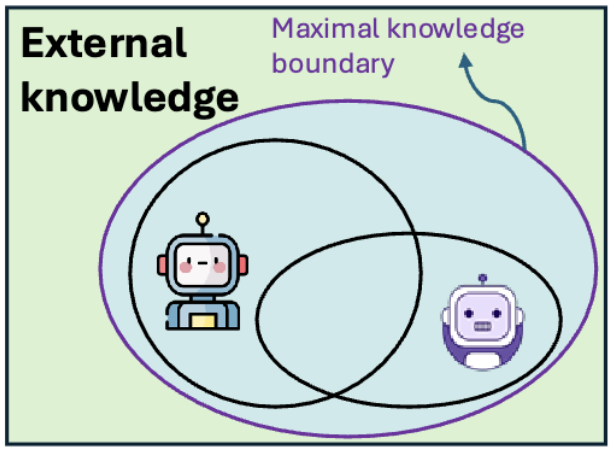


(a) Knowledge Boundary (b) Knowledge Expansion (c) New Knowledge Discovery

Principle 2: Different LLMs have Different Knowledge Boundaries.

- **Lemma 2.1:** Each model has its own knowledge boundary and decision boundary.
- **Lemma 2.2:** There exist minimal and maximal knowledge (and decision) boundaries across *all* models.





Principle 3: Dynamic Conservation of Knowledge

- **Lemma 3.1:** At any time step t, the total world knowledge W_t is fixed and identical across all models.

- **Lemma 3.2:** For any task or query q and model m, there exists a minimal and fixed epistemic effort $N(q, m)$ allocated between internal and external sources, that is necessary to solve the task, such as $N(q, m) = K_{int} + K_{ext}$.

Principle 3: Dynamic Conservation of Knowledge

- **Lemma 3.1:** At any time step t, the total world knowledge W_t is fixed and identical across all models.
- **Lemma 3.2:** For any task or query q and model m, there exists a minimal and fixed epistemic effort $N(q,m)$ allocated between internal and external sources, that is necessary to solve the task, such as $N(q,m) = K_{int} + K_{ext}$.
  - **Task-Model dependency Optimization**: $N(q,m)$ is jointly determined by the complexity of the task and the capabilities of the model.
  - **Capability Equivalence via Dynamic Offloading:** Even models with limited internal capacity can achieve same performance by dynamically offloading reasoning or retrieval steps to more capable tools or agents. There is no difference between 8B ($K_{ext} \rightarrow \mathrm{N}$) and 70B ($K_{int} \rightarrow \mathrm{N}$) from Agent perspective considering models as one of tools.
  - **Agent Objective:** Pursuing the optimal behavior that minimize interactions while managing latency, cost, and constraints, besides the final correctness.

A Roadmap to Autonomous Agent

- **Agentic Pretraining**: Next tool prediction, As research trends toward unified agent architectures, modeling all forms of interaction (API calls, UI navigation, or environment manipulation) as structured, learnable outputs opens the door to a new kind of scaling law: one that governs knowledge acquisition, not just compression.
  - Unified Format: $\tau = (t_1, k_1, t_2, k_2, \ldots, t_n, k_n)$
  - Data Collection: It is extremely challenging to collect massive pretraining interaction corpus.

A Roadmap to Autonomous Agent

- **Agentic Pretraining**: Next tool prediction, As research trends toward unified agent architectures, modeling all forms of interaction (API calls, UI navigation, or environment manipulation) as structured, learnable outputs opens the door to a new kind of scaling law: one that governs knowledge acquisition, not just compression.

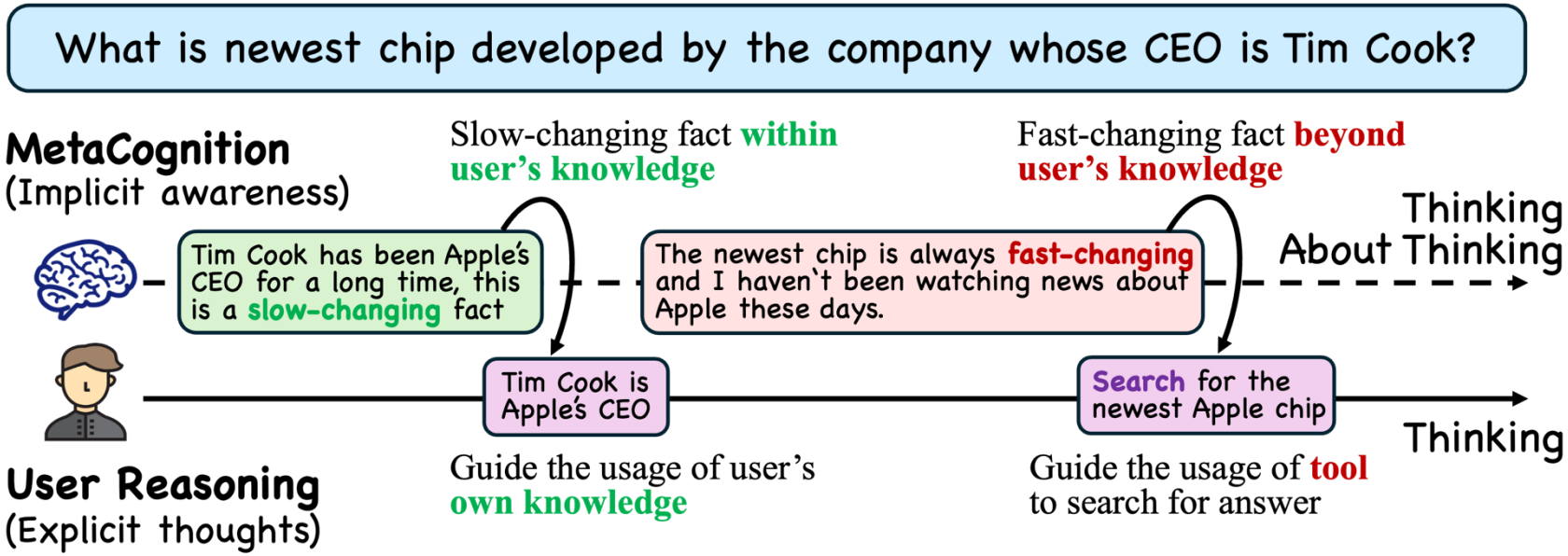
- **Agentic Supervised-finetuning:** It is important to collect model-task-specific trajectories instead of collecting one trajectory for all models due to lemma 2.1. Additionally, it is more effective to leverage the lemma 2.2 by utilizing maximal knowledge boundary to build one-fits-all dataset.

- **Agent Reinforcement Learning:** Reinforcement learning (RL) offers a more promising path for aligning a model's decision-making with its own knowledge boundary, as agents can learn from experience how to adaptively use tools. The key challenge lies in designing reward functions that go beyond correctness

- **Agent Prompting:** Once the model is trained, previous numerous studies utilize prompt engineering to develop task-specific agentic workflows across various domains. Despite achieving exceptional performance on complex tasks, few of these approaches rigorously evaluate behavioral optimality, such as internal cognitive tool overuse (i.e., overthinking) or external physical tool overuse (i.e., overacting).

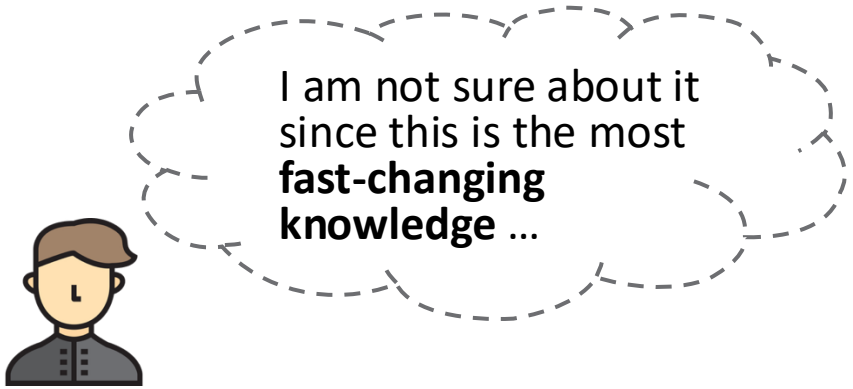
Agentic SFT -- SMART

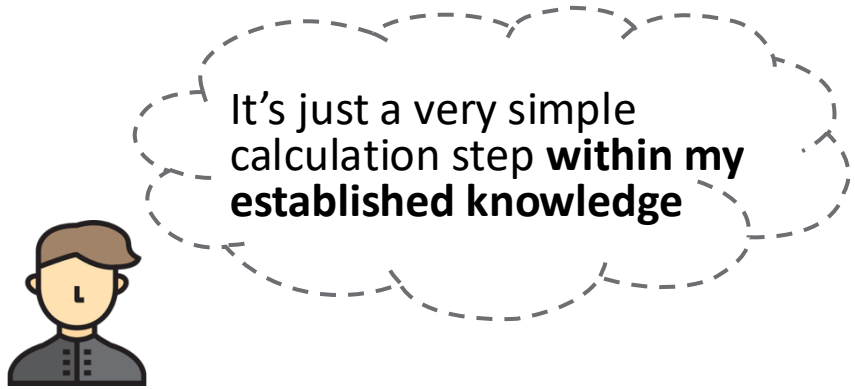
- **Metacognition** in human:
  - People often rely on intuitive feelings of certainty or uncertainty as heuristic cues to guide their meta-reasoning decisions
  - Simply: Thinking about how to “think”



SMART-Enhanced Reasoning

- Calibration of metacognition needs training on model's awareness of its **knowledge boundary**
  - Reasoning chain should integrate *what model knows* and *what it is generally not good at*





SMART-Enhanced Reasoning

- We adapt three established dataset to create the reasoning chain:
  - Math: ***simple arithmetic*** v.s. **challenging calculation**
    (Adapted from MATH)
  - Intention: ***commonsense*** v.s. **user specific intentions**
    (Adapted from Intention-in-Interaction)
  - Time: ***never-changing facts*** v.s. **fast-changing facts**
    (Adapted from FreshQA)

Code

AskUser

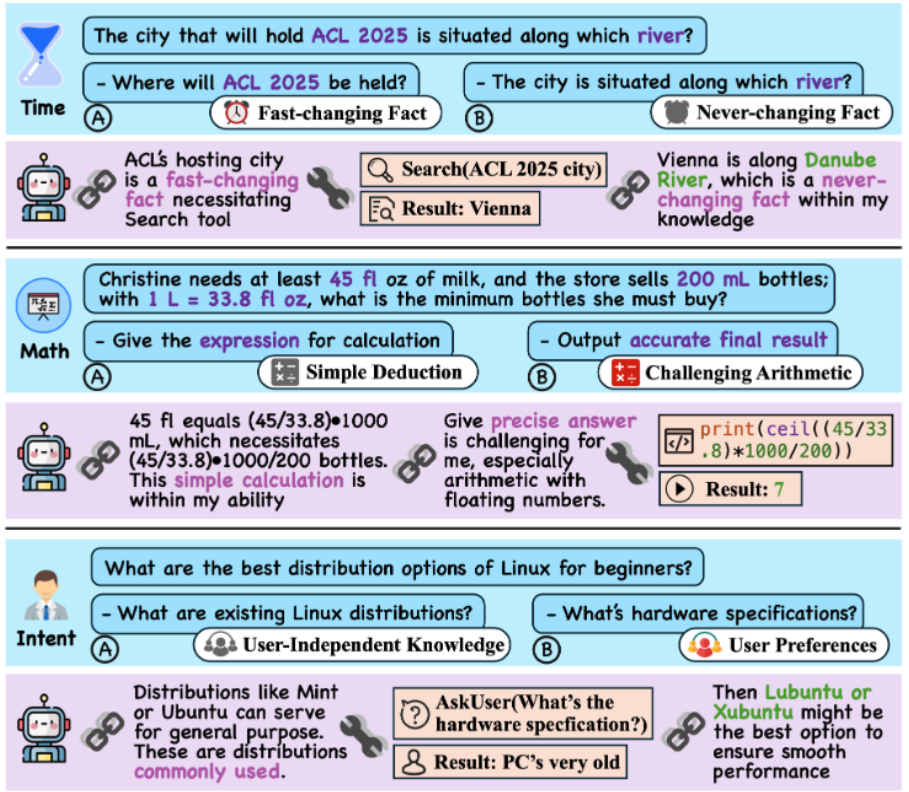
Search

SMART-Enhanced Reasoning

- Collect the dataset in the following format, where external tools are only invoked when the knowledge is unknown (i.e., challenging calculation, user specific intentions, fast-changing facts):

$$(t_1, k_1, t_2, k_2, \ldots, t_n, k_n)$$

- With SMART-ER, we train **SMARTAgent** that could perform smarter tool use, only use tools when necessary, but still achieves higher performance



SMART-Enhanced Reasoning

- SMARTAgent achieves **higher accuracy** with **lower tool call number** and **higher confidence in decision**, thus *mitigating tool overuse*

Method	Model	Math (MATH)		Time (FreshQA)		Intention (Intention-in-Interaction)		
		Tool Used↓ (*Times*)	Accuracy↑ (%)	Tool Used↓ (*Times*)	Accuracy↑ (%)	Tool Used↓ (*Times*)	Missing Details Recovery↑ (Lv3 / Lv2, %)	Summarized Intention Coverage↑ (%)
Open-Source								
Normal Reasoning Trained	*Mistral-7B*	0.00	17.00	0.00	48.00	0.00	41.86 / 43.84	-
	Llama-3.1-8B	0.00	41.00	0.00	48.00	0.00	38.37 / 42.49	-
Base Model Reasoning Prompt	*Mistral-7B*	0.00	17.25	0.00	29.00	0.00	37.21 / 33.06	-
	Llama-3.1-8B	0.00	53.00	0.00	26.00	0.00	40.70 / 25.76	-
	Mistral-Nemo(12B)	0.00	47.00	0.00	33.00	0.00	44.19 / 28.37	-
	Mistral-Small(24B)	0.00	72.25	0.00	34.00	0.00	41.86 / 31.82	-
	Llama-3.1-70B	0.00	70.00	0.00	36.00	0.00	41.86 / 29.24	-
Base Model Tool Prompt	*Mistral-7B*	3.90	13.25	1.67	49.00	3.80	48.84 / 21.70	63.04
	Llama-3.1-8B	1.93	51.00	2.05	56.00	3.77	54.76 / 25.90	70.20
	Mistral-Nemo(12B)	2.35	46.00	1.19	59.00	1.80	31.35 / 5.82	59.27
	Mistral-Small(24B)	1.55	76.00	1.73	62.00	2.52	45.74 / 33.62	78.20
	Llama-3.1-70B	3.53	67.50	2.08	63.00	2.71	45.74 / 35.96	61.68
SMARTAgent	*Mistral-7B*	0.60$_{\downarrow 3.30}$	22.75$_{\uparrow 5.50}$	1.00$_{\downarrow 0.67}$	64.00$_{\uparrow 15.00}$	3.60$_{\downarrow 0.20}$	74.42$_{\uparrow 25.58}$ / 65.44$_{\uparrow 21.60}$	81.76$_{\uparrow 18.72}$
	Llama-3.1-8B	0.88$_{\downarrow 1.05}$	54.75$_{\uparrow 1.75}$	1.05$_{\downarrow 1.00}$	67.00$_{\uparrow 11.00}$	3.80$_{\uparrow 0.03}$	**81.40**$_{\uparrow 26.64}$ / 67.41$_{\uparrow 24.92}$	78.28$_{\uparrow 8.08}$
	Mistral-Nemo(12B)	0.82$_{\downarrow 1.53}$	49.50$_{\uparrow 2.50}$	1.00$_{\downarrow 0.19}$	**70.00**$_{\uparrow 11.00}$	3.34$_{\uparrow 1.54}$	77.91$_{\uparrow 33.72}$ / 62.15$_{\uparrow 33.78}$	82.30$_{\uparrow 23.03}$
	Mistral-Small(24B)	0.79$_{\downarrow 0.76}$	69.75$_{\downarrow 6.25}$	1.00$_{\downarrow 0.73}$	66.00$_{\uparrow 4.00}$	3.89$_{\uparrow 1.37}$	74.42$_{\uparrow 28.68}$ / **68.87**$_{\uparrow 35.25}$	84.99$_{\uparrow 6.79}$
	Llama-3.1-70B	0.94$_{\downarrow 2.59}$	**72.50**$_{\uparrow 2.50}$	1.01$_{\downarrow 1.07}$	66.00$_{\uparrow 3.00}$	3.51$_{\uparrow 0.80}$	68.60$_{\uparrow 22.86}$ / 58.15$_{\uparrow 22.19}$	**86.09**$_{\uparrow 24.41}$
Tool Used Macro-Average Decrease (%)		**24.00**				Performance Macro-Average Increase (%)		**37.10**
Closed-Source								
Base Model Reasoning Prompt	*GPT-4o-mini*	0.00	73.00	0.00	44.00	0.00	45.35 / **32.41**	-
	GPT-4o	0.00	**79.50**	0.00	47.00	0.00	38.37 / 28.54	-
Base Model Tool Prompt	*GPT-4o-mini*	2.55	54.50	1.06	56.00	1.91	**50.00** / 26.90	76.44
	GPT-4o	0.27	79.25	1.01	**65.00**	1.17	40.70 / 15.61	**86.80**

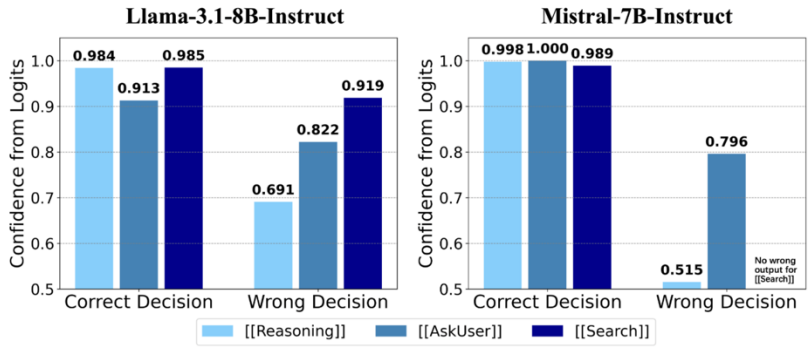
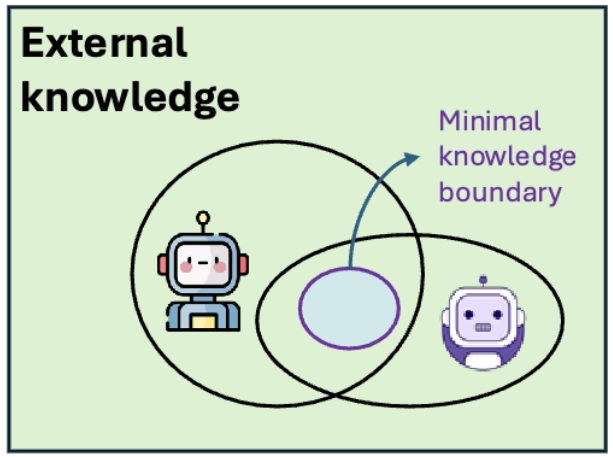
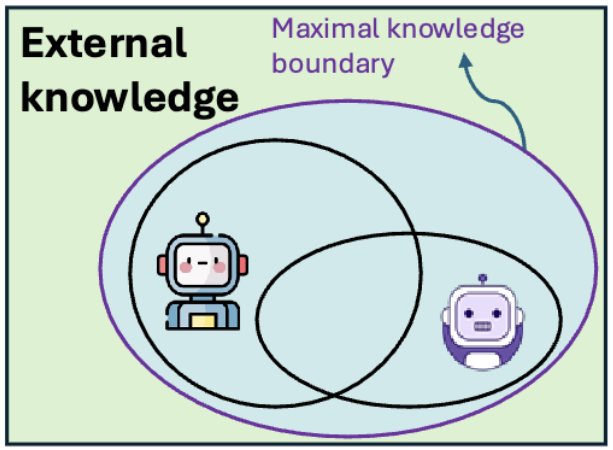


Figure 5: Confidence analysis shows that **SMART** effectively enhances the model's decision-making confidence in selecting the correct reasoning approaches.

What's SMART Anyway?

- Each LLM has different knowledge boundary
- SMART-ER *ensures* certain knowledge is **what all LLMs do not know**
- This **One-fit-for-all strategy** is approximating **Maximal Knowledge Boundary (lemma 2.2)**





Problems of SMART

- Different error still exists
- Tool overuse is not fully mitigated due to coarse-grained approximation
- **Limited Generalization**

Error Type (Explanation)	**Case / Model Action**	**Wrong Reason**	**Common Seen**
Repetitive Tool Calls Uses the same query to call the tool for multiple times.	**Last Call:** Search(current richest person) **Reasoning:** several people are mentioned instead of one richest, search again… **Tool Call:** Search(current richest person)	The model fails to extract the most useful information and instead relies on repetitive calls.	*Domain*: Time Tool Prompt
Ignorance of Feedback Overlooks tool feedback and fails to correct erroneous behavior.	**Last Output:** Error! Traceback: function 'ceil' not found **Tool Call:** Code(```print(ceil(45/33.8•5))```)	The error persists due to the absence of 'from math import ceil,' causing an incorrect call.	*Domain*: Math Tool Prompt, SMARTAgent
Tool Calls on Simple Subgoal Invokes tool calls for subgoals that are considered trivial by the user.	**Reasoning:** I need to use code to ensure the accuracy of my calculation. **Tool Call:** Code(```print(30•40/2)```)	Still using tool calls on simple calculation to ensure accuracy.	*Domain*: Math Tool Prompt, SMARTAgent
Inaccurate Tool Call Arguments Employs imprecise arguments that causes deviations in the solution chain.	**Query:** Find the next music festival happening in my city. **Tool Call:** AskUser(what's your favorite music)	Ask about not-related trivial details instead of where the city is, date or time frame, etc.	*Domain*: Intention Tool Prompt

Agentic RL – OTC-PO

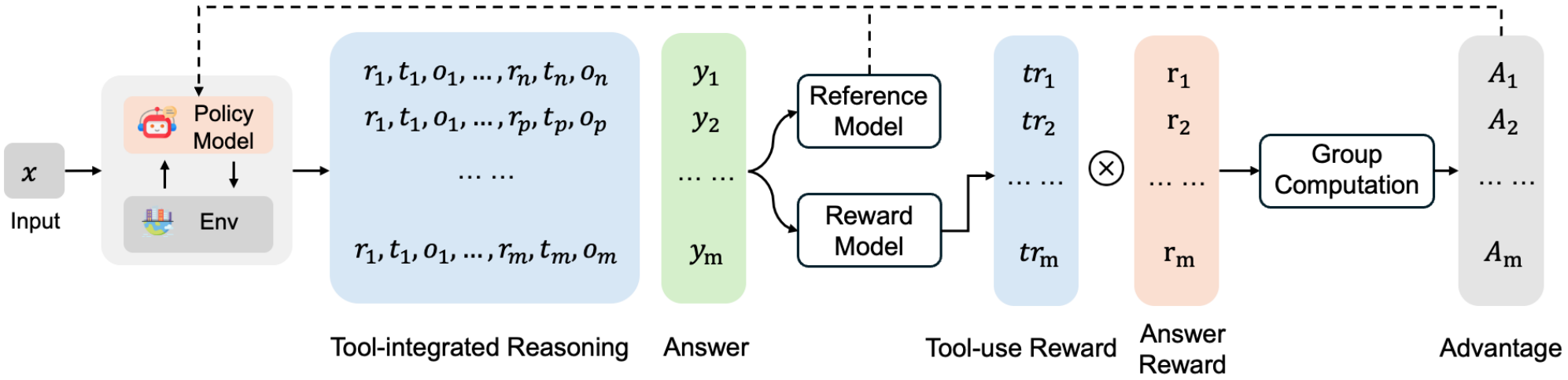
Can we effectively align an agent's tool use boundary to its knowledge boundary via RL, so that smarter tool use could be achieved from experience?

Agentic RL – OTC-PO

We start from one fundamental assumption that given one problem and one LLM, there exist an **optimal number of external tools required**, defined as **minimal number** of tool calls to solve the problem correctly.

Solution: add tool-use reward as a **coefficient** of (outcome reward + format reward)

Why tool-use reward? → Tool overuse and underuse brings serious efficiency issues, especially considering the cost of various tool calls in terms of time, money and computation.



Acting Less is Reasoning More! Teaching Model to Act Efficiently

Agentic RL – OTC-PO

- We are **the first** to define this problem as follows: Here is a tool-integrated reasoning trajectory:

$$\tau_k = (r_0, tc_0, o_0), (r_1, tc_1, o_1), ...(r_k, tc_k, o_k),$$

where r_i, tc_i, o_i denotes the reasoning, tool call and returned observation respectively. The objective of task is to provide the correct answer with minimal cost of tools given the question q and model M.

$$\arg\min_{\tau} \ \text{Cost}(\tau) \quad \text{subject to} \quad \mathcal{M}(q, \tau) = \hat{a},$$

- We are **the first** to define tool productivity (TP) as the fraction between benefits and cost.

$$\text{TP} = \frac{\sum_{i=1}^{N} \mathbb{I}\{y_i = \hat{y}_i\}}{\sum_{i=1}^{N} tc_i}$$

where I is the indicator function which equals 1 if the generated answer is the ground truth answer.

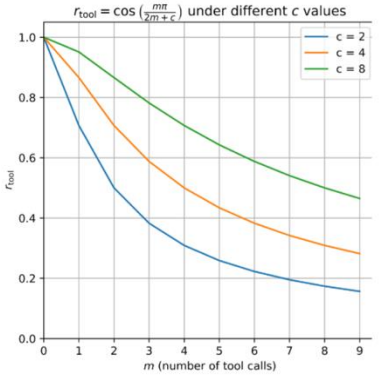
Reward Design -- OTC-PO

- **OTC-PPO**

$$r_{tool} = \cos\left(\frac{m * \pi}{2m + c}\right)$$

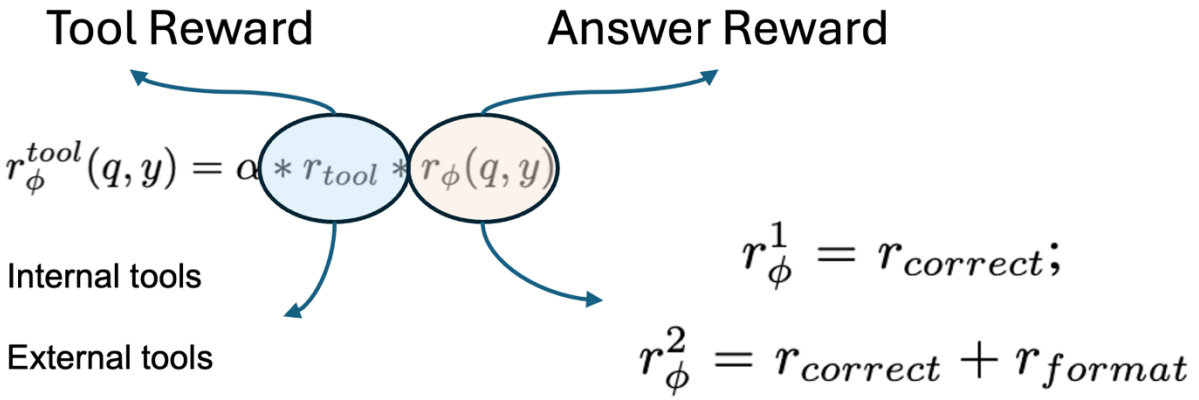
- **OTC-GRPO**

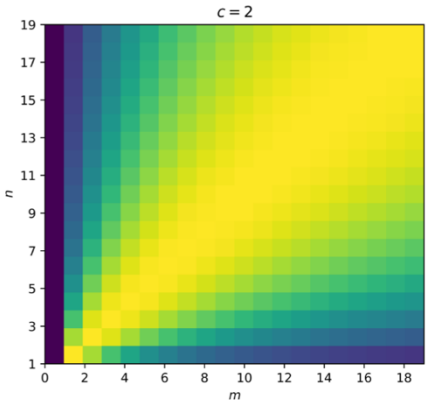
$$r_{tool} = \begin{cases} 1 & \text{if } f(m,n) = n = 0 \\ \cos\left(\frac{m*\pi}{2m+c}\right) & \text{if } n = 0 \\ \sin\left(\frac{f(m,n)*\pi}{2n}\right) & \text{otherwise} \end{cases} \qquad f(m,n) = \begin{cases} 0, & \text{if } m = 0 \text{ and } n = 0 \\ m, & \text{if } n = 0 \\ \frac{2nm}{m+n}, & \text{otherwise} \end{cases}$$



OTC-PPO

- **Unified Tool-integrated Reward Function**





OTC-GRPO

Acting Less is Reasoning More! Teaching Model to Act Efficiently

Agentic RL – OTC-PO

Models	NQ			HotpotQA		
	EM (↑)	TC (↓)	TP (↑)	EM (↑)	TC (↓)	TP (↑)
Qwen2.5-3B(-Base)						
R1-Base	0.226	-	-	0.201	-	-
SFT	0.249	-	-	0.186	-	-
RAG	0.348	1.0	0.348	0.255	1.0	0.255
IRCoT	0.111	10.0	0.011	0.164	10.0	0.016
Search-R1-PPO	0.403	1.738	0.232	0.279	1.716	0.163
Search-R1-GRPO	0.404	1.426	0.283	0.312	1.802	0.173
OTC-PPO	0.355	1.010 (▼ 41.9%)	0.351 (▲ 51.3%)	0.260	1.026 (▼ 40.2%)	0.253 (▲ 55.2%)
OTC-GRPO	0.444	1.008 (▼ 29.3%)	0.440 (▲ 55.5%)	0.365	1.387 (▼ 23.0%)	0.263 (▲ 52.0%)
Qwen2.5-7B(-Base)						
R1-Base	0.270	-	-	0.242	-	-
SFT	0.318	-	-	0.217	-	-
RAG	0.349	1.0	0.349	0.299	1.0	0.299
IRCoT	0.224	9.999	0.022	0.133	9.982	0.013
Search-R1-PPO	0.449	3.282	0.136	0.380	3.741	0.102
Search-R1-GRPO	0.399	1.697	0.235	0.341	2.109	0.162
OTC-PPO	0.446	1.040 (▼ 68.3%)	0.429 (▲ 215.4%)	0.383	1.464 (▼ 60.9%)	0.262 (▲ 156.9%)
OTC-GRPO	0.444	0.990 (▼ 41.7%)	0.448 (▲ 90.6%)	0.366	1.005 (▼ 52.3%)	0.364 (▲ 124.7%)

Models	AIME24			AIME25		
	EM (↑)	TC (↓)	TP (↑)	EM (↑)	TC (↓)	TP (↑)
Qwen2.5-Math-1.5B(-Base)						
Qwen2.5-Math-1.5B-Ins	10.0	-	-	10.0	-	-
Qwen2.5-Math-1.5B-Ins-TIR	13.3	1.1	12.1	13.3	1.4	9.5
ToRL-GRPO	23.3	2.2	10.6	23.3	2.3	10.1
OTC-GRPO	20.0	1.1 (▼ 50.0%)	18.2 (▲ 71.7%)	20.0	1.1 (▼ 41.2%)	18.2 (▲ 80.2%)
Qwen2.5-Math-7B(-Base)						
Qwen2.5-Math-7B-Ins	10.0	-	-	16.7	-	-
Qwen2.5-Math-7B-Ins-TIR	26.7	1.6	16.4	16.7	1.4	12.2
Base-RL	33.3	-	-	6.7	-	-
ToRL-GRPO	36.7	2.1	17.5	26.7	2.1	12.7
OTC-GRPO	36.7	0.7 (▼ 66.7%)	52.4 (▲ 199.4%)	23.3	0.8 (▼ 61.9%)	29.1 (▲ 129.1%)

- Simply optimize answer correctness via RL bring serious over-optimization problem such as **tool overuse issue**.
- OTC-PO significantly **reduce the tool calls and improve the tool productivity**, especially when the size of model increases.

Acting Less is Reasoning More! Teaching Model to Act Efficiently

Agentic RL – OTC-PO

Question: Are Giuseppe Verdi and Ambroise Thomas both Opera composers ?
Answer: Yes

Search-R1-PPO: <think> I need to determine if Giuseppe Verdi and Ambroise Thomas are both opera composers. </think>
<search> Giuseppe Verdi ***</search>***
<observation> ... </observation>
<search> Ambroise Thomas ***</search>***
<observation> ... </observation>
<search> Are Giuseppe Verdi and Ambroise Thomas both Opera composers? ***</search>***
<observation> ... </observation>
<think> Based on the observations, both Giuseppe Verdi and Ambroise Thomas are described as composers, specifically opera composers. </think>
<answer> yes </answer> **(Tool Calls = 3)**

Undesirable tool use behavior

Cognitive Offloading *fosters dependency on external tools, causes memory capabilities to diminish, and inhibits the development of internal reasoning..*

Search-R1-GRPO: <think> To answer this question, I need to check if both Giuseppe Verdi and Ambroise Thomas are composers of opera. I don\'t have immediate access to this knowledge, but I can conduct a search to find out. </think>
<search> who are Giuseppe Verdi and Ambroise Thomas and their works ***</search>***
<observation>…</observation>
<think>From the information gathered, … This confirms Thomas as an opera composer. Next, I will search on Giuseppe Verdi. </think>
<search> who is Giuseppe Verdi ***</search>***
<observation>…</observation>
<think>Giuseppe Verdi is named as an Italian opera composer of prominent stature. This confirms Verdi as an opera composer. Answering the initial question: Question: Are Giuseppe Verdi and Ambroise Thomas both Opera composers? </think>
<answer> yes </answer> **(Tool Calls = 2)**

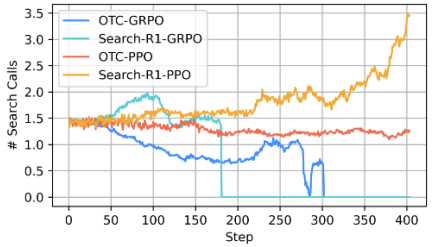
Undesirable tool use behavior

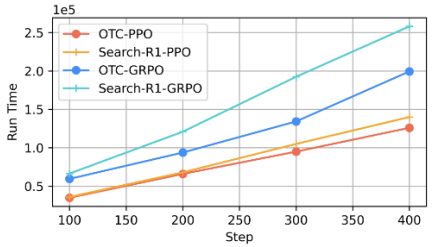
OTC-PPO: <think> First, I need to determine if Giuseppe Verdi and Ambroise Thomas are both opera composers. </think>
<search> Giuseppe Verdi and Ambroise Thomas both Opera composers ***</search>***
<observation> ... </observation>
<think> Based on the search results, Giuseppe Verdi and Ambroise Thomas are both known as opera composers. </think>
<answer> yes </answer> **(Tool Calls = 1)**

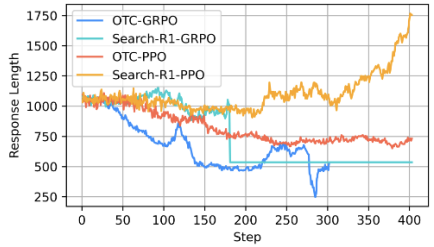
OTC-GRPO: To answer the question of whether Giuseppe Verdi and Ambroise Thomas are both Opera composers, let's first reason through the information:
1. Giuseppe Verdi is a well-known Italian opera composer. He is famous for his numerous operas, including "Rigoletto," "La Traviata," and "Aida."
2. Ambroise Thomas is a French opera composer. He is known for his operas such as "Mignon" and "Hamlet."
Based on this reasoning, both Giuseppe Verdi and Ambroise Thomas are indeed Opera composers.
<answer> yes </answer> **(Tool Calls = 0)**

- **Cognitive offloading** appears more often in larger LLMs.
- **(Minimizing external tool calls = maximizing internal reasoning) = Smart Agent.**

Agentic RL – OTC-PO







Models	TriviaQA		PopQA		2Wiki		Musique		Bamboogle	
	EM (↑)	TC (↓)	EM (↑)	TC (↓)	EM (↑)	TC (↓)	EM (↑)	TC (↓)	EM (↑)	TC (↓)
Qwen2.5-3B(-Base)										
Search-R1-PPO	0.566	1.580	0.425	1.631	0.258	1.675	0.051	1.922	0.063	1.766
Search-R1-GRPO	0.587	1.455	0.345	1.542	0.257	1.991	0.084	2.263	0.203	1.859
OTC-PPO	0.551	**1.008**	0.409	**1.009**	0.235	**1.050**	0.045	1.051	0.063	**1.016**
OTC-GRPO	**0.608**	1.046	**0.441**	1.030	**0.341**	1.561	**0.124**	1.734	**0.266**	1.547
Qwen2.5-7B(-Base)										
Search-R1-PPO	0.596	3.353	0.420	3.315	0.326	4.116	0.135	4.294	0.375	3.641
Search-R1-GRPO	0.578	1.704	0.411	1.754	0.340	2.521	0.130	2.616	0.203	1.859
OTC-PPO	**0.623**	1.066	0.425	1.083	**0.363**	1.868	**0.152**	1.942	**0.391**	1.828
OTC-GRPO	0.597	**0.430**	**0.431**	**0.739**	0.311	**0.938**	0.130	**1.224**	0.250	**0.781**

Simple

Faster

Generalizable

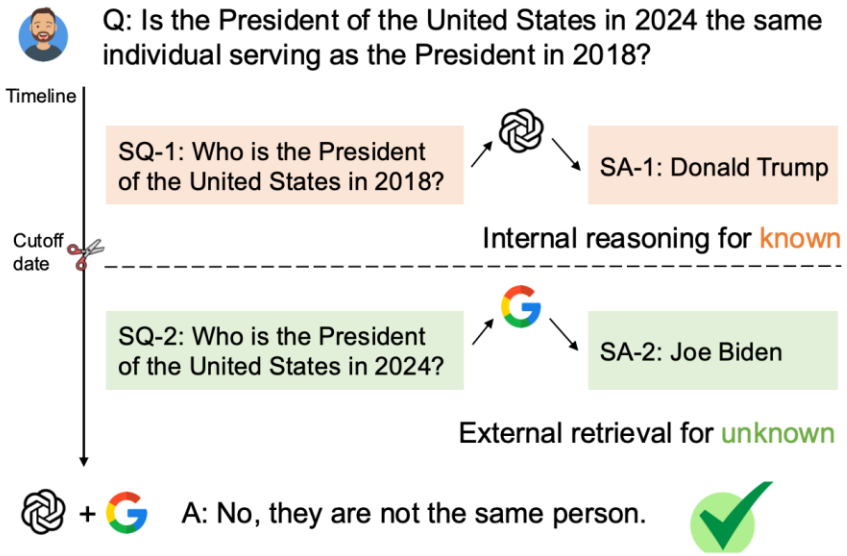
Scalable

Acting Less is Reasoning More! Teaching Model to Act Efficiently

Agentic Prompting – Self-DC

By the 1st principle: Given a LLM, its knowledge boundary is fixed at time *t*.

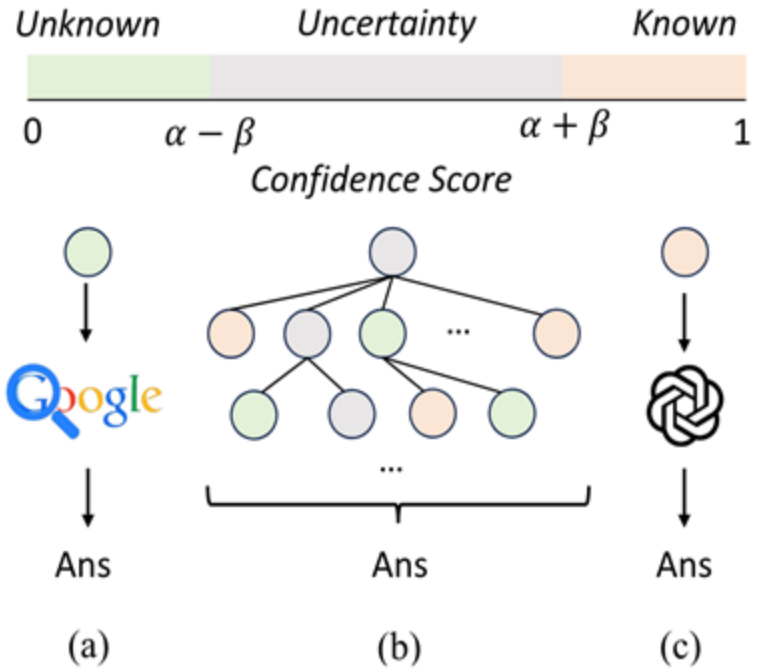
Thus, given one LLM and one question, there are four cases.



- ***Single Known.*** The question contains no sub-questions and can be solved using internal knowledge of LLMs, such as with the generate-then-read method.
- ***Single Unknown.*** The question contains no sub-questions and can only be solved using external knowledge, such as with the retrieve-then-read method.
- ***Compositional Known.*** The question contains several sub-questions, and each sub-question is *Single Known*.
- ***Compositional Unknown.*** The question contains several sub-questions, and at least one sub-question is *Single Unknown*.

Self-DC: When to Reason and When to Act? Self Divide-and-Conquer for Compositional Unknown Questions

Agentic Prompting – Self-DC



Our proposed **Self-DC** framework, including a) retrieve-then-read for unknown questions, b) decompose-and-combination for uncertain questions; and c) generate-then-read for known questions.

By the 2nd principle: different LLMs have different knowledge boundaries.

Step1: knowledge boundary assessment for different LLMs, i.e., uncertainty estimation such as prompting LLMs to generate confidence scores or multiple sampling. (***monitor***)

Step2: divide-and-conquer (***control***)

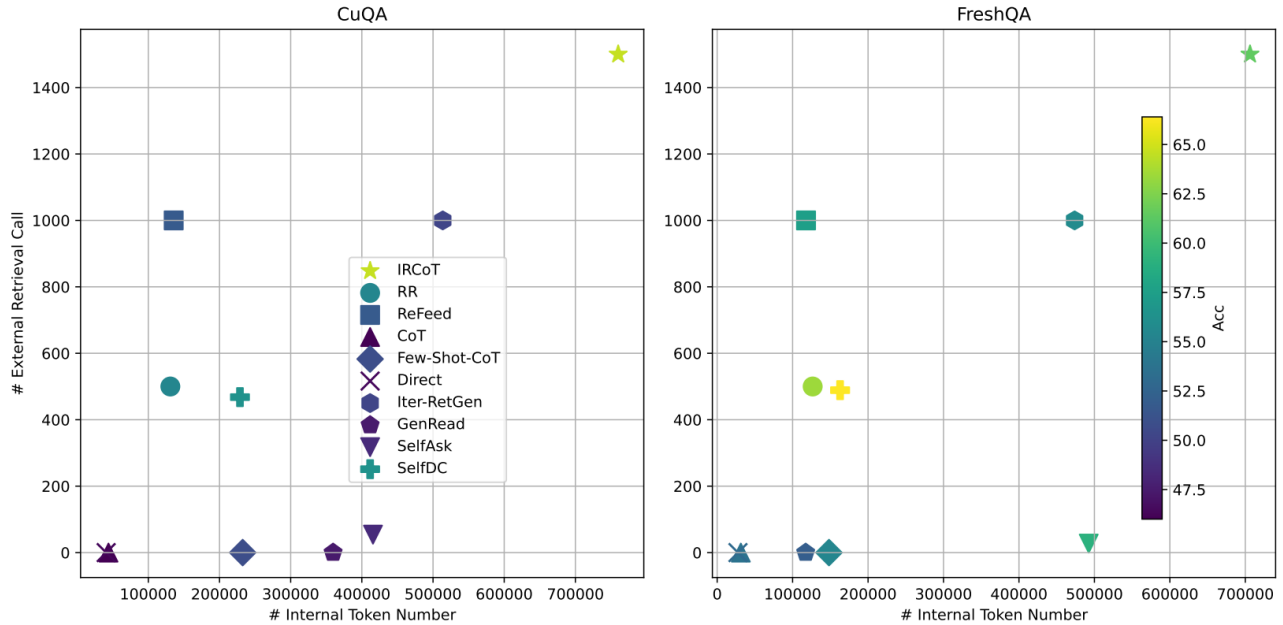
This is **the first work to consider the relationship between reasoning and acting** in terms of trade-off between effectiveness and efficiency.

Self-DC: When to Reason and When to Act? Self Divide-and-Conquer for Compositional Unknown Questions

Agentic Prompting – Self-DC

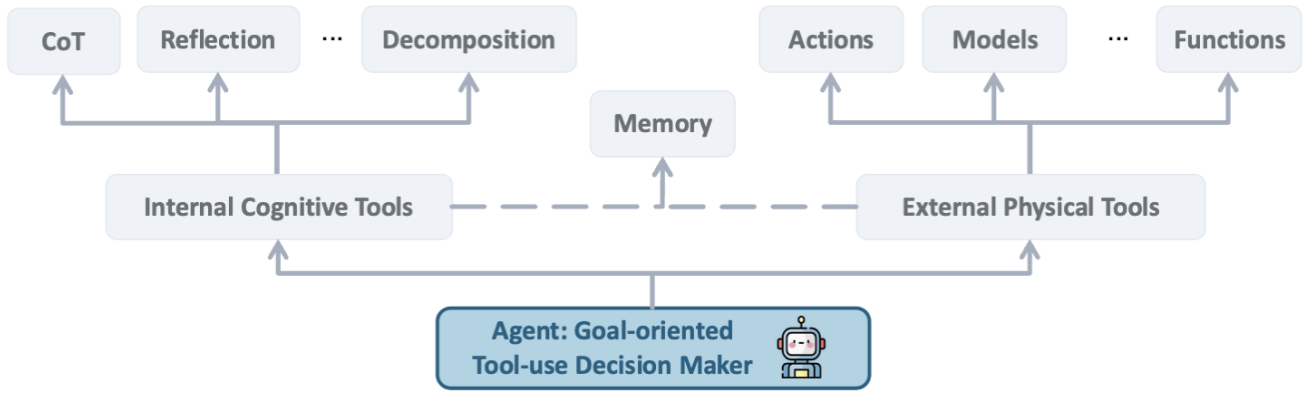
- Self-DC achieves **better trade-off between efficiency and effectiveness** than retrieval-based methods.

Methods	#R	CuQA			FreshQA		
		EM	F1	Acc†	EM	F1	Acc†
w/o retrieval							
Direct	0	29.0	19.4	46.4	27.2	17.3	53.0
CoT	0	28.8	18.2	46.0	29.2	18.1	53.8
Few-shot-CoT*	0	43.0	3.2	50.8	35.0	9.1	55.4
GenRead	0	29.6	29.2	47.4	26.8	27.7	52.0
w/ retrieval							
RR	n	32.0	31.6	55.4	35.2	32.6	63.4
REFEED	$2n$	26.2	33.5	51.8	28.8	34.5	57.4
IRCoT	$3n$	**47.8**	13.5	**64.6**	34.2	17.8	61.4
Self-Ask*	0-n	19.8	3.8	48.4	5.6	9.8	59.0
ITER-RETGEN*	$2n$	23.4	12.6	50.9	31.2	21.1	55.8
Self-DC (*verb*)	0-n	34.0	32.2	53.8	30.2	30.2	59.8
Self-DC (*prob*)	0-n	36.4	**36.5**	56.4	**37.4**	**36.6**	**66.4**



Self-DC: When to Reason and When to Act? Self Divide-and-Conquer for Compositional Unknown Questions

Future Direction



If the agent already fulfill the task, what should we pursue further?

1. Maximining Both Internal and External Tools ⇨ Over-optimization Problem and Not Efficient
2. Minimizing Both Internal and **External Tools** ⇨ Hard to train and maybe not effective ✓
3. Maximining Internal and **Minimizing External Tools** ⇨ OpenAI o3 ✓
4. Minimizing Internal and Maximizing External Tools ⇨ Counter-intuitive and also waste the reasoning capabilities of LLMs

$$r_\phi^{tool}(q,y) = \alpha * r_{tool} * r_\phi(q,y)$$

OTC-PO Can do both 2 and 3.

Future Direction

- Cost of Diverse Tool (i.e., **Reward**). Both internal cognitive tools and external physical tools, in terms of time, money and constraints.
- Space of Diver Tool (i.e., **Action**), including both internal cognitive tools and external physical tools, maybe just starting from different APIs / Actions with reasoning as a whole.
- Better RL algorithms, such as StarPO, a trajectory-level optimization method in RAGEN from Zihan.
- More applications: AI for science, …
- More complex: knowledge overlap / conflict → please refer appendix in our theory of agent paper.
- …

References

1. Toward a Theory of Agents as Tool-Use Decision-Makers
2. ReAct: Synergizing Reasoning and Acting in Language Models
3. SMART: Self-Aware Agent for Tool Overuse Mitigation
4. Self-DC: When to Reason and When to Act? Self Divide-and-Conquer for Compositional Unknown Questions
5. ToolRL: Reward is All Tool Learning Needs
6. AdaCtrl: Towards Adaptive and Controllable Reasoning via Difficulty-Aware Budgeting
7. Training Language Models to Reason Efficiently
8. Acting Less is Reasoning More ! Teaching Model to Act Efficiently
9. ……

Thank You!