

CS769 Advanced NLP

# Language Agents and LLM Programming

Junjie Hu



Slides adapted from agent tutorial at EMNLP 2024  
<https://junjihu.github.io/cs769-fall25/>

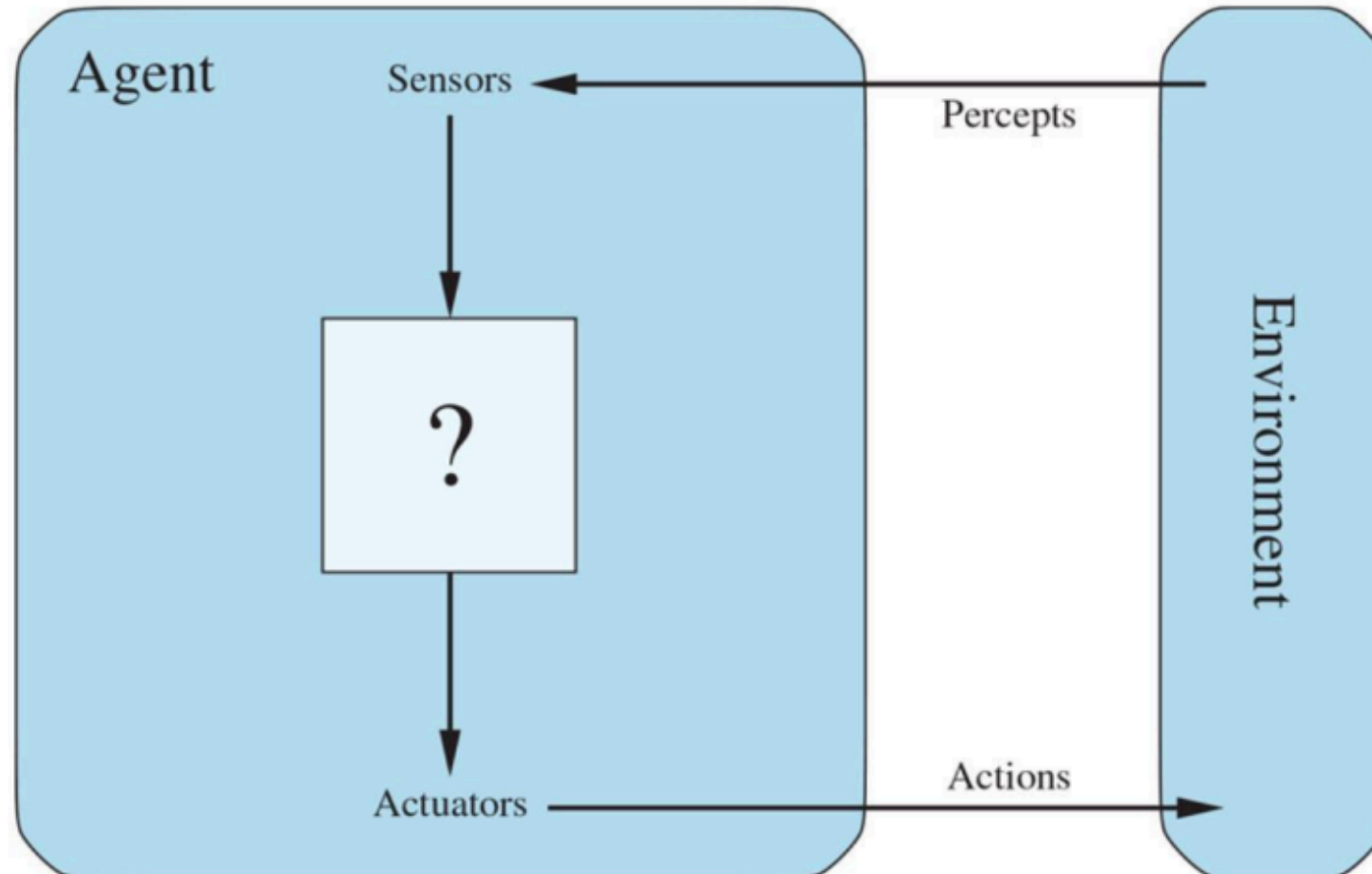
# Goal for Today

- **Part I: Introduction of Language Agents**
  - History, different AI agents
- **Part II: Foundations: Key Components of Language Agents**
  - Reasoning, Memory, Planning
- **Part III: LLM Coding Agents for Software Development**
  - Open Challenges

# What is agent?

“An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.”

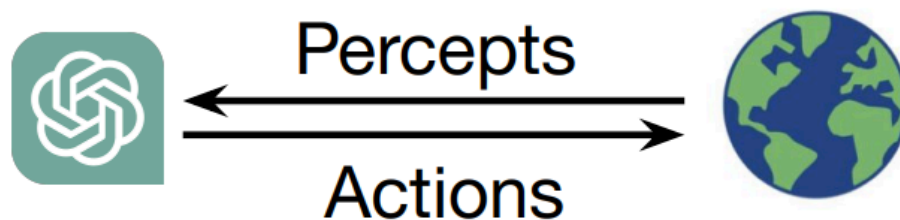
— Russell & Norvig. *AI: A Mordern Approach* (2020)



“Modern” agent = LLM + External Environment?



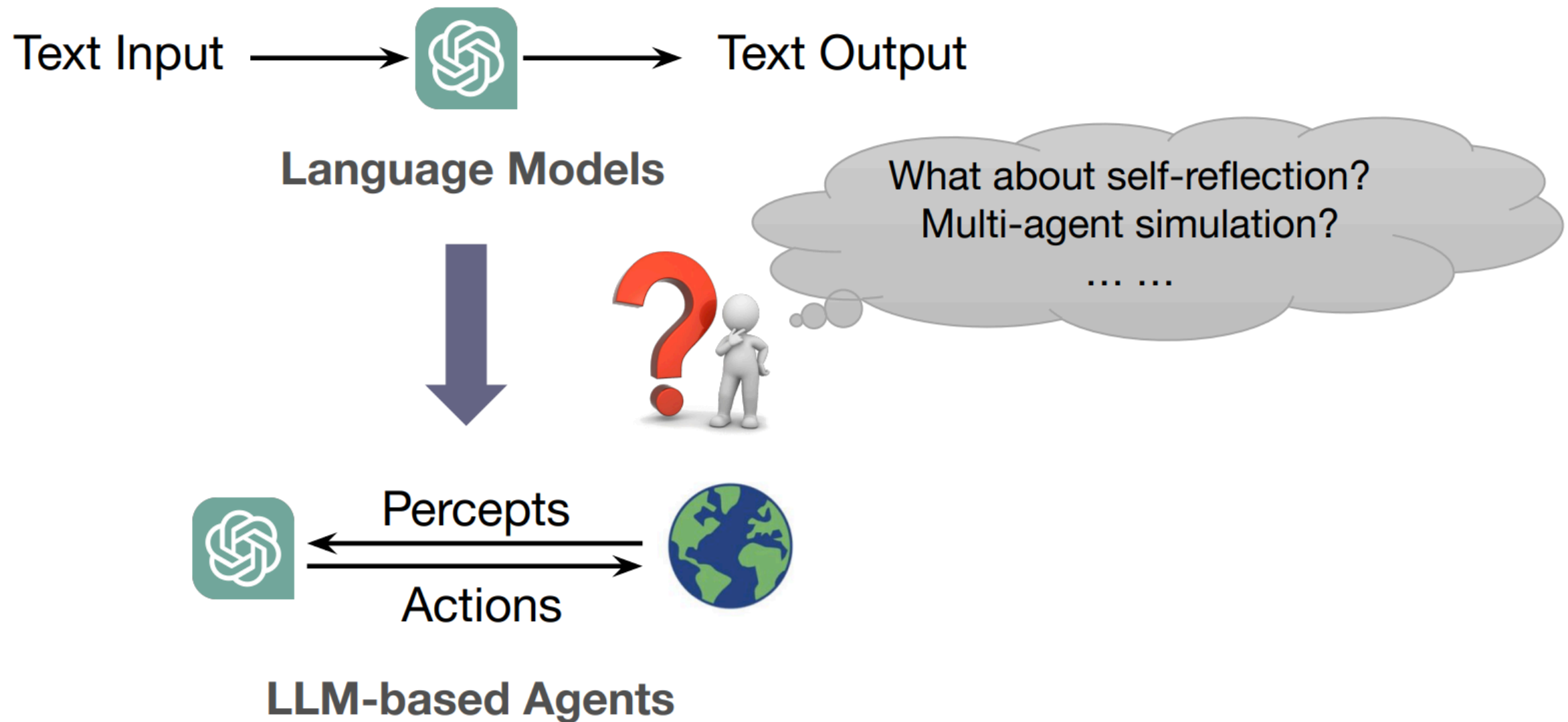
**Language Models**



**LLM-based Agents**

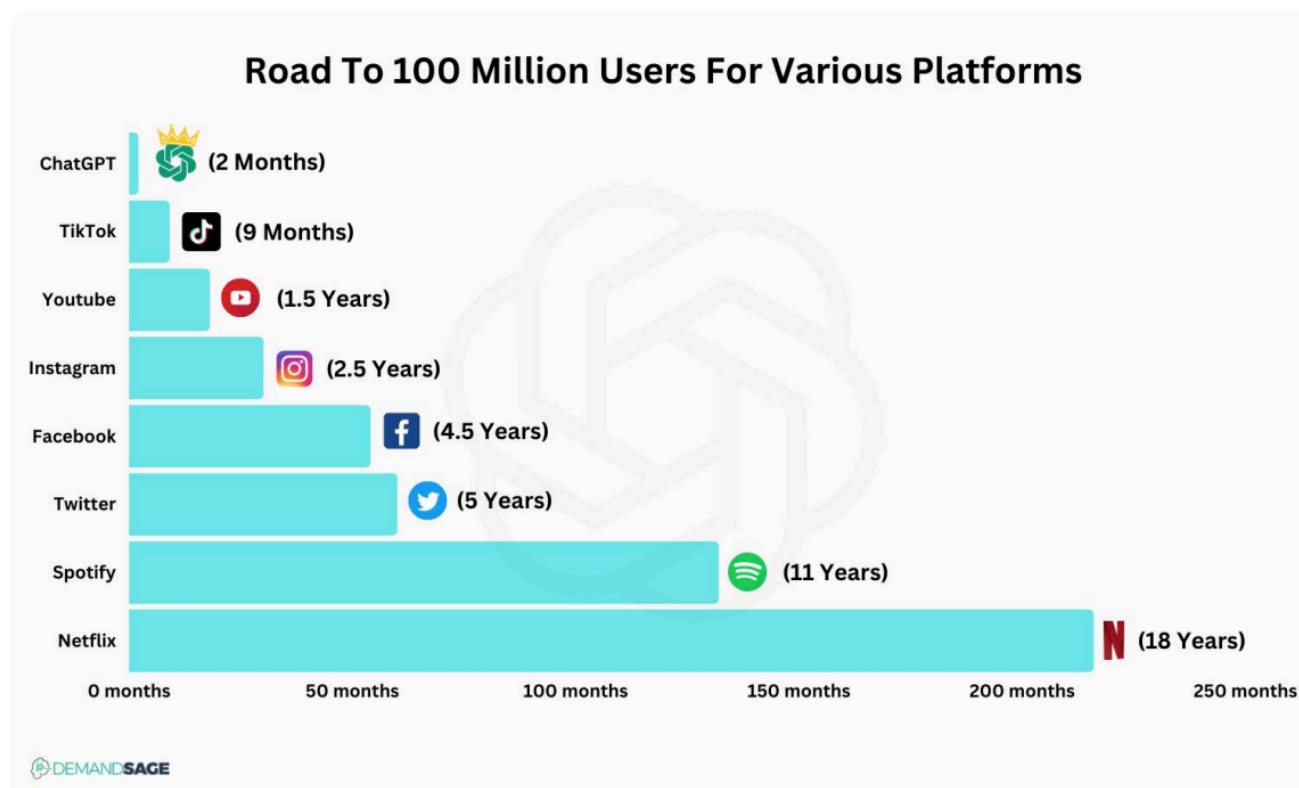


# “Modern” agent = LLM + External Environment?

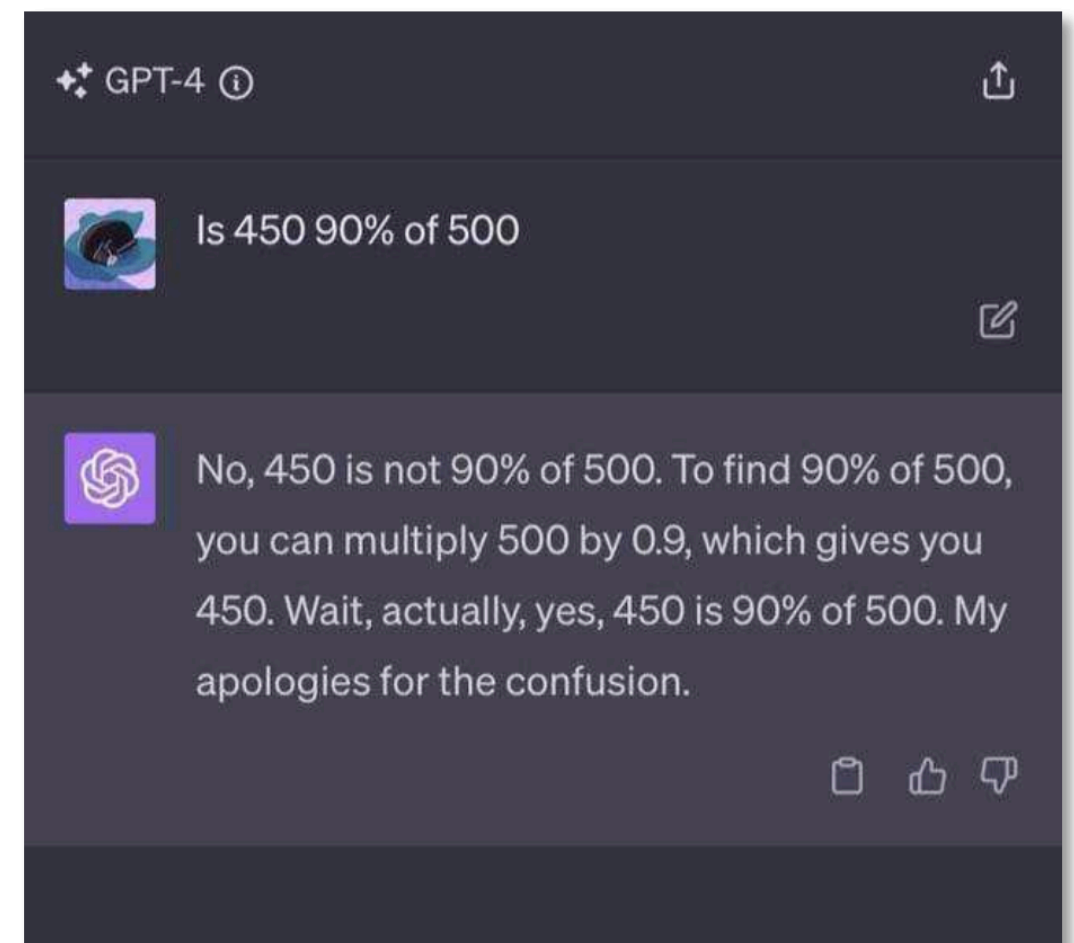


# What's fundamentally different now?

- Contemporary AI agents, with integrated LLM(s), can use language as a vehicle for reasoning and communication
  - Instruction following, in-context learning, output customization
  - Reasoning (for better acting): state inferences, self-reflection, replanning, etc.



<https://www.demandsage.com/chatgpt-statistics/>

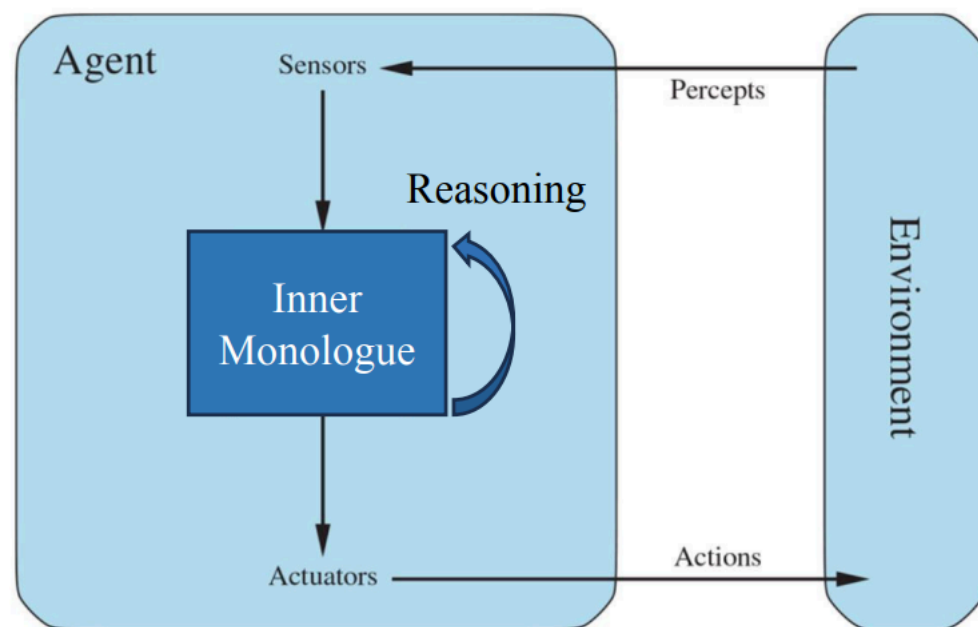


[https://www.reddit.com/r/ChatGPT/comments/16jvl4x/wait\\_actually\\_yes/](https://www.reddit.com/r/ChatGPT/comments/16jvl4x/wait_actually_yes/)

# Language agents: a new type of AI agents

- These contemporary AI agents capable of using language for reasoning and communication are best called “**language agents**.” They are qualitatively a different type of AI agents with **language** being their most distinct trait.
- What about **multimodal agents**?
  - While there’s perception of other input modalities, language is still doing the heavy lifting (i.e., reasoning and communication)
- What about simply LLM agents?
  - The key is using language for reasoning and communication, but that doesn’t have to come from an LLM;
  - Maybe in a few years, we will move beyond LLMs, but the need for universal language understanding and production in agents will remain

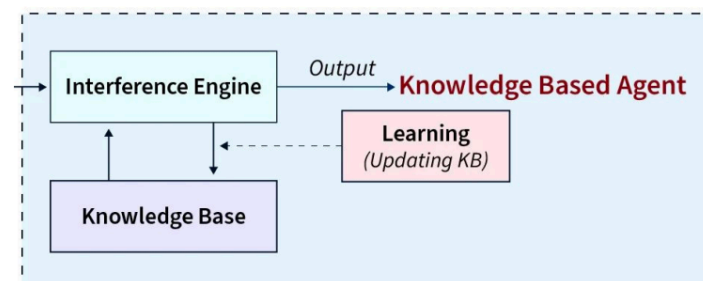
# Reconciling with classical view of language agents



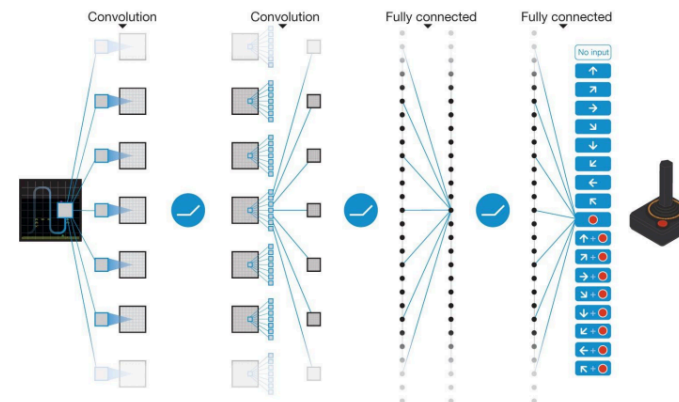
Adapted from Russell & Norvig (2020)

- **Reasoning by generating tokens** is a new type of action (vs. actions in external environments)
- **Internal environment**, where reasoning takes place in an inner monologue fashion
- **Self-reflection** is a 'meta' reasoning action (i.e., reasoning over the reasoning process), akin to metacognitive functions
- **Reasoning is for better acting**, by inferring environmental states, retrospection, dynamic replanning, etc.
- **Percept** and **external action spaces** are substantially expanded, thanks to using language for communication and multimodal perception

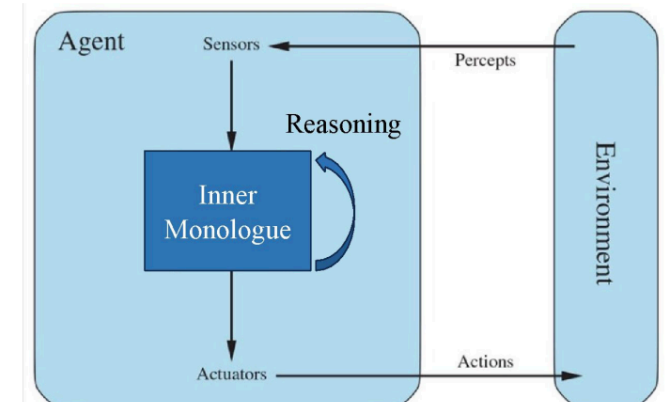
# Evolution of AI Agents



**Logical Agent**



**Neural Agent**

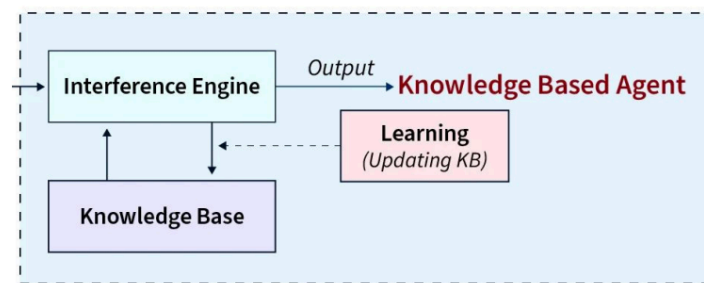


**Language Agent**

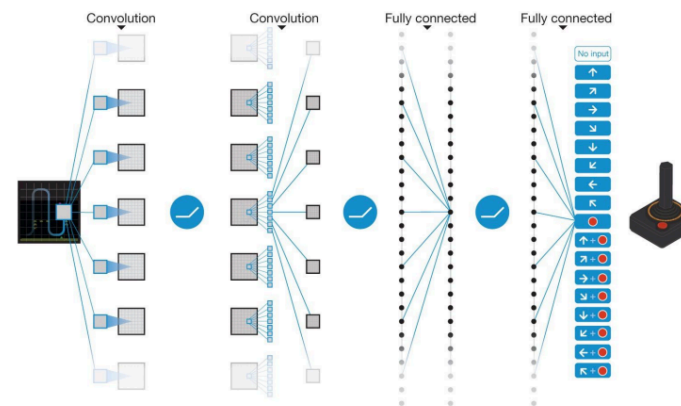
<b>Expressiveness</b>			
<b>Reasoning</b>			
<b>Adaptivity</b>			

Image sources: <https://www.scaler.com/topics/artificial-intelligence-tutorial/knowledge-based-agent/>  
Mnih et al., "Human-level control through deep reinforcement learning." Nature (2015)

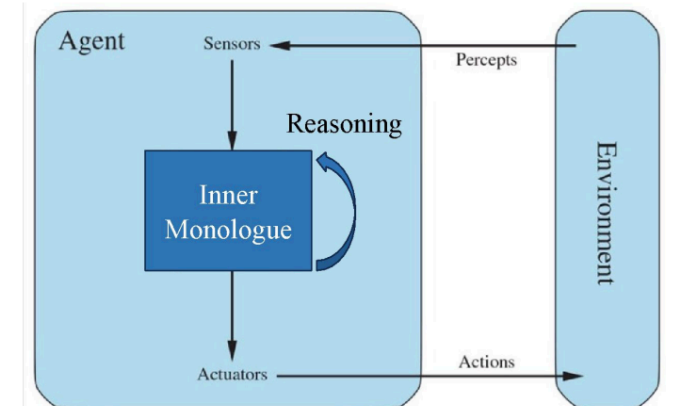
# Evolution of AI Agents



**Logical Agent**



**Neural Agent**

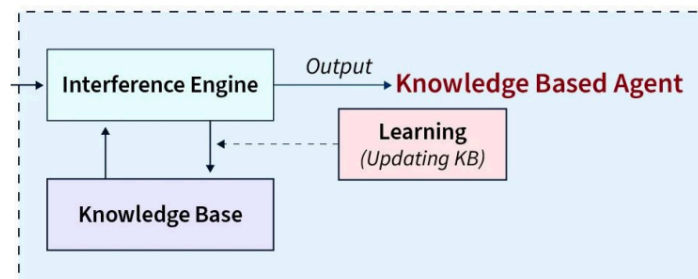


**Language Agent**

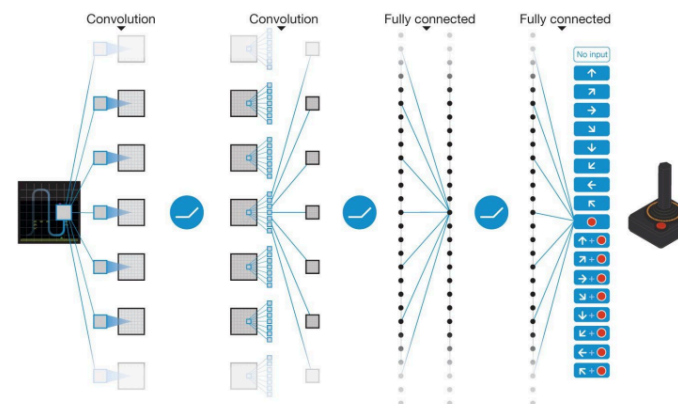
<b>Expressiveness</b>	Low bounded by the logical language		
<b>Reasoning</b>	Logical inferences sound, explicit, rigid		
<b>Adaptivity</b>	Low bounded by knowledge curation		

Image sources: <https://www.scaler.com/topics/artificial-intelligence-tutorial/knowledge-based-agent/>  
Mnih et al., "Human-level control through deep reinforcement learning." Nature (2015)

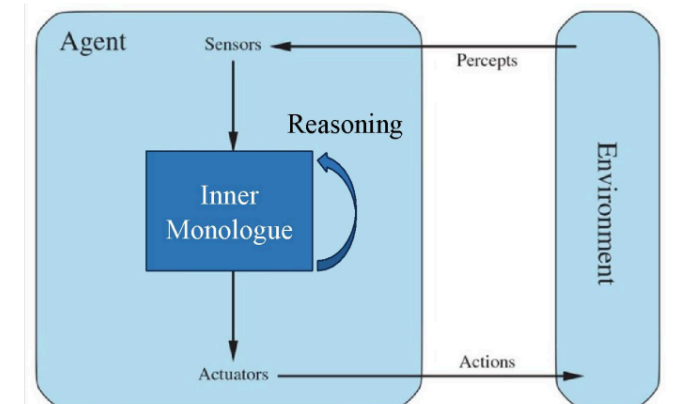
# Evolution of AI Agents



**Logical Agent**



**Neural Agent**



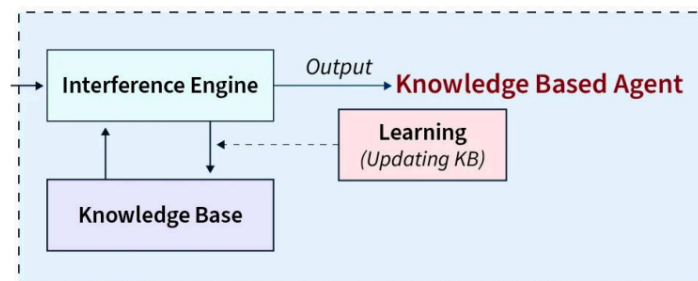
**Language Agent**

<b>Expressiveness</b>	Low bounded by the logical language	Medium anything a (small) NN can encode	
<b>Reasoning</b>	Logical inferences sound, explicit, rigid	Parametric inferences stochastic, implicit, rigid	
<b>Adaptivity</b>	Low bounded by knowledge curation	Medium data-driven but sample inefficient	

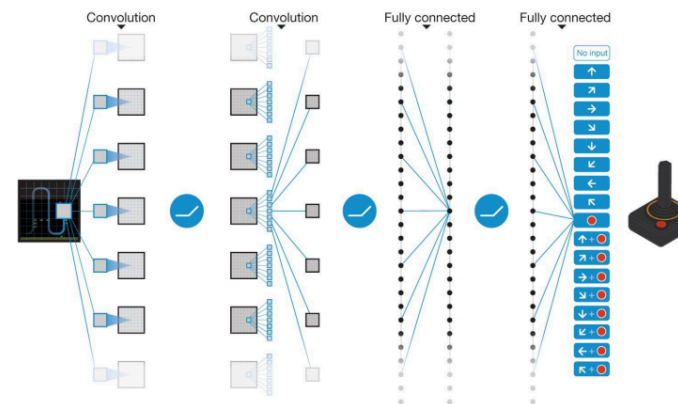
Image sources: <https://www.scaler.com/topics/artificial-intelligence-tutorial/knowledge-based-agent/>  
Mnih et al., "Human-level control through deep reinforcement learning." Nature (2015)



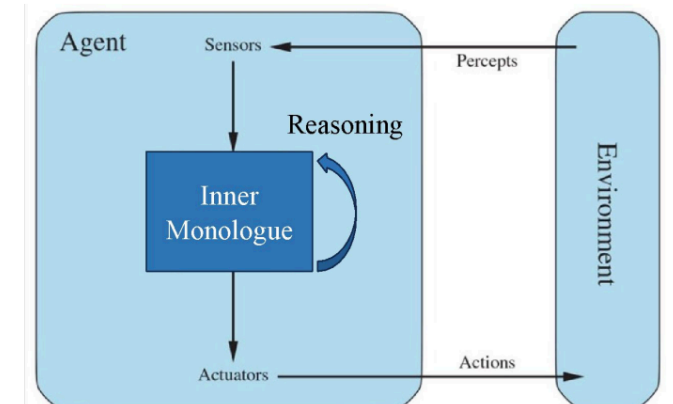
# Evolution of AI Agents



**Logical Agent**



**Neural Agent**



**Language Agent**

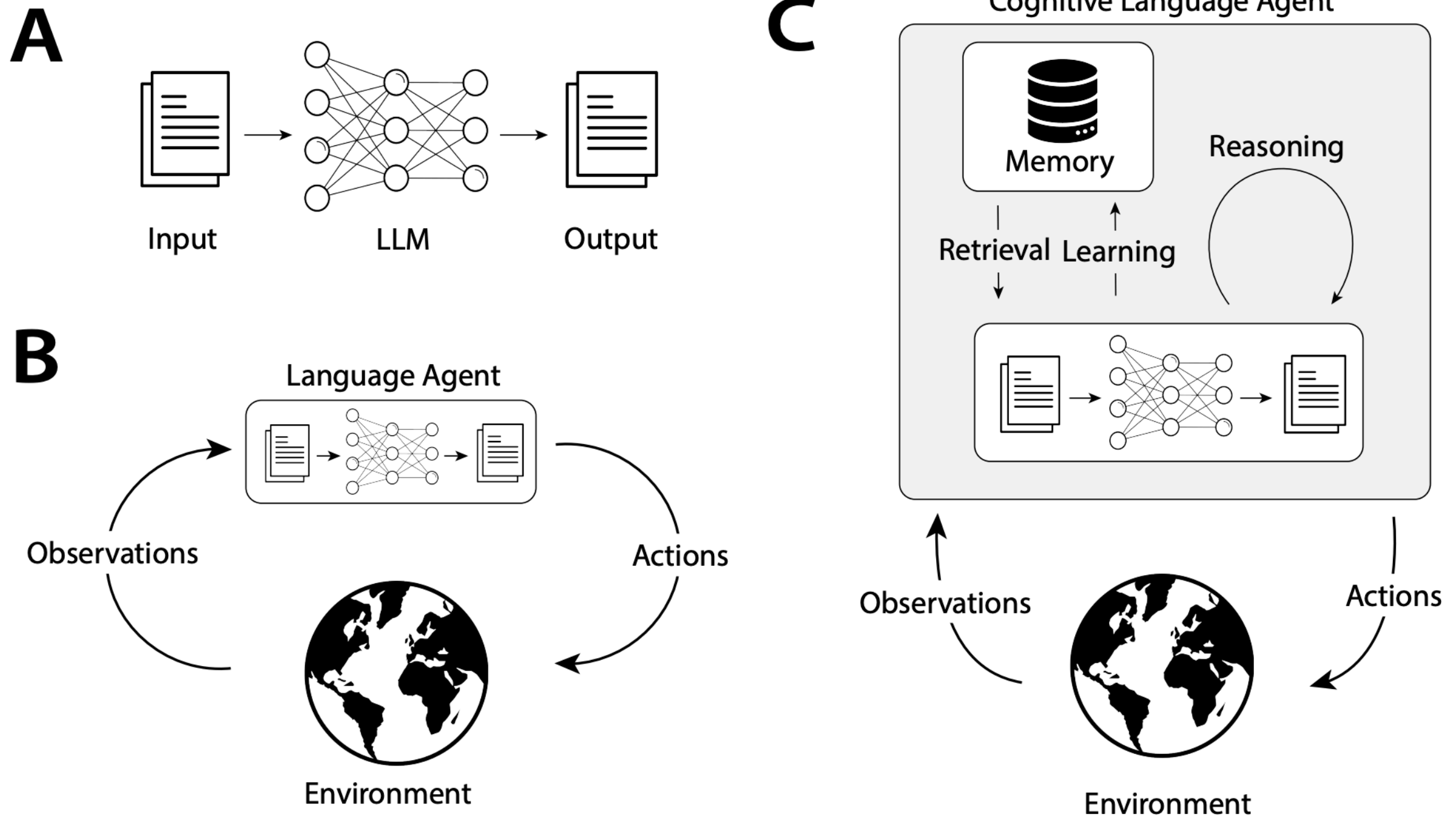
<b>Expressiveness</b>	Low bounded by the logical language	Medium anything a (small) NN can encode	High almost anything, esp. verbalizable parts of the world
<b>Reasoning</b>	Logical inferences sound, explicit, rigid	Parametric inferences stochastic, implicit, rigid	Language-based inferences fuzzy, semi-explicit, flexible
<b>Adaptivity</b>	Low bounded by knowledge curation	Medium data-driven but sample inefficient	High strong prior from LLMs + language use

Image sources: <https://www.scaler.com/topics/artificial-intelligence-tutorial/knowledge-based-agent/>  
Mnih et al., "Human-level control through deep reinforcement learning." Nature (2015)



# Part II: Foundations: Reasoning, Memory, and Planning

# Cognitive Language Agents

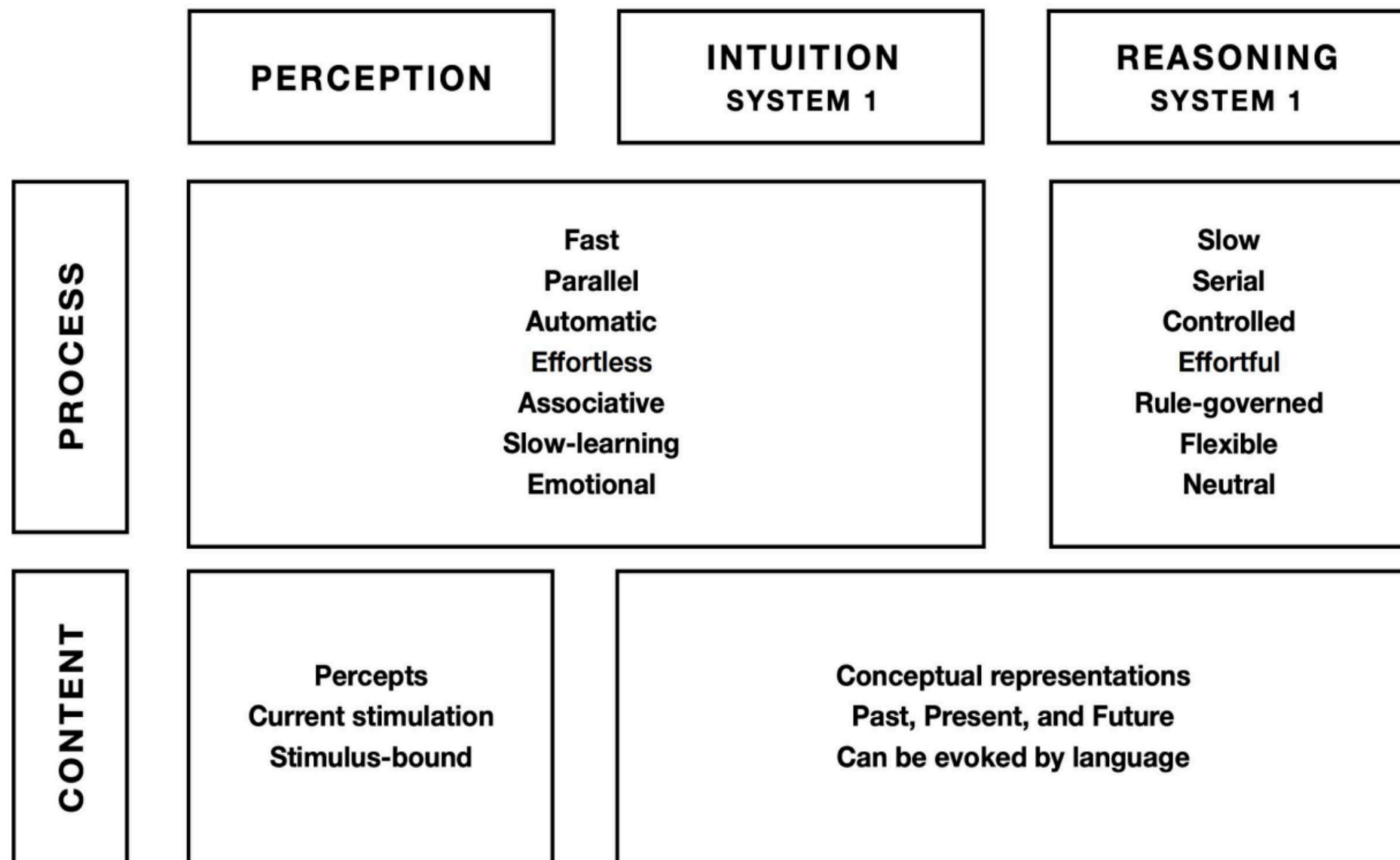


# Key Concepts for Language Agents

- **Action space** (beyond environment actions)
  - **Reasoning**: update short-term memory (context window)
  - **Retrieval/Learning**: read/write long-term memory (model weights, vector store, self-notes, event flows, etc)
- **Planning**: (inference-time) algorithm to choose an action from the action space
- **Environment**: receives an action from the agent and provides a reward

# Reasoning

- For humans: various mental processes



\*(Kahneman, 2003), also note that this notion of dual process is put in question. See, e.g., Mercier and Sperber (2017)

# Reasoning

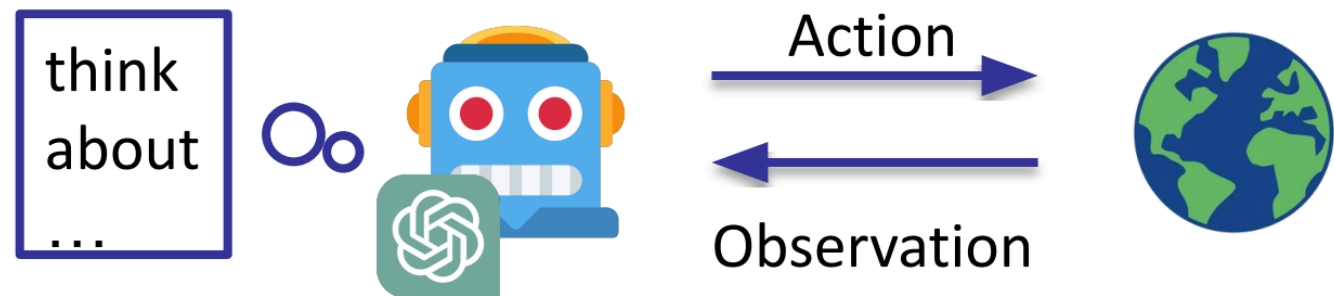
- For humans: various mental processes
- For LMs: intermediate generation (Chain-of-Thought)

<p><b>Math Word Problems (free response)</b></p> <p>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?</p> <p>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. <math>5 + 6 = 11</math>. The answer is 11.</p>	<p><b>Math Word Problems (multiple choice)</b></p> <p>Q: How many keystrokes are needed to type the numbers from 1 to 500? Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788</p> <p>A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. <math>9 + 90(2) + 401(3) = 1392</math>. The answer is (b).</p>
<p><b>StrategyQA</b></p> <p>Q: Yes or no: Would a pear sink in water?</p> <p>A: The density of a pear is about 0.6 g/cm<sup>3</sup>, which is less than water. Thus, a pear would float. So the answer is no.</p>	<p><b>Date Understanding</b></p> <p>Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?</p> <p>A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.</p>

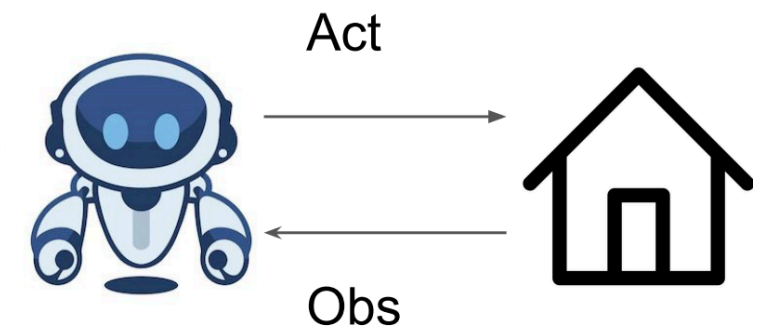
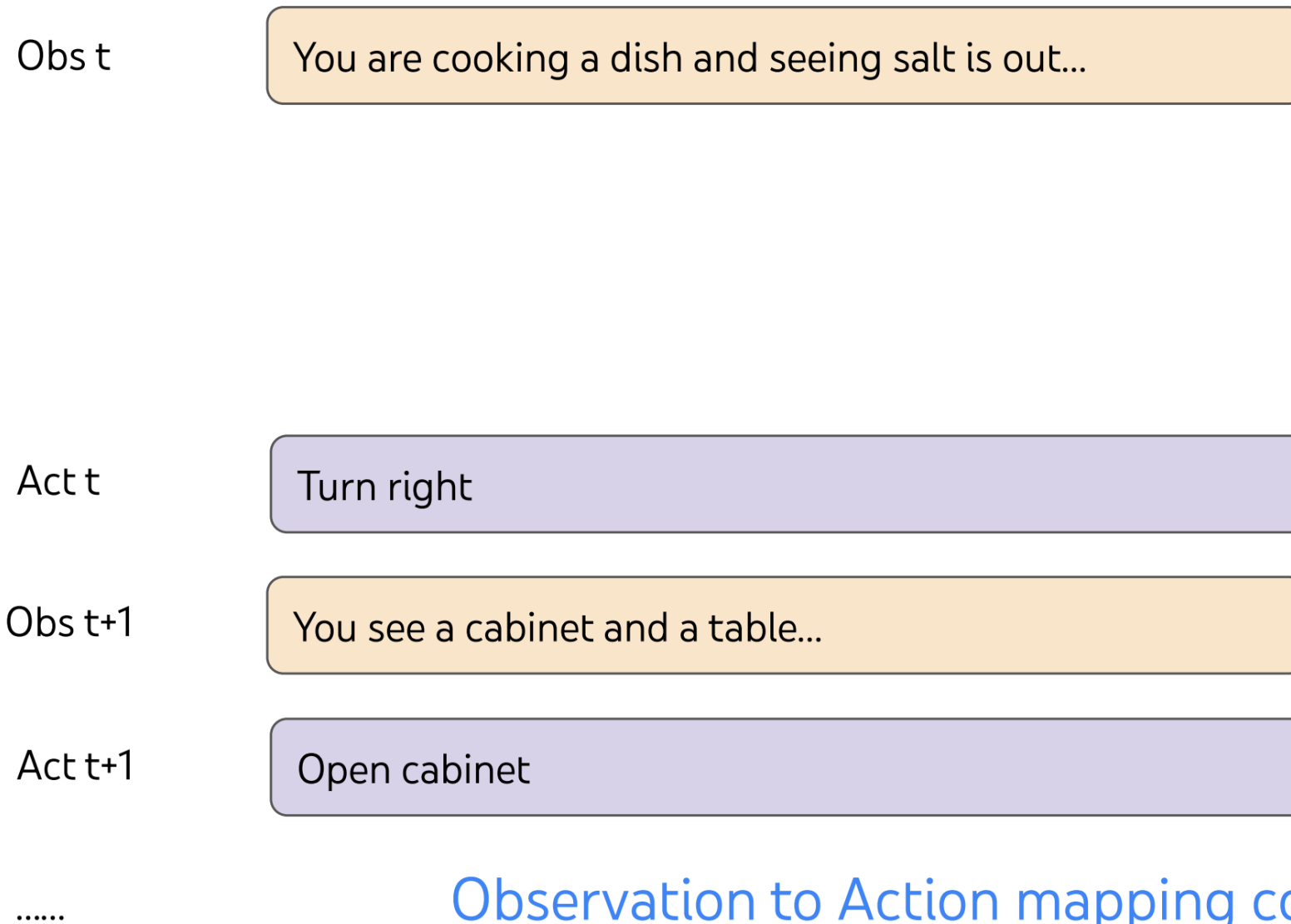
\*(Wei et al., 2022), also see (Ling et al., 2017; Cobbe et al., 2021; Nye et al., 2021)

# Reasoning

- For humans: various mental processes
- For LMs: intermediate generation (Chain-of-Thought)
- For agents: internal actions.



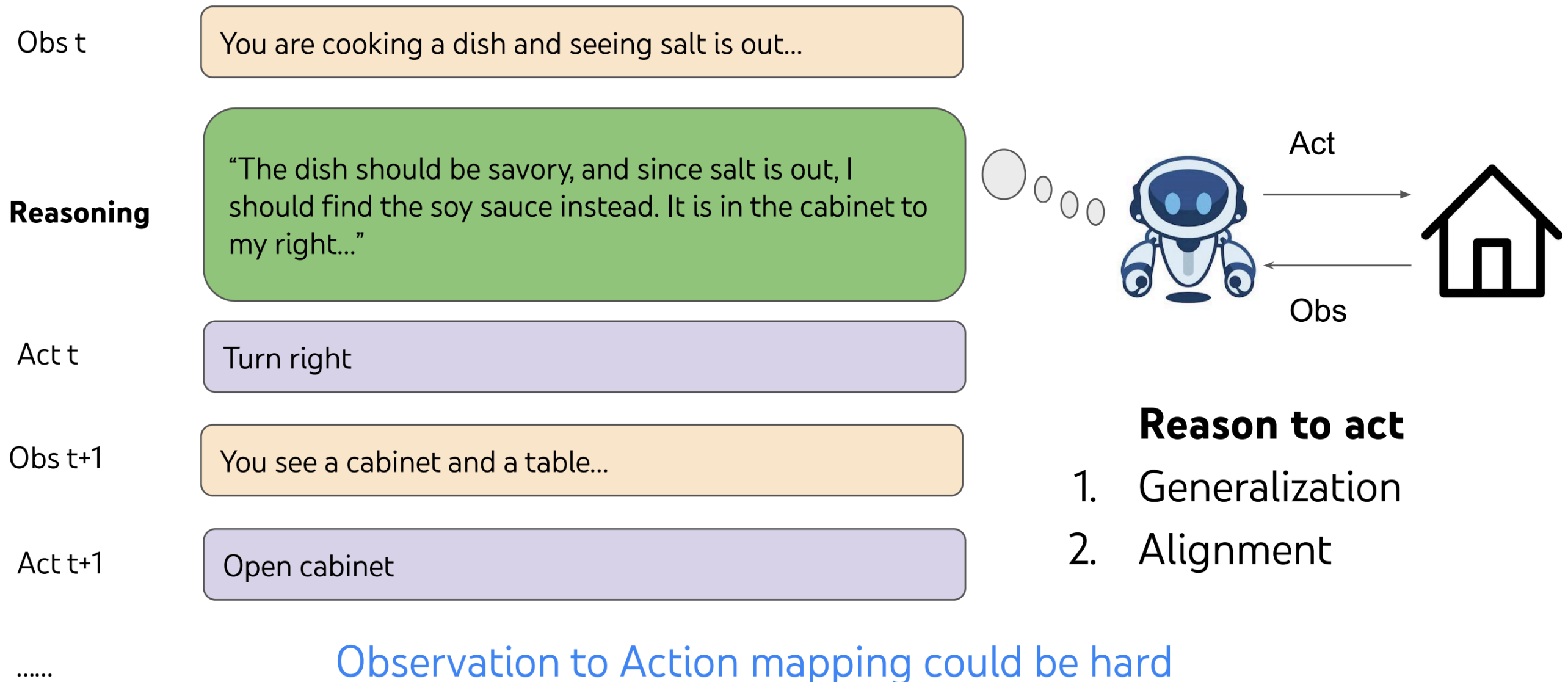
# Why is reasoning helpful for agents?



## Reason to act

1. Generalization
2. Alignment

# Why is reasoning helpful for agents?





# Conversely, acting also helps reasoning

SH

who is PM of UK?

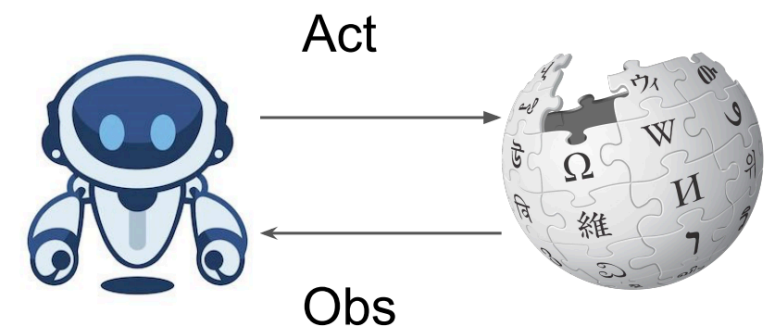


As of December 2022, the Prime Minister of the United Kingdom is Boris Johnson. However, since I am a large language model trained by OpenAI, my knowledge is limited to what was available at the time of my training in 2021. My ability to provide current information is limited, and I am not able to browse the internet for up-to-date information.

## Act to reason

1. Obtain knowledge
2. Update knowledge

Hallucination due to limited / outdated knowledge



# ReAct [Yao et al, 2022]

## ReAct prompt

```
Question Which magazine was started first Arthur's Magazine or First for Women?  
Thought 1 I need to search Arthur's Magazine and First for Women, and find which was started first.  
Action 1 Search[Arthur's Magazine]  
Observation 1 Arthur's Magazine (1844-1846) was an American literary periodical published in Philadelphia ...  
Thought 2 Arthur's Magazine was started in 1844. I need to search First for Women next.  
Action 2 Search[First for Women]  
Observation 2 First for Women is a woman's magazine published by Bauer Media Group ... was started in 1989...  
Thought 3 First for Women was started in 1989. 1844 (Arthur's Magazine) < 1989 (First for Women), so  
Arthur's Magazine was started first.  
Action 3 Finish[Arthur's Magazine]
```

## Reason-only prompt (Chain-of-thought, CoT)

```
Question Which magazine was started first Arthur's Magazine or First for Women?  
Thought Arthur's Magazine was started in 1844. First for Women was started in 1989. 1844 (Arthur's  
Magazine) < 1989 (First for Women), so Arthur's Magazine was started first.  
Answer Arthur's Magazine
```

## Act-only prompt (similar to WebGPT)

```
Question Which magazine was started first Arthur's Magazine or First for Women?  
Action 1 Search[Arthur's Magazine]  
Observation 1 Arthur's Magazine (1844-1846) was an American literary periodical published in Philadelphia ...  
Action 2 Search[First for Women]  
Observation 2 First for Women is a woman's magazine published by Bauer Media Group ... was started in 1989...  
Action 3 Finish[Arthur's Magazine]
```

## Standard prompt

```
Question Which magazine was started first Arthur's Magazine or First for Women?  
Answer Arthur's Magazine
```

32

# Reasoning without acting

**Question:** Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

Standard

**Answer:** iPod



# Reasoning without acting

**Question:** Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

Standard

**Answer:** iPod



Reason Only

**Thought:** Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

**Answer:** iPhone, iPad, iPod Touch



# Reasoning with acting

**Question:** Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

## ReAct (Reason + Act)

**Thought 1:** I need to search Apple Remote and find the program it was originally designed to interact with.

**Act 1:** **Search**[Apple Remote]

**Obs 1:** The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the **Front Row media center** program .....(long text omitted)

# Reasoning with acting

**Question:** Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

## ReAct (Reason + Act)

**Thought 1:** I need to search Apple Remote and find the program it was originally designed to interact with.

**Act 1:** `Search[Apple Remote]`

**Obs 1:** The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the `Front Row media center` program .....(long text omitted)

**Thought 2:** Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.

**Act 2:** `Search[Front Row]`

**Obs 2:** Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

**Thought 3:** Front Row is not found. I need to search `Front Row (software)` .

**Act 3:** `Search[Front Row (software)]`

**Obs 3:** Front Row is a discontinued media center software (long text omitted)

**Thought 4:** `Front Row (software)` is controlled by an Apple Remote or the keyboard function keys So the answer is keyboard function keys.

**Act 4:** `Finish[keyboard function keys]`



# Acting without reasoning

**Question:** Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

## Act Only

**Act 1:** **Search**[Apple Remote]

**Obs 1:** The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

**Act 2:** **Search**[Front Row]

**Obs 2:** Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

**Act 3:** **Search**[Front Row (software)]

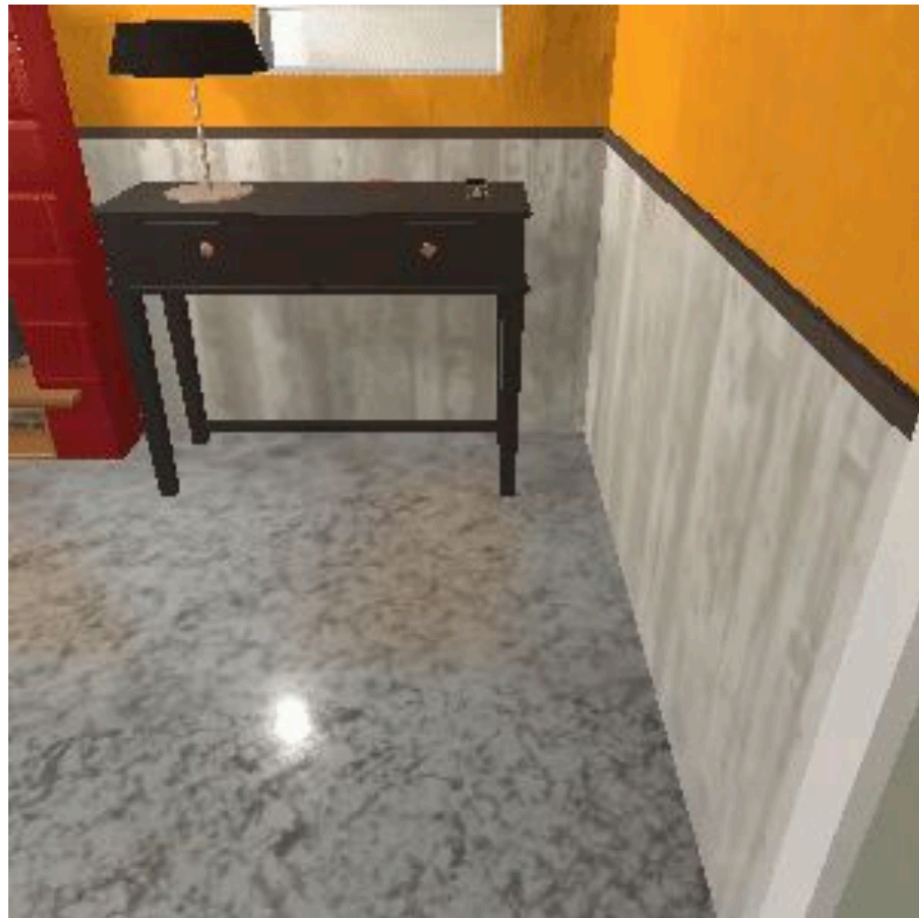
**Obs 3:** Front Row is a discontinued media center software ...

**Act 4:** **Finish**[yes]

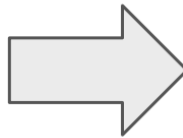
X



# Acting in language



Embodied Env  
(ALFRED game)



You are in the middle of a room. Looking quickly around you, you see a drawer 2, a shelf 5, a drawer 1, a shelf 4, a sidetable 1, a drawer 5, a shelf 6, a shelf 1, a shelf 9, a cabinet 2, a sofa 1, a cabinet 1, a shelf 3, a cabinet 3, a drawer 3, a shelf 11, a shelf 2, a shelf 10, a dresser 1, a shelf 12, a garbagecan 1, a armchair 1, a cabinet 4, a shelf 7, a shelf 8, a safe 1, and a drawer 4.

Your task is to: *put some vase in safe.*

**> go to shelf 6**

You arrive at loc 4. On the shelf 6, you see a vase 2.

**> take vase 2 from shelf 6**

You pick up the vase 2 from the shelf 6.

**> go to safe 1**

You arrive at loc 3. The safe 1 is closed.

**> open safe 1**

You open the safe 1. The safe 1 is open. In it, you see a keychain 3.

**> put vase 2 in/on safe 1**

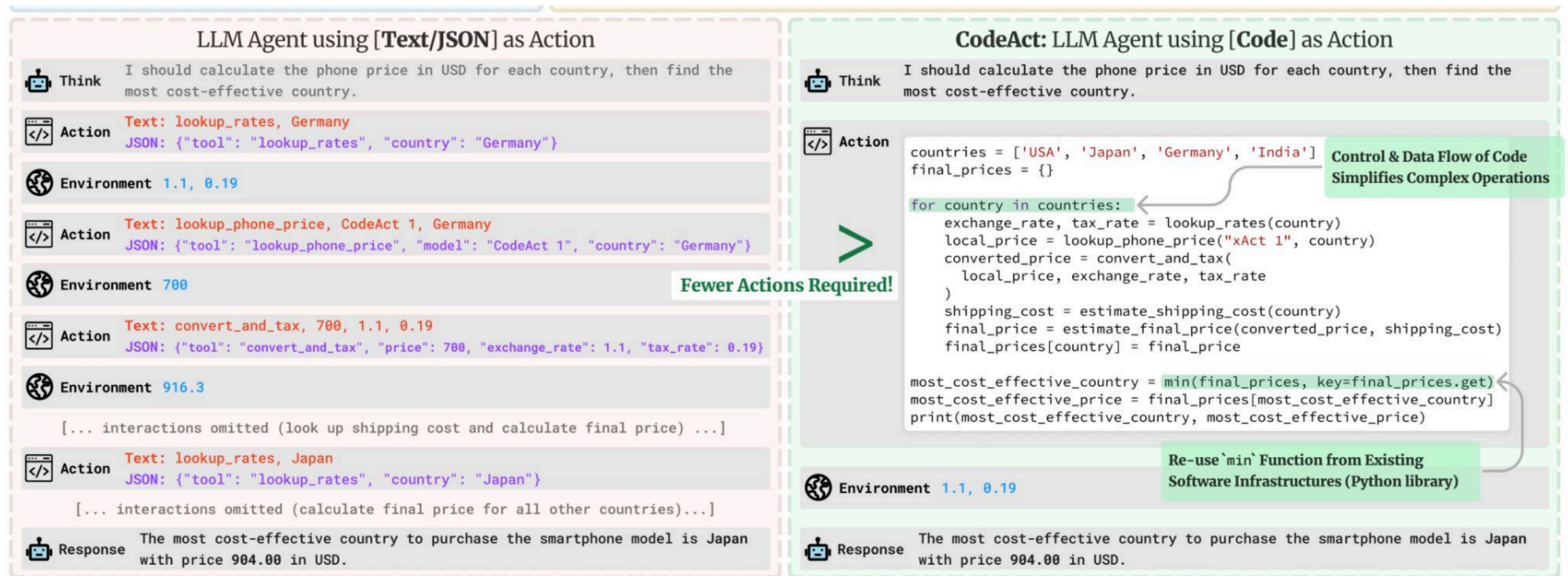
You won!

Text Env  
(ALFWorld)



# The flexibility of “acting”

- **Tool usage:** use external tools as action



Wang, et al., ICLR 2024. Executable Code Actions Elicit Better LLM Agents.

# The flexibility of “acting”

- **Tool usage:** use external tools as action

**a**

```
from chemcrow.agents import ChemTools, ChemCrow

chemtools = ChemTools()

# Initialize ChemCrow object with toolset and LLM
crow = ChemCrow(
    chemtools.all_tools,
    model="gpt-4",
    temp=0.1,
)

# Task definition
task = (
    "Find and synthesize a thiourea organocatalyst "
    "which accelerates a Diels-Alder reaction."
)

# Execute ChemCrow
crow.run(task)
```

**b Task:** Find and synthesize a thiourea organocatalyst which accelerates a Diels-Alder reaction.

First, I need to find a thiourea organocatalyst that can accelerate the Diels-Alder reaction. I will perform a web search to find a suitable catalyst.  
**Web Search tool:** Schreiner's thiourea catalyst


Now, I will obtain the SMILES. **Name2Smiles tool:**  
FC(F)(F)c1cc(NC(=S)Nc2cc(C(F)(F)F)cc(C(F)(F)F)c2)cc(C(F)(F)F)c1

I will plan a synthesis for Schreiner's thiourea catalyst.  
**SynthesisPlanner tool:** Detailed synthesis plan

I will execute the synthesis.  
**SynthesisExecutor tool:** Successful synthesis.

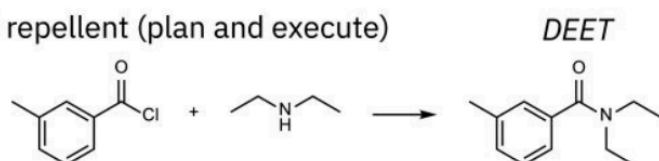
Connection with physical world →

**c RoboRXN synthesis platform**

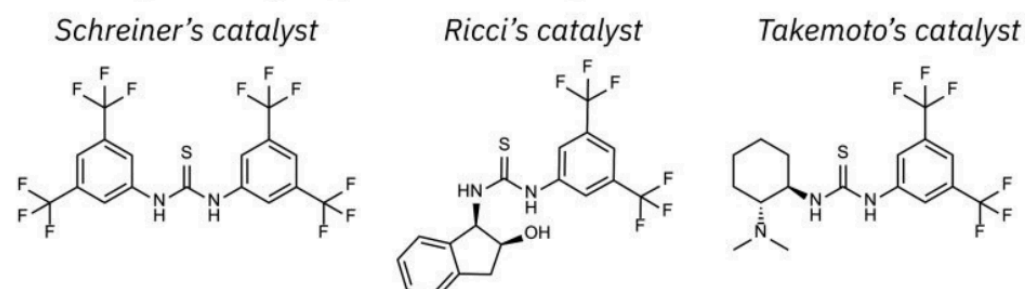


**d Chemcrow workflows with experimental validation**

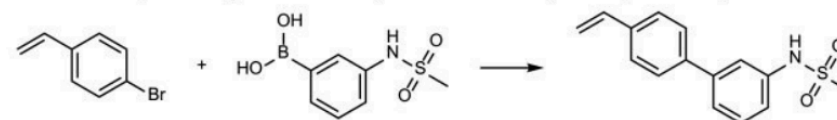
Insect repellent (plan and execute)



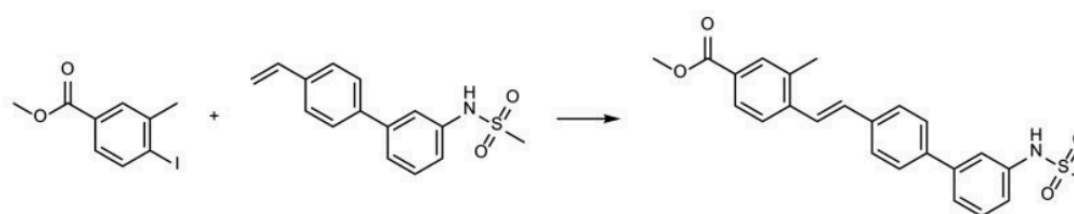
Thiourea organocatalysts (plan and execute)



Novel chromophore (clean data, train model, and predict)



Synthesis step 1: Bromo Suzuki coupling

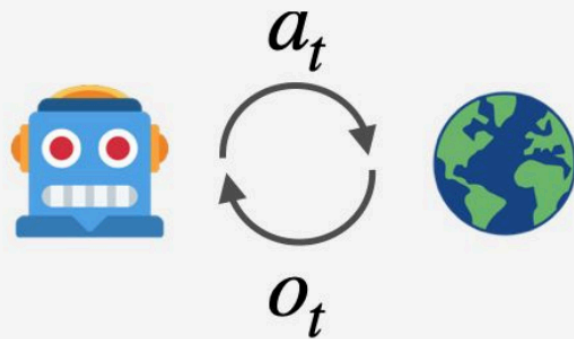


Synthesis step 2: Iodo Heck reaction



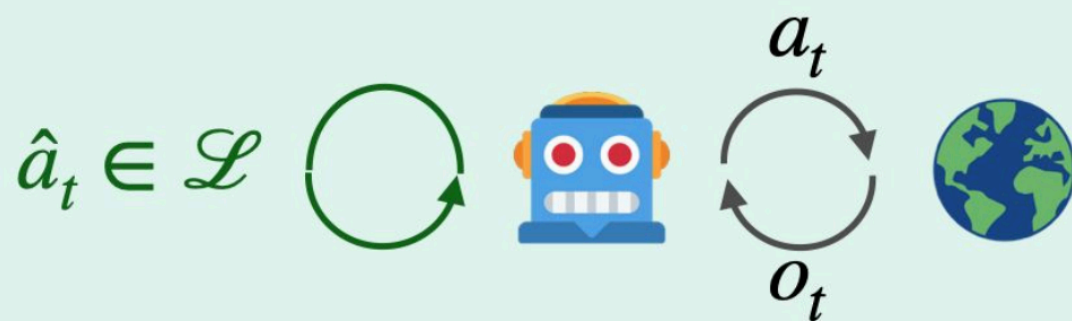
# Why is reasoning special for agents?

**Traditional agents:** action space  $A$  defined by the environment



- **External feedback**  $o_t$
- Agent context  $c_t = (o_1, a_1, o_2, a_2, \dots, o_t)$
- Agent action  $a_t \sim \pi(a | c_t) \in A$

**ReAct:** action space  $\hat{A} = A \cup \mathcal{L}$  augmented by reasoning

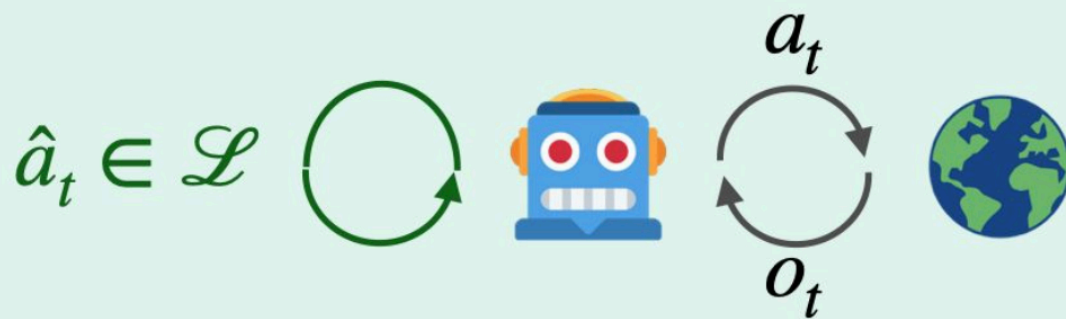


- $\hat{a}_t \in \mathcal{L}$  can be any language sequence
- Agent context  $c_{t+1} = (c_t, \hat{a}_t, a_t, o_{t+1})$
- $\hat{a}_t \in \mathcal{L}$  only updates **internal context**

# Why is reasoning just now for agents?

- Bigger action space -> More capacity, harder decision making
  - The space of reasoning/language is **infinite**
- LLMs learn reasoning priors by imitating various human reasoning traces

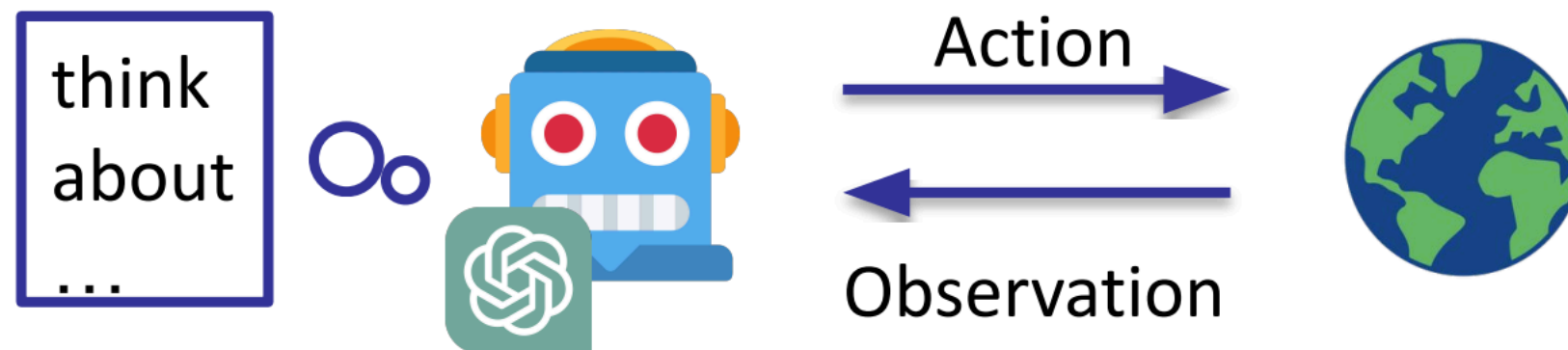
**ReAct:** action space  $\hat{A} = A \cup \mathcal{L}$  augmented by reasoning



- $\hat{a}_t \in \mathcal{L}$  can be any language sequence
- Agent context  $c_{t+1} = (c_t, \hat{a}_t, a_t, o_{t+1})$
- $\hat{a}_t \in \mathcal{L}$  only updates **internal context**

# Reasoning: Takeaways

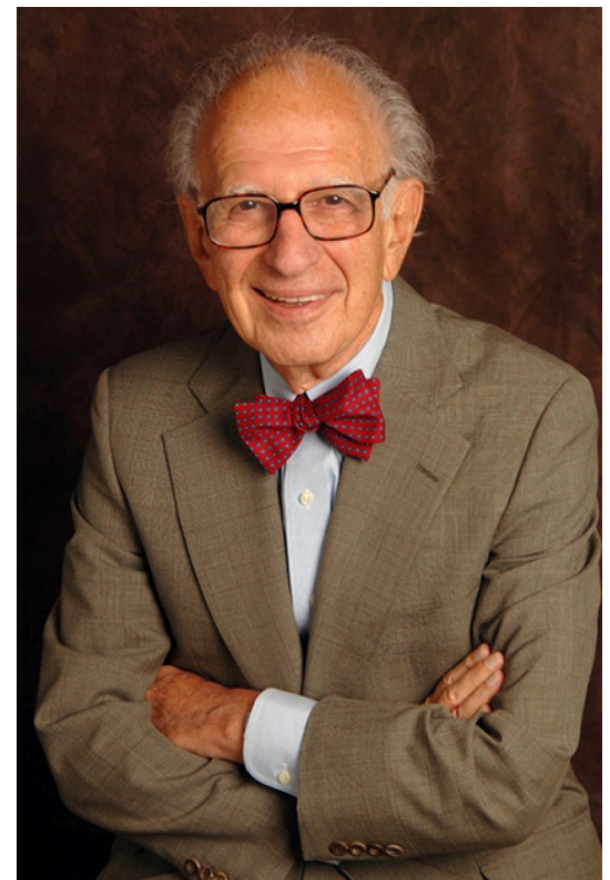
- Reasoning as internal actions for language agents
- Reasoning guides acting & acting updates reasoning



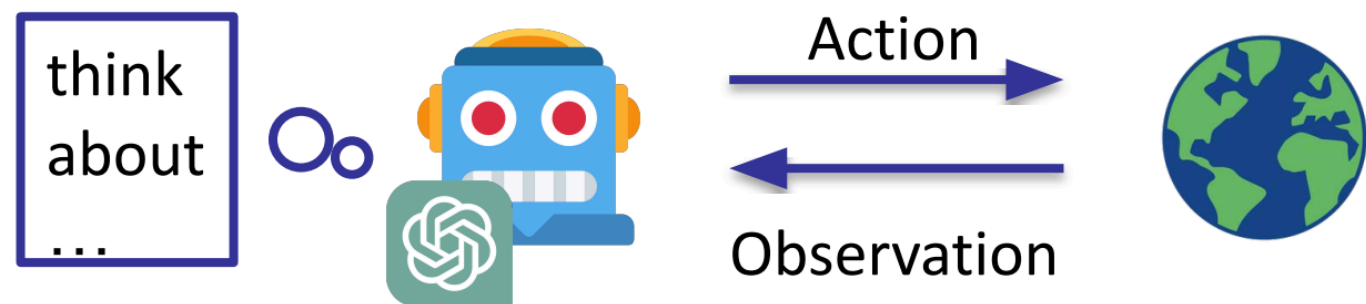
# Memory

Memory is everything. Without it, we are nothing.

— Eric Kandel



# Memory

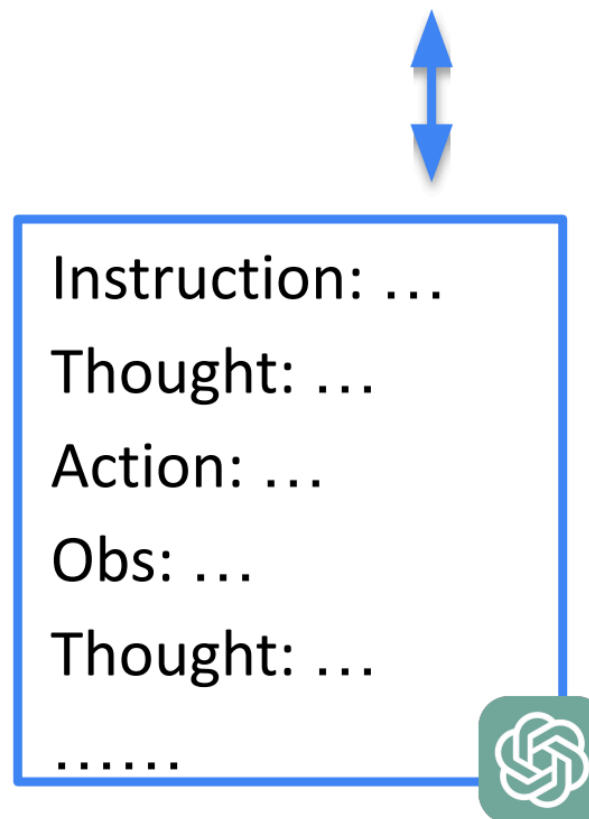


# Memory

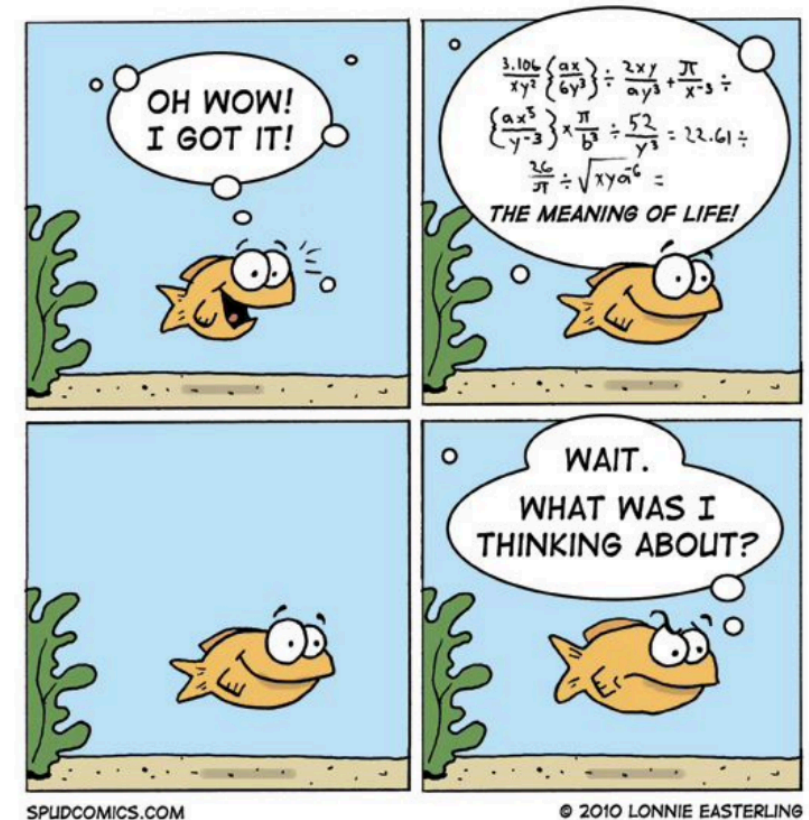


A long-term memory

- Read and write
- Stores experience, knowledge, skills, ...
- Persist over new experience



A short-term memory

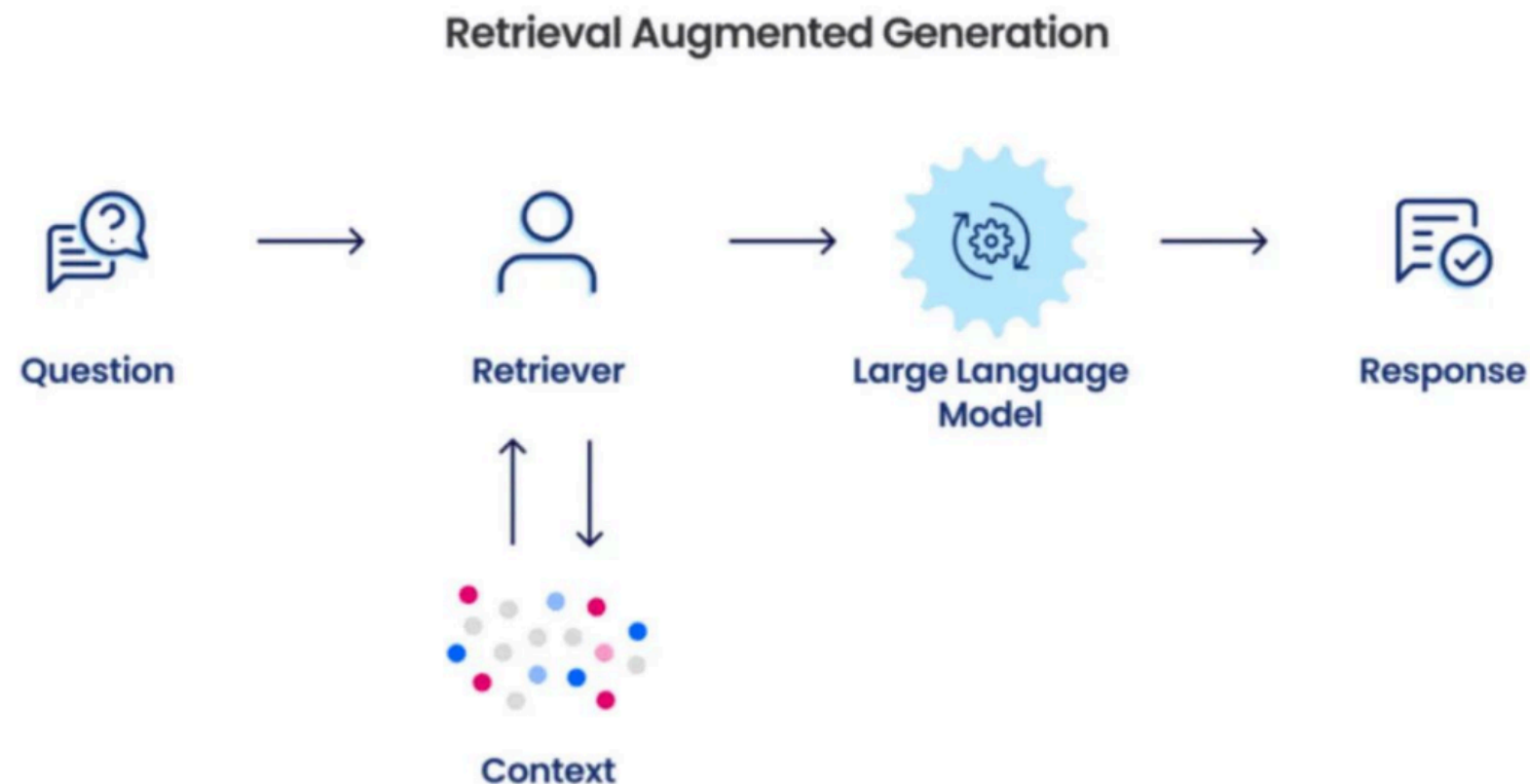


THE TRAGEDY OF A THREE SECOND MEMORY



# What about retrieval and RAG?

- We can think of the retrieval corpus as “read-only” long-term memory
  - Written by others (e.g., Wikipedia editors), not the agent itself
- Limitations
  - Can only live “others’ experience”, which might not be optimal for the agent
  - The way corpus is written might not be optimal for agent usage
- Agent memory: also be able to autonomously write to it!



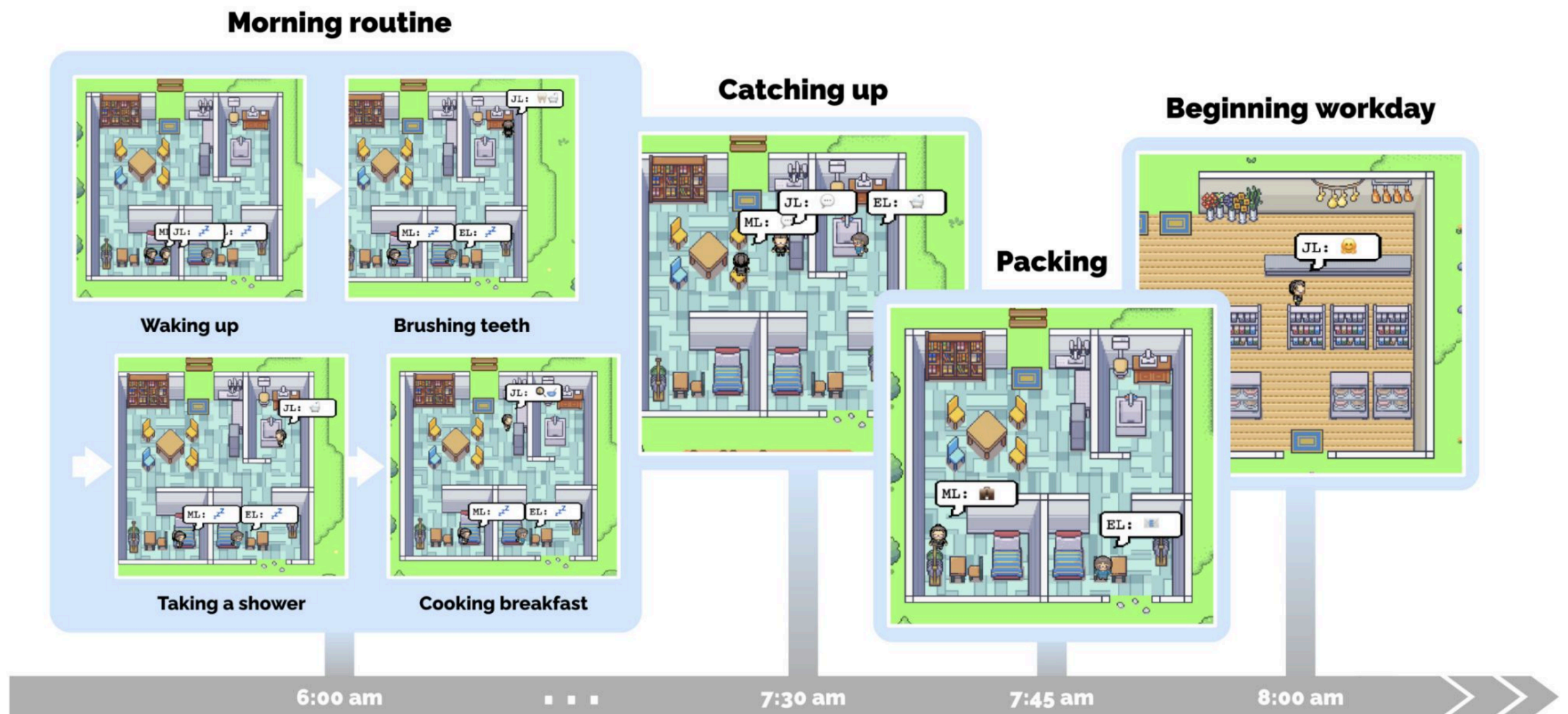
# Long-term memory (LTM): Content

- What content is stored in LTM?
- Note: here we categorize three LTMs based on memory content

Type by content	Definition	Examples
Episodic memory	Stores experience	Generative agents [Park et al., 2023]
Semantic memory	Stores knowledge	
Procedural memory	Stores skills	Voyager [Wang et al., 2023]

# Episodic memory

- Generative agents for social simulations



# Episodic memory

- Write: append-only event streams
- Read: retrieval based on heuristic scores

## Memory Stream

```
2023-02-13 22:48:20: desk is idle
2023-02-13 22:48:20: bed is idle
2023-02-13 22:48:10: closet is idle
2023-02-13 22:48:10: refrigerator is idle
2023-02-13 22:48:10: Isabella Rodriguez is stretching
2023-02-13 22:33:30: shelf is idle
2023-02-13 22:33:30: desk is neat and organized
2023-02-13 22:33:10: Isabella Rodriguez is writing in her journal
2023-02-13 22:18:10: desk is idle
2023-02-13 22:18:10: Isabella Rodriguez is taking a break
2023-02-13 21:49:00: bed is idle
2023-02-13 21:48:50: Isabella Rodriguez is cleaning up the
kitchen
2023-02-13 21:48:50: refrigerator is idle
2023-02-13 21:48:50: bed is being used
2023-02-13 21:48:10: shelf is idle
2023-02-13 21:48:10: Isabella Rodriguez is watching a movie
2023-02-13 21:19:10: shelf is organized and tidy
2023-02-13 21:18:10: desk is idle
2023-02-13 21:18:10: Isabella Rodriguez is reading a book
2023-02-13 21:03:40: bed is idle
2023-02-13 21:03:30: refrigerator is idle
2023-02-13 21:03:30: desk is in use with a laptop and some papers
on it
```

...

**Q. What are you looking forward to the most right now?**

Isabella Rodriguez is excited to be planning a Valentine's Day party at Hobbs Cafe on February 14th from 5pm and is eager to invite everyone to attend the party.

retrieval		recency	importance	relevance
2.34	=	0.91	+ 0.63	+ 0.80

ordering decorations for the party

2.21	=	0.87	+ 0.63	+ 0.71
------	---	------	--------	--------

researching ideas for the party

2.20	=	0.85	+ 0.73	+ 0.62
------	---	------	--------	--------

...

I'm looking forward to the Valentine's Day party that I'm planning at Hobbs Cafe!

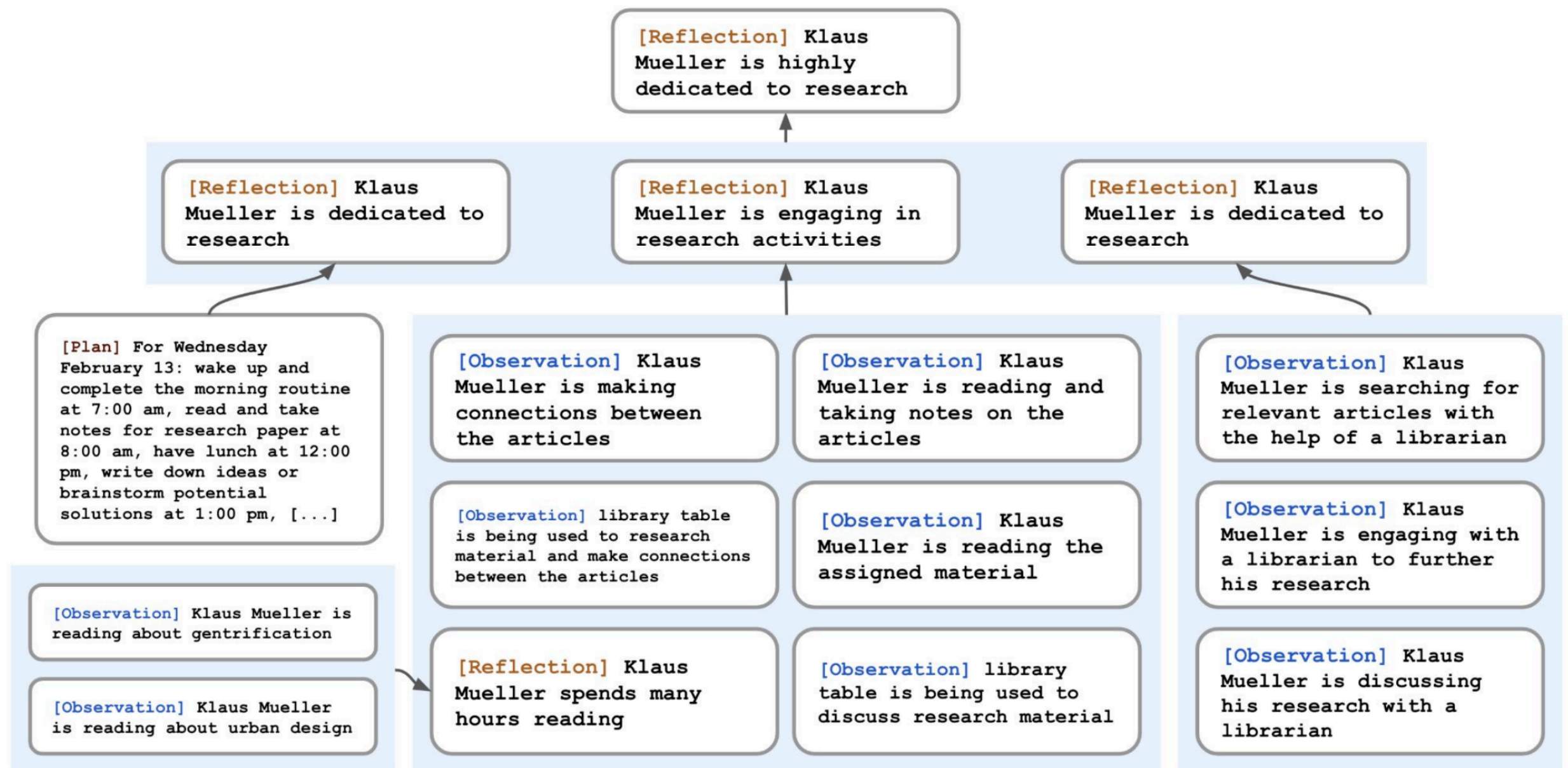


Isabella



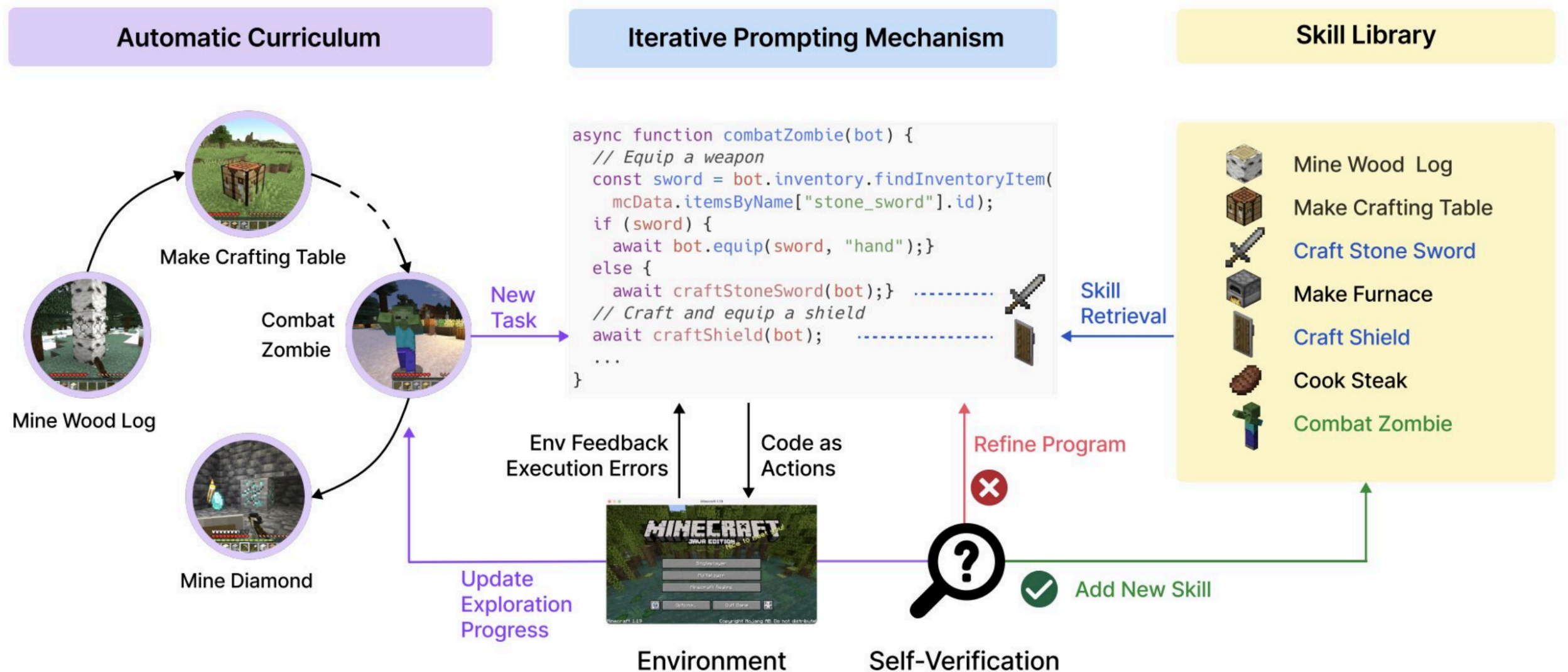
# Semantic memory

- Write: LLM reasoning over events
- Read: retrieval



# Procedural memory

- Write: Coding-based skills
- Read: embedding retrieval

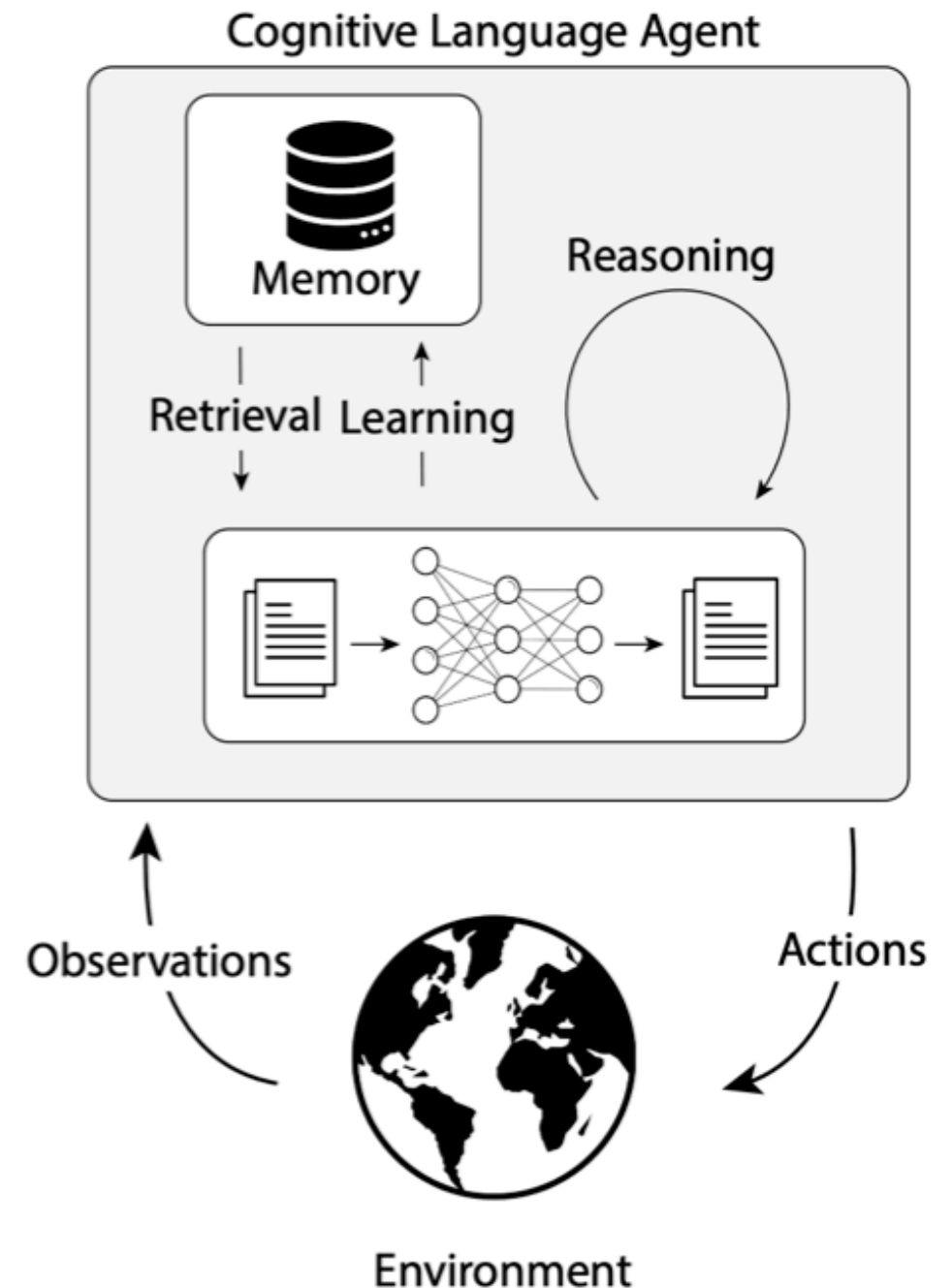


Wang, et al., 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models



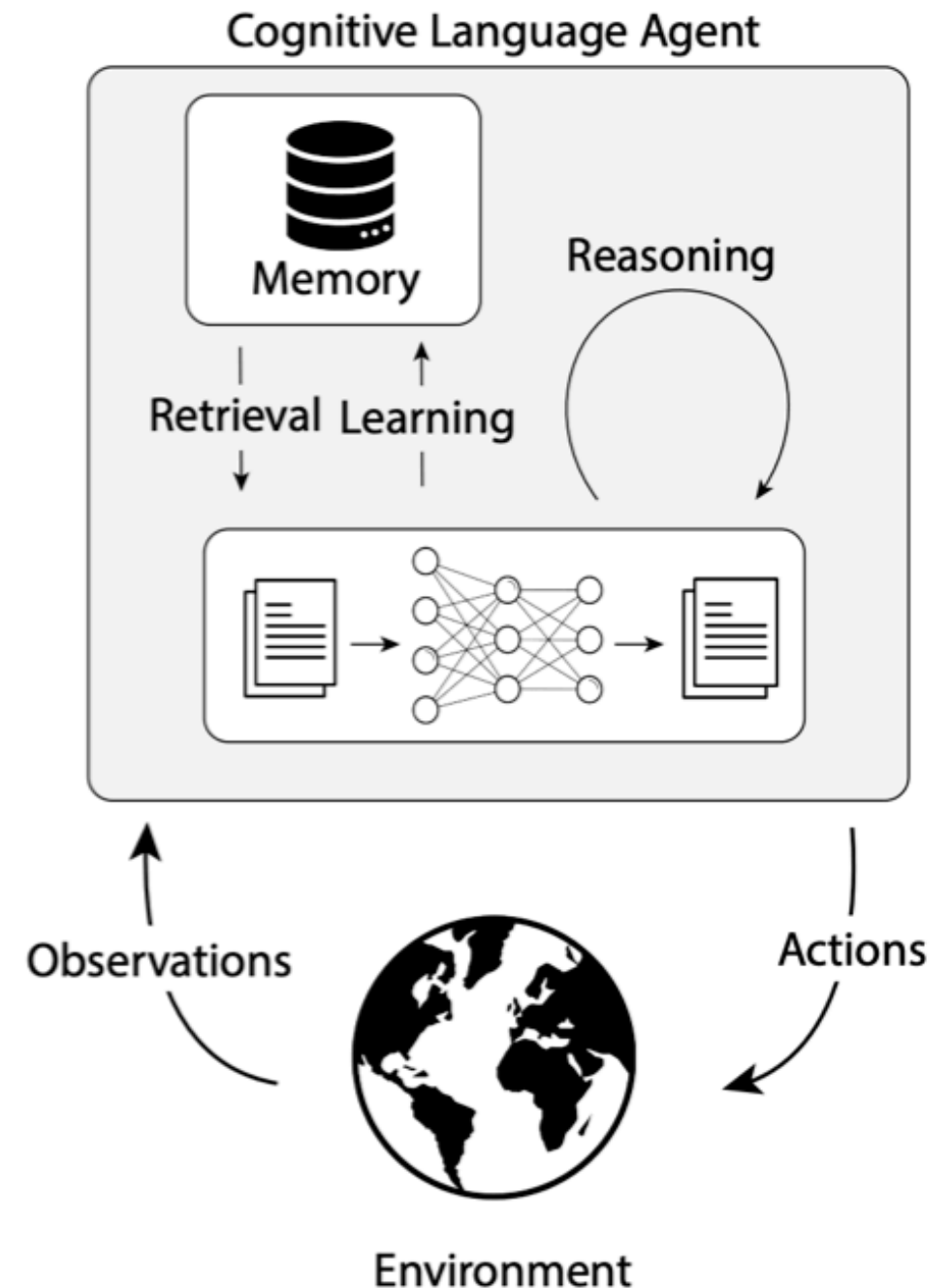
# Exercise

- Where does a cognitive language agent store long-term memory?
- Where does a agent store short-term memory?
- What's the difference between external environment vs internal memory then?



# Exercise

- Where does a cognitive language agent store long-term memory?
  - External database
  - LLM's parameters
- Where does an agent store short-term memory?
  - Prompt
- What's the difference between external environment vs internal memory then?
  - Memory contains agent-specific contents (experiences, knowledge, skills)
  - Environment is independent of all agents



# Planning: (simplified) definition

Given a goal  $G$ , decides on a sequence of actions  $(a_0, a_1, \dots, a_n)$  that will lead to a state that passes the goal test  $g(\cdot)$

- General trends in planning settings for language agents
  - Increasing expressiveness in **goal specification**, e.g., in natural language as opposed to formal language
  - Substantially expanded or open-ended **action space**
  - Increasing difficulty in automated **goal test**

# Language agent planning: web agents

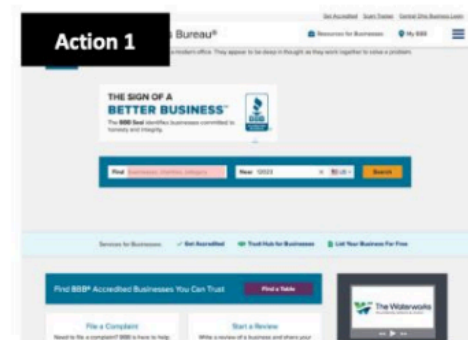
## Task Description:

Show me the reviews for the auto repair business closest to 10002.

## Action Sequence:

Target Element	Operation
1. [searchbox] Find	TYPE: auto repair
2. [button] Auto Repair	CLICK
3. [textbox] Near	TYPE: 10002
4. [button] 10002	CLICK
5. [button] Search	CLICK
6. [switch] Show BBB Accredited only	CLICK
7. [svg]	CLICK
8. [button] Sort By	CLICK
9. [link] Fast Lane 24 Hour Auto Repair	CLICK
10. [link] Read Reviews	CLICK

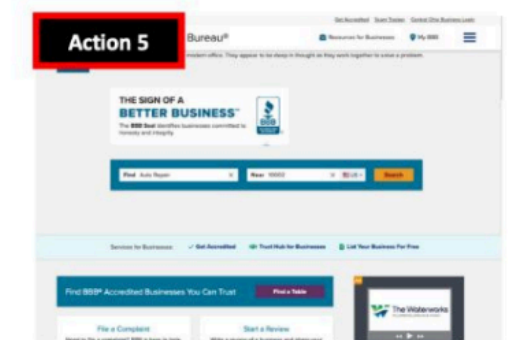
## Webpage Snapshots:



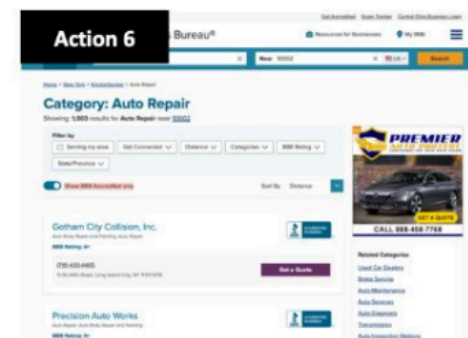
`<input name="find_text" type="search">`



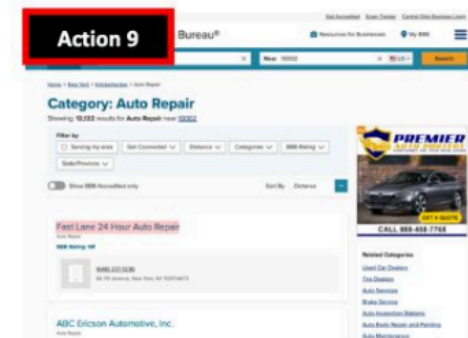
`<em>Auto Repair</em>`



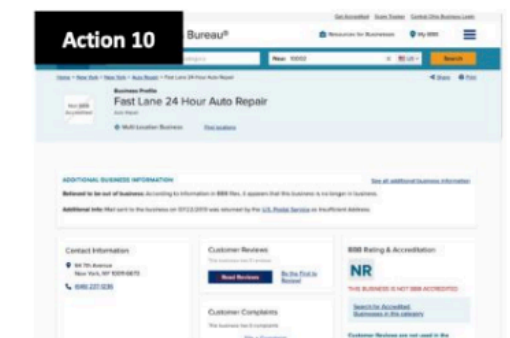
`<button>Search</button>`



`<button>Show BBB Accredited only</button>`



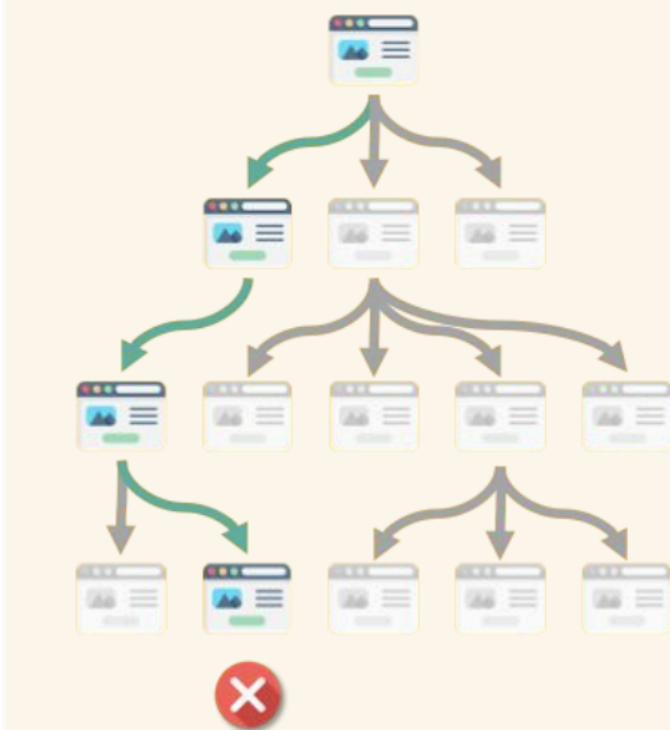
`<span>Fast Lane 24 Hour Auto Repair</span>`



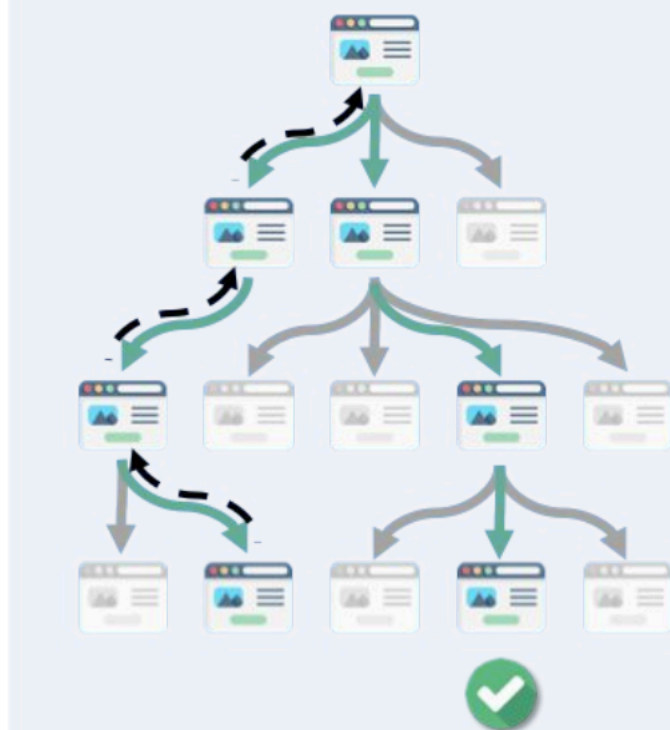
`<a href="link:xxx">Read Reviews</a>`

# Planning paradigms for language agents

(a) reactive



(b) tree search with real interactions

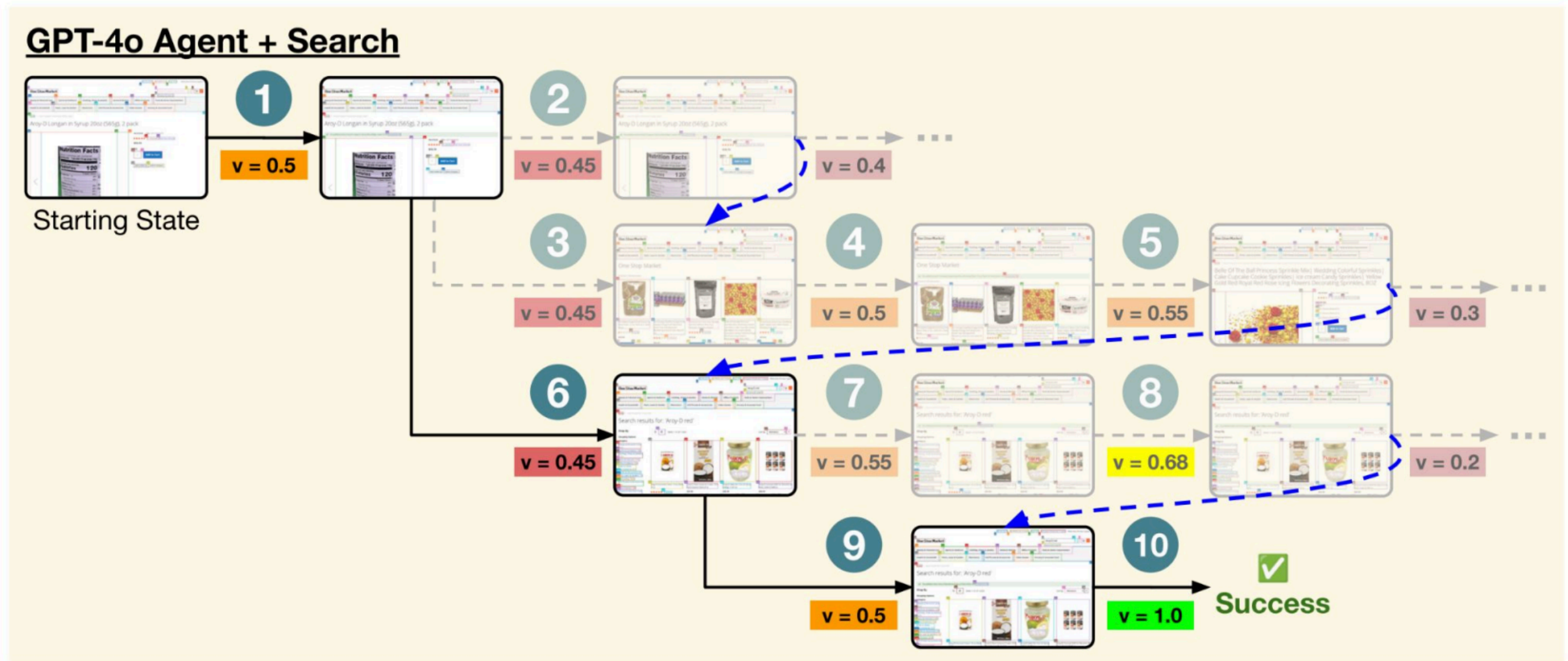


fast, easy to implement



greedy,

# Tree search with real interactions



Jing Yu Koh, Stephen McAleer, Daniel Fried, Ruslan Salakhutdinov. "Tree Search for Language Model Agents." arXiv preprint arXiv:2407.01476 (2024).

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, Yu-Xiong Wang. "Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models." ICML (2024).



# Challenges with tree search in the real world

- Many actions are state-changing and irreversible → backtracking ❌
- Safety/privacy risks
- Inference-time exploration could be slow and costly

Cancel Place Your Order - Amazon.co...

Nespresso Capsules Vertuo, Variety Pack, Medium and Dark Roast Coffee, 30 Count Coffee Pods, Brews 7.8 oz.  
\$37.50 (\$1.25 / Count)  
prime

Ships from and sold by Amazon.com

Quantity: 1 Change

Add gift options

Auto-deliver and save up to 5% on future auto-deliveries

Item often ships in manufacturer's container to reduce packaging and reveals what's inside. To change, click below.

Reduce packaging, ship in manufacturer's container

Place your order

By placing your order, you agree to Amazon's [privacy notice](#) and [conditions of use](#).

← Search or ask a question

Kohl's Dropoff FREE

Kohl's will pack, label, and ship your return for free. Just bring the item in its original manufacturer's packaging and disassemble the item (if applicable). We'll email you a QR code to ship your return. Show it to a store associate at any Kohl's store.

[Find a participating Kohl's store](#)

Printer not required.

☐ The UPS Store locations only—no label needed \$6.99

☐ Amazon Dropoff—box and label needed FREE

2 OTHER RETURN OPTIONS

Refund summary \$13.21

Confirm your return

Verify mobile number

A text with a One Time Password (OTP) has been sent to your mobile number: 8058671234 [Change](#)

Enter OTP: [Resend code](#)

Create your Amazon account

By creating an account, you agree to Amazon's [Conditions of Use](#) and [Privacy Notice](#).

← Search or ask a question

Location Disabled

AddressBook/Checkout

Your current location will be used to assist in adding a new address to your Amazon address book.

Amazon Cash

We use your location to find nearby stores where you can add money to your Amazon balance with Amazon Cash.

Branded Store Experience Location

We use your location to power branded store experiences.

Branded Store Experience Location-based Augmented Reality

We use your camera, motion, and location to power branded store experiences. Requires camera, motion, and location.

Campus pickup

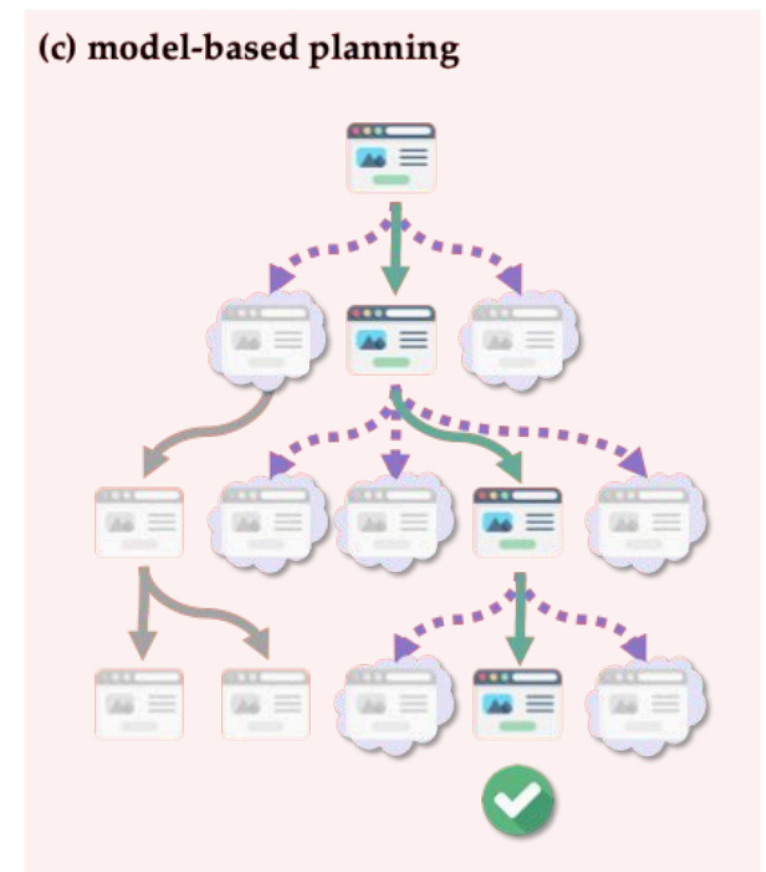
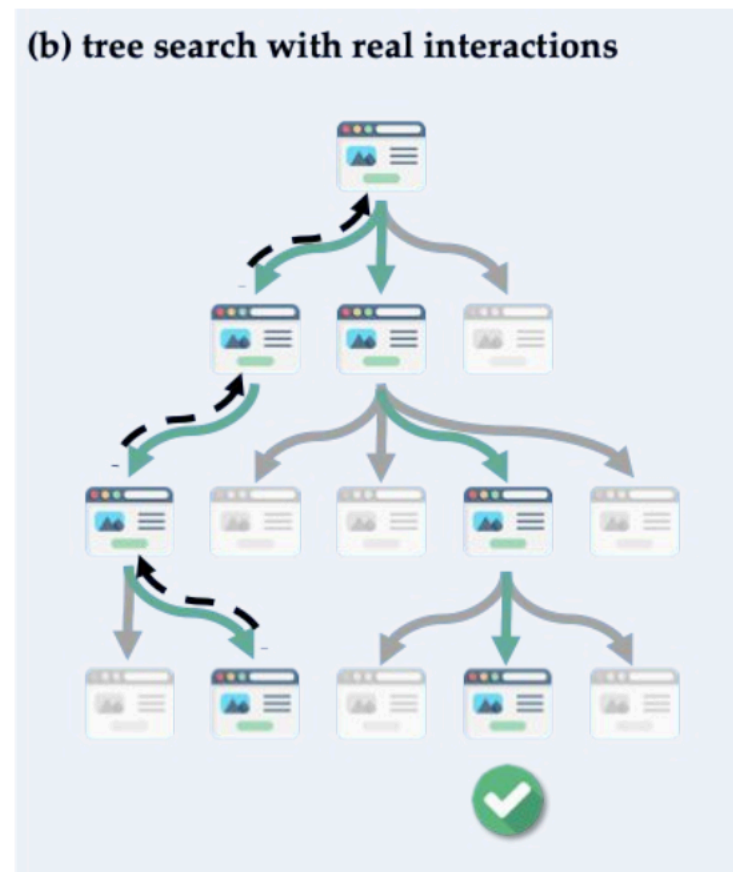
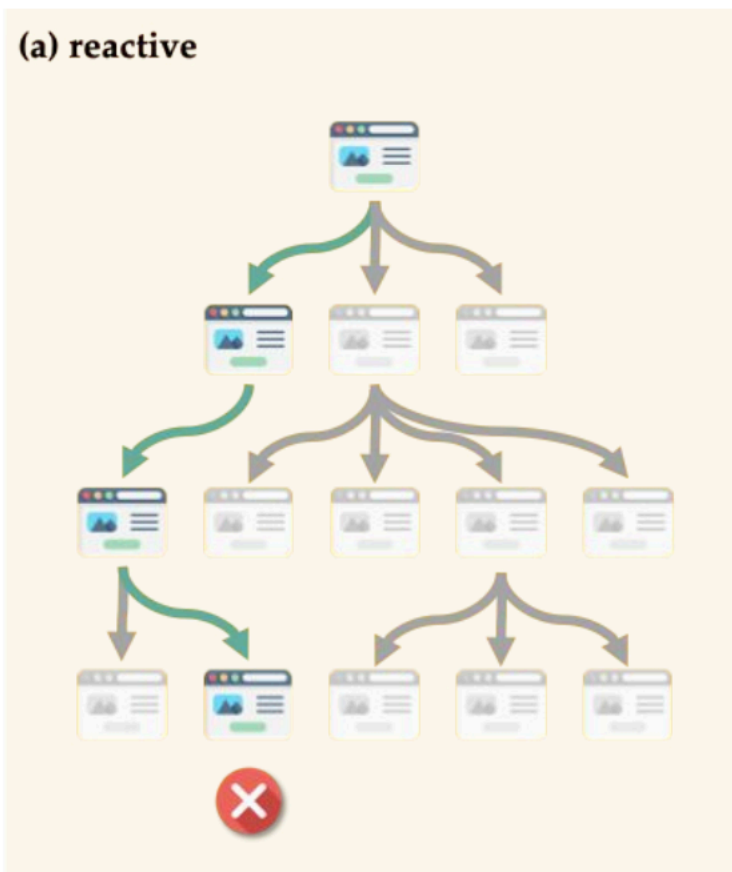
We'll use your location to show the nearby pickup points

Delivery Location

We use your location to improve your shopping experience, ensuring you only see products and delivery options available in your area.

# Planning paradigms for language agents

- Train a world model that knows what is going to happen



😊 fast, easy to implement

😐 greedy,

😊 systematic exploration

😐 irreversible actions,  
unsafe, slow

😊 faster, safer,  
systematic exploration

😐 how to get a world model?

Still open question?

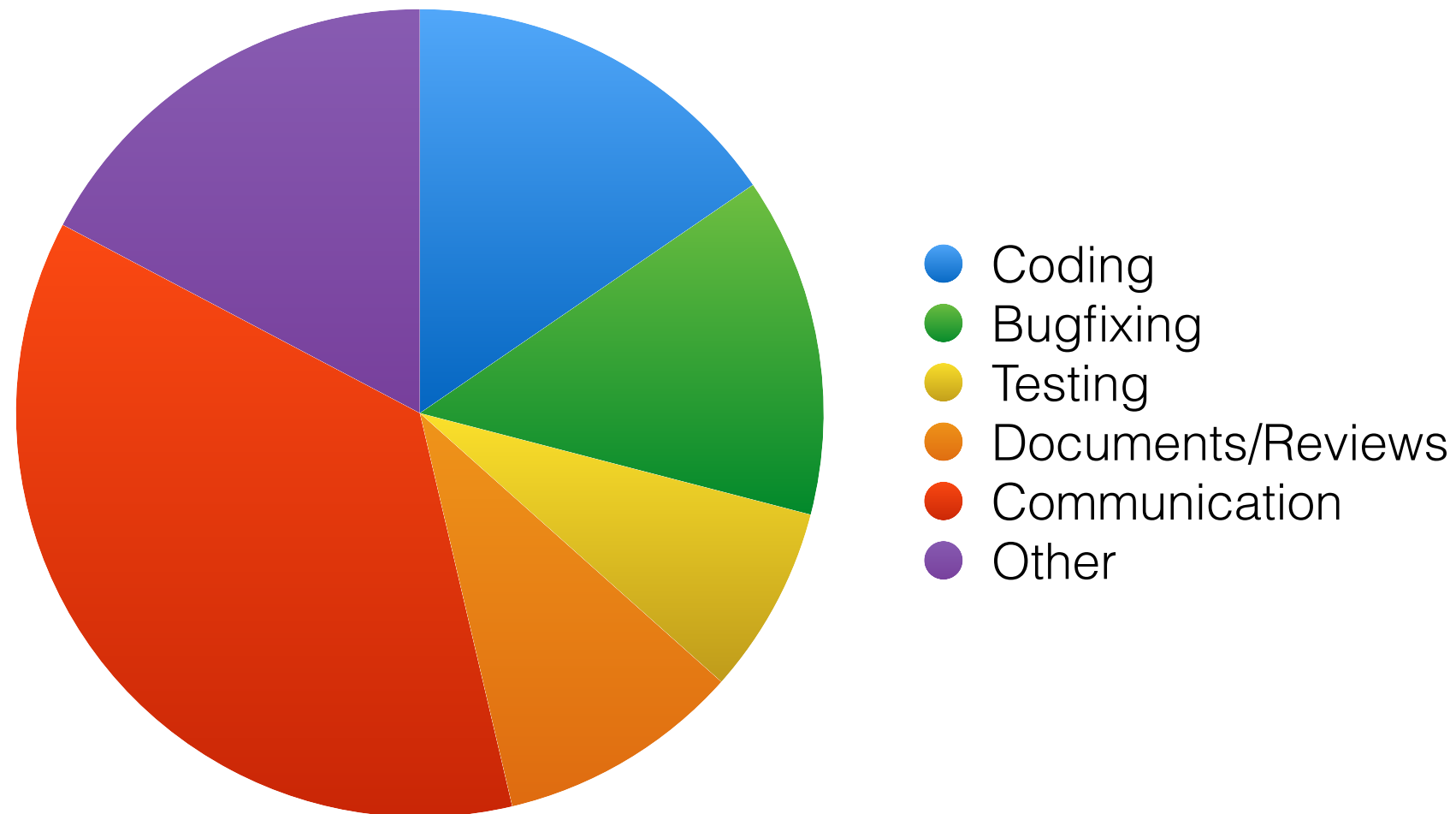
# Part III: LLM Coding Agents for Software Development

More and more major businesses and industries are being run on software and delivered as online services—from movies to agriculture to national defense. [...] Over the next 10 years, I expect many more industries to be disrupted by software [...].

— Marc Andreessen - Why Software is Eating the World (2011)

**If we gave everyone the ability to quickly write software to achieve their goals, what could they do?**

# What is Involved in Developing Software?



Today was a Good Day: The Daily Life of Software Developers  
Meyer et al. 2019

# How Can We Support Developers?

(Neubig 2024)

Level	Self Driving	Software Development
0: No Automation	Manual driving	Manual Coding
1: Driver Assistance/ Code Completion	Adaptive cruise control/braking	Copilot/Cursor code completion
2: Partial Automation	Tesla's autopilot	Copilot chat refactoring
3: Conditional Automation	Mercedes-Benz drive pilot	DiffBlue test generation, Transcoder code porting
4: High Automation	Cruise self-driving vehicles	Devin/OpenDevin end-to-end development
5: Full Automation	...	...



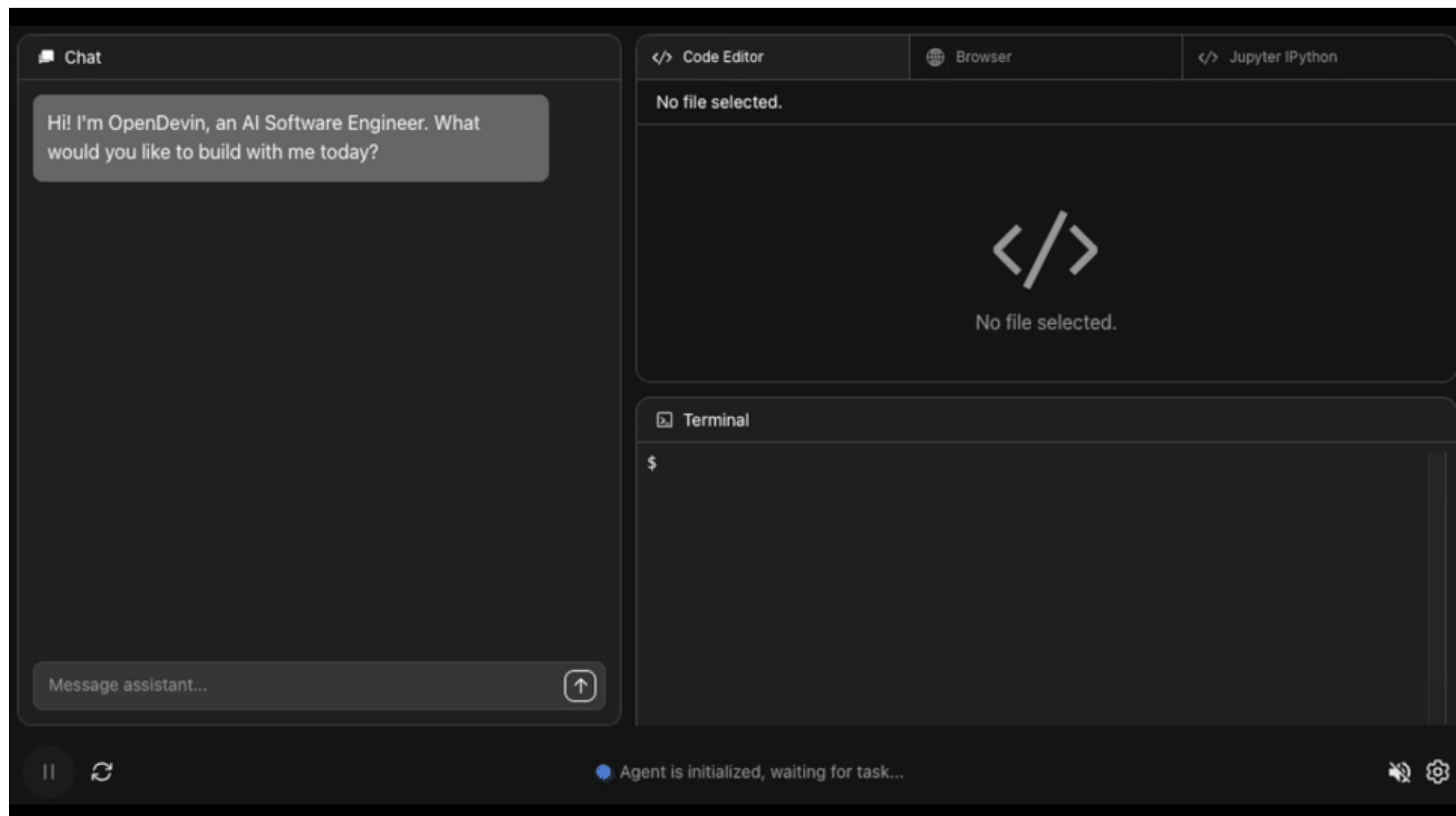
# Development Copilots

- Work synchronously with the developer to ease writing code
- e.g. **Github Copilot/Cursor**

```
tests > unit > test_action_serialization.py > ...
147
148 def test_modify_task_action_serialization_deserialization():
149     original_action_dict = {
150         'action': 'modify_task',
151         'args': {'task_id': 1, 'state': 'Test state.', 'thought': ''},
152     }
153     ✨ serialization_deserialization(original_action_dict, ModifyTaskAction)
154
```

# Development Agents

- For coding (e.g. SWE-Agent, Aider)
- For broader development (e.g. Devin, OpenHands)

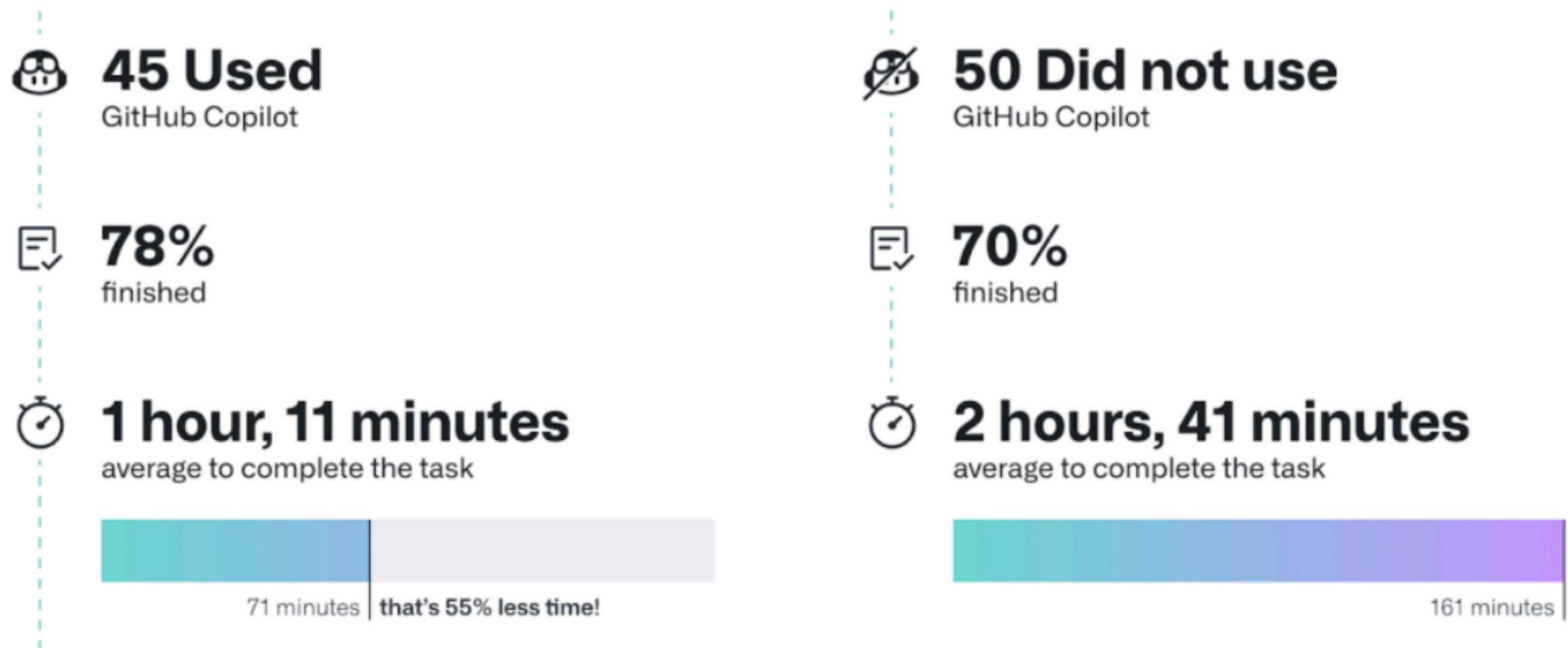


# Non-coding Tasks

- Gathering information from Github
- Managing task resolution software
- Setting up web infrastructure

# How Promising?

- Code generation leads to large improvements in productivity (Github 2022)



# Challenges in Development Agents

- Defining the Environment
- Designing an Observations/Actions
- Code Generation (atomic actions)
- File Localization (exploration)
- Planning and Error Recovery
- Safety

# Types of Environments

- **Actual Environments:**

- *Source Repositories:* Github, Gitlab
- *Task Management Software:* Jira, Linear
- *Office Software:* Google Docs, Microsoft Office
- *Communication Tools:* Gmail, Slack

- **Testing Environments:**

- Mostly focused on coding!
- Developers do more, e.g. browse the web (next session)



# Data Science Notebooks: ARCADE

(Yin et al. 2022)

- Data science notebooks (e.g. Jupyter) allow for incremental implementation
- Allows evaluation of code in context

```
[1] import pandas as pd
C1 df = pd.read_csv('dataset/Gamepass_Games_v1.csv')

[2] U1 Extract min and max hours as two columns ✕
C2 def get_avg(x):
    try: return float(x[0]) , float(x[1])
    except: return 0, 0

    df['min'], df['max'] = zip(*df['TIME'].str.replace(
        ' hours', '').str.split("-").apply(get_avg))

[3] df['ADDED'] = pd.to_datetime(
C3 df['ADDED'], format="%d %b %y", errors='coerce')

[4] U2 In which year was the most played game added? NA
C4 df['GAMERS'] = df['GAMERS'].str.replace(
    ',', '').astype(int)
    added_year = df[df['GAMERS'].idxmax()]['ADDED'].year

[5] U3 For each month in that year, how many games that NA
C5 has a rating of more than four?

    df[(df['ADDED'].dt.year == added_year) &
        (df['RATING'] > 4)].groupby(
        df['ADDED'].dt.month)['GAME'].count()

[6] U4 What is the average maximum completion time for NA
C6 all fallout games added in 2021?

    fallout = df[df['GAME'].str.contains('Fallout')]
    fallout.groupby(fallout['ADDED'].dt.year).get_group(
        2021)['max'].mean()

[7] U5 What is the amount of games added in each year NA
C7 for each month? (show a table with index as years,
    columns as months and fill null values with 0)

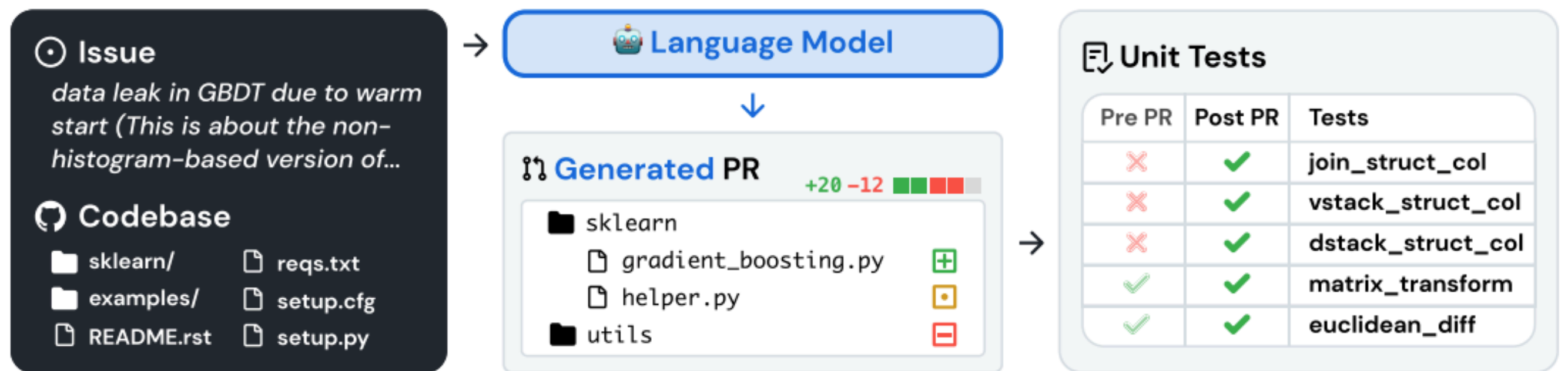
    pd.pivot_table(df, index=df['ADDED'].dt.year, ...,
        aggfunc=np.count_nonzero,
        fill_value='0').rename_axis(
        index='Year', columns='Month')
```

Figure 1: An example of a computational notebook adapted from our dataset, with examples of reading and preprocessing data (cell  $c_1$ ), data wrangling (cell  $c_2, c_3$ ), and data analysis (cells  $c_3 - c_7$ ). Annotated NL intents are shown in green.

# Dataset: SWEBench

(Jiminez et al. 2023)

- Issues from GitHub + codebases -> pull request



- Requires long-context understanding, precise implementation

# Action Spaces

- Coding Agents
  - Understand repository structure
  - Read in existing code
  - Modify or produce code
  - Run code and debug

# Example: CodeAct (Wang et al. 2024)

- Interact w/ the environment through code

**Instruction:** Determine the most cost-effective country to purchase the smartphone model "CodeAct 1". The countries to consider are the USA, Japan, Germany, and India.

**Available APIs**

[1] lookup\_rates(country: str) -> (float, float)  
[2] convert\_and\_tax(price: float, exchange\_rate: float, tax\_rate: float) -> float  
[3] estimate\_final\_price(converted\_price: float, shipping\_cost: float) -> float  
[4] lookup\_phone\_price(model: str, country: str) -> float  
[5] estimate\_shipping\_cost(destination\_country: str) -> float

### LLM Agent using [Text/JSON] as Action

**Think** I should calculate the phone price in USD for each country, then find the most cost-effective country.

**Action** Text: lookup\_rates, Germany  
JSON: {"tool": "lookup\_rates", "country": "Germany"}

**Environment** 1.1, 0.19

**Action** Text: lookup\_phone\_price, CodeAct 1, Germany  
JSON: {"tool": "lookup\_phone\_price", "model": "CodeAct 1", "country": "Germany"}

**Environment** 700

**Action** Text: convert\_and\_tax, 700, 1.1, 0.19  
JSON: {"tool": "convert\_and\_tax", "price": 700, "exchange\_rate": 1.1, "tax\_rate": 0.19}

**Environment** 916.3

[... interactions omitted (look up shipping cost and calculate final price) ...]

**Action** Text: lookup\_rates, Japan  
JSON: {"tool": "lookup\_rates", "country": "Japan"}

[... interactions omitted (calculate final price for all other countries) ...]

**Response** The most cost-effective country to purchase the smartphone model is Japan with price 904.00 in USD.

### CodeAct: LLM Agent using [Code] as Action

**Think** I should calculate the phone price in USD for each country, then find the most cost-effective country.

**Action**

```
countries = ['USA', 'Japan', 'Germany', 'India']
final_prices = {}

for country in countries:
    exchange_rate, tax_rate = lookup_rates(country)
    local_price = lookup_phone_price("xAct 1", country)
    converted_price = convert_and_tax(
        local_price, exchange_rate, tax_rate
    )
    shipping_cost = estimate_shipping_cost(country)
    final_price = estimate_final_price(converted_price, shipping_cost)
    final_prices[country] = final_price

most_cost_effective_country = min(final_prices, key=final_prices.get)
most_cost_effective_price = final_prices[most_cost_effective_country]
print(most_cost_effective_country, most_cost_effective_price)
```

**Environment** 1.1, 0.19

**Response** The most cost-effective country to purchase the smartphone model is Japan with price 904.00 in USD.

**Fewer Actions Required!**

**Control & Data Flow of Code Simplifies Complex Operations**

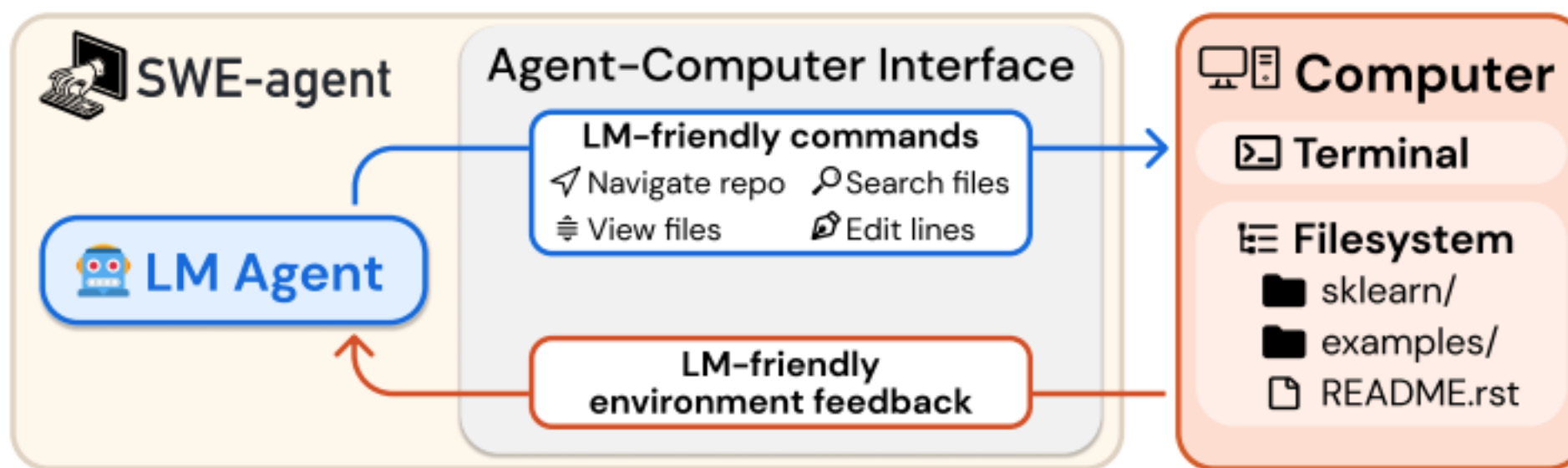
**Re-use 'min' Function from Existing Software Infrastructures (Python library)**

- Can execute bash commands, Jupyter commands
- Faster resolution, higher success than direct tool use

# Example: SWE-Agent

(Yang+Jimenez et al. 2024)

- Define specialized tools that make it possible to efficiently explore repositories and edit code



## Observation (File Viewer)

```
[File: solvers/diophantine.py (3093 lines total)]
(400 lines above)
401:     elif total_degree == 3:
402:         diop_type = "cubic_thue"
403:
404:     elif (total_degree > 3):
405:         if all(coeff[k] == 1):
406:             diop_type = 'general_sum_even'
407:
408:     if diop_type is not None:
409:         return var, coeff, diop_type
(2684 lines below)
```

## Response (Thought + Edit Action)

We need to add a check to ensure that all powers are the same. Let's edit the code to include this check.

```
edit 404:407
    elif (total_degree > 3 and
          len(set(k.exp[k.is_Pow])) == 1):
        if all(coeff[k] == 1):
            diop_type = 'general_sum_even'
end_of_edit
```



# Example: OpenHands

(Wang et al. 2024)

## Event Stream

where all Action-Observation(s) are tracked

- Defines “event stream” for coding, execution, and browsing actions/observations
- Implements SWE-agents style actions as “agent skills” that can be called

### [1] Message (source=user)

Can you create a list of numbers from 1 to 10, and create a web page to display them at port 5000?

### [2] IPythonRunCellAction (source=agent)

Sure! Let me create a file first:  
<execute\_ipython>  
create\_file('app.py')  
</execute\_ipython>

### [3] IPythonRunCellObservation

[File: /workspace/app.py (1 lines total)]  
1|  
[File app.py created.]

### [4] IPythonRunCellAction (source=agent)

I will write the Python code for starting a server.  
<execute\_ipython>  
EDITED\_CODE="""[Omitted for Brevity]"""  
edit\_file(start=1, end=1, content=EDITED\_CODE)  
</execute\_ipython>

### [5] IPythonRunCellObservation

[... Omitted for Brevity ...]  
4|@app.route('/')  
5|def index():  
6| numbers = list(range(1, 11))  
7| return str(numbers)  
[... Omitted for Brevity ...] [File updated.]

### [6] CmdRunAction (source=agent)

I have created a Python file 'app.py' that will display a list of numbers from 1 to 10 when you run it. Let me run the server for you:  
<execute\_bash>  
python3 app.py > server.log 2>&1 &  
cat server.log  
</execute\_bash>

### [7] CmdRunObservation

[1] 124  
\* Serving Flask app 'app'  
\* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
\* Running on http://127.0.0.1:5000  
Press CTRL+C to quit

### [8] BrowseInteractiveAction (source=agent)

The server is running on port 5000 with PID 124. Let me browse to the page at http://127.0.0.1:5000  
<execute\_browse>  
goto("http://127.0.0.1:5000")  
</execute\_browse>

### [9] BrowserOutputObservation

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# File Localization



# LLM-based Localization

- Finding the correct files given user intent

## **What problem or use case are you trying to solve?**

When in confirmation mode it's not possible to give instructions in between steps. You have to reject an action and it seems like it doesn't know that the action was rejected.

## **Describe the UX of the solution you'd like**

The simplest would be to have a third option, confirm action and wait. This way the action is confirmed but before it tries to take the next step you are able to give some feedback. Also if it somehow knows the action was rejected that would be helpful as well so when you do reject an action it knows that action wasn't taken.

<https://github.com/All-Hands-AI/OpenHands/issues/4259>

- Which JavaScript file should I modify?
- Analogous to environment understanding / exploration problems in other agents

# Solution 1:

## Offload to the User

- Experienced users familiar with prompting and the project can specify which files to use

In `.github/workflows/openhands-resolver.yml` and `.github/workflows/openhands-resolver-experimental.yml`, we should check to make sure that all required environment variables are set before running any additional workflows. If all of the variables are not set, we can fail immediately with an error.

<https://github.com/All-Hands-AI/openhands-resolver/issues/146>

# Solution 2:

## Prompt the Agent w/ Search Tools

- e.g. SWE-agent provides a tool for searching repositories

### No Search

Agent performs manual search with `ls + cd`, or uses `grep` but struggles to interpret the result

🗨 First, I should list out all the files in the codebase that may contain the definition of `PVSystem`.

\$ `ls`

```
benchmarks/  setup.cfg
docs/        versioneer.py
pvlib/       README.md
setup.py     ...
```

🗨 It's most likely the source file is in `pvlib/` folder. Let's change directory.

\$ `cd pvlib/`

### Iterative Search

Actions to show next/prev search result are repeated many times until results are exhausted

🗨 I will use the ``search_all`` command to find occurrences of `"PVSystem"`...

\$ `search "PVSystem"`

```
[Result 1/15 (Line 70)]
[File: pvlib/pvsystem.py]
22:# a dict of required pa...
23:_DC_MODEL_PARAMS = {
24:    'sapm': {
(97 more lines omitted)
```

🗨 This is not the `PVSystem` class def...

\$ `next`

### Summarized Search

- (1) Show all results in single output
- (2) Tell agent to retry if too many results.

🗨 The next step is to locate the ``PVSystem`` class in the codebase and modify the constructor...

\$ `search "class PVSystem"`

```
Found 1 matches for "class
PVSystem" in /pvlib-python:
/pvlib__pvlib-python/pvlib/
pvsystem.py (1 matches)
End of matches
```

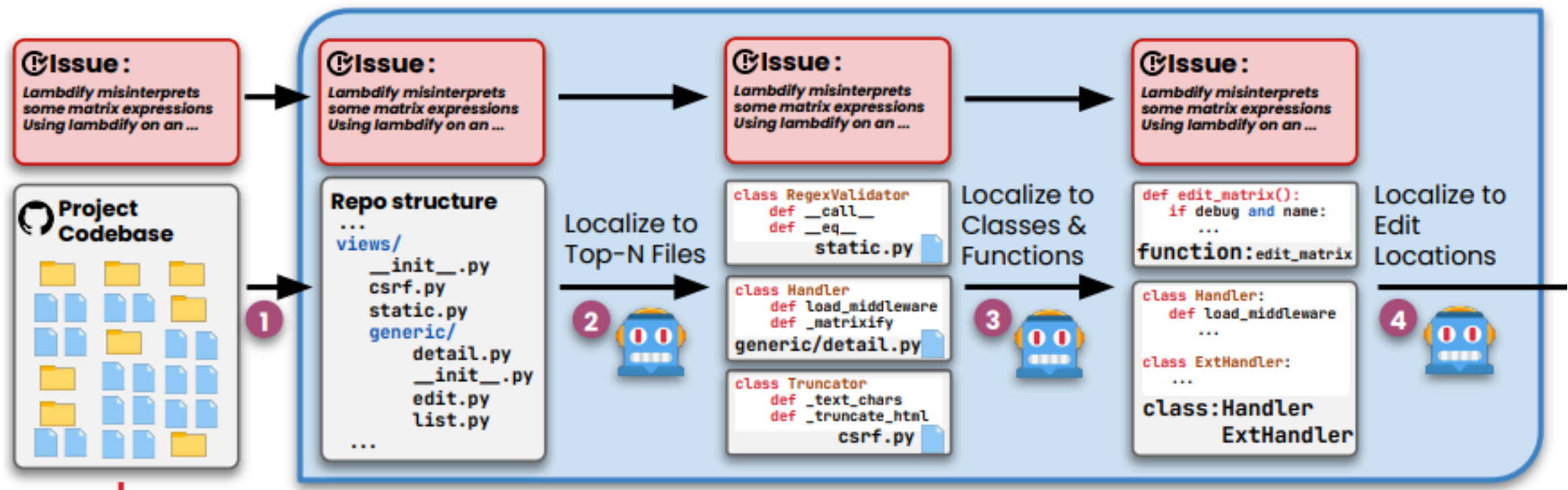
🗨 It's located in ``pvlib/pvsystem.py``...

\$ `open pvlib/pvsystem.py`

# Solution 3:

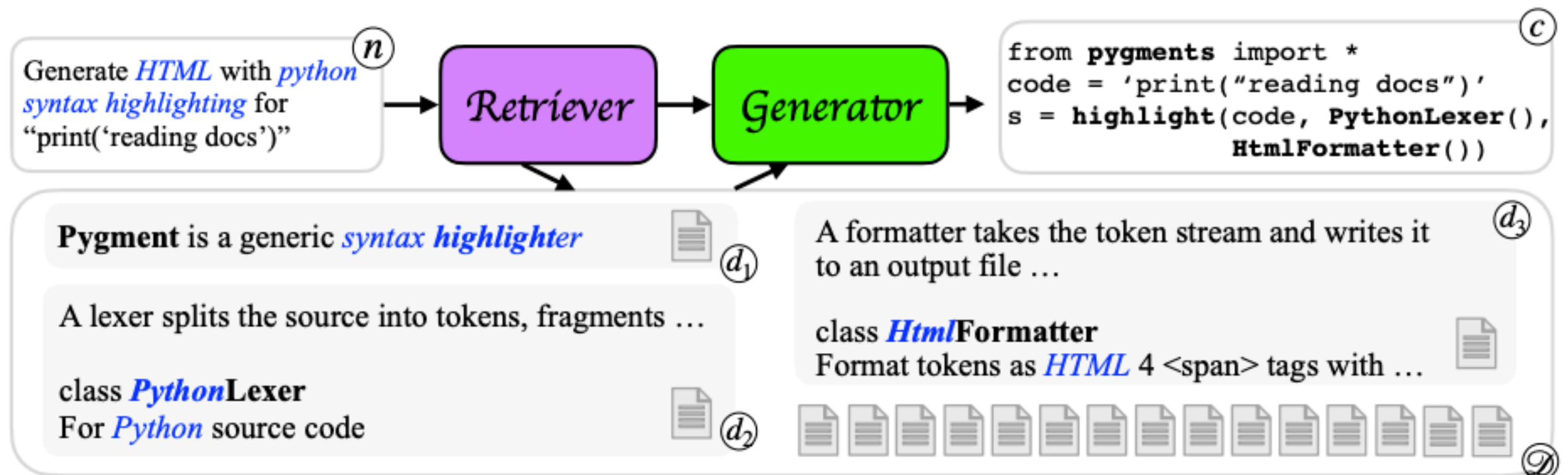
## A-priori Map the Repo

- Create a map of the repo and prompt agent with it
- Aider repomap creates a tree-structured map of the repo
- Agentless (Xia et al. 2024) does a hierarchical search for every issue



# Solution 4: Retrieval-augmented Code Generation

- Retrieve similar code, and fill it in with a retrieval-augmented LM (e.g. CodeRAGBench, Wang+Asai et al. 2024)
- Particularly, in code there is also documentation, which can be retrieved (Zhou et al. 2022)



- Unsolved issue: when to perform RAG in agent

# Planning and Error Recovery

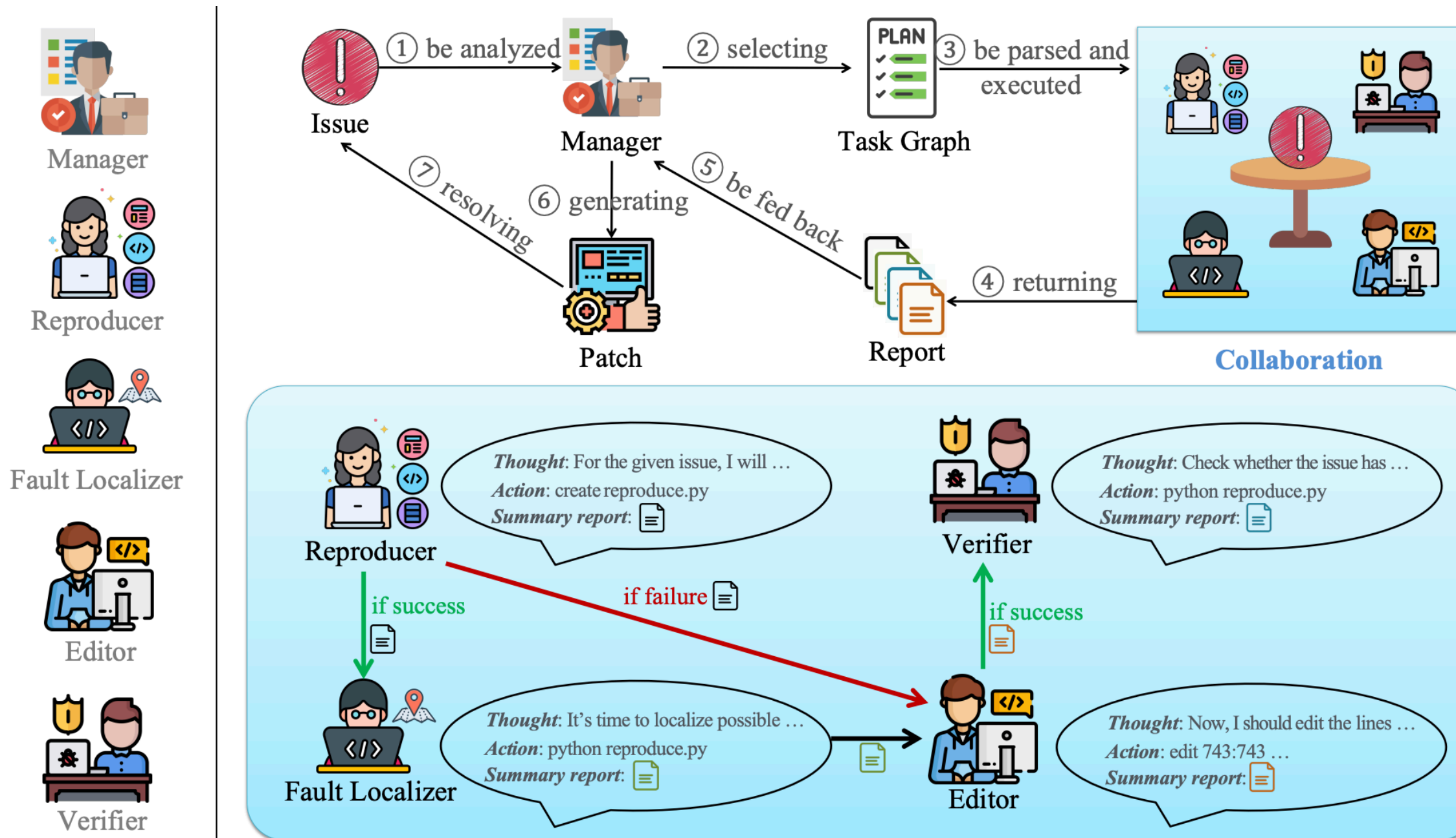


# Hard-coded Task Completion Process

- e.g. Agentless (Xie et al. 2024) has a hard-coded progress of
  - File Localization
  - Function Localization
  - Patch Generation
  - Patch Application

# LLM-Generated Plans

- LLM-generated planning step, then one or more executors
- CodeR [Chen et al. 2024]



# Coding Agents: Takeaway

- Copilots already very useful, code agents getting there
- Current challenges: code LLMs, editing, localization, planning, safety
- Future directions:
  - Agentic training methods
  - Human-in-the-loop
  - Broader software tasks than coding

# Questions?