Artificial Intelligence

CSC411

Lecture Set-03

22 BSCS

Dr. Ali Asghar Manjotho

Assistant Professor, CSE-MUET

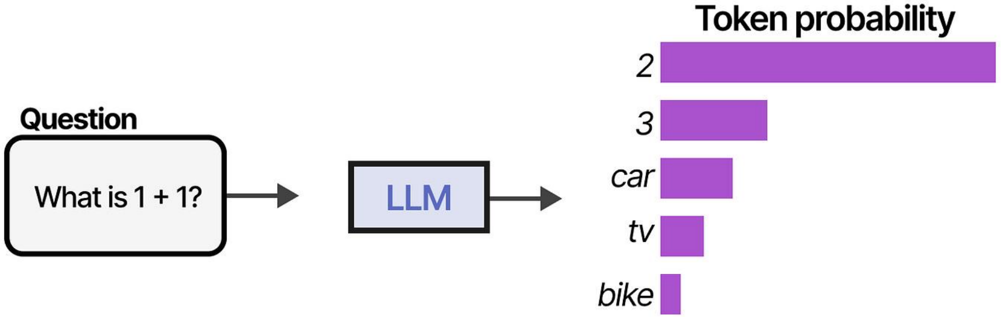
ali.manjotho@faculty.muet.edu.pk
ali.manjotho.ali@gmail.com

Contents

- Modern AI Agents (LLM Agents)
- Memory
  - Short-Term Memory
  - Long-Term Memory
- Tools

Modern AI Agents (LLM Agents)

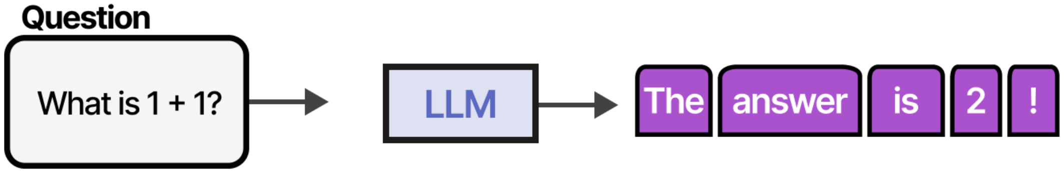
- Modern AI agents are also referred to as **LLM Agents**.
- An LLM agent is an AI system that goes beyond simple text production.
- It uses a large language model (LLM) as its central computational engine, allowing it to carry on conversations, do tasks, reason, and display a degree of autonomy.
- Traditionally, an LLM does nothing more than **next-token prediction**.



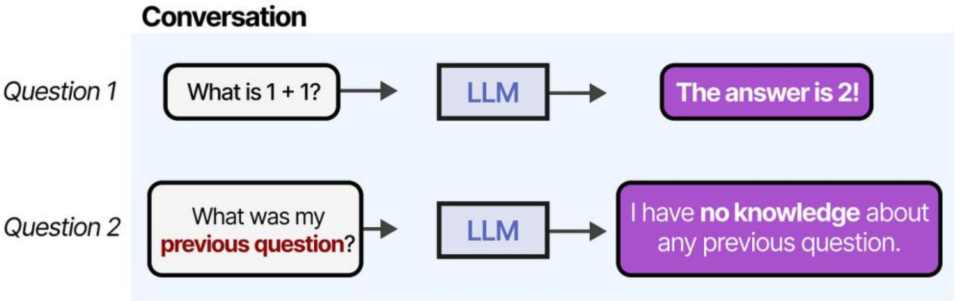
By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

Modern AI Agents (LLM Agents)

- By sampling many tokens in a row, we can mimic conversations and use the LLM to give more extensive answers to our queries.



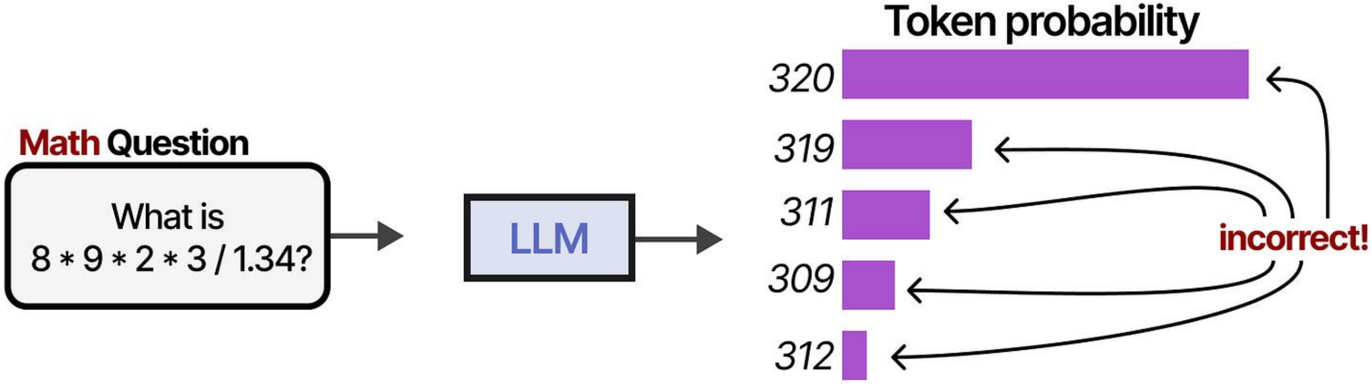
- However, when we continue the "conversation", any given LLM will showcase one of its main disadvantages. It does not remember conversations!


Conversation
Question 1
What is 1 + 1?
LLM
The answer is 2!
Question 2
What was my previous question?
LLM
I have no knowledge about any previous question.

By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

Modern AI Agents (LLM Agents)

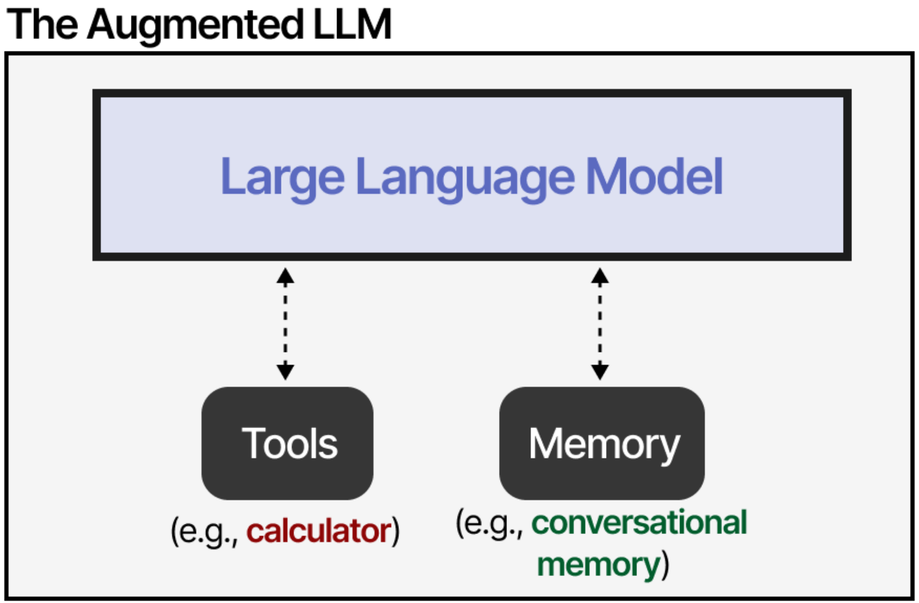
- There are many other tasks that LLMs often fail at, including basic math like multiplication and division:



- We can compensate for their disadvantage through external tools, memory, and retrieval systems.

Modern AI Agents (LLM Agents)

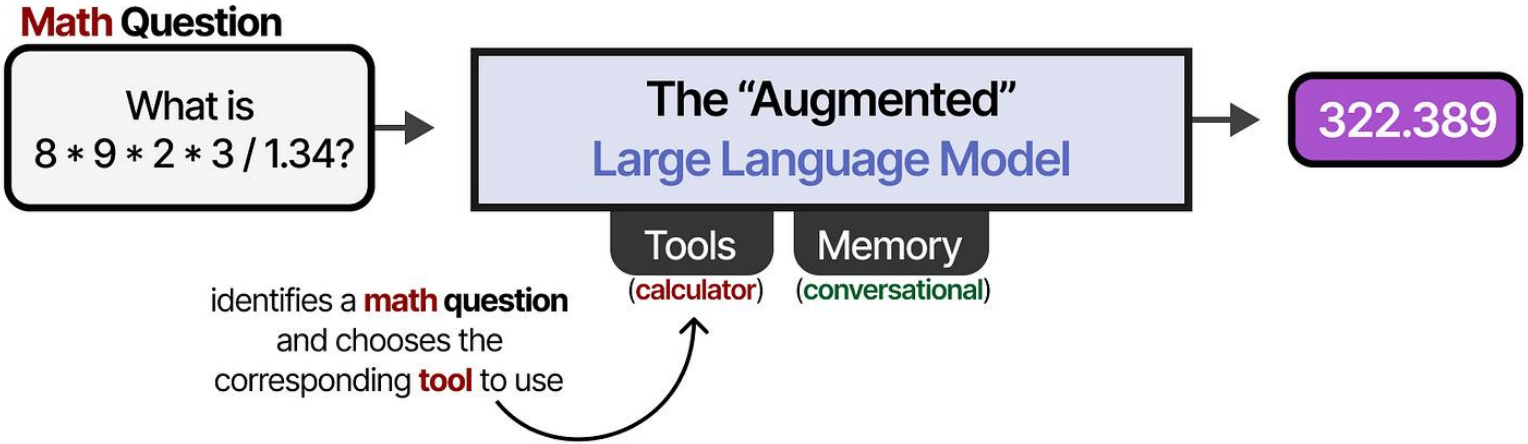
- Through external systems, the capabilities of the LLM can be enhanced. Anthropic calls this "**The Augmented LLM**".



By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

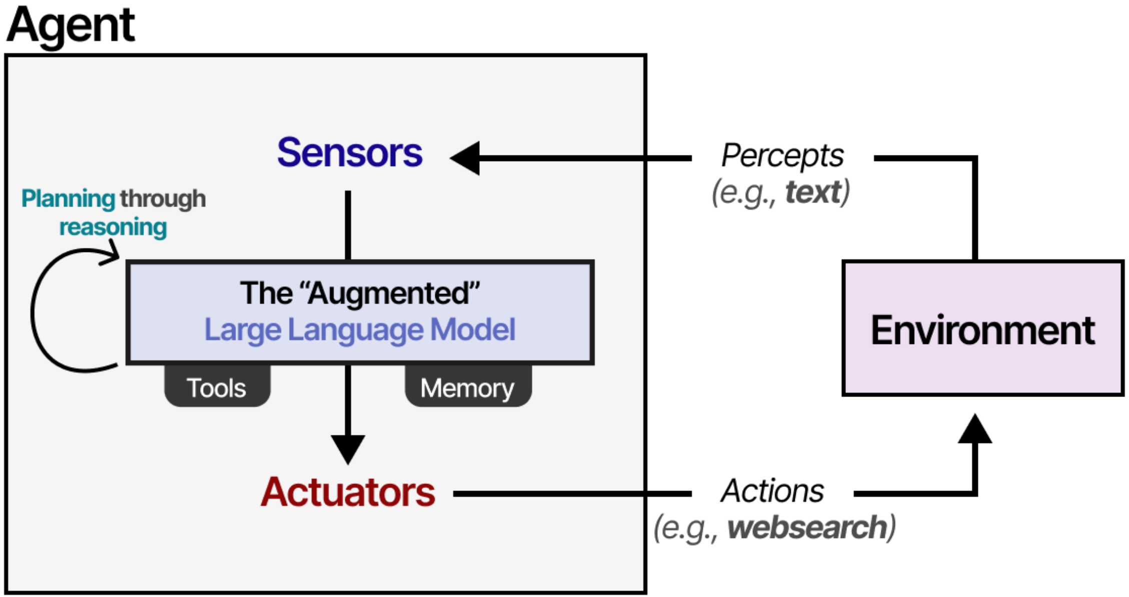
Modern AI Agents (LLM Agents)

- For instance, when faced with a math question, the LLM may decide to use the appropriate tool (a calculator).



Modern AI Agents (LLM Agents)

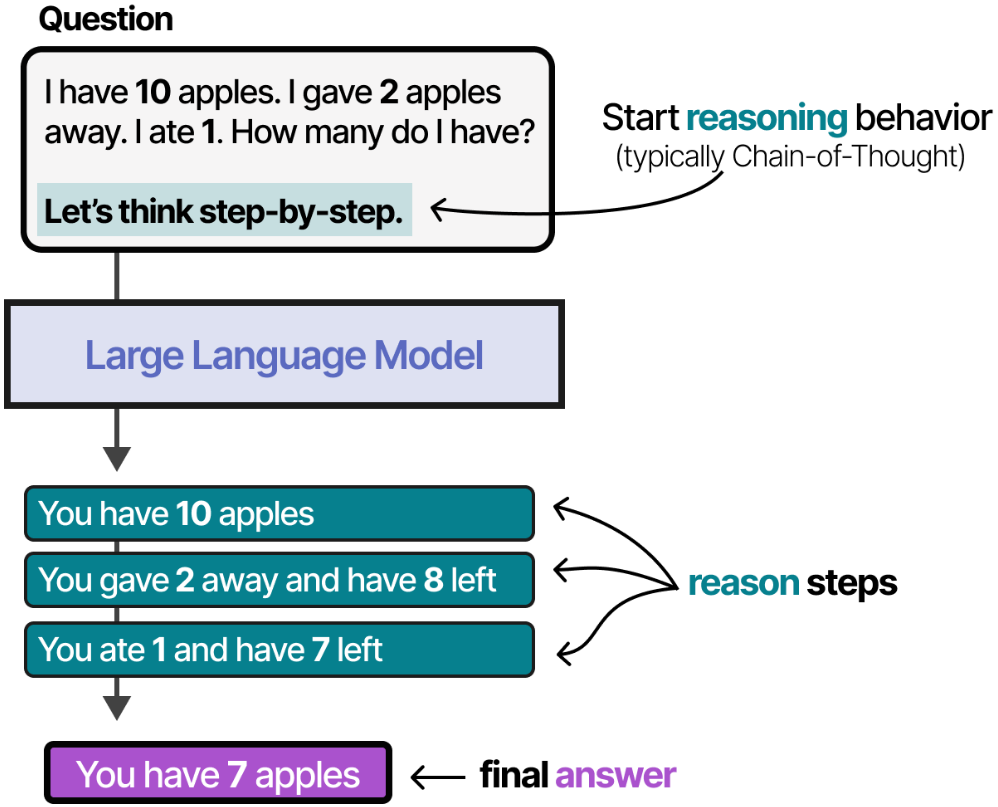
- Using the “Augmented” LLM, the Agent can observe the environment through textual input (as LLMs are generally textual models) and perform certain actions through its use of tools (like searching the web)..


Agent
Sensors
Planning through reasoning
The “Augmented” Large Language Model
Tools
Memory
Actuators
Percepts (e.g., text)
Environment
Actions (e.g., websearch)

Modern AI Agents (LLM Agents)

- To select which actions to take, the LLM Agent has a vital component: its ability to plan.

- For this, LLMs need to be able to "**reason**" and "**think**" through methods like **chain-of-thought**.



Question

I have **10** apples. I gave **2** apples away. I ate **1**. How many do I have?

Let's think step-by-step.

Start **reasoning** behavior
(typically Chain-of-Thought)

Large Language Model

You have **10** apples

You gave **2** away and have **8** left

You ate **1** and have **7** left

reason steps

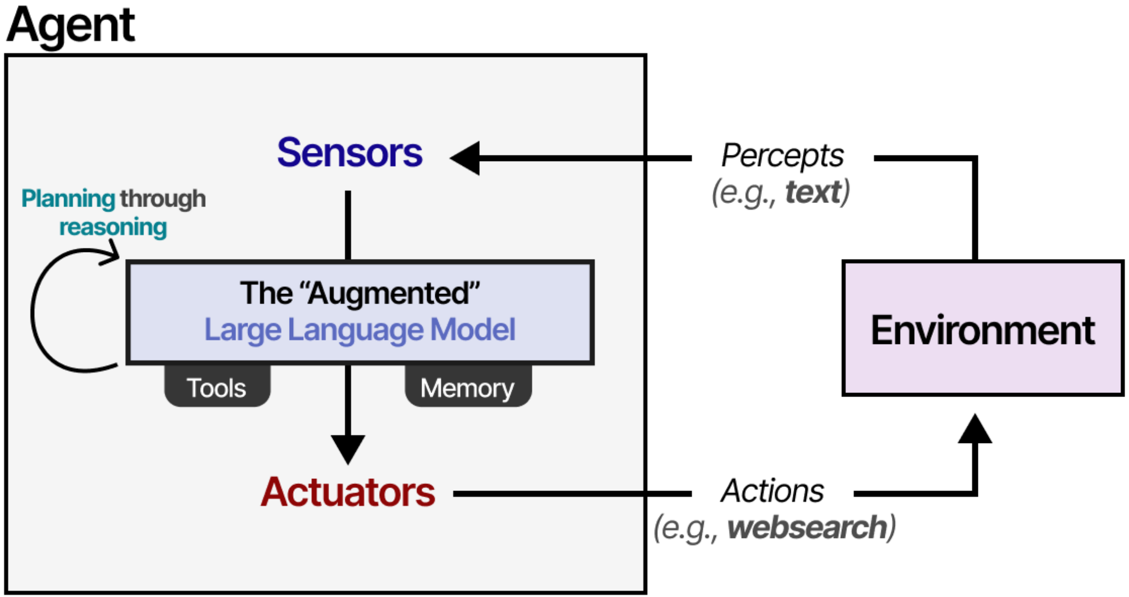
You have **7** apples

final answer

By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

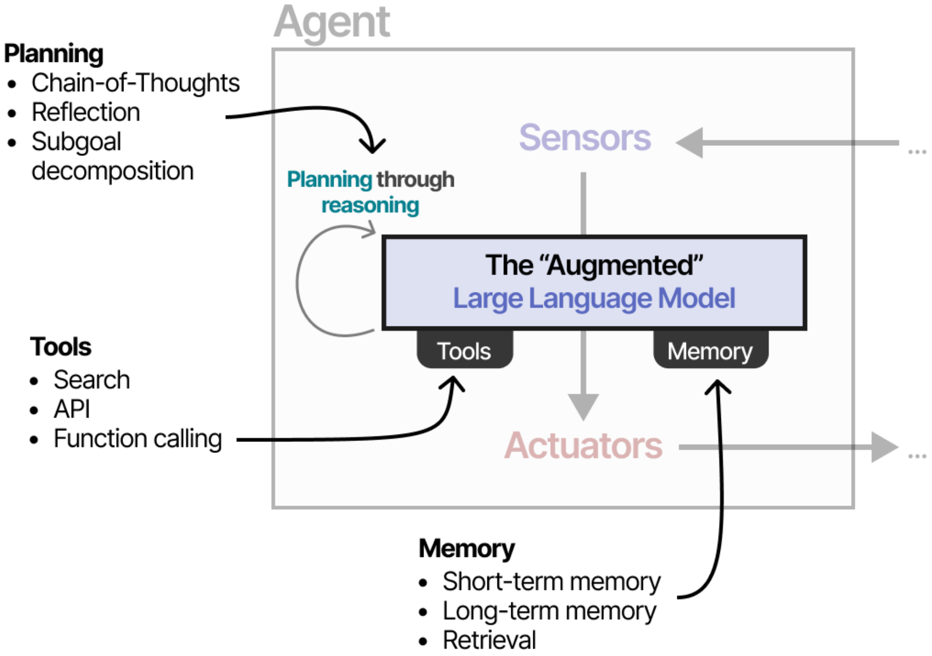
Modern AI Agents (LLM Agents)

- Using the “Augmented” LLM, the Agent can observe the environment through textual input (as LLMs are generally textual models) and perform certain actions through its use of tools (like searching the web)..


Agent
Sensors
Planning through reasoning
The “Augmented” Large Language Model
Tools
Memory
Actuators
Percepts (e.g., text)
Environment
Actions (e.g., websearch)

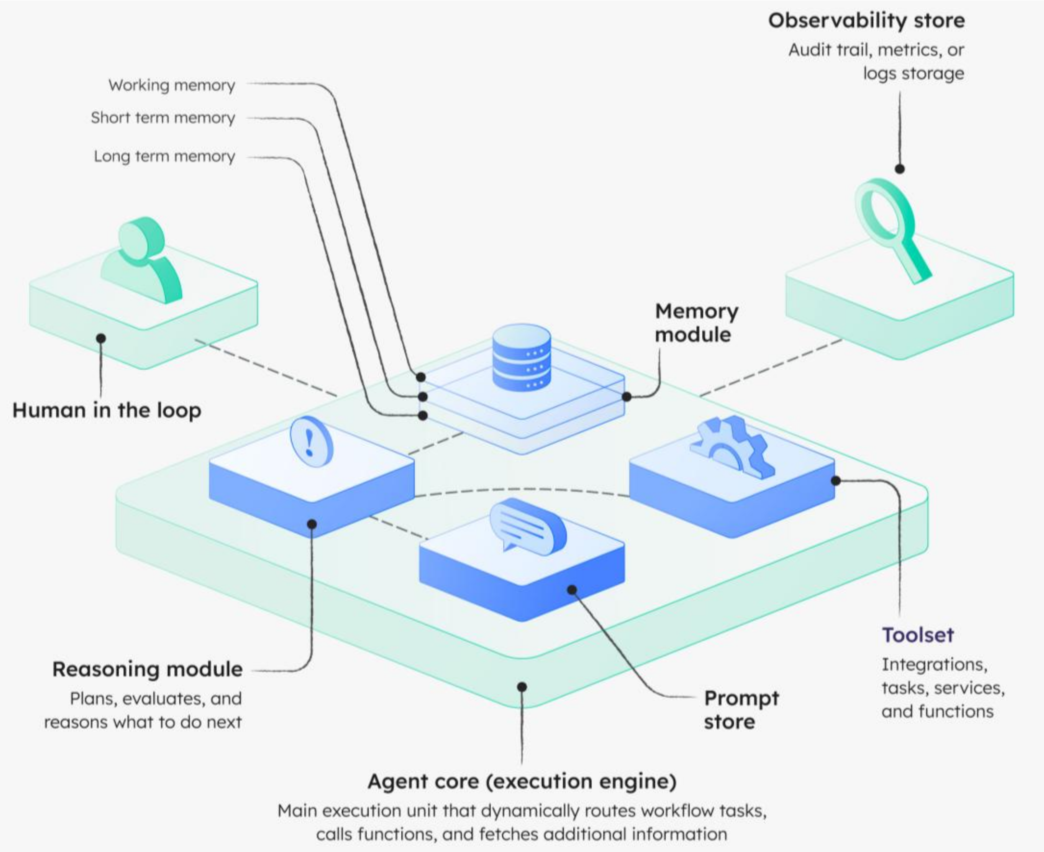
Modern AI Agents (LLM Agents)

- This planning behavior allows the Agent to understand the situation (**LLM**), plan next steps (**planning**), take actions (**tools**), and keep track of the taken actions (**memory**).



By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

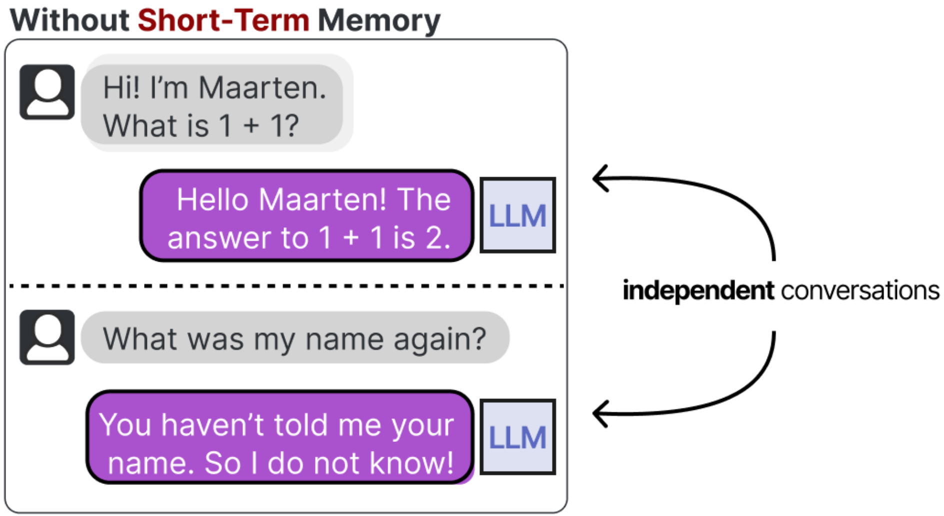
Modern AI Agents



Observability store
Audit trail, metrics, or logs storage
Working memory
Short term memory
Long term memory
Memory module
Human in the loop
Toolset
Integrations, tasks, services, and functions
Reasoning module
Plans, evaluates, and reasons what to do next
Prompt store
Agent core (execution engine)
Main execution unit that dynamically routes workflow tasks, calls functions, and fetches additional information

Memory

- LLMs are **forgetful** systems, or more accurately, do not perform any memorization at all when interacting with them.
- For instance, when you ask an LLM a question and then follow it up with another question, it will not remember the former.



Without Short-Term Memory

Hi! I'm Maarten. What is 1 + 1?

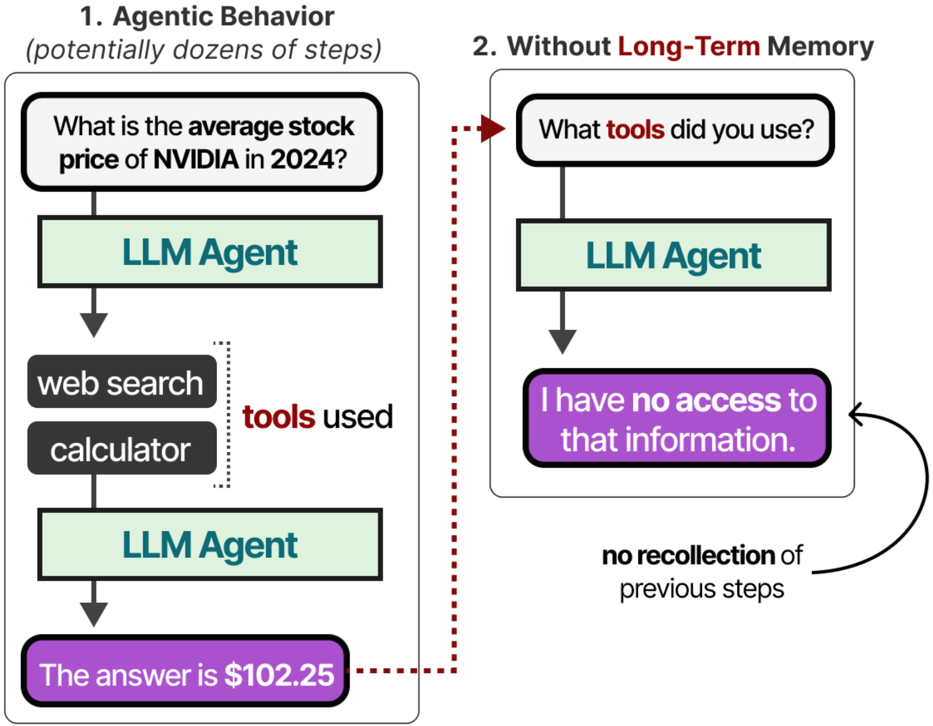
Hello Maarten! The answer to 1 + 1 is 2.

What was my name again?

You haven't told me your name. So I do not know!

Memory

- We typically refer to this as **short-term memory**, also called **working memory**, which functions as a buffer for the (near) immediate context. This includes recent actions the LLM Agent has taken.
- However, the LLM Agent also needs to keep track of potentially dozens of steps, not only the most recent actions.

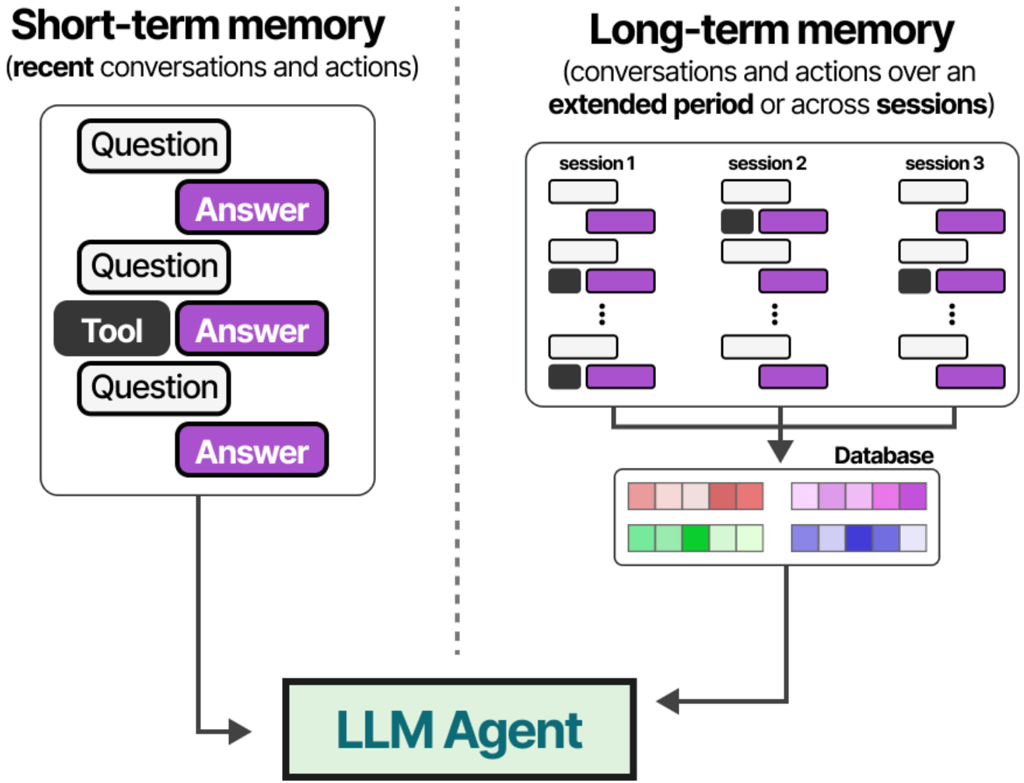


1. Agentic Behavior
(potentially dozens of steps)
What is the average stock price of NVIDIA in 2024?
LLM Agent
web search
calculator
tools used
LLM Agent
The answer is $102.25
2. Without Long-Term Memory
What tools did you use?
LLM Agent
I have no access to that information.
no recollection of previous steps

By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

Memory

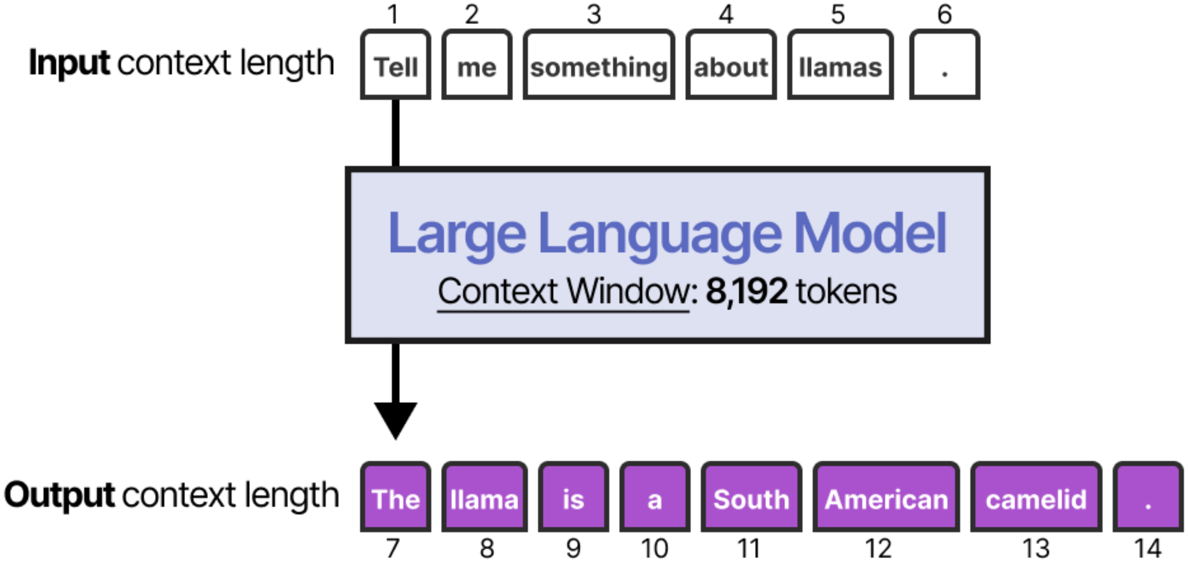
- This is referred to as **long-term memory** as the LLM Agent could theoretically take dozens or even hundreds of steps that need to be memorized.



By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

Short-Term Memory

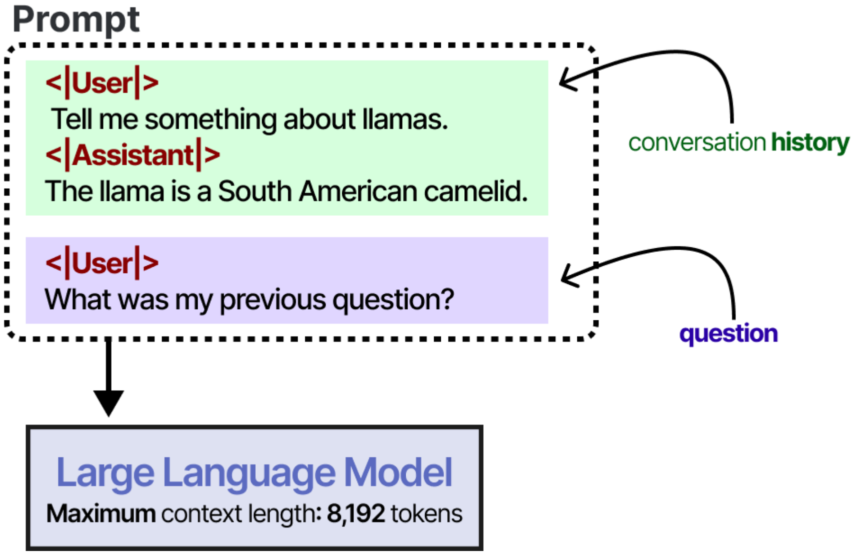
- The most straightforward method for enabling short-term memory is to use the model's **context window**, which is essentially the number of tokens an LLM can process.



By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

Short-Term Memory

- The context window tends to be at least **8192 tokens** and sometimes can scale up to hundreds of thousands of tokens.
- A large context window can be used to track the full conversation history as part of the input prompt.

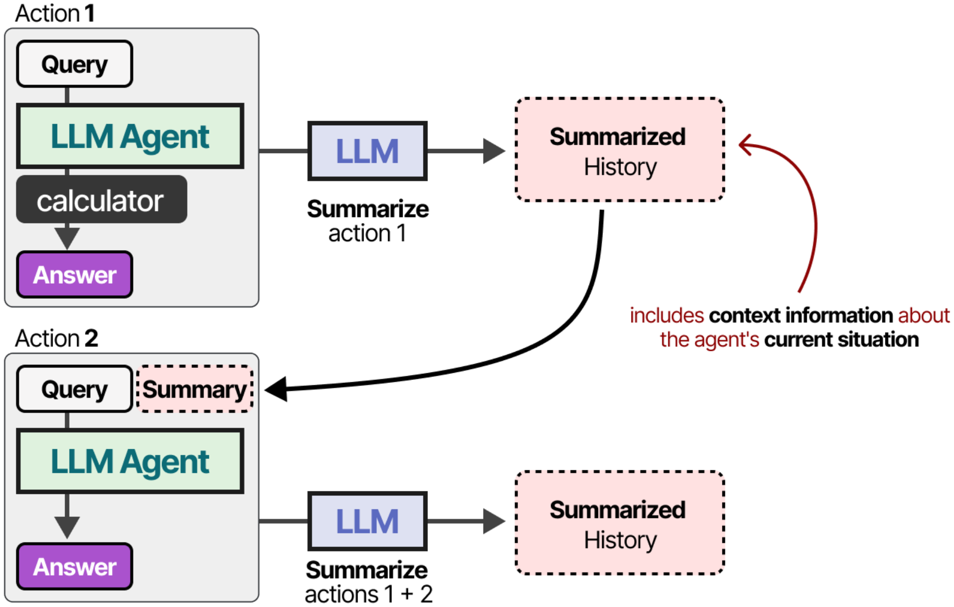


Prompt
<|User|>
Tell me something about llamas.
<|Assistant|>
The llama is a South American camelid.
<|User|>
What was my previous question?
conversation history
question
Large Language Model
Maximum context length: 8,192 tokens

By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

Short-Term Memory

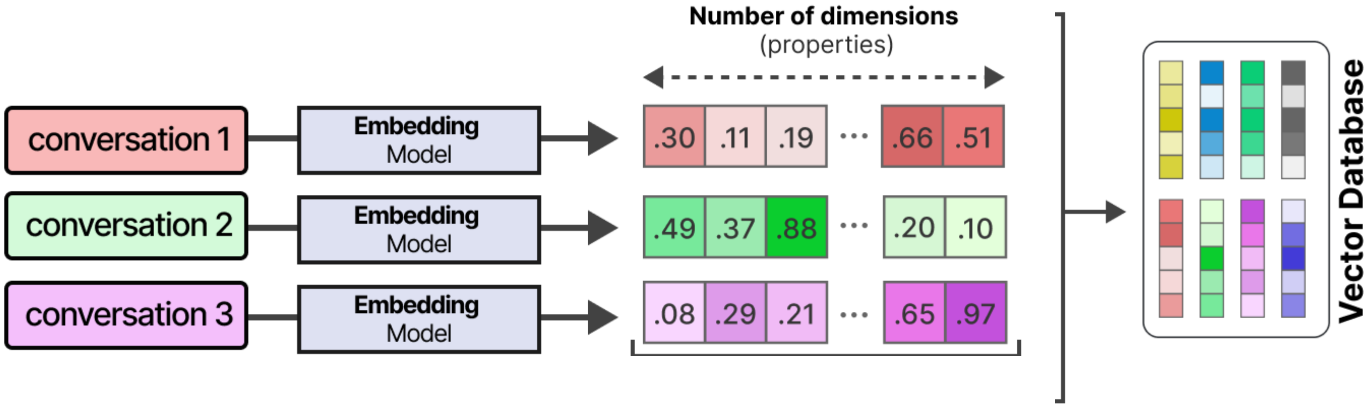
- This works as long as the conversation history **fits** within the **LLM's context window** and is a nice way of mimicking memory.
- However, instead of actually memorizing a conversation, we essentially "tell" the LLM what that conversation was.
- For models with a smaller context window, or when the conversation history is large, we can instead use another LLM to **summarize** the conversations that happened thus far.



By continuously summarizing conversations, we can keep the size of this conversation small. It will reduce the number of tokens while keeping track of only the most vital information.

Long-Term Memory

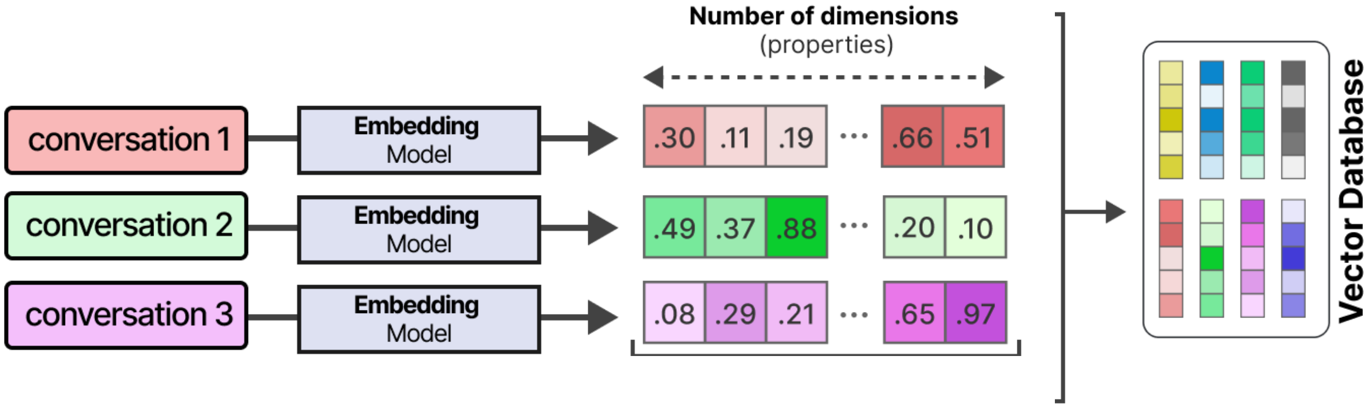
- Long-term memory in LLM Agents includes the agent's **past action space** that needs to be retained over an **extended period**.
- A common technique to enable long-term memory is to store all previous interactions, actions, and conversations in an **external vector database**.
- To build such a database, conversations are first **embedded** into numerical representations that capture their meaning.



By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

Long-Term Memory

- Long-term memory in LLM Agents includes the agent's **past action space** that needs to be retained over an **extended period**.
- A common technique to enable long-term memory is to store all previous interactions, actions, and conversations in an **external vector database**.
- To build such a database, conversations are first **embedded** into numerical representations that capture their meaning.

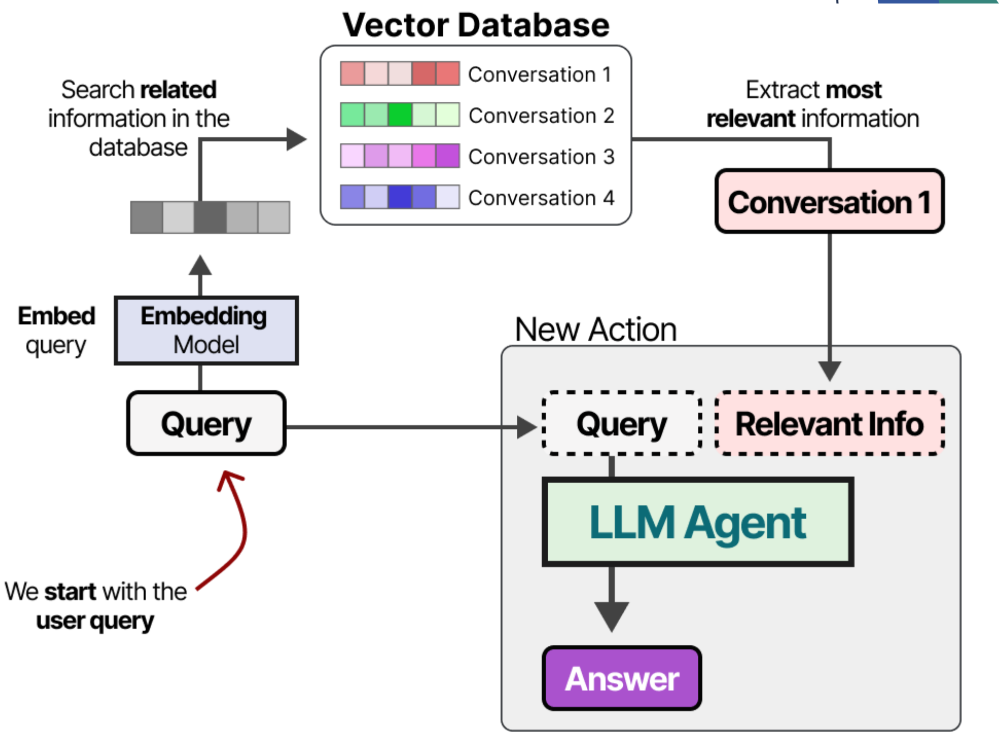


By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

Long-Term Memory

- After building the database, we can embed any given prompt and find the most relevant information in the vector database by comparing the prompt embedding with the database embeddings.

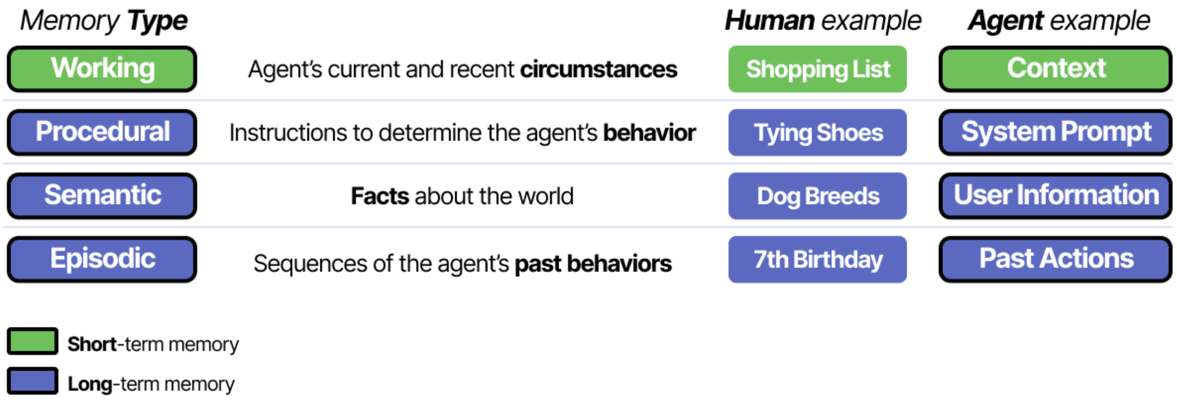
- This method is often referred to as **Retrieval-Augmented Generation (RAG).**



By Dr. Ali Asghar Manjotho, Assistant Professor, CSE-MUET

Long-Term Memory

- Long-term memory can also involve retaining information from different sessions. For instance, you might want an LLM Agent to remember any research it has done in previous sessions.
- Different types of information can also be related to different types of memory to be stored. In psychology, there are numerous types of memory to differentiate, but the Cognitive Architectures for Language Agents paper coupled four of them to LLM Agents.



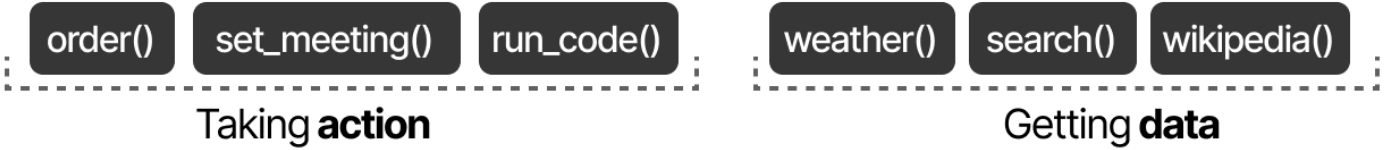
*Memory **Type***		***Human** example*	***Agent** example*
🟩 Working	Agent's current and recent **circumstances**	🟩 Shopping List	🟩 Context
🟦 Procedural	Instructions to determine the agent's **behavior**	🟦 Tying Shoes	🟦 System Prompt
🟦 Semantic	**Facts** about the world	🟦 Dog Breeds	🟦 User Information
🟦 Episodic	Sequences of the agent's **past behaviors**	🟦 7th Birthday	🟦 Past Actions

🟩 **Short**-term memory

🟦 **Long**-term memory

Tools

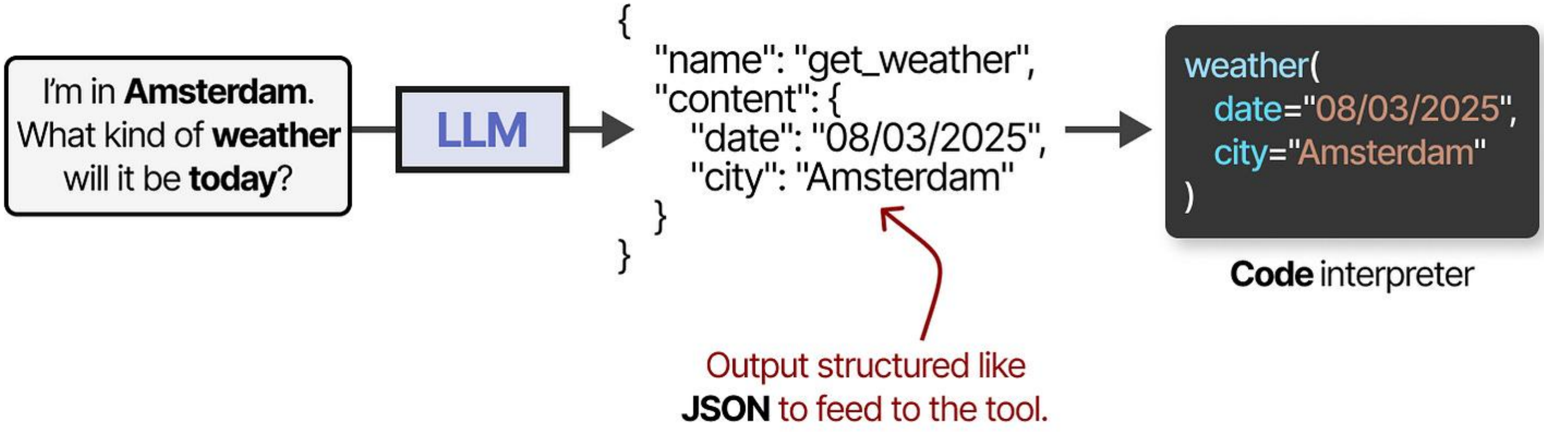
- Tools allow a given LLM to either interact with an external environment (such as databases) or use external applications (such as custom code to run).



- Tools generally have two use cases: fetching data to retrieve up-to-date information and taking action like setting a meeting or ordering food.

Tools

- To actually use a tool, the LLM has to generate text that fits with the API of the given tool. We tend to expect strings that can be formatted to JSON so that it can easily be fed to a code interpreter.

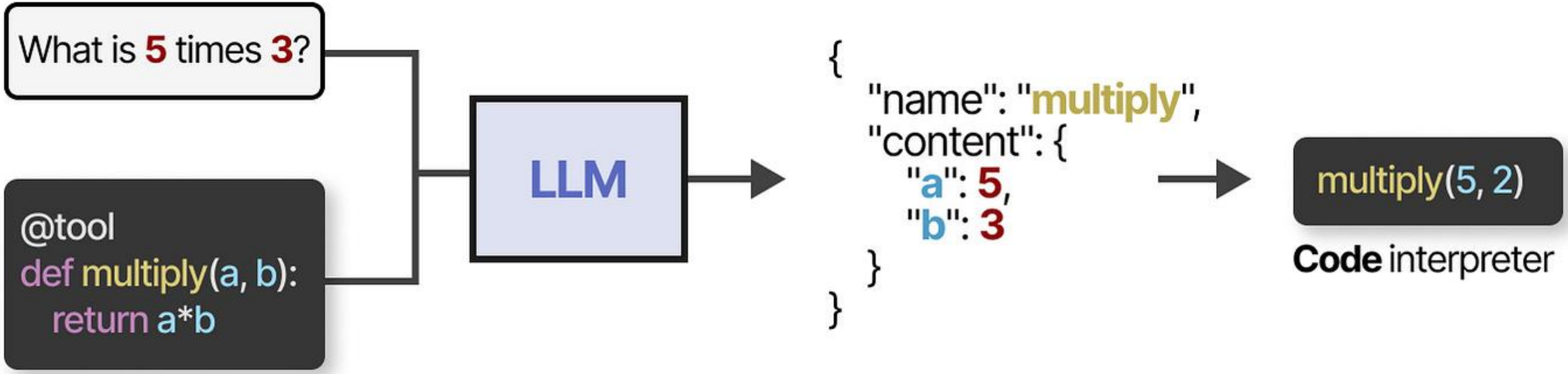

I'm in **Amsterdam.** What kind of **weather** will it be **today**?
LLM
Code interpreter
Output structured like JSON to feed to the tool.

```
{
   "name": "get_weather",
   "content": {
      "date": "08/03/2025",
      "city": "Amsterdam"
   }
}
```

```
weather(
   date="08/03/2025",
   city="Amsterdam"
)
```

Tools

- You can also generate custom functions that the LLM can use, like a basic multiplication function. This is often referred to as **function calling**.



What is 5 times 3?

```
@tool
def multiply(a, b):
    return a*b
```

LLM

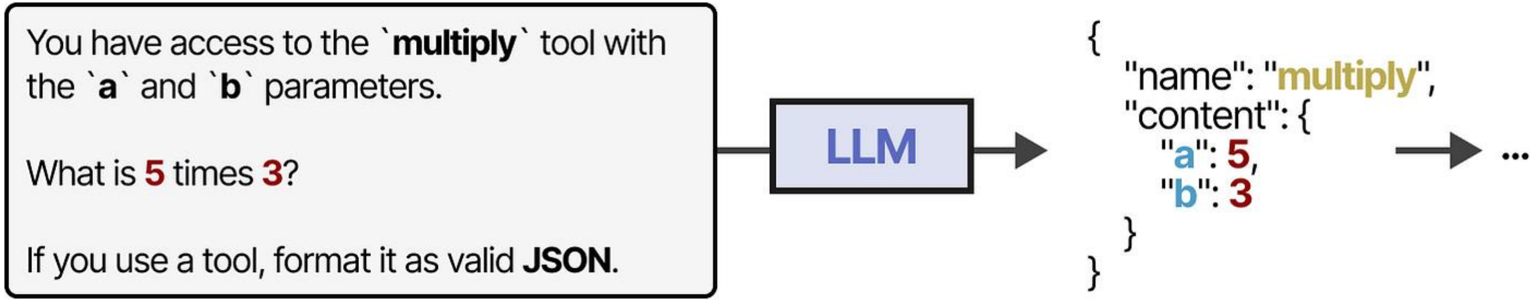
```
{
    "name": "multiply",
    "content": {
        "a": 5,
        "b": 3
    }
}
```

```
multiply(5, 2)
```

Code interpreter

Tools

- Some LLMs can use any tools if they are prompted correctly and extensively. Tool-use is something that most current LLMs are capable of.



You have access to the `**multiply**` tool with the `**a**` and `**b**` parameters.

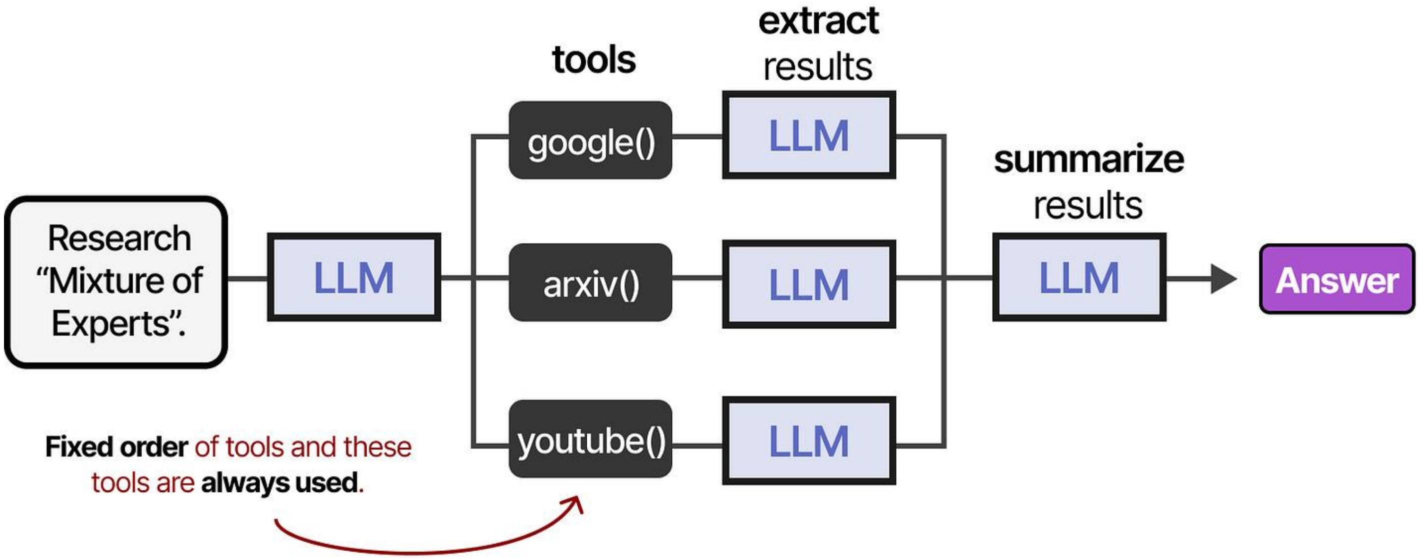
What is **5** times **3**?

If you use a tool, format it as valid **JSON**.

```
{
   "name": "multiply",
   "content": {
      "a": 5,
      "b": 3
   }
}
```

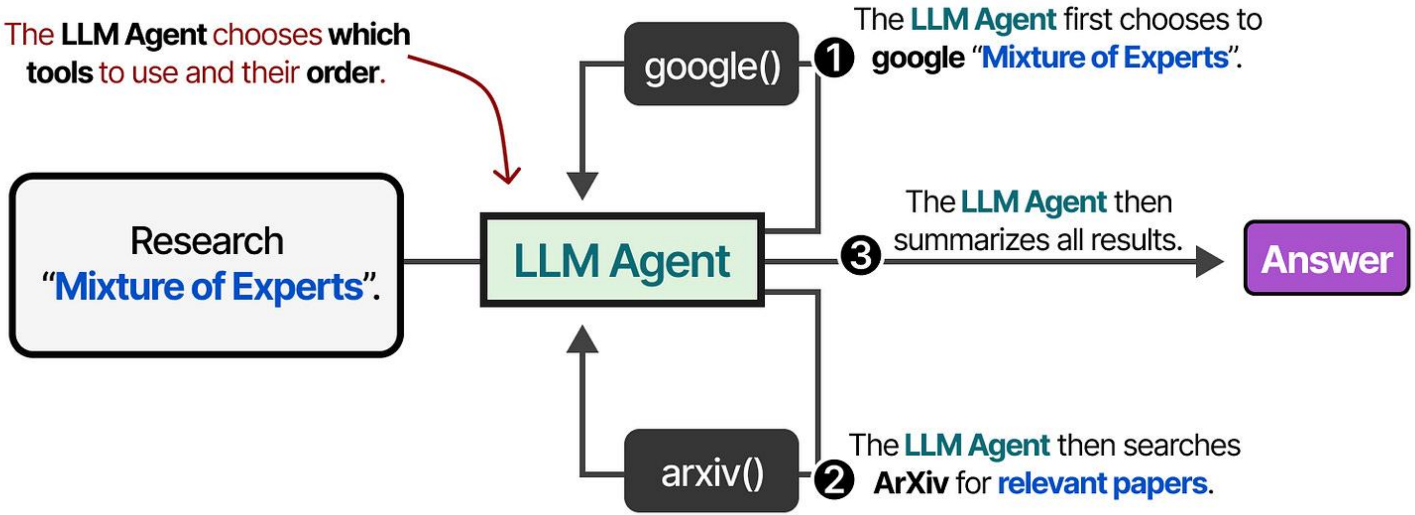
Tools

- A more stable method for accessing tools is by fine-tuning the LLM.
- Tools can either be used in a given order if the agentic framework is fixed.


extract
results
tools
summarize
results
Research "Mixture of Experts".
LLM
google()
arxiv()
youtube()
LLM
LLM
LLM
LLM
Answer
Fixed order of tools and these tools are **always used**.

Tools

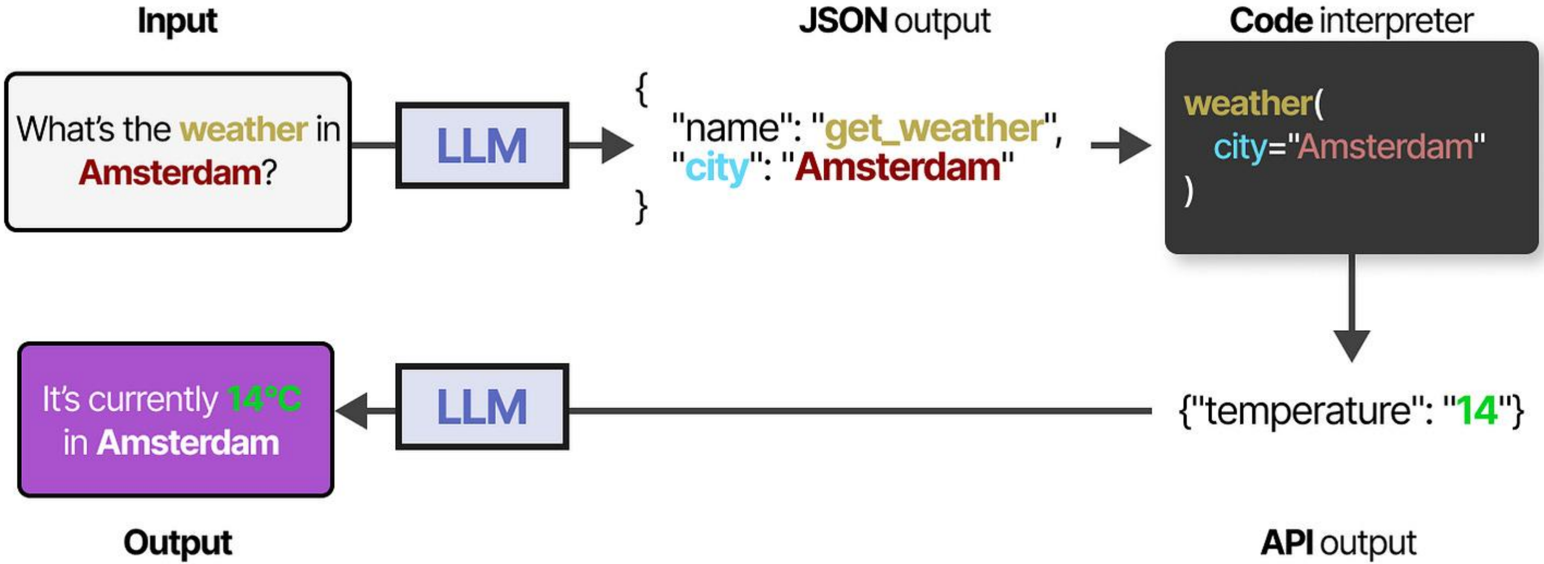
- LLM can also autonomously choose which tool to use and when. LLM Agents, like the above image, are essentially sequences of LLM calls (but with autonomous selection of actions/tools/etc.).



The LLM Agent chooses which tools to use and their order.
google()
❶ The LLM Agent first chooses to google "Mixture of Experts".
Research "Mixture of Experts".
LLM Agent
❸ The LLM Agent then summarizes all results.
Answer
arxiv()
❷ The LLM Agent then searches ArXiv for relevant papers.

Tools

- In other words, the output of intermediate steps is fed back into the LLM to continue processing.


Input
What's the weather in Amsterdam?
LLM
JSON output
{
"name": "get_weather",
"city": "Amsterdam"
}
Code interpreter
weather(
city="Amsterdam"
)
{"temperature": "14"}
API output
LLM
It's currently 14°C in Amsterdam
Output