Trustworthy AI Systems

-- Large Language Model Agent

Instructor: Guangjing Wang

guangjingwang@usf.edu

Last Lecture

- Recurrent Neural Network
- Attention
- Transformers
- Pretrained Foundation Model

This Lecture

- LLM Agent

- External Lecture: LLM Agents: Brief History and Overview

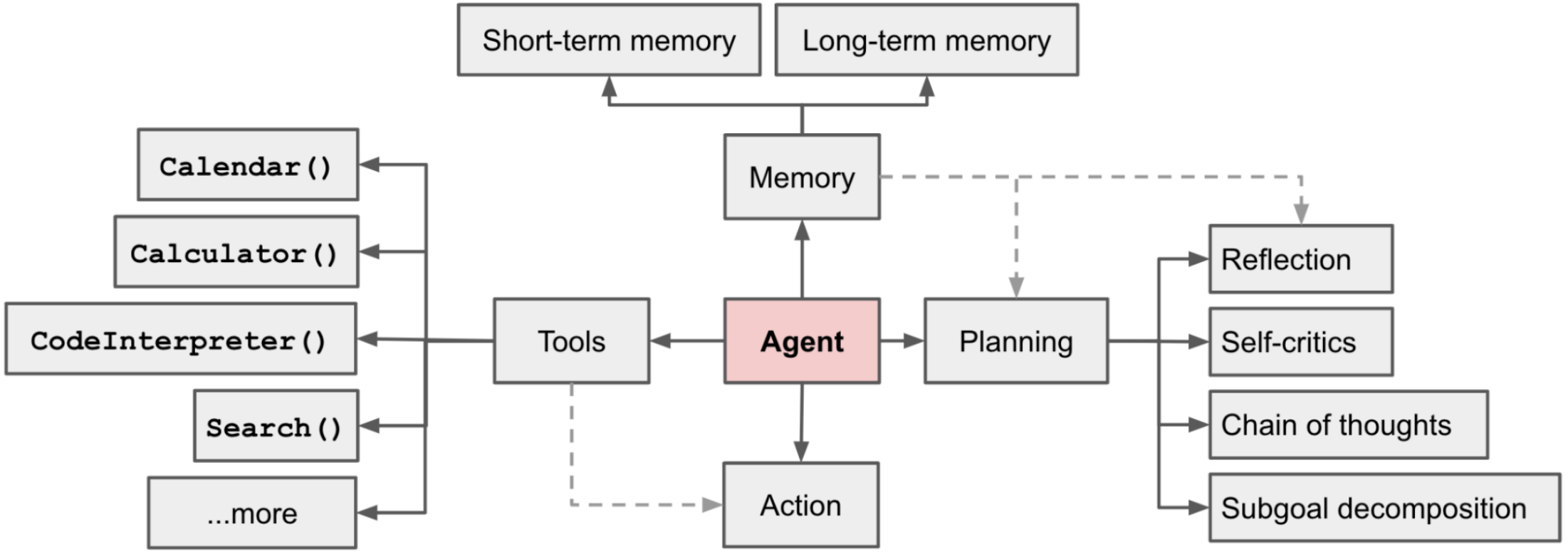
LLM Agent

- A system that can use a Large Language Model (LLM) to reason through a problem, create a plan to solve the problem, and execute the plan with the help of a set of tools.

- LLM model example: https://github.com/meta-llama/llama-models/tree/main

- LLM Agent can response the inquiry by using planning, tailored focus, memory, using different tools, and breaking down a complex question into simpler sub-parts.

LLM-powered Agent System



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

An Application Example

- Consider a LLM application that is designed to help financial analysts answer questions about the performance of a company.
    - “What were the three takeaways from the Q2 earnings call from FY23? Focus on the technological moats that the company is building”.

How do we develop a solution to answer a question like above?

Agent Core

- The agent core is the central coordination module that manages the core logic and behavioral characteristics of an Agent.

```
template = """GENERAL INSTRUCTIONS
Your task is to answer questions. If you cannot answer the question, request a
helper or use a tool. Fill with Nil where no tool or helper is required.

AVAILABLE TOOLS
- Search Tool
- Math Tool

AVAILABLE HELPERS
- Decomposition: Breaks Complex Questions down into simpler subparts

CONTEXTUAL INFORMATION
<No previous questions asked>

QUESTION
How much did the revenue grow between Q1 of 2024 and Q2 of 2024?

ANSWER FORMAT
{"Tool_Request": "<Fill>", "Helper_Request "<Fill>"}"""
```

Basic template of how the different modules of an agent are assembled in its core.

https://developer.nvidia.com/blog/introduction-to-llm-agents/

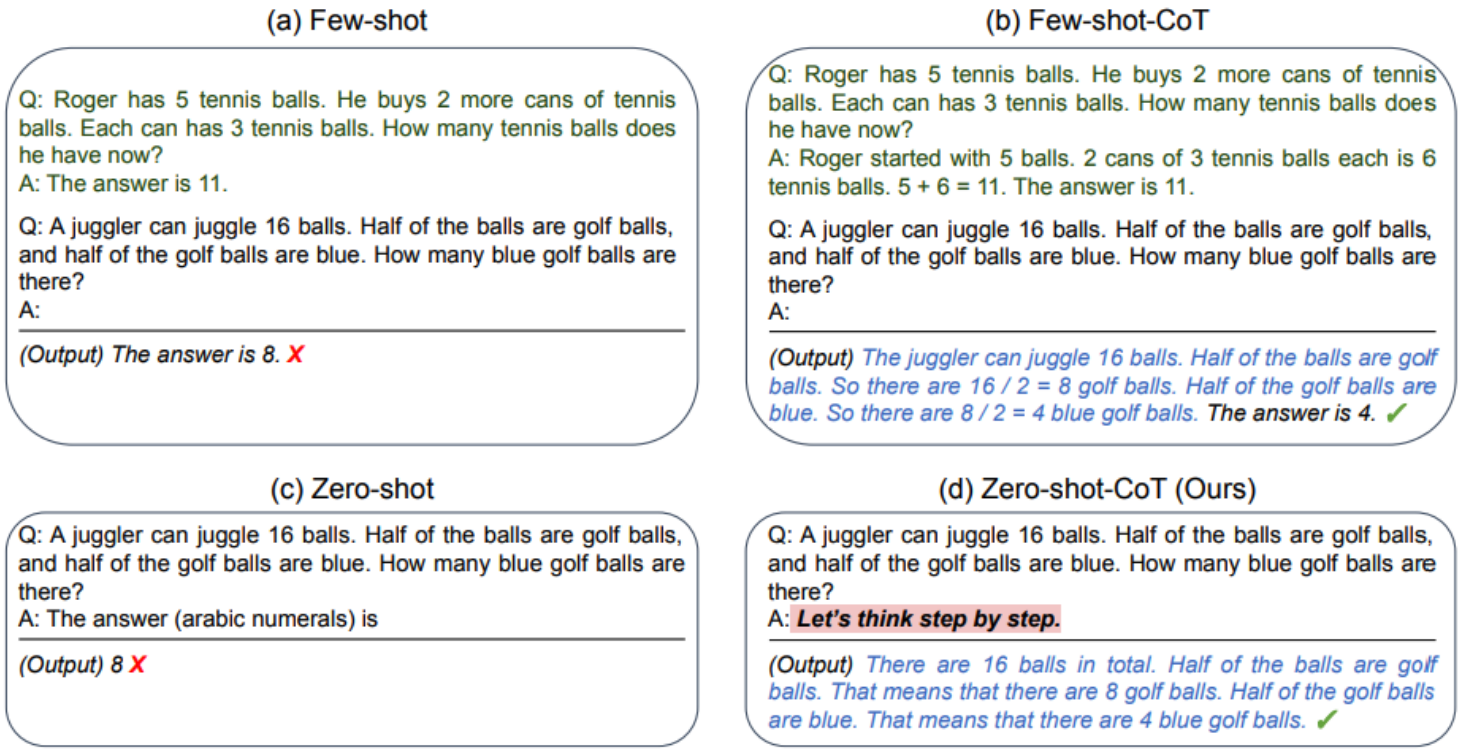
Planning Module

- Task and question decomposition
  - “What were the three takeaways from NVIDIA’s last earnings call?”
    - “Which technological shifts were discussed the most?”
    - “Are there any business headwinds?”
    - “What were the financial results?”
  - Each of these questions can be further broken into subparts.
  - A specialized AI agent must guide this decomposition.

- Self-Reflection: Techniques such as Chain of Thought, and Graph of thought have served as critic– or evidence-based prompting frameworks.

Chain-of-Thought (CoT)

- A series of intermediate reasoning steps to significantly improve the ability of LLM to perform complex reasoning.



https://arxiv.org/abs/2205.11916

Memory Module

- **Short-term memory:** A ledger of actions and thoughts that an agent goes through to attempt to answer a single question from a user: the agent's "train of thought." (e.g., in-context learning)

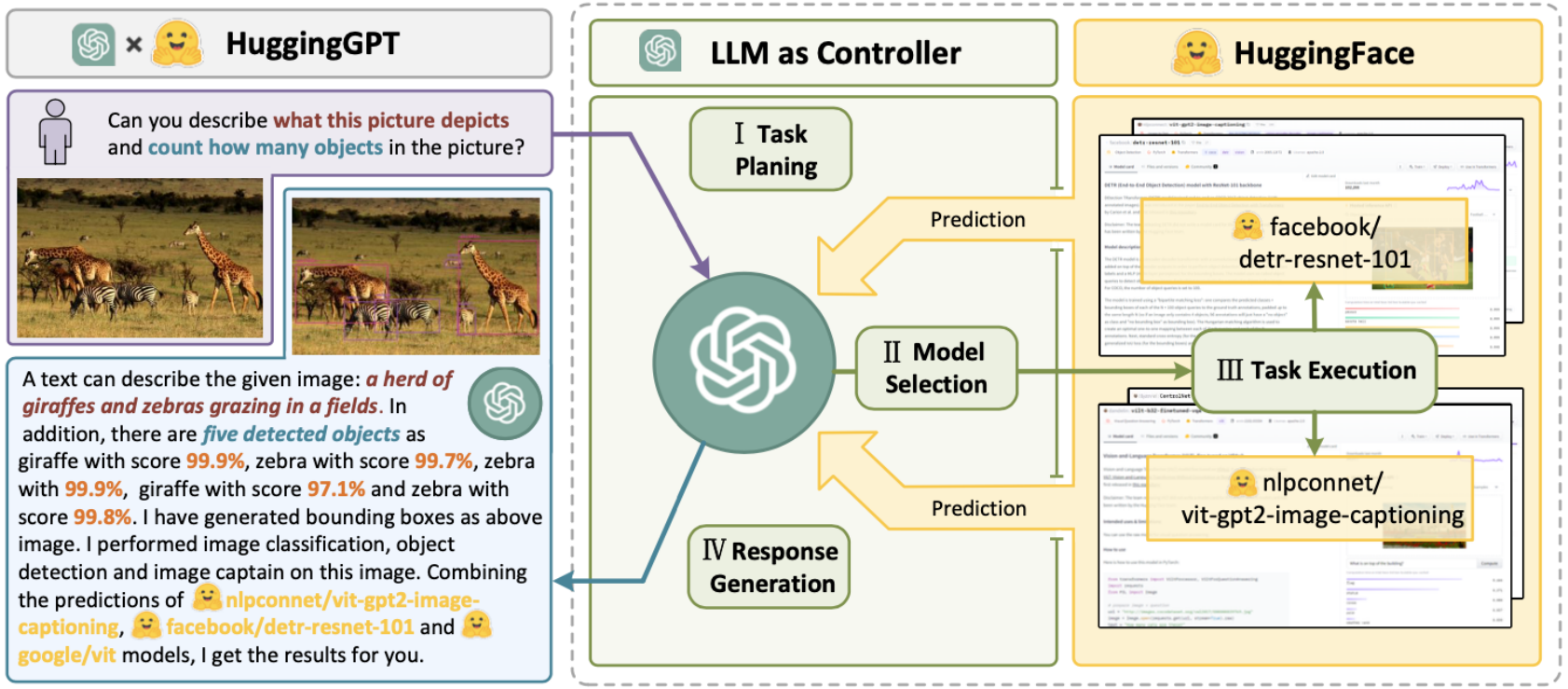
- **Long-term memory:** A ledger of actions and thoughts about events that happen between the user and agent. It is a log book that contains a conversation history stretching across weeks or months.

Memory requires a composite score, which is made up of semantic similarity, importance, recency, and other application-specific metrics.

Tools

- Tools are well-defined executable workflows that agents can use to execute tasks
  - A Retrieval Augmented Generation (RAG) pipeline to generate context aware answers
  - A code interpreter to solve complex programmatically tasks
  - An API to search information over the internet
  - Any simple API service like a weather API or an API for an Instant messaging application
  - …

Example: HuggingGPT



Using ChatGPT as the task planner to select models available in HuggingFace platform according to the model descriptions and summarize the response based on the execution results.

https://arxiv.org/pdf/2303.17580

HuggingGPT: Task Planning

- Using few-shot examples to guide LLM to do task parsing and planning.

```
The AI assistant can parse user input to several tasks: [{"task": task, "id",
task_id, "dep": dependency_task_ids, "args": {"text": text, "image": URL, "audio":
URL, "video": URL}}]. The "dep" field denotes the id of the previous task which
generates a new resource that the current task relies on. A special tag "-task_id"
refers to the generated text image, audio and video in the dependency task with id
as task_id. The task MUST be selected from the following options: {{ Available Task
List }}. There is a logical relationship between tasks, please note their order. If
the user input can't be parsed, you need to reply empty JSON. Here are several
cases for your reference: {{ Demonstrations }}. The chat history is recorded as {{
Chat History }}. From this chat history, you can find the path of the user-
mentioned resources for your task planning.
```

HuggingGPT: Model Selection

- LLM distributes the tasks to expert models.

```
Given the user request and the call command, the AI assistant helps the user to
select a suitable model from a list of models to process the user request. The AI
assistant merely outputs the model id of the most appropriate model. The output
must be in a strict JSON format: "id": "id", "reason": "your detail reason for the
choice". We have a list of models for you to choose from {{ Candidate Models }}.
Please select one model from the list.
```

HuggingGPT: Task Execution

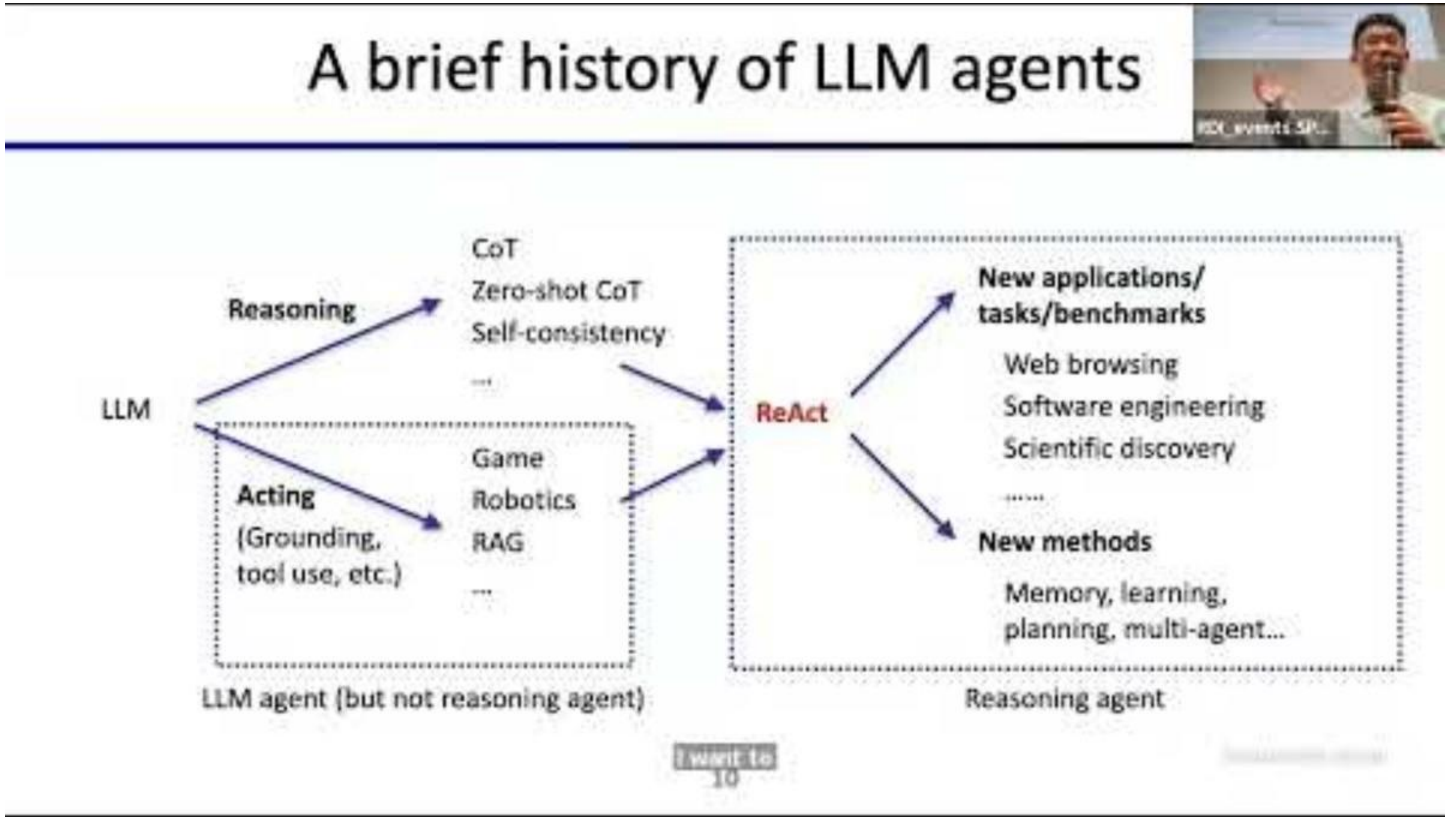
- Expert models execute on the specific tasks and log results.

```
With the input and the inference results, the AI assistant needs to describe the process and results. The previous stages can be formed as - User Input: {{ User Input }}, Task Planning: {{ Tasks }}, Model Selection: {{ Model Assignment }}, Task Execution: {{ Predictions }}. You must first answer the user's request in a straightforward manner. Then describe the task process and show your analysis and model inference results to the user in the first person. If inference results contain a file path, must tell the user the complete file path.
```

Challenges for Practical Usage

- Efficiency improvement is needed as both LLM inference rounds and interactions with other models slow down the process;
- It relies on a long context window to communicate over complicated task content;
- Stability improvement of LLM outputs and external model services.
- The reliability of model outputs is questionable, as LLMs may make formatting errors and occasionally exhibit rebellious behavior (e.g. refuse to follow an instruction)

LLM Agents: Brief History and Overview



https://www.youtube.com/watch?v=RM6ZArd2nVc

References

- https://developer.nvidia.com/blog/introduction-to-llm-agents/
- https://www.promptingguide.ai/
- https://lilianweng.github.io/posts/2023-06-23-agent/
- https://nips.cc/virtual/2023/tutorial/73948