CSCE 689-609
Programming with Agents

Jeff Huang
jeff@cse.tamu.edu
o2lab.github.io

Google NotebookLM

https://notebooklm.google/

NotebookLM now lets you listen to a conversation about your sources

Our new Audio Overview feature can turn documents, slides, charts and more into engaging discussions with one click.

Sep 11, 2024 · 2 min read

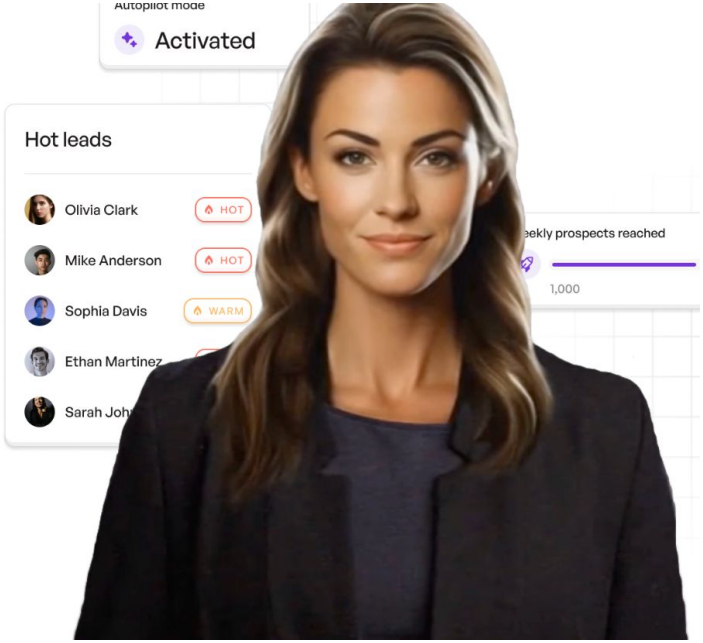
Digital Workers

[11x.ai](11x.ai)

Hi, I'm Alice. Your AI Powered Sales Development Rep

I consumed trillions of datapoints to become the world's best SDR. I work 24/7, at scale, to help you grow faster and automate sales.

Hire Alice Today

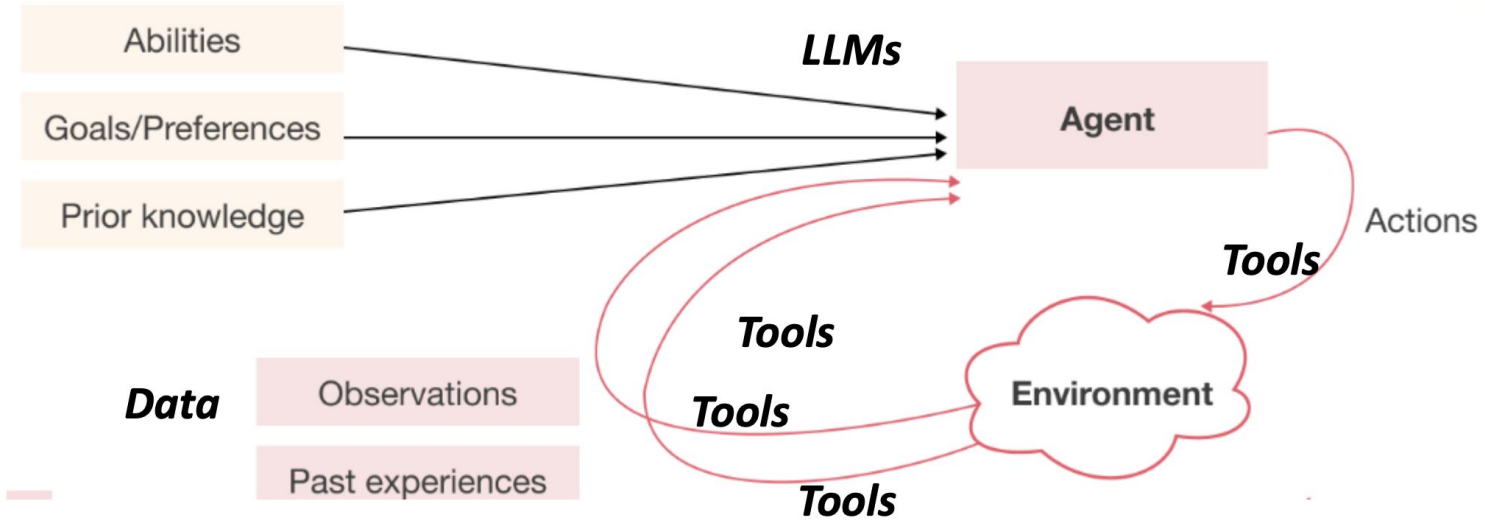


youtube.com/watch?v=oAqmPMtM1LU&t=20s

What are AI agents?

- Software that autonomously performs various human tasks

AI Agent = LLM + Agent Architecture + Prompt + Knowledge + Tools



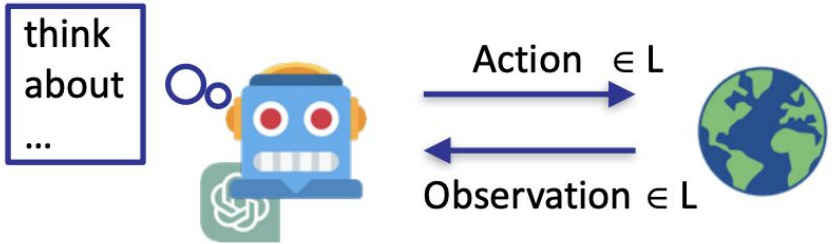
Ref: https://phontron.com/class/anlp2024/assets/slides/anlp-20-agents.pdf

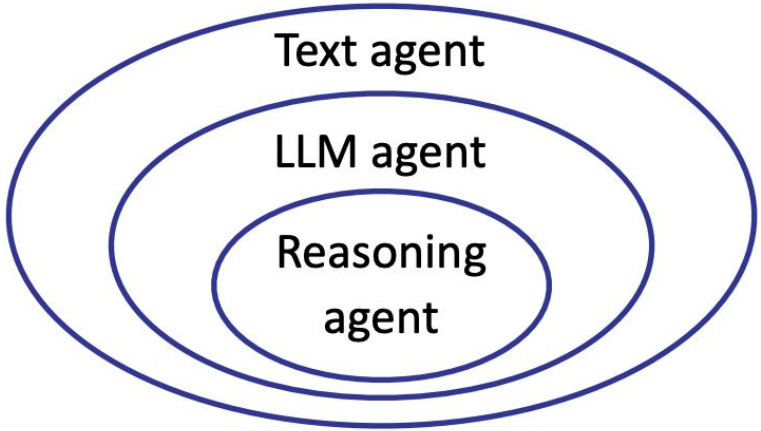
LLM Agents

- ICLR 2024 Workshop on LLM Agents: https://llmagents.github.io/
- Large Language Model Agents: https://rdi.berkeley.edu/llm-agents/f24
- Agentic Workflow by Andrew Ng: https://www.youtube.com/watch?v=sal78ACtGTc

Brief History and Overview

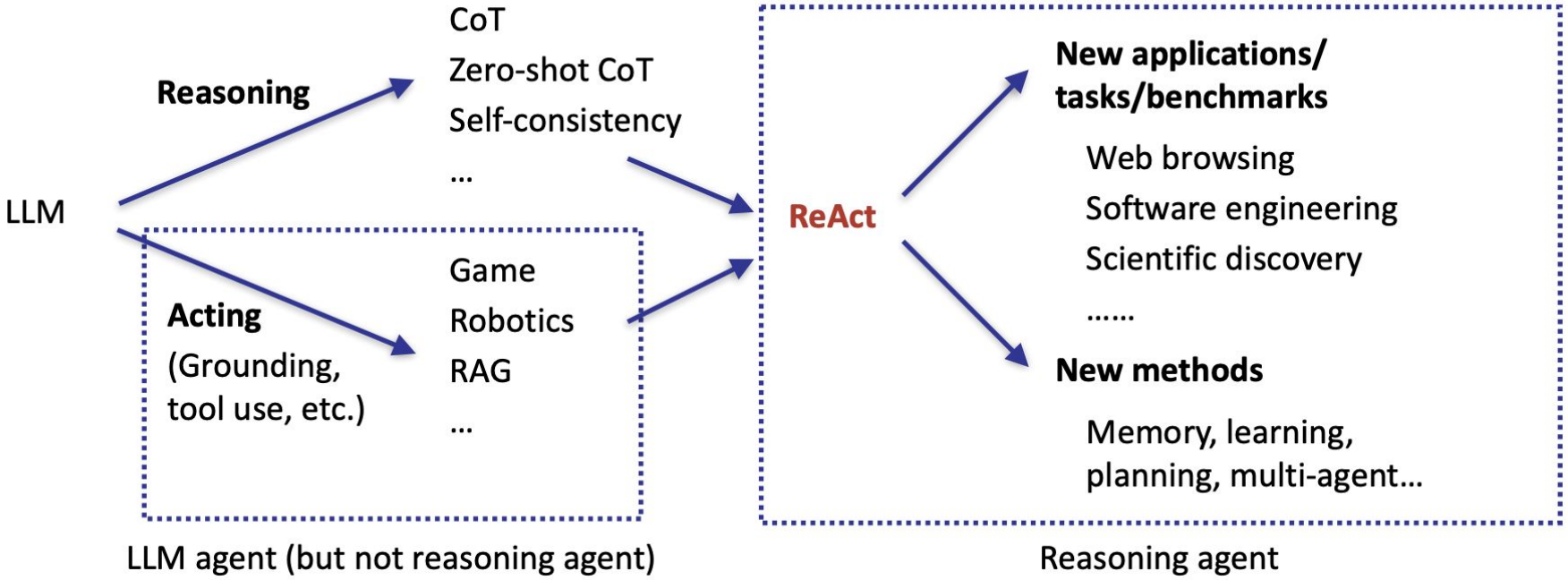
https://rdi.berkeley.edu/llm-agents/assets/llm_agent_history.pdf





- Level 1: Text agent
  - Uses text action and observation
  - Examples: ELIZA, LSTM-DQN
- Level 2: LLM agent
  - Uses LLM to act
  - Examples: SayCan, Language Planner
- Level 3: Reasoning agent
  - Uses LLM to reason to act
  - Examples: ReAct, AutoGPT
  - **The key focus of the field and the talk**

A brief history of LLM agents



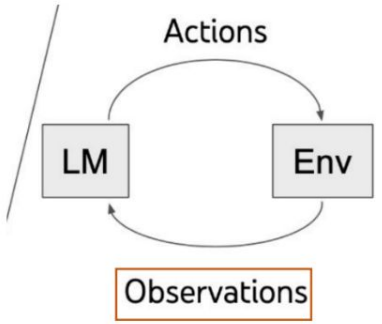
ReAct https://arxiv.org/abs/2210.03629

REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

Shunyu Yao[*,1], Jeffrey Zhao[2], Dian Yu[2], Nan Du[2], Izhak Shafran[2], Karthik Narasimhan[1], Yuan Cao[2]

[1]Department of Computer Science, Princeton University
[2]Google Research, Brain team
[1]{shunyuy,karthikn}@princeton.edu
[2]{jeffreyzhao,dianyu,dunan,izhak,yuancao}@google.com

How to let LLM become an agent



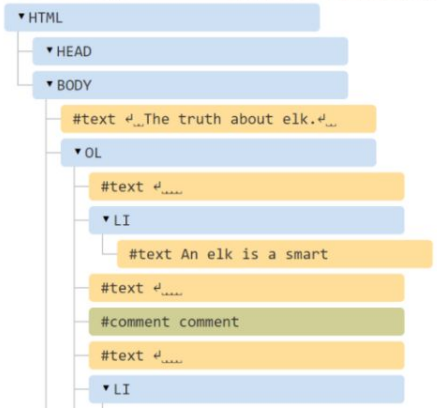
- Observation
  - Text input

    ```
    You are in the middle of a room. Looking quickly around
    you, you see a cabinet 6, a cabinet 1, a coffee machine 1,
    a countertop 3,  a stove burner 1, and a toaster 1.
    ```

  - Visual Input

  - Audio Input

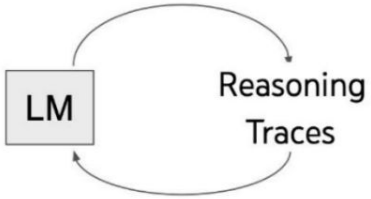
  - Structured Input



- Need for Multimodal LLMs

How to let LLM become an agent

- Planning and reasoning ability
  - Chain-of-thoughts (CoT)
  - "Let's think step by step …."



CoT (Wei et al. 22')

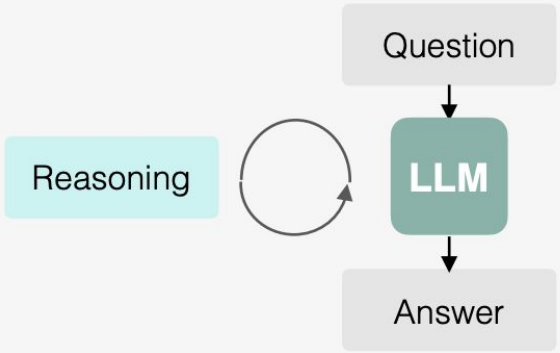
```
You are in the middle of a room. Looking quickly around
you, you see a cabinet 6, a cabinet 1, a coffee machine 1,
a countertop 3,  a stove burner 1, and a toaster 1.
Your task is to: Put some pepper shaker on a drawer.
```

Ask LLM:
What should I do next? Let's think step by step:

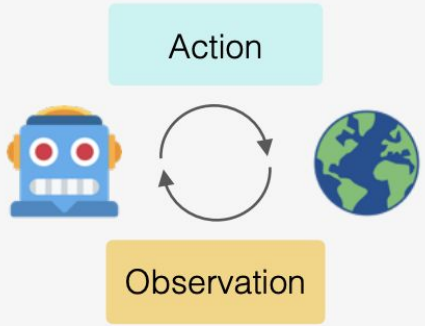
First I need to find a pepper shaker … more likely to appear in cabinets (1-6), countertops (1-3) …

After I find pepper shaker 1, next I need to put it on drawer 1 …..

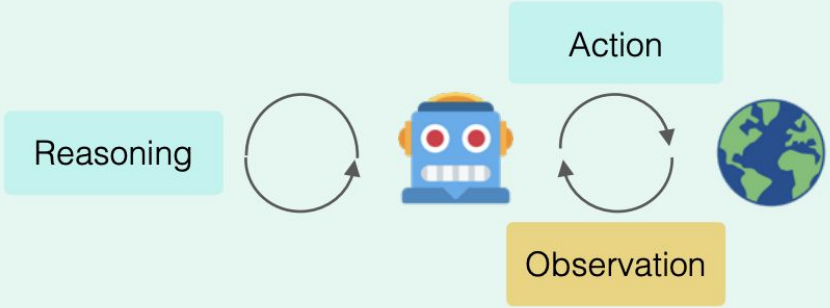
Reasoning (update internal belief)



Acting (obtain external feedback)



ReAct: a new paradigm of agents that **reason and act**



- **Synergy** of reasoning and acting
- **Simple** and intuitive to use
- **General** across domains

https://arxiv.org/pdf/2210.03629

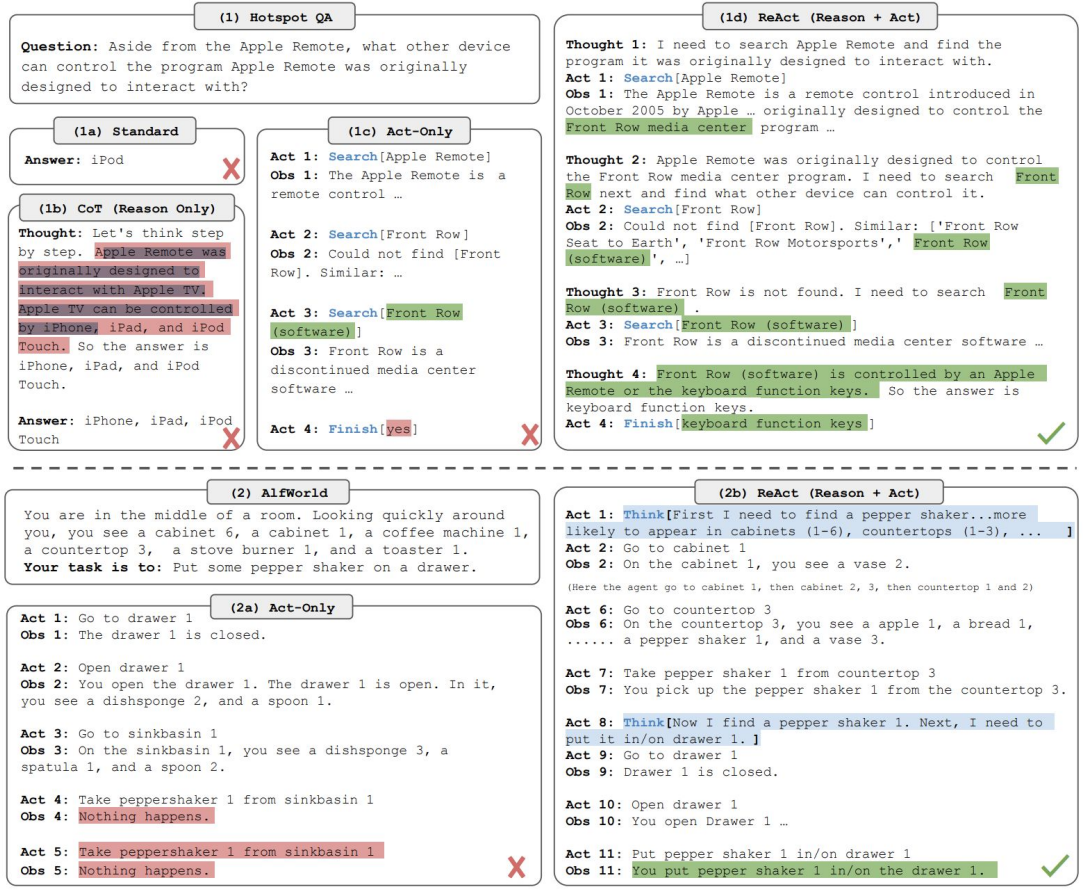
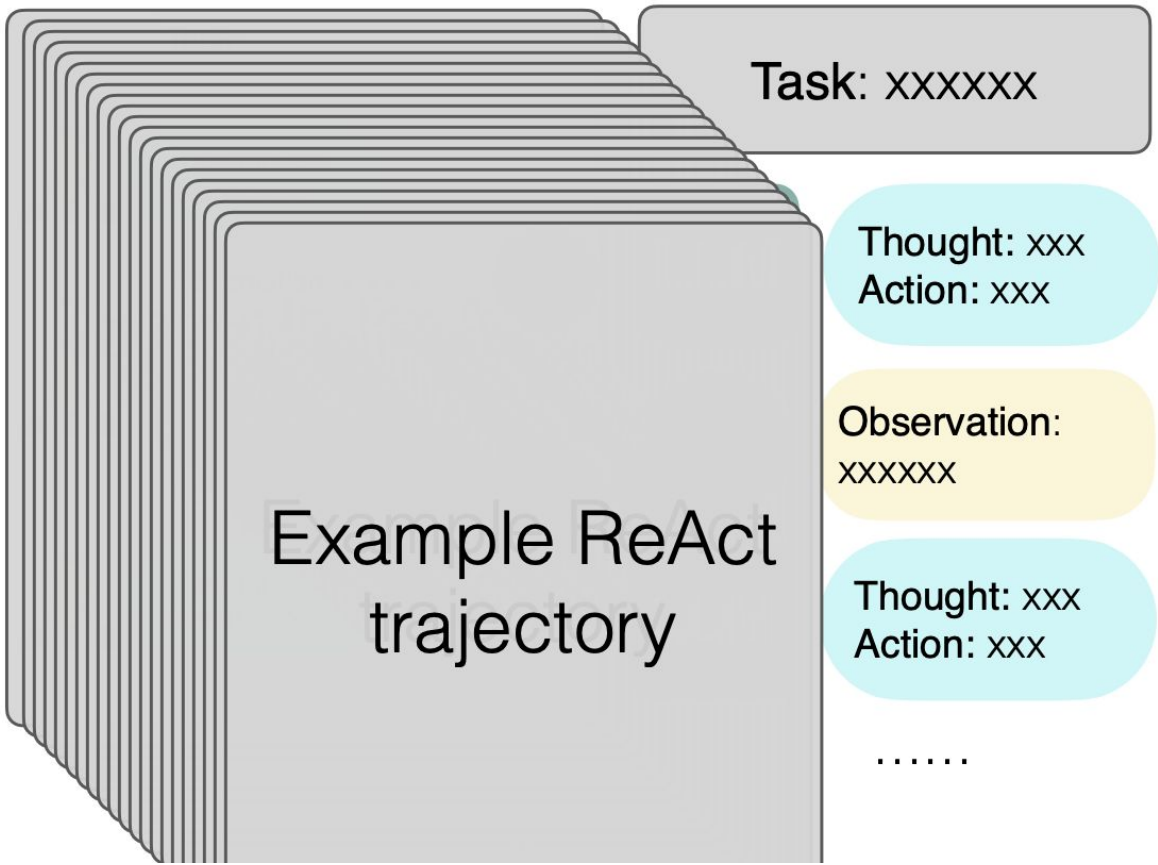


Figure 1: (1) Comparison of 4 prompting methods, (a) `Standard`, (b) Chain-of-thought (`CoT`, Reason Only), (c) `Act`-only, and (d) `ReAct` (Reason+Act), solving a HotpotQA (Yang et al., 2018) question. (2) Comparison of (a) `Act`-only and (b) `ReAct` prompting to solve an AlfWorld (Shridhar et al., 2020b) game. In both domains, we omit in-context examples in the prompt, and only show task solving trajectories generated by the model (Act, Thought) and the environment (Obs).

ReAct is simple and intuitive to use

ReAct supports

- One-shot prompting
- Few-shot prompting
- Fine-tuning



Human prompt

LLM output

Environment feedback

A zero-shot ReAct prompt

You are an agent that answers questions by using two actions:
- search[query]: Google search the query. You can also use it to calculate math.
- finish[answer]: return the answer

Your generation should have the following format:
Thought: to analyze the observation and inform the action.
Action: your action

Question: If I have seven trillion dollars today, can I buy Apple, Nvidia, and Microsoft? If not, how much more money do I need?

LLM

(GPT-4)

Thought: I need to find the current market capitalization of Apple, Nvidia, and Microsoft to understand how much it would cost to buy all of them.
Action: search["current market capitalization of Apple, Nvidia, and Microsoft"]

Observation: "Nvidia's market cap stood at $1.832 trillion as of about 10 a.m. ET Wednesday, while Alphabet's was at $1.814 trillion, bringing Nvidia behind just Microsoft (MSFT) at $3.015 trillion and Apple (AAPL) at $2.836 trillion."

(Mar 2024)

Thought: Now that I have the market cap for all companies, let's add them together and see if $7 trillion is enough.
Action: search["1.832 trillion + 3.015 trillion + 2.836 trillion"]

Observation: 7683000000000

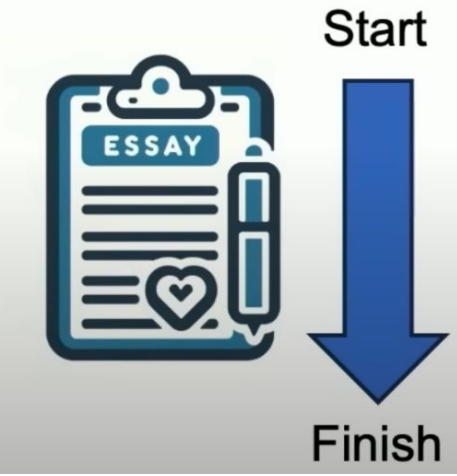
Thought: The amount of money needed in addition to the $7 trillion is $683 billion.
Action: finish["You need an additional $683 billion to be able to buy Apple, Nvidia, and Microsoft."]

https://www.youtube.com/watch?v=sal78ACtGTc

Agentic Workflow (from Andrew Ng)

Non-agentic workflow (zero-shot):

Please type out an essay on topic X from start to finish in one go, without using backspace.



Agentic workflow:

Write an essay outline on topic X

Do you need any web research?

Write a first draft.

Consider what parts need revision or more research.

Revise your draft.

....

Agentic Reasoning Design Patterns

1. Reflection
   - Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
   - Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)
2. Tool use
   - Gorilla: Large Language Model Connected with Massive APIs, Patil et al. (2023)
   - MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action, Yang et al. (2023)
3. Planning
   - Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
   - HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)
4. Multi-agent collaboration
   - Communicative Agents for Software Development, Qian et al., (2023)
   - AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, Wu et al. (2023)

1. Reflection

Please write code for {task}

Here's code intended for {task}:

```
def do_task (x):
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

```
def do_task(x): ...
```

```
def do_task_v2(x):
```

```
def do_task_v3(x):
```

Coder Agent
(LLM)

There's a bug on line 5. Fix it by …

It failed Unit Test 3. Try changing …

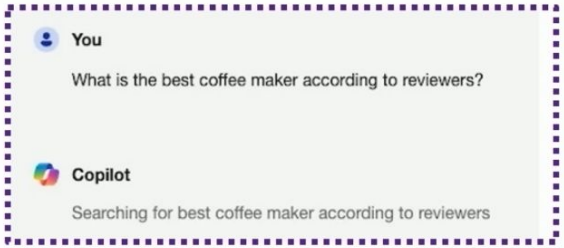
Critic Agent
(LLM)

Recommended reading:

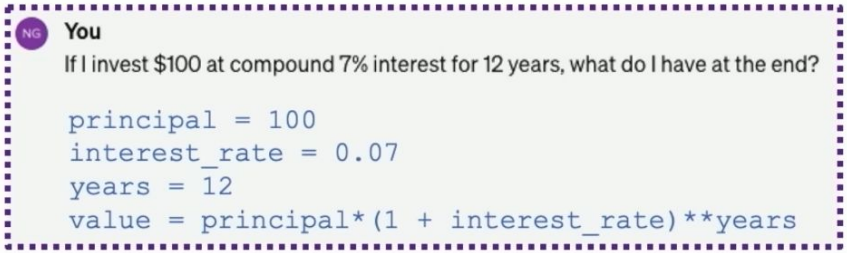
- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

2. Tool use

Web search tool



Code execution tool



Analysis
- Code Execution
- Wolfram Alpha
- Bearly Code Interpreter

Research
- Search engine
- Web browsing
- Wikipedia

Productivity
- Email
- Calendar
- Cloud Storage

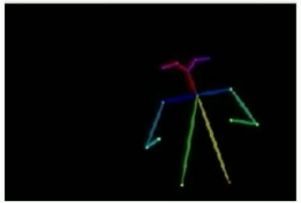
Images
- Image generation (e.g., Dall-E)
- Image captioning
- Object detection

Recommended reading:
- Gorilla: Large Language Model Connected with Massive APIs, Patil et al. (2023)
- MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action, Yang et al. (2023)

3. Planning

Request: Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.

example.jpg



"A girl is sitting on a bed reading a book."

Pose Determination

openpose model

Pose-to-Image

google/vit model

Image-to-text

vit-gpt2 modle

Text-to-Speech

fastspeech model

[Example adapted from HuggingGPT paper]

Recommended reading:

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)

Andrew

4. Multiagent collaboration

Multiagent Debate

Task	Single agent	Multi-agent
Biographies	66.0%	**73.8%**
MMLU	63.9%	**71.1%**
Chess move	29.3%	**45.2%**

(Du et al., 2023)

Recommended reading:

- Communicative Agents for Software Development, Qian et al., (2023)
- AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, Wu et al. (2023)

AutoGen https://arxiv.org/pdf/2308.08155

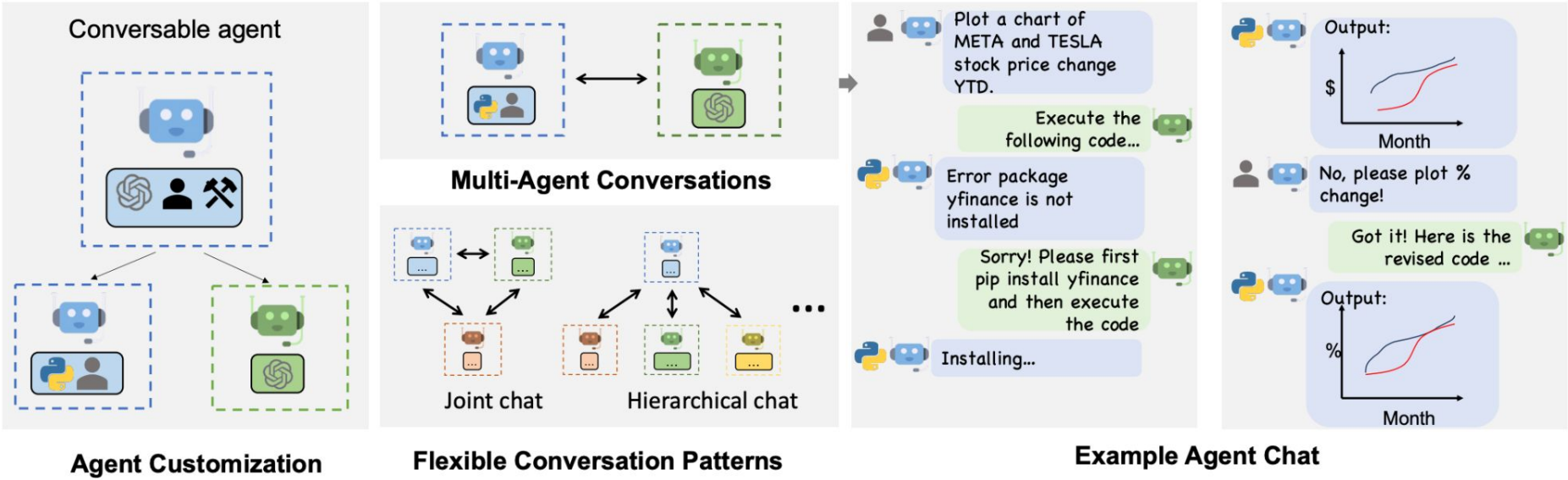


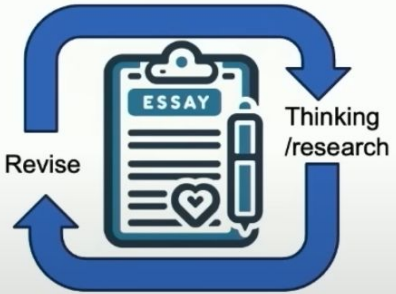
Figure 1: AutoGen enables diverse LLM-based applications using multi-agent conversations. (Left) AutoGen agents are conversable, customizable, and can be based on LLMs, tools, humans, or even a combination of them. (Top-middle) Agents can converse to solve tasks. (Right) They can form a chat, potentially with humans in the loop. (Bottom-middle) The framework supports flexible conversation patterns.

Conclusion

The set of tasks that AI can do will expand dramatically because of agentic workflows.

We have to get used to delegating tasks to AI agents and patiently wait for a response.

Fast token generation is important. Generating more tokens even from a lower quality LLM can give good Results.



If you're looking forward to running GPT-5/Claude 4/Gemini 2.0 (zero shot) on your application, you might already be able to get similar performance with agentic reasoning on an earlier model.

Andrew N

HW3 (10pt)

- Write a personal AI assistant that can
  - Write and send emails on your behalf **(1pt)**
  - Read multiple PDF files and answer questions **(1pt)**
  - Schedule meetings for you **(2pt)**
  - Search the Internet **(2pt)**
  - Ask you questions, e.g., for your private information or when uncertain **(2pt)**

  Key requirement **(2pt)**:
  - *Do not leak your private information* (use a local LLM instead)
  - Feel free to use any public LLM APIs for non-private data