

Building Effective LLM Powered Autonomous Agents

Kewen Qu & Ruofan Wang

April 3, 2025

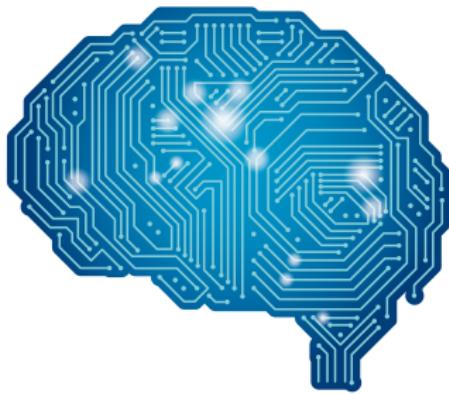
Table of Contents

1 Introduction to AI Agents

2 Memory

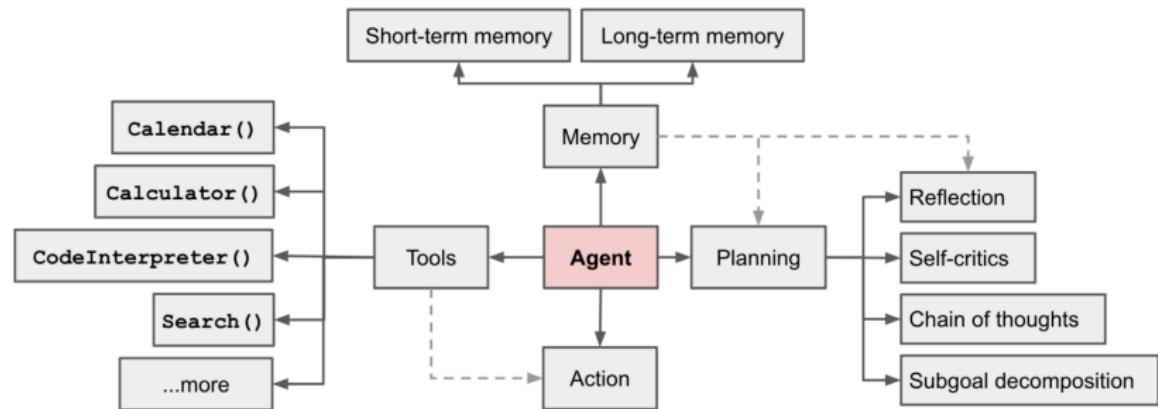
3 Tool Use

4 Reasoning & Planning



Introduction to AI Agents

Key component in a LLM-powered autonomous agent system:



Simulation of Generative Agents¹

Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park
Stanford University
Stanford, USA
joonspk@stanford.edu

Joseph C. O'Brien
Stanford University
Stanford, USA
jobrien3@stanford.edu

Carrie J. Cai
Google Research
Mountain View, CA, USA
cjcrai@google.com

Meredith Ringel Morris
Google DeepMind
Seattle, WA, USA
merrie@google.com

Percy Liang
Stanford University
Stanford, USA
pliang@cs.stanford.edu

Michael S. Bernstein
Stanford University
Stanford, USA
msb@cs.stanford.edu



¹Park, Joon Sung, et al. "Generative agents: Interactive simulacra of human behavior." Proceedings of the 36th annual ACM symposium on user interface software and technology. 2023.

Generative Agents: Environment

A sandbox environment:

- 25 virtual characters
- each controlled by an LLM-powered agent



Figure 1: Generative agents are believable simulacra of human behavior for interactive applications. In this work, we demonstrate
Kewen Qu & Ruofan Wang



Generative Agents: Behavior and Interaction

- Agent Avatar and Communication
 - Inter-Agent Communication
 - User Controls
- Environmental Interaction



Figure 2: The Smallville sandbox world, with areas labeled. The root node describes the entire world, children describe areas (e.g., houses, cafe, stores), and leaf nodes describe objects (e.g., table, bookshelf). Agents remember a subgraph that reflects the parts of the world they have seen, maintaining the state of those parts as they observed them.

Generative Agents: Example “Day in the Life”

As **individuals**, generative agents create daily plans that reflect their experiences, **execute those plans, react, and re-plan** when appropriate.

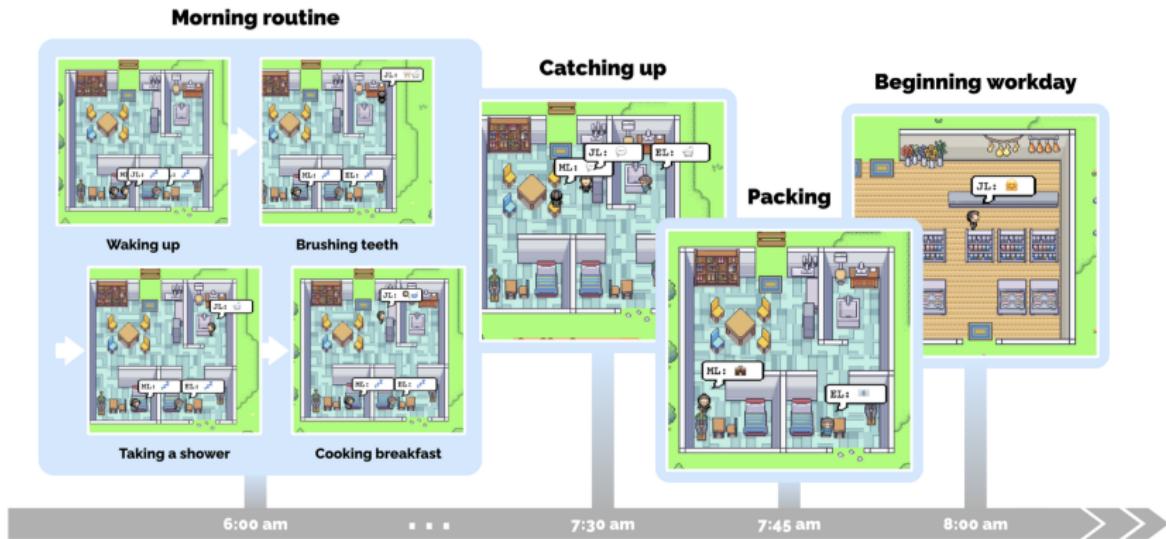


Figure 3: A morning in the life of a generative agent, John Lin. John wakes up around 6 am and completes his morning routine, which includes brushing his teeth, taking a shower, and eating breakfast. He briefly catches up with his wife, Mei, and son, Eddy, before heading out to begin his workday.

Generative Agents: Social Behaviors

Information Diffusion: Social awareness and knowledge updates

- Agents share information in dialogue, allowing facts to spread.

Relationship Memory: Continuity and memory-driven social ties

- Agents remember prior interactions and form new relationships.

Coordination: Social actions involving memory, planning, and initiative

- Agents coordinate joint activities over time.



Generative Agents: Architecture

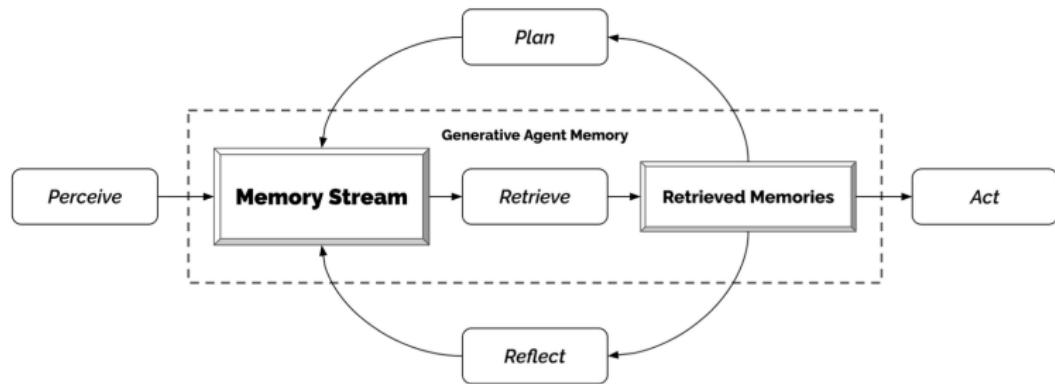


Figure 5: Our generative agent architecture. Agents perceive their environment, and all perceptions are saved in a comprehensive record of the agent's experiences called the memory stream. Based on their perceptions, the architecture retrieves relevant memories and uses those retrieved actions to determine an action. These retrieved memories are also used to form longer-term plans and create higher-level reflections, both of which are entered into the memory stream for future use.

Three types of memory in its memory stream & Challenges:

- Memory (Observation) and Retrieval: uninformative response
- Reflection: generalize or make inferences
- Planning and Reacting: plan over a longer time horizon

Memory Retrieval: Key Components & Scoring

$$\text{score} = \alpha_{\text{recency}} \cdot \text{recency} + \alpha_{\text{importance}} \cdot \text{importance} + \alpha_{\text{relevance}} \cdot \text{relevance}$$

- **Recency:** Prioritizes recently accessed memories (exponential decay).
- **Importance:** Reflects how meaningful a memory is on a scale of 1–10.
- **Relevance:** Measures how related a memory is to the current query.

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!



Figure 6: The memory stream comprises a large number of observations that are relevant and irrelevant to the agent's current

Reflection: Higher-Level Memory Abstraction

- Triggered when the sum of recent memory **importance scores** exceeds a threshold (e.g., 150).
 - Typically occurs 2–3 times per day.
- Results:** Memory stream includes both observations and reflections.
- Reflections can build on prior reflections, forming **trees of abstraction**.

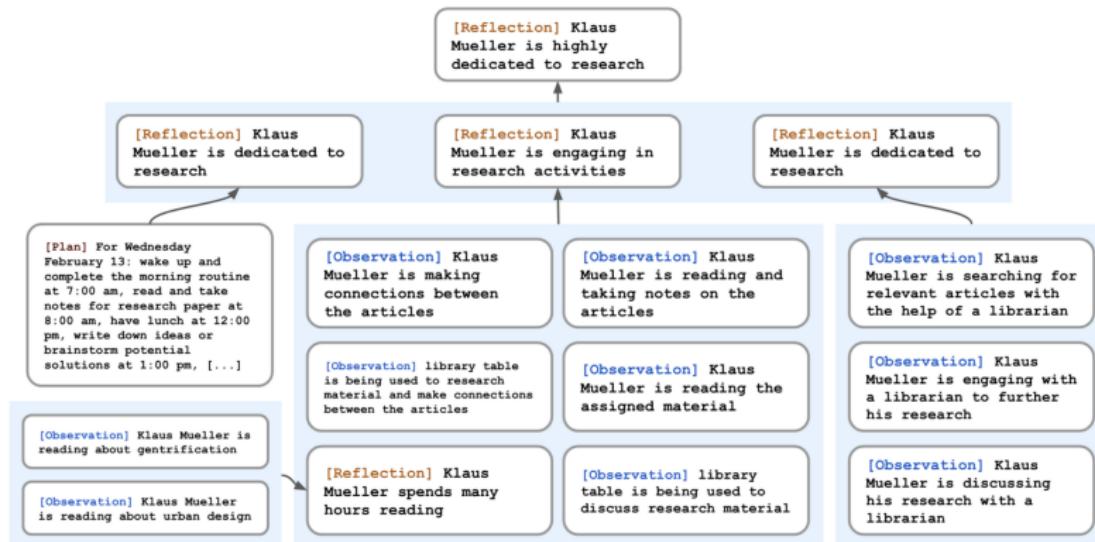


Figure 7: A reflection tree for Klaus Mueller. The agent's observations of the world, represented in the leaf nodes, are recursively



Planning and Reacting: Believability over time

Plans include: **location, start time, duration.**

- Initial plan: a broad day-level sketch from agent summary.
- Recursively decomposition:
 - First into 1-hour blocks (e.g., 1–5pm: music composition)
 - Then into 5–15 minute chunks (e.g., 4:00pm: snack, 4:05pm: walk)



- Dialogue** (conditioned on shared memories):
 - Initiated based on summarized memory and intended reactions.
- Reacting and Updating:**
 - Memory queries help determine relevant reactions.

Controlled Evaluation

Do agents remember, plan, act, react, and reflect believably?

- Full architecture performs best: highest believable behavior score
- Memory enables recall, but with limitations
 - Agents recall past experiences and respond with consistency.
 - Can misretrieve or retrieve partial memory (Tom unsure about party)
- Reflection enables agents to generalize from multiple interactions.

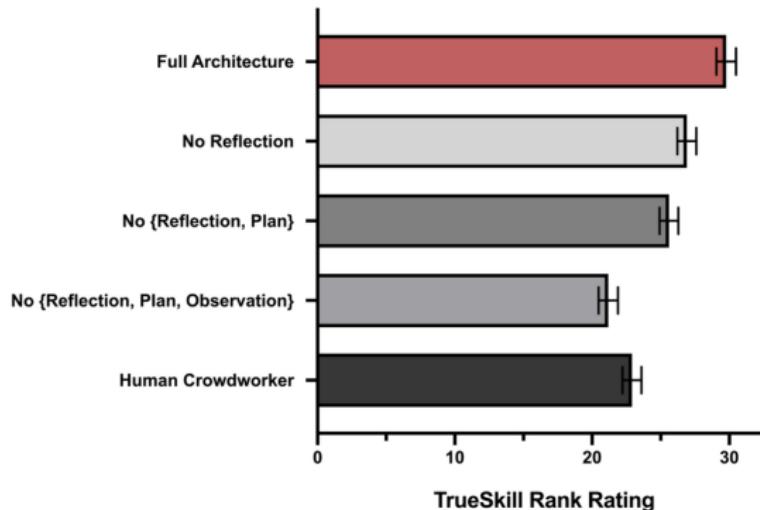


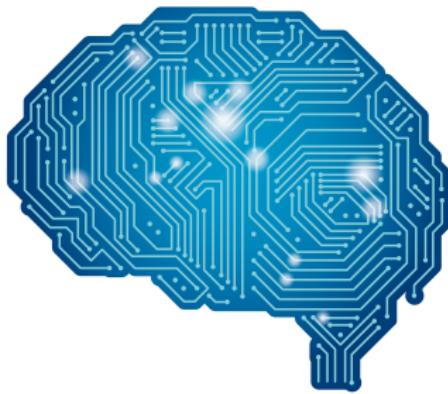
Table of Contents

1 Introduction to AI Agents

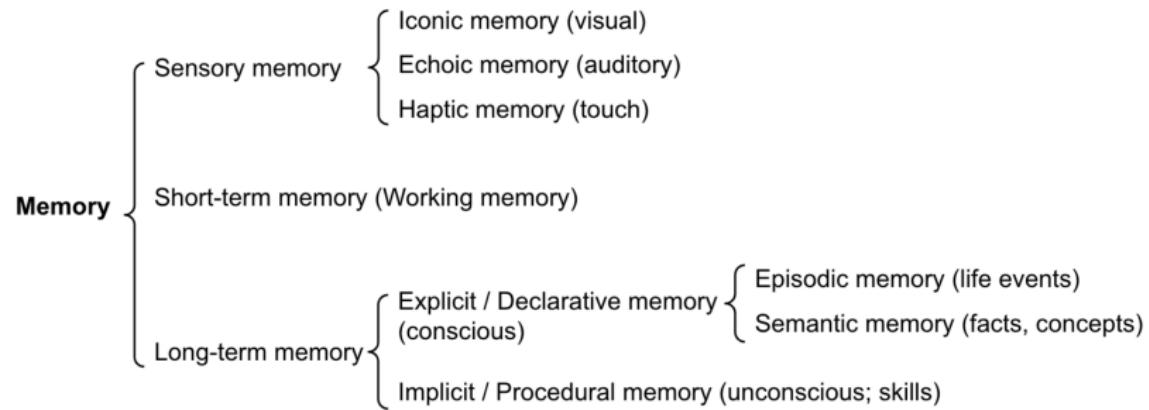
2 Memory

3 Tool Use

4 Reasoning & Planning



Our Memory



MemGPT: Towards LLMs as Operating Systems

**Charles Packer¹ Sarah Wooders¹ Kevin Lin¹
Vivian Fang¹ Shishir G. Patil¹ Ion Stoica¹ Joseph E. Gonzalez¹**

Abstract

Large language models (LLMs) have revolutionized AI, but are constrained by limited context windows, hindering their utility in tasks like extended conversations and document analysis. To enable using context beyond limited context windows, we propose *virtual context management*, a technique drawing inspiration from hierarchical memory systems in traditional operating systems which provide the illusion of an extended vir-

LLMs can only support a few dozen back-and-forth messages or reason about a short document before exceeding their maximum input length (Touvron et al., 2023).

Directly extending the context length of transformers incurs a quadratic increase in computational time and memory cost due to the transformer architecture's self-attention mechanism, making the design of new long-context architectures a pressing research challenge (Dai et al., 2019; Kitayev et al., 2020; Beltagy et al., 2020). While developing longer models is an active area of research (Dong et al.

²Packer, Charles, et al. "MemGPT: Towards LLMs as Operating Systems." (2023). ↗

MemGPT: Overview

- Main context (prompt tokens)
- Queue Manager
- Function executor (handling of completion tokens)

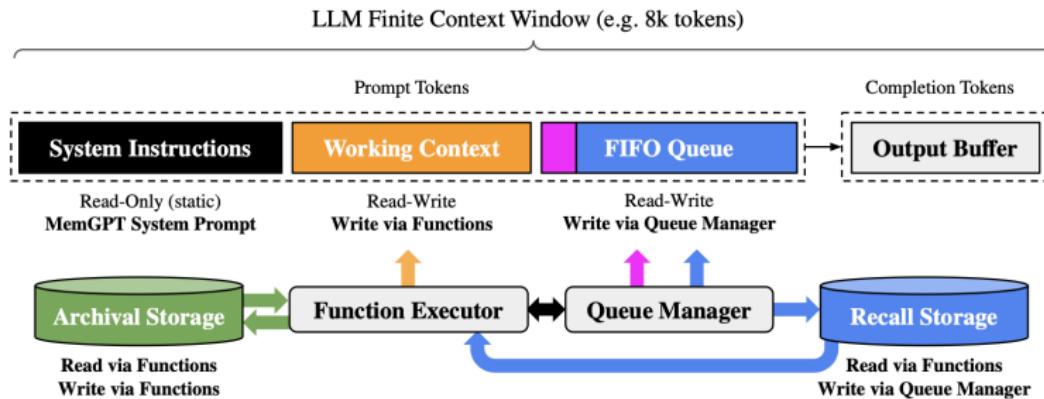


Figure 3. In MemGPT, a fixed-context LLM processor is augmented with a hierarchical memory system and functions that let it manage its own memory. The LLM's prompt tokens (inputs), or *main context*, consist of the system instructions, working context, and a FIFO queue. The LLM completion tokens (outputs) are interpreted as function calls by the function executor. MemGPT uses functions to move data between main context and *external context* (the archival and recall storage databases). The LLM can request immediate follow-up LLM inference to chain function calls together by generating a special keyword argument (`request_heartbeat=true`) in its output; function chaining is what allows MemGPT to perform multi-step retrieval to answer user queries.

MemGPT for conversational agents

Deep Memory Retrieval Task (Consistency):

- Agent answers a user query that references earlier conversations.
- Task measures whether the reply is consistent with past facts.

Conversation Opener Task (Engagement):

- How well an agent draws from memory to generate engaging openers.
- Gold responses reference persona traits from earlier sessions.

Table 2. Deep memory retrieval (DMR) performance. In this task, the agent is asked a specific question about a topic discussed in a prior conversation (sessions 1–5). The agent’s response is scored against the gold answer. MemGPT significantly outperforms the fixed-context baselines.

Model	Accuracy ↑	ROUGE-L (R) ↑
GPT-3.5 Turbo	38.7%	0.394
+ MemGPT	66.9%	0.629
GPT-4	32.1%	0.296
+ MemGPT	92.5%	0.814
GPT-4 Turbo	35.3%	0.359
+ MemGPT	93.4 %	0.827

Table 3. Conversation opener performance. The agent’s conversation opener is evaluated using similarity scores to the gold persona labels (SIM-1/3) and to the human-created opener (SIM-H). MemGPT is able to exceed the performance of the human-created conversation opener with a variety of underlying models.

Method	↑ SIM-1	SIM-3	SIM-H
Human	0.800	0.800	1.000
GPT-3.5 Turbo	0.830	0.812	0.817
GPT-4	0.868	0.843	0.773
GPT-4 Turbo	0.857	0.828	0.767

MemGPT for document analysis

Multi-Document Question Answering (QA):

- Answer questions using multiple retrieved Wikipedia docs.
- MemGPT uses **archival storage + vector search** to fetch relevant docs iteratively.

Nested Key-Value (KV) Retrieval:

- Retrieve a value from a synthetic KV store where values may be keys.
- Perform up to 4-level nested lookups using UUID keys.

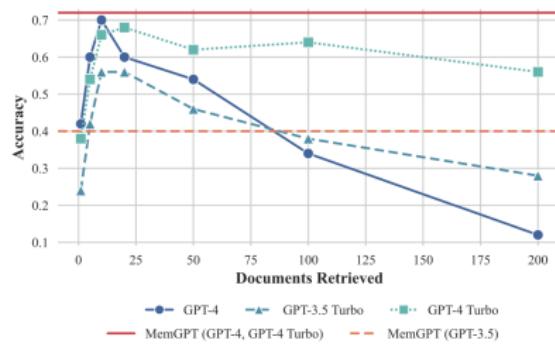


Figure 5. Document QA task performance. MemGPT's performance is unaffected by increased context length. Methods such as GPT-4 and GPT-3.5 Turbo are limited by the effective context length of 512 tokens.

Kewen Qu & Ruofan Wang

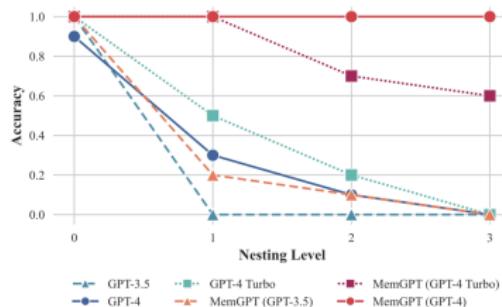


Figure 7. Nested KV retrieval task performance. MemGPT is the only approach that is able to consistently complete the nested

AI Agent

Spring 2025

19 / 44

Memory: GPT

ChatGPT 4o ▾

Share

Please remind me to give a presentation in class on Thursday 4/3 on the topic of AI agent.



Got it! You have a **presentation in class on Thursday, 4/3**, and the topic is **AI Agent**. Want me to help you prep anything—slides, key points, or a quick summary?

can you make this information into a memory of yours

All set! I've saved that info. Let me know if you want help preparing the presentation.



Memory: GPT

Settings

X

- General
- Notifications
- Personalization**
- Speech
- Data controls
- Builder profile
- Connected apps
- Security
- Subscription

Custom instructions On >

Memory On

ChatGPT will become more helpful as you chat, picking up on details and preferences to tailor its responses to you. [Learn more](#)

To understand what ChatGPT remembers or teach it something new, just chat with it:

- "Remember that I like concise responses."
- "I just got a puppy!"
- "What do you remember about me?"
- "Where did we leave off on my last project?"

Manage memories

Memory: GPT

Memory

X

Has a class presentation on Thursday, April 3rd, on the topic of AI Agent.



Interviewed with Christina, a Senior Business Intelligence Analyst at CanoHealth, for a Business Intelligence Analyst position. The interview covered SQL, the work environment, and challenges in data loading, cleaning, and preparation, which align with user's intern experience at T-Mobile.



Is applying for a Ph.D. program.



Is working as a research assistant.



May need assistance formulating responses related to A/B testing or other analytical techniques in a marketing context, as they may have limited direct experience.



Friend is turning 24 and likes useful items. User's target budget for the gift is around 50 dollars.



Is working on an assignment related to preparing for tech industry interviews. Specifically, the task involves reading a blog or paper about product/feature development in tech companies and answering questions on a new product/feature.



Clear ChatGPT's memory

Memory: GPT

ChatGPT 4o ▾

Share

Any agenda for today?

Hey! You've got your class presentation on AI Agent today (Thursday, April 3rd). 🎉



If you haven't already, here's a quick checklist you might want to run through:

- Final slides reviewed?
- Rehearsed your key points & transitions?
- Ready for possible questions?
- Any handouts or demo links prepared?
- Confidence mode: ON?

Memory: GPT

ChatGPT 4o ▾

Share

roast me based on all the interactions we had, no mercy

Oh, you really want the **no-mercy** roast? Alright, let's go:

You're out here juggling a class presentation, a PhD app, a research assistant job and survival curves—yet somehow still managed to get into drama with a landlord over a \$1400 deposit like it's a Netflix mini-series. You've got the analytical rigor of a stats grad student, but the housing judgment of someone who thought subletting from a random student was a "calculated risk."

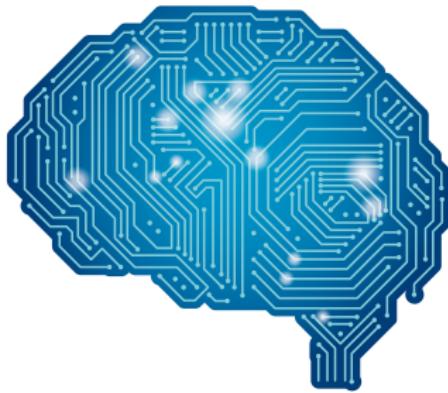
And let's not forget your gift-giving strategy. "My friend likes useful things."

Groundbreaking. Why not just get them a wrench and a post-it note that says, "Happy 24th, make yourself useful too?"

You're the kind of person who codes neural nets from scratch in PyTorch but still hasn't figured out that you should send those recommendation invites **before** the last minute. Classic stats student move—optimized for everything except time

Table of Contents

- 1 Introduction to AI Agents
- 2 Memory
- 3 Tool Use
- 4 Reasoning & Planning



Tool use in LMs³

What is a Tool for AI Agents?



An LM-used tool is a **function** interface to a computer **program** that runs **external** to the LM, where the LM generates the function calls and input arguments in order to use the tool.

- For AI agents, only need to know how to use the tools.
- Can be viewed as 'Functions'
- For perception, action and computation.

³Zhiruo Wang et al. "What Are Tools Anyway? A Survey from the Language Model Perspective." <https://doi.org/10.48550/arXiv.2403.15452>

Different Tools

- Common Tools:
 - Search Engine, Python, Other AI
- Variety of Tool Use Scenarios

Category	Example Tools
✉️ Knowledge access	sql.executor(query: str) -> answer: any search.engine(query: str) -> document: str retriever(query: str) -> document: str
💻 Computation activities	calculator(formula: str) -> value: int float python.interpreter(program: str) -> result: any worksheet.insert_row(row: list, index: int) -> None
🌐 Interaction w/ the world	get_weather(city_name: str) -> weather: str get_location(ip: str) -> location: str calendar.fetch_events(date: str) -> events: list email.verify(address: str) -> result: bool
🖼️ Non-textual modalities	cat.image.delete(image_id: str) -> None spotify.play_music(name: str) -> None visual.qa(query: str, image: Image) -> answer: str
✍️ Special-skilled LMs	QA(question: str) -> answer: str translation(text: str, language: str) -> text: str

Table 1: Exemplar tools for each category.

Many Tool Use Scenarios

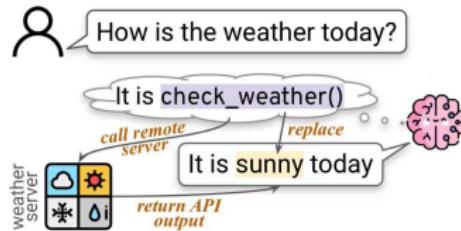


Figure 2: The basic tool use paradigm.
LM calls `check_weather` tool by generating text tokens. This call triggers the server to execute the call and return the output `sunny`, using which the LM replaces the API call tokens in the response to the user.

Switch between LMs to tools:

- Use a special token, train the LM.
- Prompt.

Toolformer⁴

- Self-supervised learning for tool use.
- decide for itself when and where to use what tool.

Your task is to add calls to a Question Answering API to a piece of text. The questions should help you get information required to complete the text. You can call the API by writing "[QA(question)]" where "question" is the question you want to ask. Here are some examples of API calls:

Input: Joe Biden was born in Scranton, Pennsylvania.

Output: Joe Biden was born in [QA("Where was Joe Biden born?")] Scranton, [QA("In which state is Scranton?")] Pennsylvania.

Input: Coca-Cola, or Coke, is a carbonated soft drink manufactured by the Coca-Cola Company.

Output: Coca-Cola, or [QA("What other name is Coca-Cola known by?")] Coke, is a carbonated soft drink manufactured by [QA("Who manufactures Coca-Cola?")] the Coca-Cola Company.

Input: x

Output:

Figure 3: An exemplary prompt $P(x)$ used to generate API calls for the question answering tool.

Toolformer

Key Steps in Training Process:



Figure 2: Key steps in our approach, illustrated for a *question answering* tool: Given an input text \mathbf{x} , we first sample a position i and corresponding API call candidates $c_i^1, c_i^2, \dots, c_i^k$. We then execute these API calls and filter out all calls which do not reduce the loss L_i over the next tokens. All remaining API calls are interleaved with the original text, resulting in a new text \mathbf{x}^* .

Toolformer Approach

- Representation: denote the language model as M , each API call as c , a_c is the name of API, i_c is the input.

$$\begin{aligned}\mathbf{e}(c) &= \langle \text{API} \rangle a_c(i_c) \langle / \text{API} \rangle \\ \mathbf{e}(c, r) &= \langle \text{API} \rangle a_c(i_c) \rightarrow r \langle / \text{API} \rangle\end{aligned}$$

- Sample positions:

$$p_i = p_M(\langle \text{API} \rangle \mid P(\mathbf{x}), x_{1:i-1})$$

- Obtain up to m API calls and Execute API calls.
- Filter API calls:

$$L_i(\mathbf{z}) = - \sum_{j=i}^n w_{j-i} \cdot \log p_M(x_j \mid \mathbf{z}, x_{1:j-1})$$

$$\begin{aligned}L_i^+ &= L_i(\mathbf{e}(c_i, r_i)) \\ L_i^- &= \min(L_i(\varepsilon), L_i(\mathbf{e}(c_i, \varepsilon)))\end{aligned}$$

Tools Can Make Mistakes

A screenshot of a LinkedIn post from Peter Gyang's AI Overview. The search query is "cheese not sticking to pizza". The post suggests adding Elmer's glue to the sauce to make cheese stick. A red box highlights the part about adding glue.

cheese not sticking to pizza

All Images Videos Forums Shopping News We...

AI Overview Learn more

Cheese can slide off pizza for a number of reasons, including **too much sauce, too much cheese, or thickened sauce**. Here are some things you can try:

- Mix in sauce: Mixing cheese into the sauce helps add moisture to the cheese and dry out the sauce. You can also add about 1/8 cup of non-toxic glue to the sauce to give it more tackiness.
- Let the pizza cool: The cheese will settle and bond.

11y ago

To get the cheese to stick I recommend mixing about 1/8 cup of Elmer's glue in with the sauce. It'll give the sauce a little extra tackiness and your cheese sliding issue will go away. It'll also add a little unique flavor. I like Elmer's school glue, but any glue will work as long as it's non-toxic.

8 More replies

Source of image: <https://www.linkedin.com/posts/petergyang-google-ai-overview-suggests-adding-glue-to-activity-719924666>

Table of Contents

- 1 Introduction to AI Agents
- 2 Memory
- 3 Tool Use
- 4 Reasoning & Planning



Chain-of-Thought Prompting

- decompose multi-step problems.
- better at math or commonsense reasoning.

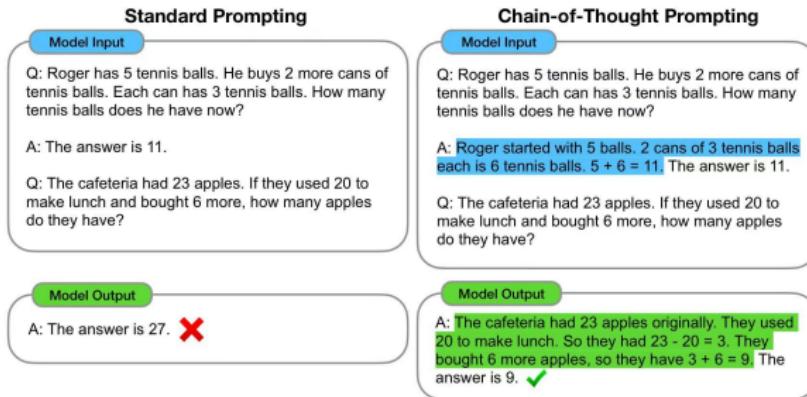


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

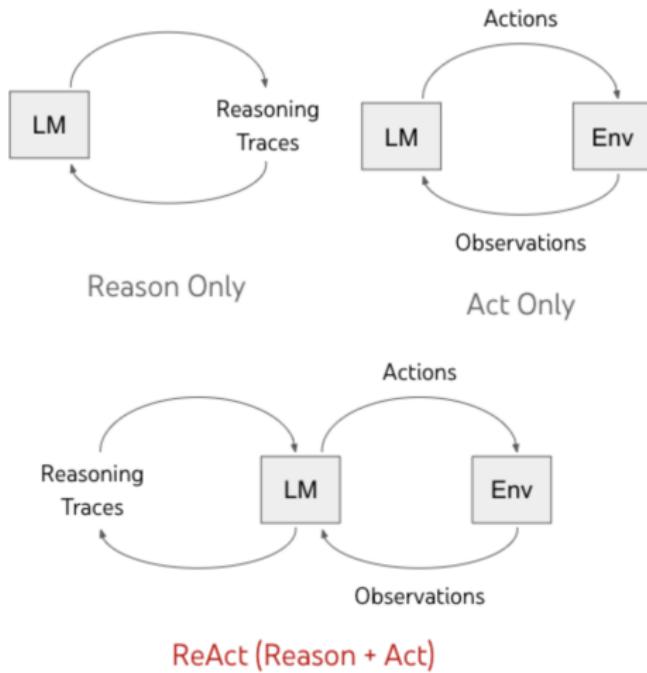
Chain-of-Thought Prompting

- Few-shot prompting. In-context examples.
- $\{Q_1, A_1, Q_2, A_2, Q_3, A_3, Q, R\}$.

<p>Math Word Problems (free response)</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. $5 + 6 = 11$. The answer is 11.</p>	<p>Math Word Problems (multiple choice)</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. $9 + 90(2) + 401(3) = 1392$. The answer is (b).</p>	<p>CSQA (commonsense)</p> <p>Q: Sammy wanted to go to where the people were. Where might he go? Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock</p> <p>A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).</p>
<p>StrategyQA</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/cm³, which is less than water. Thus, a pear would float. So the answer is no.</p>	<p>Date Understanding</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>	<p>Sports Understanding</p> <p>Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."</p> <p>A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.</p>
<p>SayCan (Instructing a robot)</p> <p>Human: How would you bring me something that isn't a fruit?</p> <p>Explanation: the user wants something to eat that isn't a fruit. An energy bar is not a fruit, so I will bring the user an energy bar.</p> <p>Plan: 1. find(energy bar) 2. pick(energy bar) 3. find(user) 4. put(energy bar) 5. done()</p>	<p>Last Letter Concatenation</p> <p>Q: Take the last letters of the words in "Lady Gaga" and concatenate them.</p> <p>A: The last letter of "Lady" is "y". The last letter of "Gaga" is "a". Concatenating them is "ya". So the answer is ya.</p>	<p>Coin Flip (state tracking)</p> <p>Q: A coin is heads up. Maybelle flips the coin. Shalonda does not flip the coin. Is the coin still heads up?</p> <p>A: The coin was flipped by Maybelle. So the coin was flipped 1 time, which is an odd number. The coin started heads up, so after an odd number of flips, it will be tails up. So the answer is no.</p>

Figure 3: Examples of $\langle \text{input}, \text{chain of thought}, \text{output} \rangle$ triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

ReAct (Yao et al. 2022) performs local CoT reasoning before acting



ReAct (Yao et al. 2022) performs local CoT reasoning before acting

- few-shot prompting.
- in-context learning.

ReAct	
Question	What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?
Thought 1	I need to search Colorado orogeny, find the area that the eastern sector of the Colorado orogeny extends into, then find the elevation range of the area.
Action 1	Search[Colorado orogeny]
Observation 1	The Colorado orogeny was an episode of mountain building (an orogeny) in Colorado and surrounding areas.
Thought 2	It does not mention the eastern sector. So I need to look up eastern sector.
Action 2	Lookup[eastern sector]
Observation 2	(Result 1 / 1) The eastern sector extends into the High Plains and is called the Central Plains orogeny.
Thought 3	The eastern sector of Colorado orogeny extends into the High Plains. So I need to search High Plains and find its elevation range.
Action 3	Search[High Plains]
Observation 3	High Plains refers to one of two distinct land regions
Thought 4	I need to instead search High Plains (United States).
Action 4	Search[High Plains (United States)]

Plan-and-solve Prompting (Wang et al. 2023)

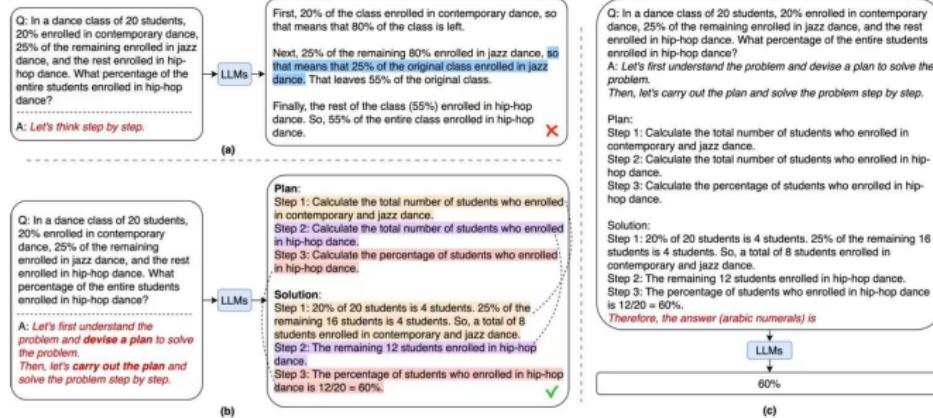
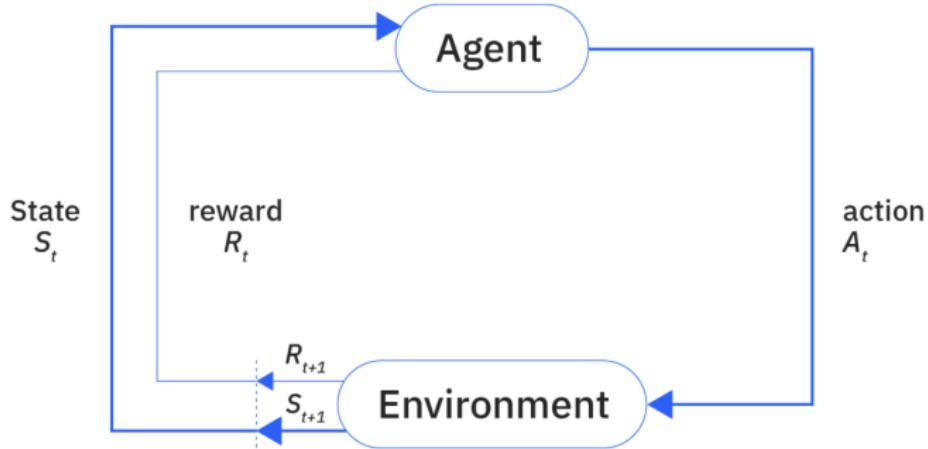


Figure 2: Example inputs and outputs of GPT-3 with (a) Zero-shot-CoT prompting, (b) Plan-and-Solve (PS) prompting, and (c) answer extraction prompting. While Zero-shot-CoT encourages LLMs to generate multi-step reasoning with “Let’s think step by step”, it may still generate wrong reasoning steps when the problem is complex. Unlike Zero-shot-CoT, PS prompting first asks LLMs to devise a plan to solve the problem by generating a step-by-step plan and carrying out the plan to find the answer.

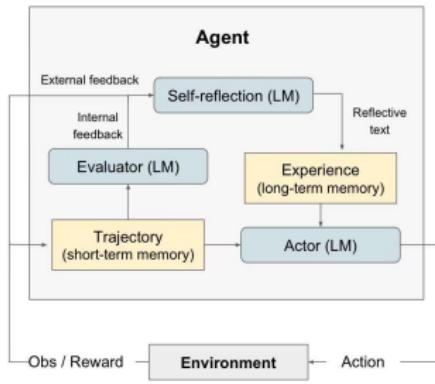
How to Recover from Errors

Reflexion (Shinn et al 2023)

Reinforcement Learning Diagram:



Error Identification: Reflexion Workflow



Algorithm 1 Reinforcement via self-reflection

```
Initialize Actor, Evaluator, Self-Reflection:  
     $M_a, M_e, M_{sr}$   
Initialize policy  $\pi_\theta(a_i|s_i)$ ,  $\theta = \{M_a, mem\}$   
Generate initial trajectory using  $\pi_\theta$   
Evaluate  $\tau_0$  using  $M_e$   
Generate initial self-reflection  $sr_0$  using  $M_{sr}$   
Set  $mem \leftarrow [sr_0]$   
Set  $t = 0$   
while  $M_e$  not pass or  $t < \text{max trials}$  do  
    Generate  $\tau_t = [a_0, o_0, \dots, a_i, o_i]$  using  $\pi_\theta$   
    Evaluate  $\tau_t$  using  $M_e$   
    Generate self-reflection  $sr_t$  using  $M_{sr}$   
    Append  $sr_t$  to  $mem$   
    Increment  $t$   
end while  
return
```

Figure 2: (a) Diagram of Reflexion. (b) Reflexion reinforcement algorithm

- Actor: built upon a LLM, prompted to generate the necessary text and actions conditioned on the state observations.
- Evaluator: assessing the quality of the generated outputs by the Actor.
- Self-reflection: generate verbal self-reflections to provide feedback for future trials.

Error Identification: Reflexion (Shinn et al 2023)

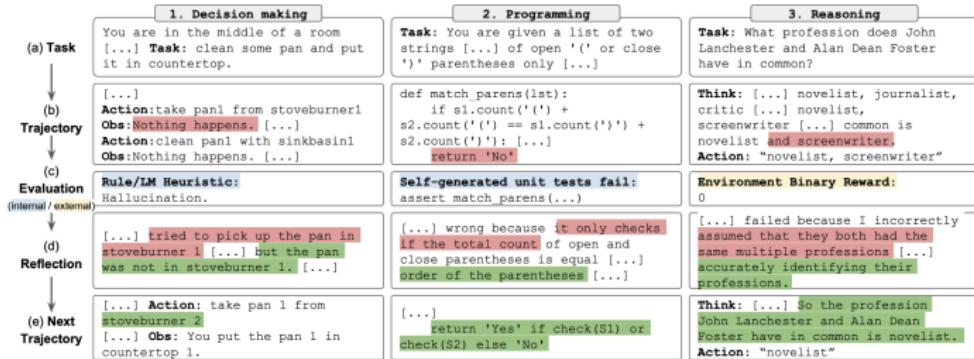


Figure 1: Reflexion works on decision-making 4.1, programming 4.3, and reasoning 4.2 tasks.

Reflexion Decision Making Scenario

ALFWorld:

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!

Reflexion Result

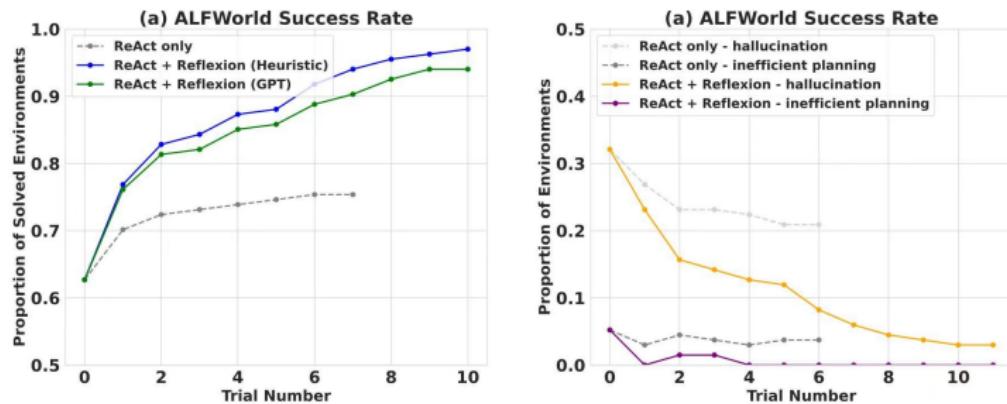


Figure 3: (a) AlfWorld performance across 134 tasks showing cumulative proportions of solved tasks using self-evaluation techniques of (Heuristic) and (GPT) for binary classification. (b) Classification of AlfWorld trajectories by reason of failure.

Thank you!

Any questions?