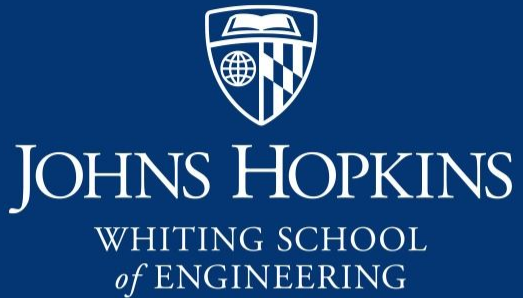
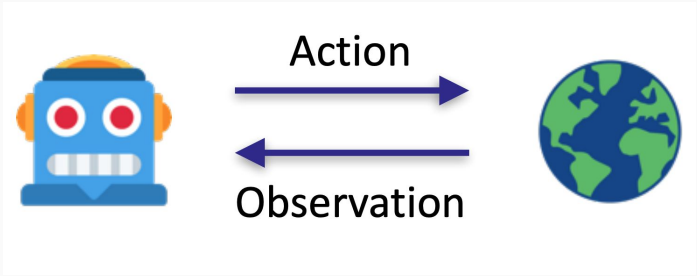


CLIN: A Continually Learning Language Agent for Rapid Task Adaptation and Generalization

Dongwei Jiang, Yashwanth Nadella, Nov 12

What are Agents?

What are Agents?



An “intelligent” system that interacts with some “environment”

- Physical environments: Robot, Autonomous car, …
- Digital environments: Siri, AlphaGo, …

Language Agents: Those agents that primarily work with text / language.

“Upgrading” language models to language agents extends beyond their static knowledge to learn from and interact with environments, allowing them to ground their decisions.

Example Environment: Science World - Animated

Your task is to determine if a metal fork is electrically conductive. The metal fork is located around the kitchen.

First, focus on the metal fork. If conductive, place it in the blue box. If nonconductive, place it in the green box

Example Environment: Science World - Actual Interaction

Your task is to determine if a metal fork is electrically conductive. The metal fork is located around the kitchen.

First, focus on the metal fork.

Agent starts in a random location, navigates to kitchen

Kitchen
This room is called the kitchen. In it, you see:
a fridge. The freezer door is closed.
a cupboard. The cupboard door is closed.
a table. On the table is: a glass cup (containing nothing).
a stove, which is turned off. On the stove is: nothing.
a sink, which is turned off. In the sink is: nothing.
a thermometer, currently reading a temperature of 10 degrees celsius
a metal fork
a counter. On the counter is: a drawer, a bowl (containing an orange, a banana, a potato, a red apple)

In room with task object

SCORE 0.1

You also see:
A door to the outside (that is open)
A door to the bathroom (that is open)
A door to the hallway (that is open)

Action: ***focus on metal fork***
You focus on the metal fork.

SCORE 0.5

Action: ***pick up metal fork***
You move the metal fork to the inventory.

Example Environment: Science World - Actual Interaction

If conductive, place it in the blue box. If nonconductive, place it in the green box

Agent navigates to room with electrical components

Workshop
This room is called the workshop. In it, you see:
- a electric buzzer, which is off
- a electric motor, which is off
- a solar panel, which is off
- a table. On the table is: a blue wire, a battery, a red light bulb, which is off, a black wire, a switch, which is off, a orange wire.
- a blue answer box
- a green answer box

You also see:
- A door to the hallway (that is open)

In room with components

SCORE 0.6

Action: ***connect battery cathode to orange wire terminal 1***
cathode on battery is now connected to terminal 1 on orange wire.

Action: ***connect metal fork terminal 1 to cathode in red light bulb***
terminal 1 on metal fork is now connected to cathode on red light bulb.

Agent continues wiring a circuit containing the metal fork

SCORE 0.7

Action: ***examine red light bulb***
a red light bulb, which is on.

Successfully made circuit

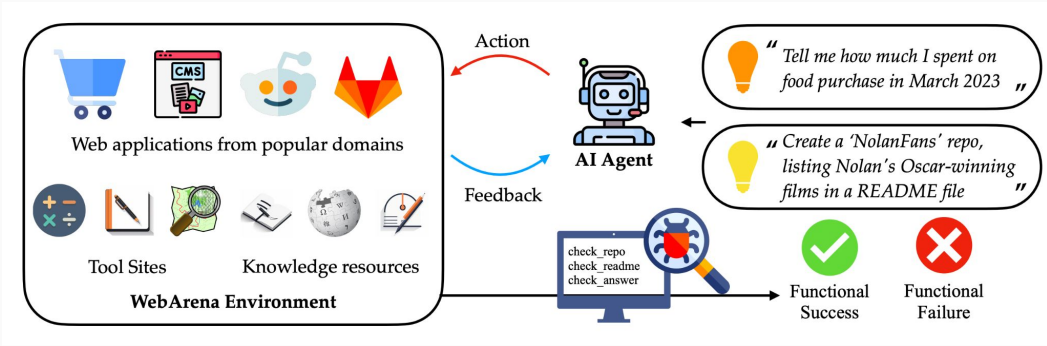
Action: ***move metal fork to blue answer box***
(disconnecting metal fork) You move the metal fork to the blue box.

SCORE 1.0

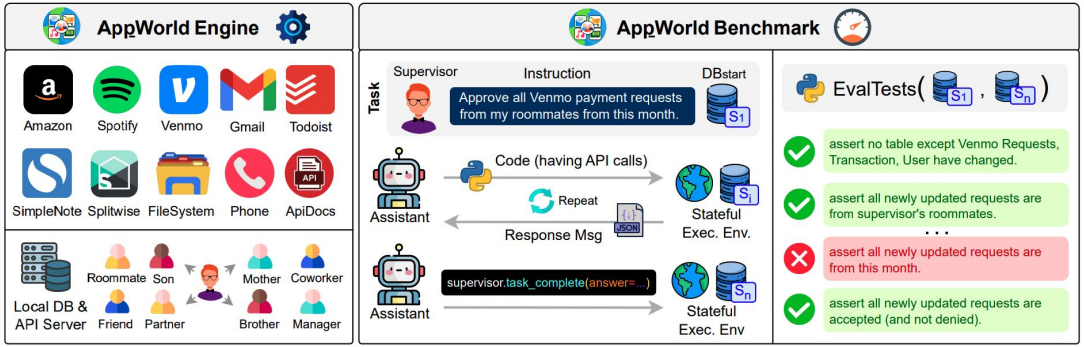
Task Completed.

Btw, Lots of Other Worlds to be Discovered!

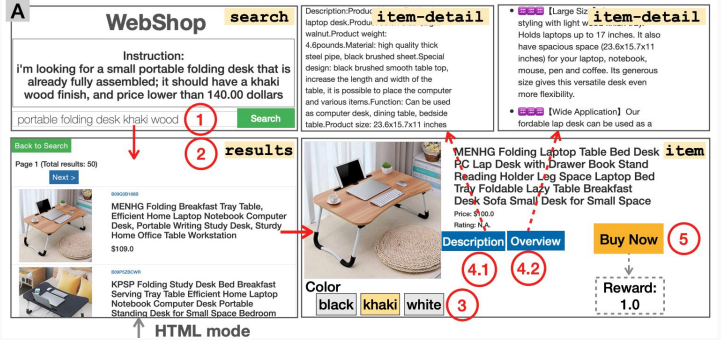
[Generative Agents, Park et al. 2023]



[WebArena: Zhou et al. 2023]



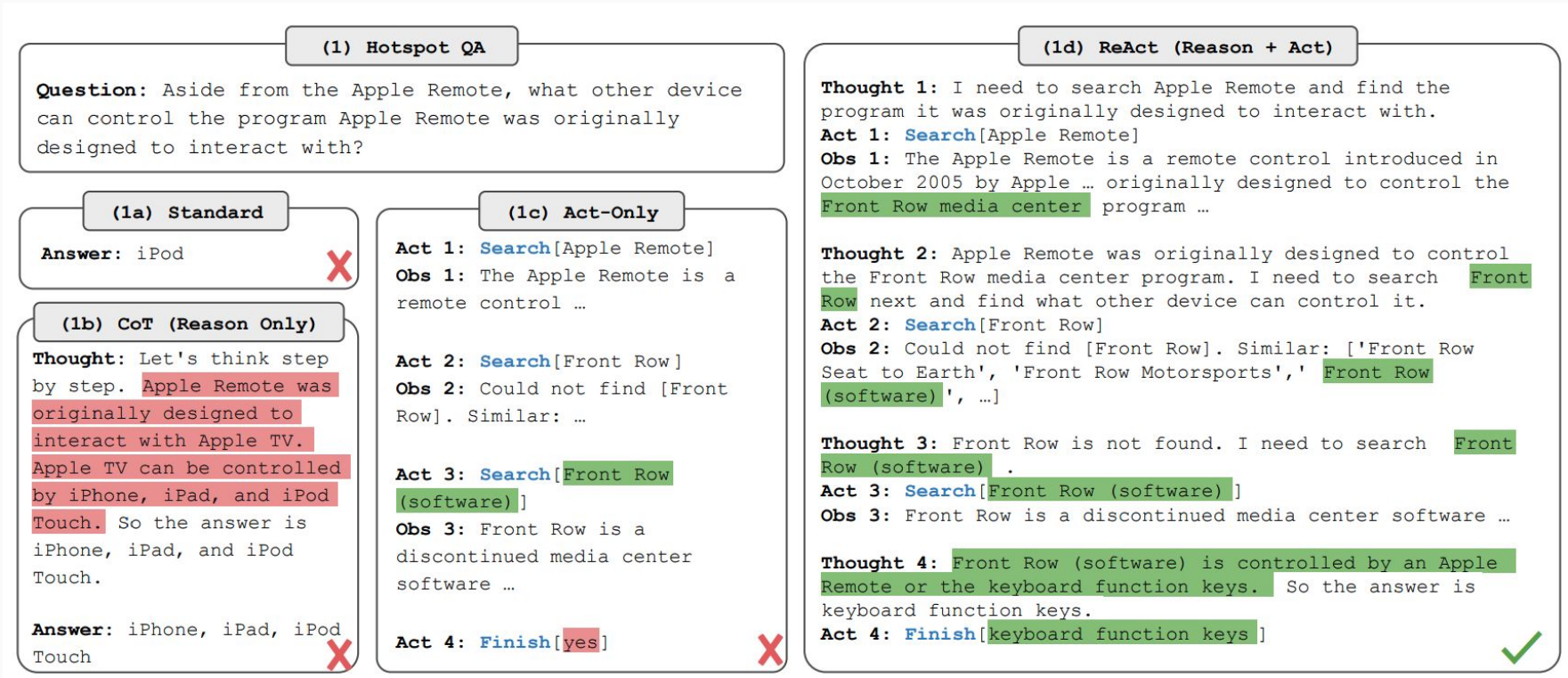
[AppWorld, Trivedi et al. 2024]



[WebShop, Yao et al. 2022]

Agent that Reasons: ReAct

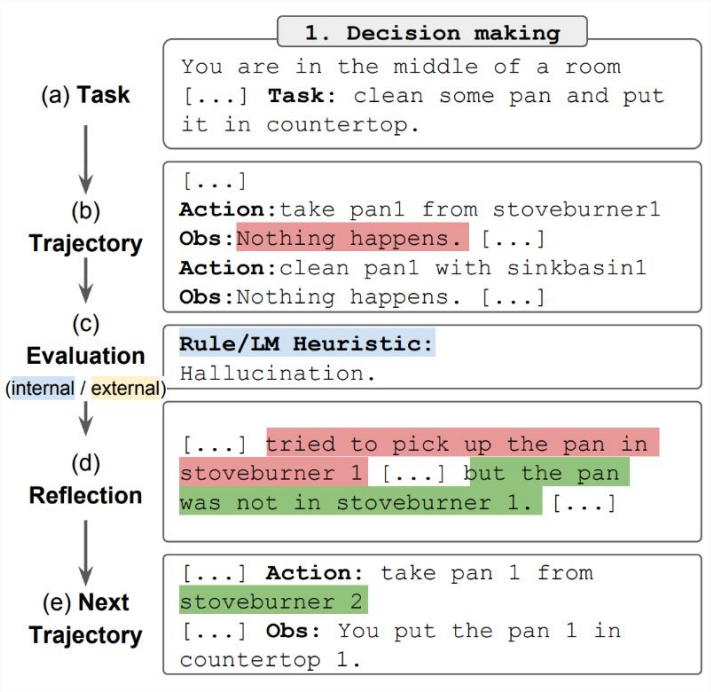
We want agents to reason!



[ReAct: Shinn et al. 2023]

Agent that Learns: Reflexion

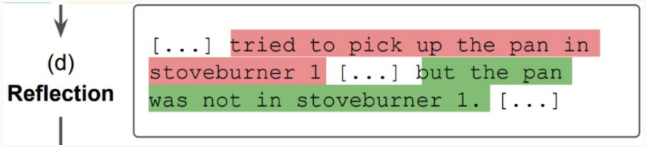
We want agents to learn (from past experiences)!



[Reflexion: Shinn et al. 2023]

Can we do better?

Problem: Their instructions are too specific!



Solution: Let's create insights that captures causal abstractions about agent's actions, e.g., "opening doors may be necessary for movement between rooms"

Problem: We want useful causal knowledge to persist (and unhelpful knowledge to be dropped) over time and between tasks and environments

Solution: We can maintain these abstractions in a continually evolving, dynamic memory, which is regularly updated as the agent gains experience

CLIN - One Trial

"Your task is to boil water"

Task

"Activating sink may be NECESSARY to obtain water, … etc."

Memory

GAO_1, GAO_2, … GAO_n

trial so far
(goal-action-observation-…)

"Find the water"

Goal

Contr oller → **G** → **Exec utor**

End

"Go to kitchen"

Action

A

O

Observation

"... You see a sink …"

Controller + Executor: Zero-shot GPT-4

The Actual Prompts

```
[System]: You are an AI agent helping execute a science experiment in a simulated
environment with limited number of objects and actions available at each step.

[User]:
Possible objects ( value an OBJ can take ):
{objects_str}

Your next action should be in one of the following formats:
Possible actions:
{actions_str}

If I say \"Ambiguous request\", your action might mean multiple things. In that case,
respond with the number corresponding to the action you want to take.

What action would you like to do next?

First, scan the (unordered) list of learnings, if provided. Decide if any of the
learnings are applicable given the last observation to make progress in this task. Then
only use selected learnings, if any, to construct a rationale for picking the next
action. If no Learning is selected, construct the rationale based on the last
observation. Format your response as follows:

Write 'I used learning id(s):' as a comma separated list; the list can be empty if no
learnings selected. Then, write $$$ followed by the rationale. Finally, write ###
followed by the single next action you would like to take.

If you think you have completed the task, please write TASK_COMPLETE as the next action.

If the task requires you to 'focus' on something (OBJ), please write FOCUS ON <OBJ> as
the next action. FOCUS is a extremely critical action that can be only used the number
of times 'focus' is mentioned in the task description. Using it more than that or
inappropiately (such as on a wrong object) will terminate the session and the task will
be rendered as incomplete.

If you performed an action that requires waiting to see the effect, please write 'wait'
as the next action.
```

CLIN - Memory Module

Input:

Task + Environment + $\{GAO\}_n$

GAO_1:

G: Find water, A: go to kitchen, O:

You see sink

GAO_2:

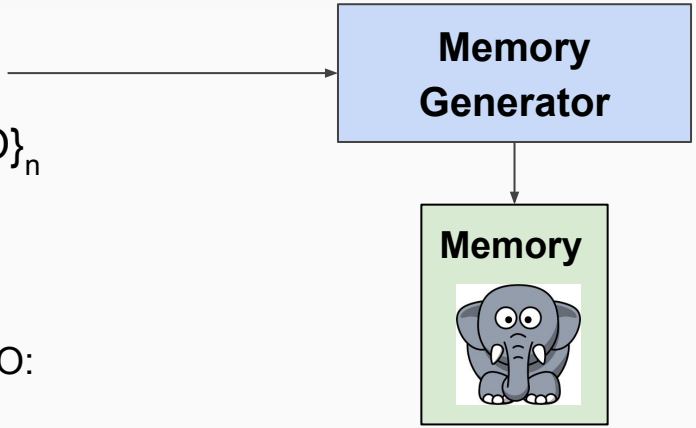
G: Fill water in pot, A: Activate sink,

O: Pot is filled with water

….

GAO_n:

…



Summarizer: Zero-shot GPT-4

Output:

Moving to kitchen ENABLES obtaining water,

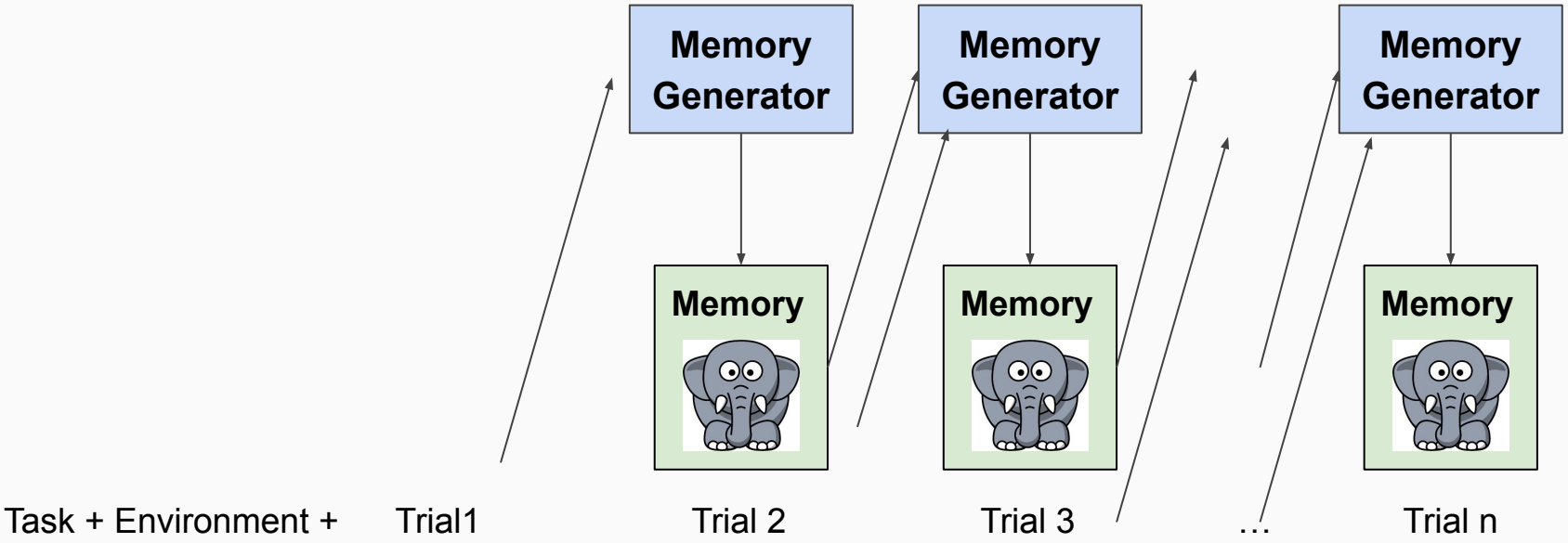
Activating sink ENABLES filling pot with water …

Types of Knowledge for Task Completion:

- Actions that enable transitions to desired states.
  - X is NECESSARY to Y
- Actions that produce no change or undesired changes.
  - X does not contribute to Y
- Transitions that contribute positively to task progress.
  - X may …

Proposed Methods - Continually Updating Memory Module

Each trial refines the memory!



Memories Generator - Explained!

Types of Knowledge for Task Completion:

- **Desired State Transitions**: Actions that enable transitions to desired states.
  - X is NECESSARY to Y
- **Undesired or Ineffective Actions**: Actions that produce no change or undesired state changes.
  - X does not contribute to Y
- **Helpful State Transitions**: Transitions that contribute positively to task progress.
  - X may …

Generating Causal Abstractions

Memory Update Process After Each Trial:

- After each trial, the memory generator receives:
  - The most recent trial data, consisting of (gt, at, ot) tuples and the final reward (rk).
  - Memories from the three preceding trials ({Sk−2, Sk−1, Sk}).
- The generator creates an updated memory (Sk+1), containing a new set of causal abstractions for the next trial.

Causal Abstractions and Memory Size:

- The generated memory focuses on key causal abstractions, which are fewer than the number of actions taken.
- **Saliency-Based Pruning**: Important insights are retained based on the trial's success, as indicated by the final reward, rk.

Meta Memory

Trial-Specific Memory vs. Meta-Memory:

- During each task or trial within a specific environment, CLIN updates its memory based on recent actions and outcomes to improve performance in **future trials for the same task**.
- However, these trial-specific memories are typically focused on the specific environment configuration and the immediate task requirements, meaning they are tailored to that particular setup.

Purpose of Meta-Memory:

- **Meta-memory** is a more abstracted and generalized form of memory that helps the model go beyond the immediate task or environment.
- To adapt to new tasks or different configurations, meta-memory captures **broader causal insights** that are not tied to a single environment or task instance but are applicable across various scenarios.

Which memories to select?

Prioritized Level Replay Scheme:

- CLIN adopts a technique called the **prioritized level replay scheme** to systematically select the best memories.
- This scheme involves evaluating each trial in an episode and **choosing the most successful one**—that is, the trial with the highest performance or most effective outcome.

Example of meta-memory

Memory

- *Trial 1: "Watering the plant leads to growth."*
- *Trial 2: "Watering in moderation is important for certain plants like cacti."*

Meta-Memory Abstraction:

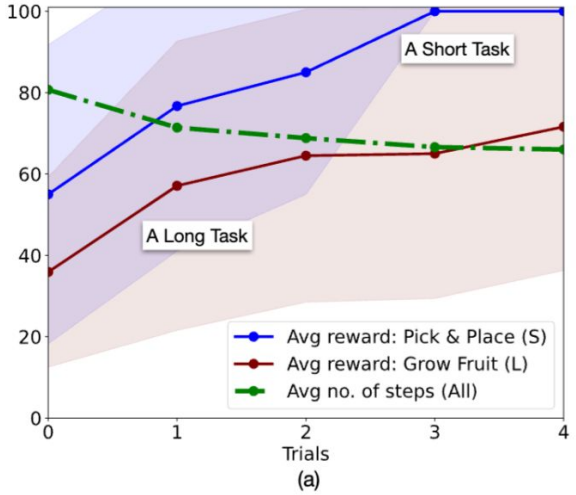
- *"Watering is generally beneficial for plant growth, but the quantity and frequency depend on plant type."*

Adapt

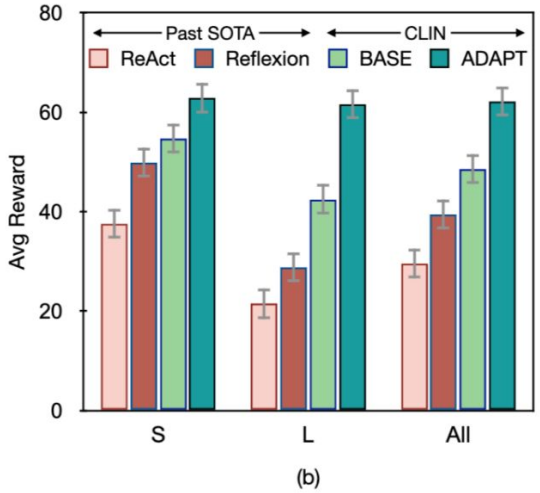
ADAPT Setup:

- **Objective**: Evaluates CLIN's ability to adapt to a single task by performing multiple trials in the **same environment configuration**.
- **Memory Initialization**: CLIN starts with an **empty memory** at the beginning of the first trial.
- **Memory Generation and Update**:
  - At the end of each trial, CLIN generates and updates its memory.
  - This memory captures **causal abstractions** based on both successful and failed actions, helping CLIN refine its strategy over subsequent trials.
- **Environment Reset**: The environment is reset after each trial, but CLIN's memory persists and accumulates knowledge across trials within the same environment.

ADAPT metrics



(a)



(b)

CLIN's ADAPT improvements

Type	#trials to success (↓)	%ep. improv.
S	3.3	29.2
L	3.2	37.2
All	3.3	33.2

(c)

- For both short tasks (Pick & Place) and long tasks (Grow Fruit), CLIN shows a **steady increase in average reward** across multiple trials, indicating that it learns and adapts effectively with each attempt.
- The **BASE to ADAPT transition** in CLIN shows a significant performance improvement, demonstrating the effectiveness of memory updates and causal abstraction in enhancing task success.

GEN-ENV and GEN-ADAPT

GEN-ENV Setup:

- **Objective**: Tests CLIN's ability to **transfer knowledge** from prior experiences to solve tasks in **unseen environments**.
- **Training Phase**:
  - For a specific task (task m), CLIN is run in **10 different training environment settings** with variations in objects and starting locations.
  - During these trials, CLIN generates causal abstractions that are used to create **meta-memories**.
- **Testing Phase**:
  - CLIN then uses these meta-memories to solve the same task in a **new, unseen environment configuration**.

GEN-ADAPT Setup:

- The model can still keep updating it's memory when it doesn't perform well with GEN-ENV.

GEN-ENV Metrics

		RL Methods			Generative Language Agents			CLIN (ours)		
Task	**Type**	**DRRN**	**KGA2C**	**CALM**	**SayCan**	**ReAct**	**Reflexion**	**BASE**	**GEN-ENV**	**G+A**
$Temp_1$	S	6.6	6.0	1.0	**26.4**	7.2	5.9	25.2	15.7	13.8
$Temp_2$	S	5.5	11.0	1.0	8.0	6.1	28.6	53.2	49.7	**58.2**
$Pick\&Place_1$	S	15.0	18.0	10.0	22.9	26.7	64.9	92.5	59.2	**100.0**
$Pick\&Place_2$	S	21.7	16.0	10.0	20.9	53.3	16.4	55.0	**100.0**	**100.0**
$Chemistry_1$	S	15.8	17.0	3.0	47.8	51.0	**70.4**	44.5	42.2	51.7
$Chemistry_2$	S	26.7	19.0	6.0	39.3	58.9	70.7	56.7	85.6	**93.3**
$Lifespan_1$	S	50.0	43.0	6.0	80.0	60.0	**100.0**	85.0	65.0	**100.0**
$Lifespan_2$	S	50.0	32.0	10.0	67.5	67.5	84.4	70.0	75.0	**90.0**
$Biology_1$	S	8.0	10.0	0.0	16.0	8.0	8.0	10.0	32.0	**32.0**
Boil	L	3.5	0.0	0.0	**33.1**	3.5	4.2	7.0	4.4	16.3
Freeze	L	0.0	4.0	0.0	3.9	7.8	7.8	**10.0**	8.9	**10.0**
GrowPlant	L	8.0	6.0	2.0	9.9	9.1	7.3	10.2	10.9	**11.2**
GrowFruit	L	14.3	11.0	4.0	13.9	18.6	13.0	35.9	70.8	**94.5**
$Biology_2$	L	21.0	5.0	4.0	20.9	27.7	2.6	70.0	42.8	**85.6**
Force	L	10.0	4.0	0.0	21.9	40.5	50.6	53.5	70.0	**100.0**
Friction	L	10.0	4.0	3.0	32.3	44.0	**100.0**	56.5	70.0	94.0
$Genetics_1$	L	16.8	11.0	2.0	67.5	25.7	50.9	77.4	84.5	**100.0**
$Genetics_2$	L	17.0	11.0	2.0	59.5	16.8	23.7	62.3	61.4	**100.0**
	S	22.1	19.1	5.2	36.5	37.6	49.9	54.7	58.3	**71.0**
	L	11.2	6.2	1.9	29.2	21.5	28.9	42.5	47.1	**68.0**
	All	16.7	12.7	3.6	32.9	29.6	39.4	48.6	52.7	**69.5**

Table 1: Comparing CLIN with baselines for **generalization across unseen environments**

- They say CLIN compared across unseen environments but isn't G+A like training on the test set?
- There is a drop in performances for a few tasks with GEN-ENV.
- GEN-ENV + ADAPT beats the SOTA by a lot

Memory Structure Importance

Causal Abstractions in CLIN:

- CLIN's memory generation focuses on **structured causal abstractions** around two relations:
  - **"Necessary"**: Actions that are essential for progress.
  - **"Does not contribute"**: Actions that do not help or may hinder task progress.

Ablation Study with Free-Form Memory:

- As a comparison, the memory generator was modified to produce **free-form advice** without enforcing a structured format.
- This free-form memory lacks the clear causal structure found in CLIN's standard memory (i.e., without "necessary" or "does not contribute" labels).

Findings:

- Using **unstructured memory** resulted in a **6-point drop in average reward** across tasks.
- This performance drop occurred in **10% of cases** compared to the structured causal abstractions used by CLIN.

Continued

Superior BASE Performance:

- CLIN achieves **higher BASE performance** than both **ReAct** and **Reflexion**, despite using the same underlying language model (**GPT-4**).

Ablation Study on Controller Module:

- The **controller module** in CLIN is responsible for generating a **goal** before determining the next action.
- When the controller is removed (**Abl-Controller-BASE** setup), CLIN's BASE performance significantly declines.

Performance Impact:

- Without the controller, CLIN's BASE performance drops in **44% of cases**.
- There is an **18-point decrease in average reward** making Abl-Controller-BASE on par with ReAct, the foundational agent for Reflexion.

Limitation 1

Limitation: Lack of Exploration:

- **Dependency on Past Experience**: CLIN's learning is limited to actions and locations it has previously encountered, so unobserved activities or areas remain unknown.
- **Impact on Task Success**: When a task-critical location or action is unknown from past trials, CLIN may struggle to complete tasks successfully.
  - *Example*: In a task to create orange paint, CLIN must find red and yellow paints in the art studio, which is not visible when starting from "outside." Without prior exploration of the art studio, CLIN tries using irrelevant items (like an orange) and fails.
  - *Another Example*: In boiling or freezing tasks, CLIN fails if it hasn't learned to measure the temperature regularly, which is crucial for these tasks.
- **Importance of Exploration**: Prior exploration of task-critical locations or actions is necessary to generate useful memory insights and improve performance in future trials.

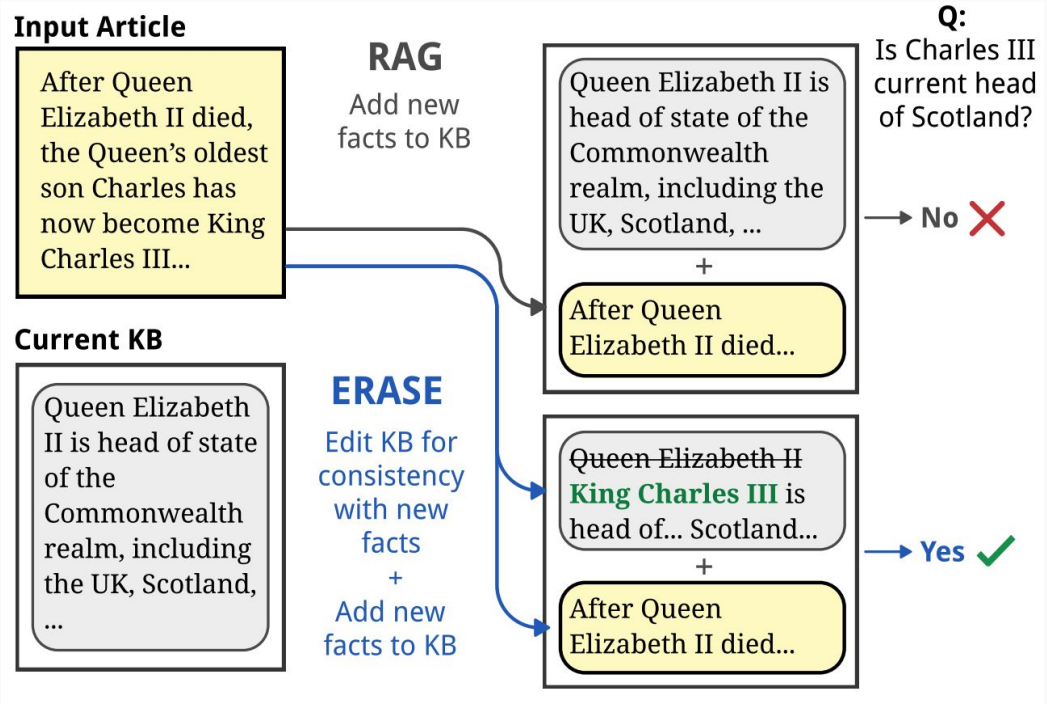
Limitation 2

Limitation: Poor Memory Retrieval:

- **Retrieval Errors in Meta-Memory**: CLIN sometimes retrieves the wrong memory insights for a task, leading to repeated failures.
  - *Example*: In a task to boil gallium, CLIN needs to use an oven or blast furnace but repeatedly retrieves the insight to “activate the stove,” which is ineffective, resulting in task failure.
- **Challenge with Varied Initial Conditions**: When initial conditions vary, it’s harder for CLIN to retrieve the most relevant insight, especially in the initial trial of generalization.
- **Future Improvement Needed**: Improved memory representation and retrieval mechanisms could help CLIN access more relevant insights, which is suggested as an area for future work.

Other Related Works

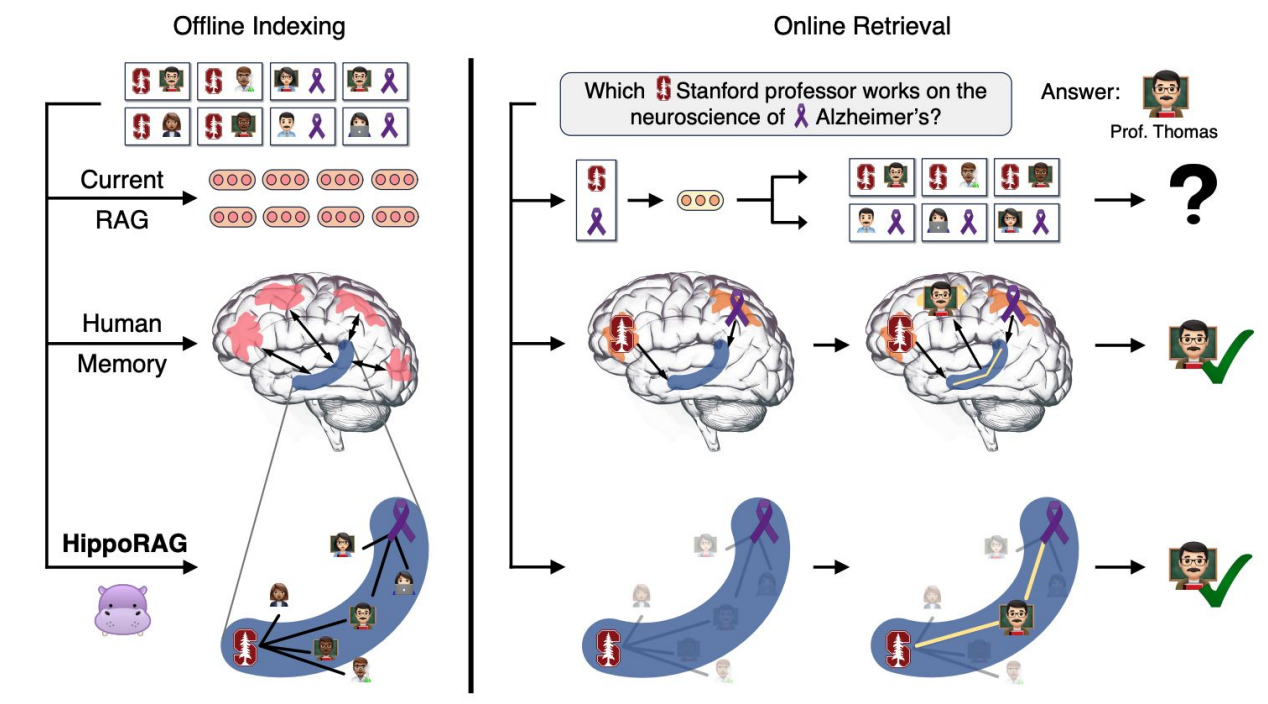
When new documents are acquired, we incrementally deleting or rewriting other entries in the knowledge.



[Language Modeling with Editable External Knowledge: Li et al. 2024]

Other Related Works

We can use hippocampal indexing theory of human long-term memory to enable deeper and more efficient knowledge integration over new experiences!



[HippoRAG: Gutiérrez et al. 2024]

Thank you!