



CS 839 Next Generation Data Systems

Lecture 8: Agentic Memory

Tianyu Li
litianyu@cs.wisc.edu

Recap

- Last lecture: Agentic Retrieval
 - What actions can the agent take?
 - How training/prompting influence the agent's strategy?
 - How are tool prompts organized, results presented, and space usage managed, etc.?

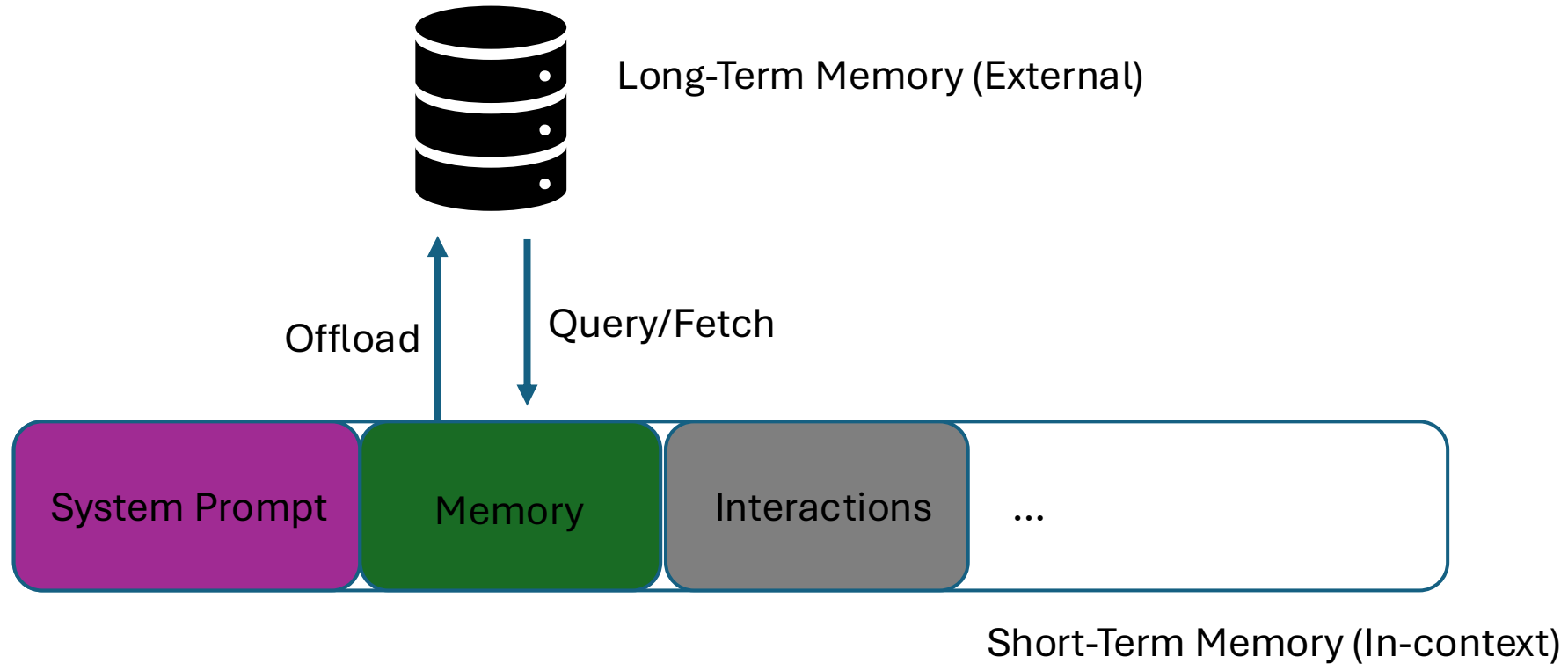
Discussion Redux

- What drawbacks of agentic retrieval have we overlooked and when do they become salient?
- Where would you place your research bet: better tools or better models? What research will be “bitter-lesson proof” in this space?
- Will the future be about smarter, more generalist models, or specialized agents and tool stack?
- What are the key missing pieces from a convincing demo to production search agents?

Agentic Memory: Motivating Scenario

- A chatbot interacts with a human with clear preferences and background information, how can it retain that information for future sessions?
- Coding agents executes similar search steps across a variety of tasks (e.g., search for `AsyncEventHandler`), how can it avoid repeated calls?
- A customer support agent processes the same workload every day and gets feedback on its actions, how can it improve without retraining the model?

Agentic Memory: Idea



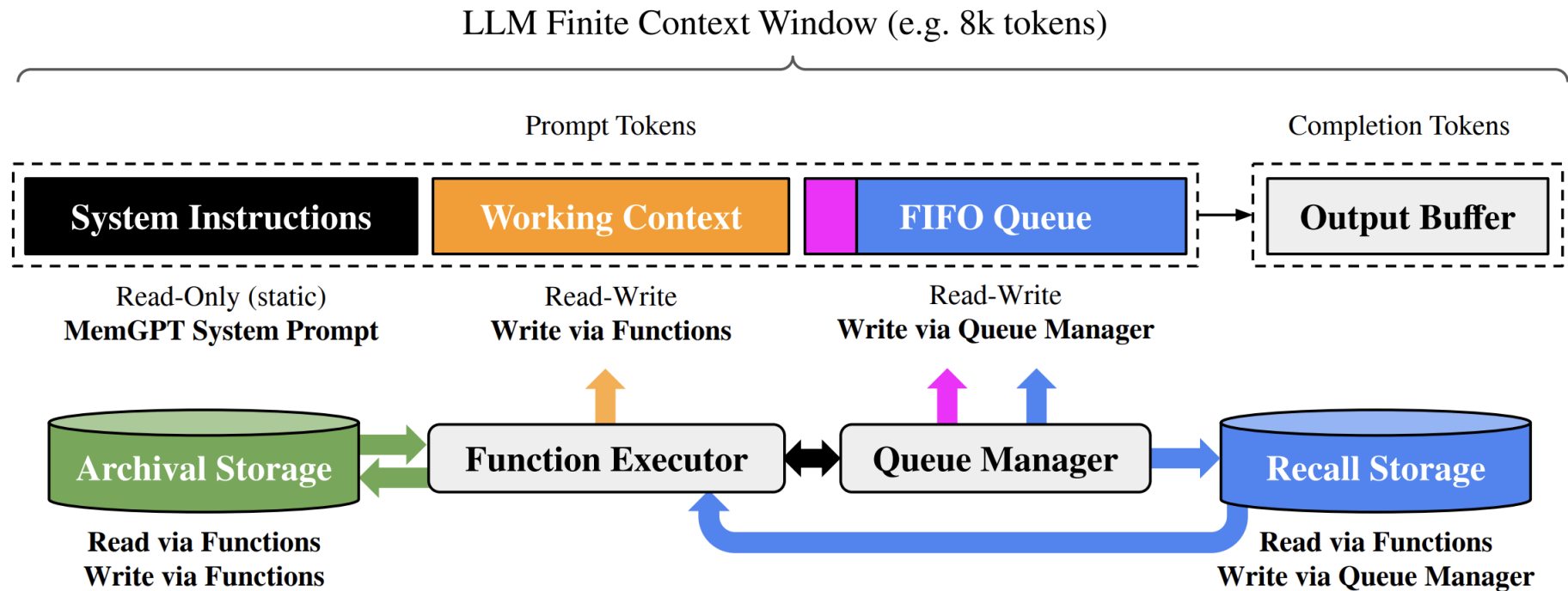
Agentic Memory: Design Space

- Thought 1: memory can be “passive”
 - Offloading strategy (e.g., sliding window + summary) in harness layer
 - Retrieval strategy (e.g., simple RAG) from external store
- Thought 2: memory can also be “agentic”
 - LLM decides what to keep in its context, what/how/when to offload, and what/how/when to retrieve using tools

Case Study: MemGPT

- One of the first projects around agent memory that spawn out of UC Berkeley (now a startup called Letta)
- Key Idea: give model tools to edit its own memory + page-based memory management

Case Study: MemGPT



Case Study: MemGPT

- External stores are organized in named text chunks
- Archival storage searchable with vector search
- Recall storage searchable with regex/date filter

Discussion -- MemGPT

- Why separate Archival/Recall Storage?
- MemGPT was designed back when context windows were a few thousand tokens. Are its designs still relevant with today's 1M+ context window?
- Are there fundamental architectural restrictions to MemGPT?
- Is MemGPT really an OS-like abstraction?

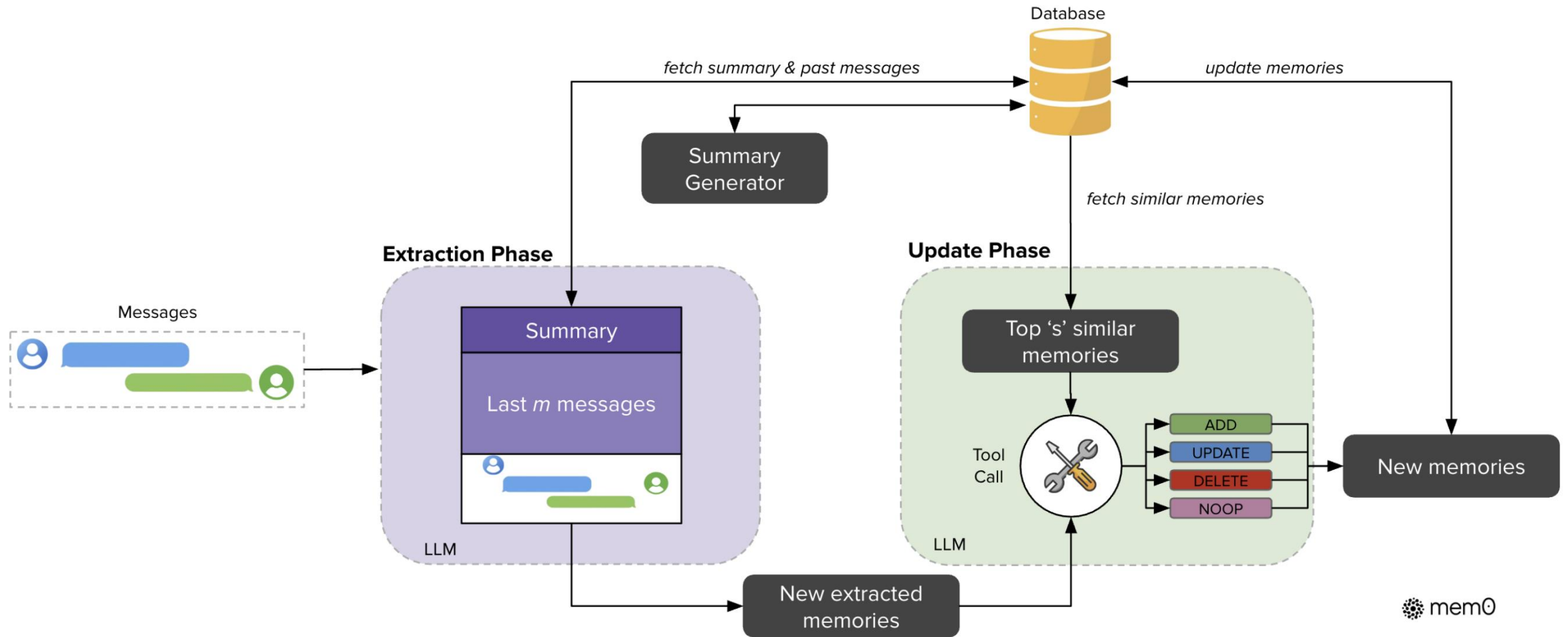
Next: External Memory Maintenance

- In MemGPT, agents can save notes and search its history
- Problem: as execution state accumulates, notes may become duplicated, overlapping, or contradictory
- How do we maintain memory over long execution horizons?

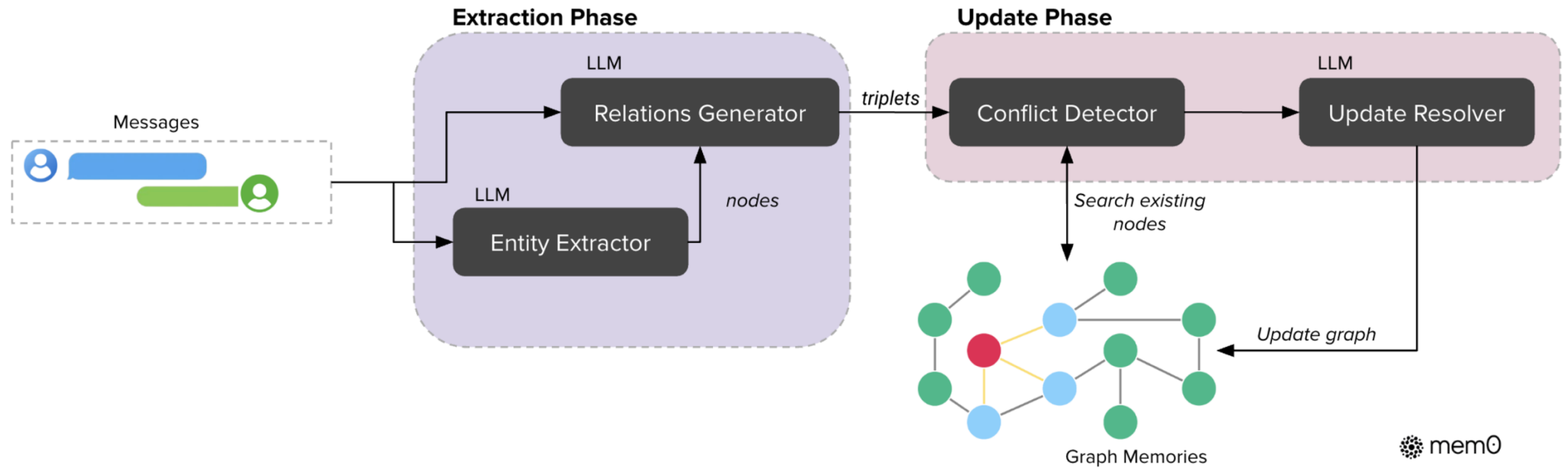
Case Study: Mem0

- Mem0: a separate startup that grew from an open-source RAG framework
- Key Idea: dedicated memory maintenance pipeline that is separate from the main agentic loop
 - Explicit attention to state reconciliation
- Usage of Mem0 is more flexible and often controlled by the harness

Case Study: Mem0



Case Study: Mem0



Discussion – Mem0

- What are the hidden costs of Mem0 compared to MemGPT?
- What are the trade-offs between the normal vector-based and graph-based backend?
- Does the experiments accurately and holistically measure Mem0 performance? Why or why not?

Next: Agent Workflow Memory

- Compared to MemGPT/Mem0, this is a much more targeted technique for acquiring reusable “skills” for agents
- Key Idea: discover common workflow shapes from past trajectories, templatize, and inject into future agent contexts

Case Study: AWM

- Setup: Prompt LLM to emit annotated traces of execution and logs task outcome (success/failure)
- Offline or online

I need to click the “Orders” link to see all orders.

click('126') # id of the button

I need to find order 0130 in the current page.

scroll(0, 200)

... ..

The current page shows order 0130.

send_msg_to_user(“Emma Lopez”)

stop()

Case Study: AWM

- “Training”: gather traces and prompt an LLM to extract templates that are common across successful traces

❖ Workflow Description d

This workflow aims to find an customer order with specified ID.

❖ Workflow Trajectory

[env desc] The current page shows..

[reason] I need to click “Orders” to..

[action] click(‘order-link-id’)

... ..

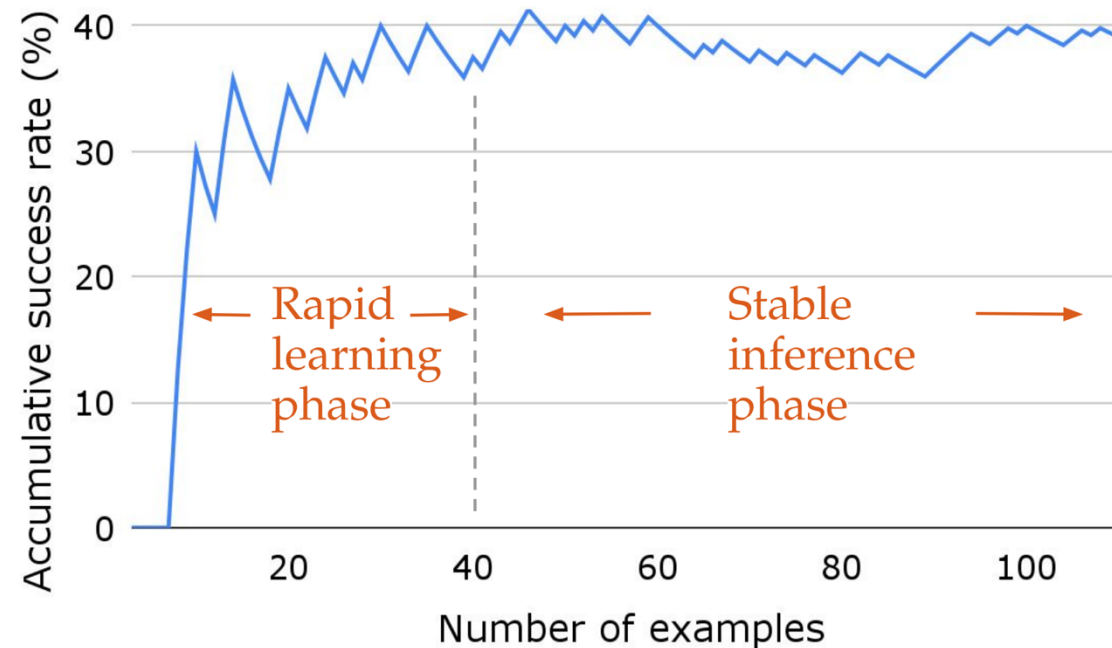
[env desc] Order {id} is shown.

[reason] Order {id} is found, I will now terminate the task.

[action] stop()

Case Study: AWM

- Extracted workflows are directly added to agent context as “memory” and rapidly improves task success rate



Discussion – AWM

- What is the key philosophical difference between AWM and MemGPT/Mem0?
- What are missed opportunities in the AWM work?
- What is the relationship between fine-tuning (a la Chroma Context-1) and AWM?