ARTIFICIAL INTELLIGENCE AND NEURAL NETWORKS

Lecture 9a – AI Agents: Memory, Tools, and Planning

Chizhi Chris ZHANG

zhangchizhi@ciomp.ac.cn
Advanced Computing and Digital Technology Research Center
University of Chinese Academy of Sciences

Spring 2026

Today's Question

What we are trying to answer

What is the difference between a chatbot that answers once and an AI agent that can carry a task through several steps?

Why this matters

For many students, "AI" now means not only asking for a paragraph, but also asking a system to search, organize, compare, call tools, and finish a workflow.

What changes from AI8

AI8 focused on generation. AI9 asks what happens when a model is not only producing content, but also planning actions, using tools, and reacting to the outside world.

From AI8 to AI9

Last time

We studied generative systems that create drafts, images, and other outputs for people to inspect and revise.

Today

We move from generation to execution. The system is no longer only creating content. It is trying to complete a multi-step objective.

One sentence

AI8 was about making content. AI9 is about getting work done.

From Assistant to Agent

Assistant mode

The user asks one question and gets one answer.

Agent mode

The system can plan several steps, call tools, inspect results, revise its approach, and continue until the task reaches a stopping point.

The key difference is not only better language. It is the move from one reply to a controlled loop of action.

Why Agents Feel New

Several things matured at once

- stronger base models
- longer context windows
- better tool APIs
- cheaper inference for repeated calls

What users now expect

People increasingly expect systems to finish workflows, not only to generate text that humans must manually turn into action.

A Student Research Story

Task

"Find three recent papers on battery recycling, summarize the methods, compare them in a table, and draft an email to my supervisor."

What an agent might do

- search for papers
- read titles and abstracts
- extract the main points
- organize a comparison
- draft the final message

That sequence of actions is exactly why the agent idea matters. The task is too long to treat as one isolated answer.

What People Often Get Wrong

A common mistake

"Agent" means full autonomy with no human oversight.

Better statement

Most useful agents are semi-autonomous. They operate inside boundaries, ask for confirmation when needed, and hand control back to people when the risk is high.

Autonomy is not all-or-nothing. Good system design decides where the line should be.

Why Workflow Completion Matters

Why a single answer is often not enough

Real tasks involve missing information, external tools, corrections, and changing constraints.

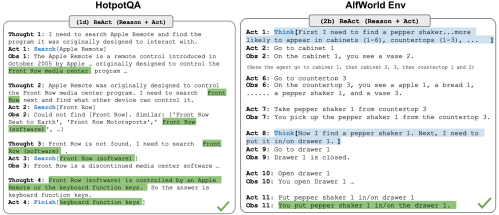
Why agents became attractive

An agent can keep state across steps and adapt when the first attempt is not enough.

The bigger lesson

The world does not usually reward one-shot eloquence. It rewards sustained, correct progress toward a goal.

The Think-Act-Observe Loop



The core idea

Reasoning and action are interleaved. The system thinks, does something, sees what happened, and then decides what to do next.

A simple travel-planning example

The agent checks the timetable, sees the last train is full, and revises the plan instead of repeating the first answer.

Why This Loop Helps

Without the loop

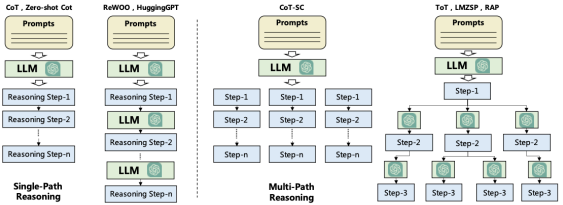
The model may produce a fluent plan that quietly ignores the real external state.

With the loop

The agent can test assumptions against the world, revise the plan, and recover when something unexpected happens.

This is the difference between imagination and controlled interaction.

Task Decomposition



Why decomposition matters

Breaking a large task into smaller steps reduces confusion, clarifies dependencies, and makes failures easier to isolate.

Why users feel the difference

“Plan my move to another city” is too vague as one giant request. Splitting it into housing, transport, budget, and school logistics makes the process easier to trust.

Planner Quality Matters

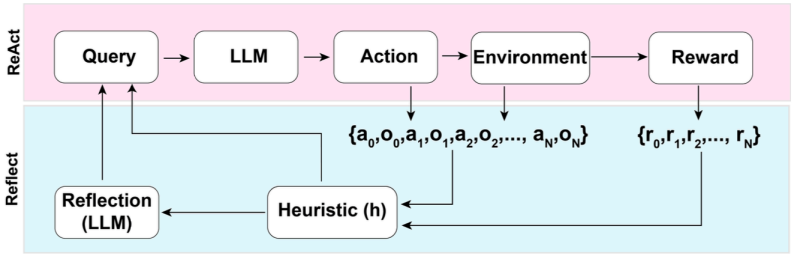
Weak planner symptoms

- repeats steps
- misses important constraints
- calls tools too early or too late
- cannot recover after errors

Better planner traits

Explicit goals, bounded substeps, checkpoints, and clear stopping conditions.

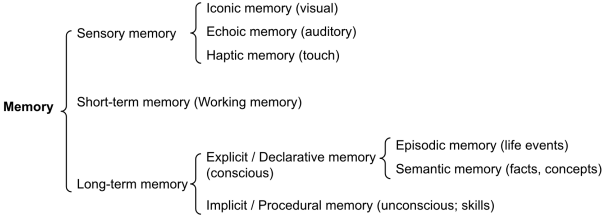
Reflection Can Improve the Plan



Why this idea matters

Agents often fail not because the goal is impossible, but because they keep repeating a weak plan. Reflection creates a chance to notice that the plan itself needs to change.

Why Agents Need Memory



Simple intuition

Without memory, every step feels like a new conversation. The system forgets what it already tried, what the user preferred, and what the tools returned.

A common failure

If an agent forgets that a booking API already returned “sold out”, it may keep repeating the same call instead of adapting.

Short-Term and Long-Term Memory

Short-term memory

Recent conversation turns, recent tool results, and the immediate task state.

Long-term memory

Stored preferences, retrieved notes, summaries of prior work, and reusable background knowledge.

Why both matter

Short-term memory keeps the current task coherent. Long-term memory keeps the agent from starting over every time.

Why Tool Use Changes the Game

Typical tools

- search
- calculators
- databases
- calendars
- code execution

Key point

Tool use lets the model stop pretending it already knows everything. It can query the outside world and work with current information.

Retrieval Becomes More Important for Agents

Role in agents

Retrieval is not only for answering questions. It also helps an agent recover instructions, documents, prior notes, and domain-specific constraints while working through a task.

A real agent habit

Before acting, a useful agent often needs to look up the current rule book, the user's files, and what happened earlier in the workflow.

Frameworks Turn Ideas into Systems

Inside the wrapper

- planning
- memory
- tool routing
- logging and retry

Why this matters

An agent is rarely just one prompt. In practice, the surrounding system decides whether the model is usable, traceable, and safe.

A Multi-Tool Story

Algorithm 1 API call process

```
Input: us ← UserStatement
if API Call is needed then
    while API not found do
        keywords ← summarize(us)
        api ← search(keywords)
        if Give Up then
            break
        end if
    end while
    if API found then
        api_doc ← api.documentation
        while Response not satisfied do
            api_call ← gen_api_call(api_doc, us)
            api_re ← execute_api_call(api_call)
            if Give Up then
                break
            end if
        end while
    end if
end if
if response then
    re ← generate_response(api_re)
else
    re ← generate_response()
end if
Output: ResponseToUser
```


How to read this page

Many useful agents are really coordinators. They move between several tools and data sources, then assemble the results into one coherent workflow.

Education Scenario

Student request

“Help me build a study plan for the final exam, based on what I already understand and what I keep getting wrong.”

Agent workflow

- inspect past errors
- identify weak topics
- schedule review sessions
- generate practice material

This is more than one reply. It is a small educational workflow.

Healthcare Support Scenario

Appropriate role

An agent can help gather guidelines, organize symptoms, structure records, and flag missing information for a clinician.

Why this can help

It reduces clerical load and helps surface useful context.

Boundary

Diagnosis, final judgment, and high-risk decisions should stay under human professional control.

Public Service Scenario

Citizen service

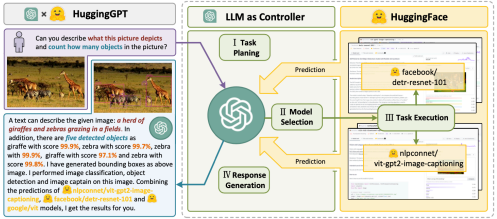
An agent can help people navigate forms, deadlines, eligibility rules, and required documents.

Why that is useful

It lowers friction in systems that are often hard to understand, especially for first-time users.

Good public-facing agents are less about sounding impressive and more about reducing confusion.

Human-Agent Teams



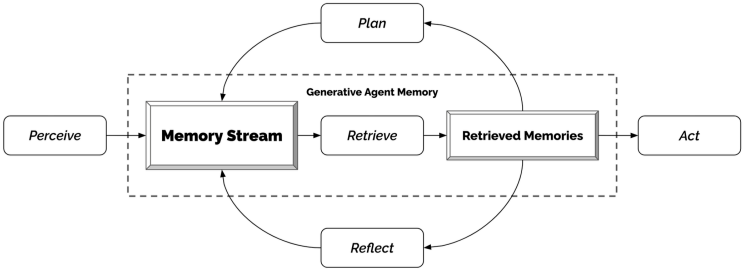
The lesson

Power grows when a system can coordinate several specialized tools or models, but the need for supervision also grows because errors can spread across the whole chain.

Why this is not science fiction anymore

A researcher may ask one system to search papers, another to summarize, and a third to draw figures, while a human still catches the bad citation.

A Social Simulation Example



Why this example is interesting

It shows that once agents have memory and goals, people naturally start asking about social behavior, coordination, and long-term interaction, not only about single-task completion.

Where Agents Fail

Planning failure

The steps are badly chosen.

Tool failure

The wrong tool is used, or the right tool is used badly.

Memory failure

The system forgets key facts or carries the wrong state forward.

Hallucination Inside Action Loops

Why this is worse than ordinary hallucination

If a model invents a fact in a plain answer, the error may stay on the page. If an agent acts on that invented fact, the error can trigger tool calls, notifications, transactions, or other downstream effects.

What this means

Action amplifies mistakes. That is why grounded tool use and confirmation gates matter.

Permission Design

Low-risk mode

Read, summarize, draft, search, and prepare options for a human.

High-risk actions

Sending money, deleting records, filing official forms, contacting third parties, or changing production systems.

Design rule

The more irreversible the action, the stronger the confirmation and logging should be.

A Minimal Governance Checklist

Traceability

Can we see what the agent did, what tools it called, and why it reached its final answer?

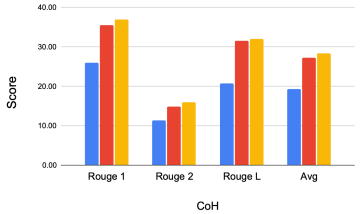
Boundaries

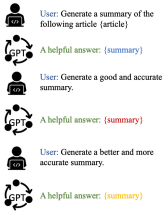
Are the allowed actions, data sources, and stop conditions clearly defined?

Fallback

When the system is uncertain or the risk is high, does it slow down and ask a human instead of improvising?

Why Benchmark Scores Are Not Enough





The lesson

An agent can look strong on isolated benchmarks and still fail in a messy environment where tools are unreliable, instructions are incomplete, and real users behave unpredictably.

What breaks in practice

A benchmark website is stable. A real office workflow has missing files, vague requests, changing permissions, and interruptions.

What Human Skills Become More Important

Human strengths that matter more

- framing the real goal
- deciding what risk is acceptable
- judging when evidence is weak
- choosing when to trust and when to intervene

Agent-era habit

Do not compete with the machine at repeating tiny steps. Compete at setting direction, checking reality, and taking responsibility for outcomes.

Why AI9 Leads to NN9

What AI9 focused on

Planning, memory, tool use, workflow completion, human oversight, and governance.

What NN9 adds

Many of the next important agents do not only read text. They also read screens, photos, charts, forms, and interfaces. That means we need multimodal models.

The agent story naturally expands into the question of how an agent can see as well as speak.

Summary

- An agent is different from a simple assistant because it plans, acts, observes, and continues across several steps.
- Memory, tools, retrieval, and decomposition are what make multi-step task completion possible.
- Reflection and retry help because many failures come from weak plans, not impossible goals.
- Real agent value appears in workflows such as study support, service navigation, and tool coordination.
- As agents gain the power to act, permission design, traceability, and human oversight become more important.

Next Time

Where the course is going

We have now described agents at the application level.

Next lecture

We will study vision-language models, multimodal reasoning, and why agents become much more capable once they can see documents, interfaces, diagrams, and images as well as read text.

Thank You