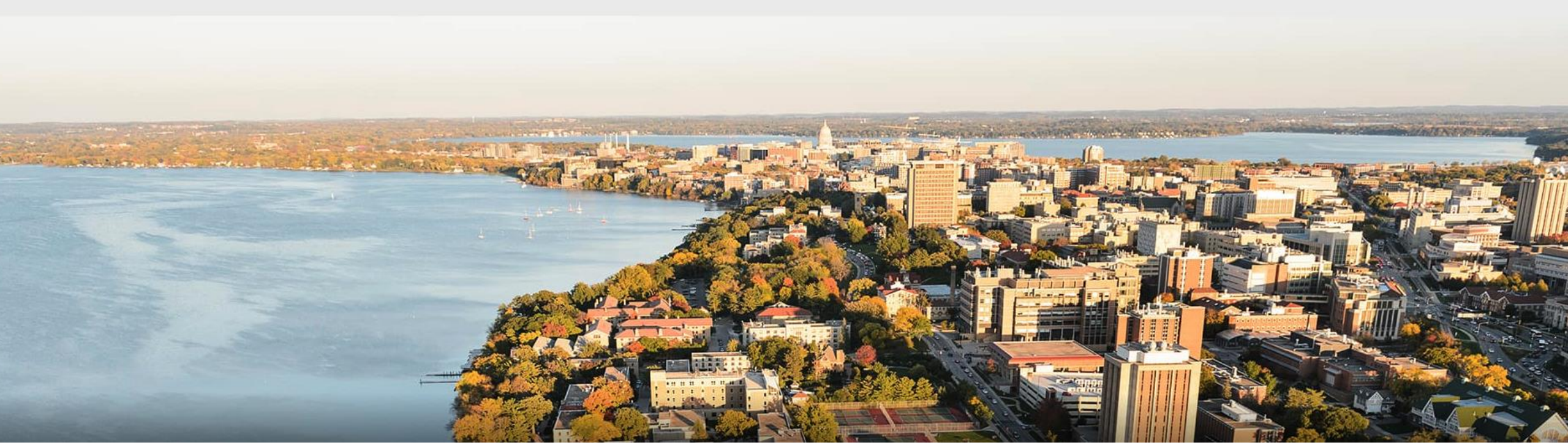


CS 839: Foundation Models
Agents

Fred Sala

University of Wisconsin-Madison

Oct. 21, 2025

Announcements

- **Logistics:** Homework 2 Due Today
- **Presentations:** Most have signed up---thank you!
- **Project information:** Coming out this week
- Class roadmap:

Tuesday Oct. 21	Agents
Thursday Oct. 23	More reasoning
Tuesday Oct. 28	Multimodal models
Thursday Oct. 30	Scaling Las

Outline

- **Introduction to LLM-Powered Agents**
  - Motivation, Goals, Differences vs classical agents / standard LMs, overall architecture and key components
- **Multiagent Systems**
  - Motivation, simulations, architectures and design, communication
- **Evaluation and Challenges**
  - New benchmarks, realism, open problems

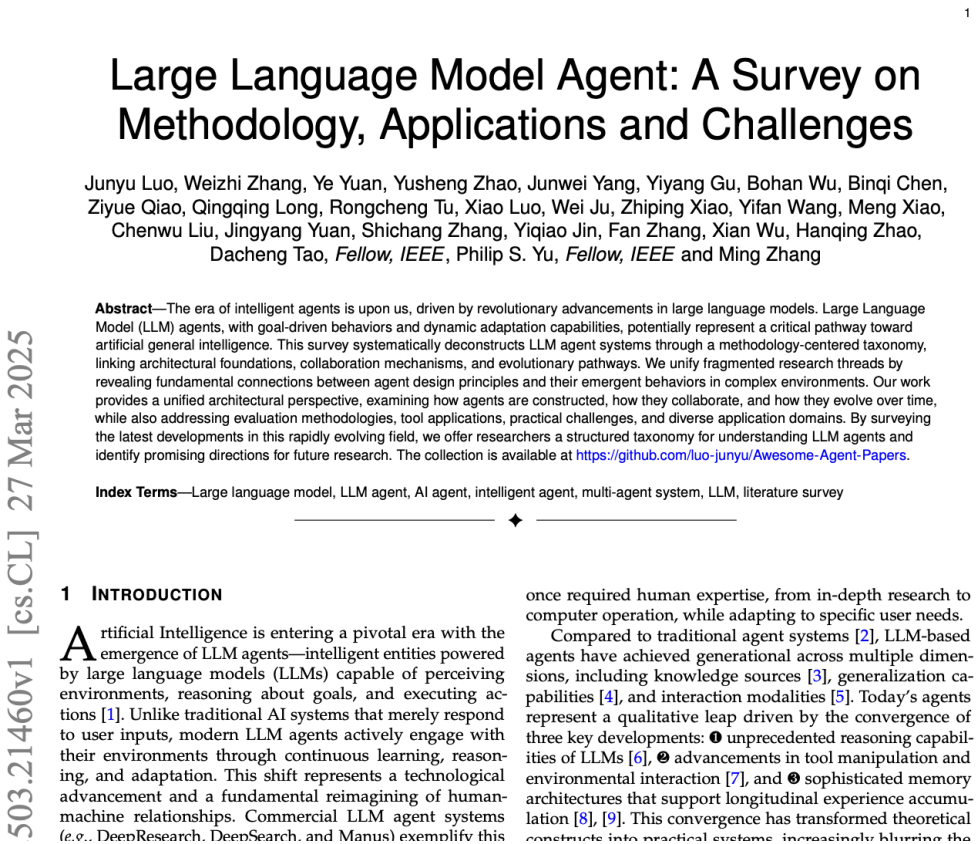
Outline

- **Introduction to LLM-Powered Agents**
  - Motivation, Goals, Differences vs classical agents / standard LMs, overall architecture and key components
- **Multiagent Systems**
  - Motivation, simulations, architectures and design, communication
- **Evaluation and Challenges**
  - New benchmarks, realism, open problems

Motivation: **From LLMs To Agents**

Standard LLMs are static

- But we want **interactive** systems
  - Want systems that can reason, use tools, and act autonomously
  - Want FMs that can be actors, not just predictors
- Useful survey: Wang et al ‘25 “Large Language Model Agent: A Survey on Methodology"



Large Language Model Agent: A Survey on Methodology, Applications and Challenges

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, Yiqiao Jin, Fan Zhang, Xian Wu, Hanqing Zhao, Dacheng Tao, *Fellow, IEEE*, Philip S. Yu, *Fellow, IEEE* and Ming Zhang

Abstract—The era of intelligent agents is upon us, driven by revolutionary advancements in large language models. Large Language Model (LLM) agents, with goal-driven behaviors and dynamic adaptation capabilities, potentially represent a critical pathway toward artificial general intelligence. This survey systematically deconstructs LLM agent systems through a methodology-centered taxonomy, linking architectural foundations, collaboration mechanisms, and evolutionary pathways. We unify fragmented research threads by revealing fundamental connections between agent design principles and their emergent behaviors in complex environments. Our work provides a unified architectural perspective, examining how agents are constructed, how they collaborate, and how they evolve over time, while also addressing evaluation methodologies, tool applications, practical challenges, and diverse application domains. By surveying the latest developments in this rapidly evolving field, we offer researchers a structured taxonomy for understanding LLM agents and identify promising directions for future research. The collection is available at https://github.com/luo-junyu/Awesome-Agent-Papers.

Index Terms—Large language model, LLM agent, AI agent, intelligent agent, multi-agent system, LLM, literature survey

1 Introduction

Artificial Intelligence is entering a pivotal era with the emergence of LLM agents—intelligent entities powered by large language models (LLMs) capable of perceiving environments, reasoning about goals, and executing actions [1]. Unlike traditional AI systems that merely respond to user inputs, modern LLM agents actively engage with their environments through continuous learning, reasoning, and adaptation. This shift represents a technological advancement and a fundamental reimagining of human-machine relationships. Commercial LLM agent systems (*e.g.*, DeepResearch, DeepSearch, and Manus) exemplify this once required human expertise, from in-depth research to computer operation, while adapting to specific user needs.

Compared to traditional agent systems [2], LLM-based agents have achieved generational across multiple dimensions, including knowledge sources [3], generalization capabilities [4], and interaction modalities [5]. Today’s agents represent a qualitative leap driven by the convergence of three key developments: ❶ unprecedented reasoning capabilities of LLMs [6], ❷ advancements in tool manipulation and environmental interaction [7], and ❸ sophisticated memory architectures that support longitudinal experience accumulation [8], [9]. This convergence has transformed theoretical constructs into practical systems, increasingly blurring the

Pre-LLMs: **Classical Agent Models**

Much of this work predates FMs/LLMs,

- A common framework: **PEAS** (Performance, Environment, Actuators, Sensors)
- Agent types:
  - Reactive, deliberative, hybrid
- Reinforcement learning is a standard paradigm for agents
  - Convenient since we already use it for LLMs

Agents vs. Models

Key differences:

- **Models**: static mappings (i.e., input-output pair)
- **Agents**: lots of additional possibilities:
  - Act, observe, remember, adapt
- The advantage of LLMs to power agents is their
  - Flexible interfaces,
  - Reasoning capabilities

Architecture of LLM-Powered Agents

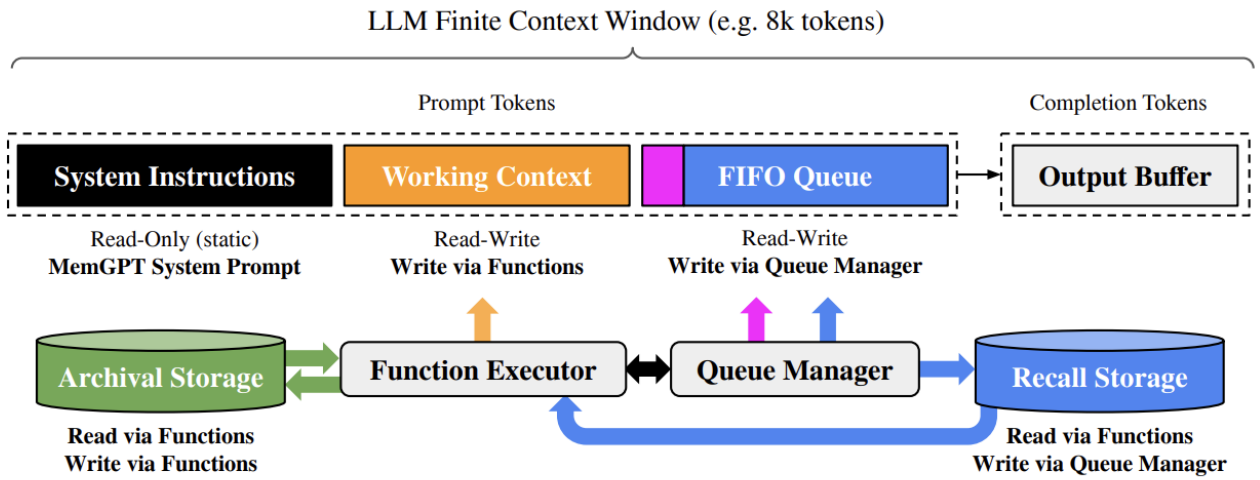
High-level components: **input**, **memory**, **planner**, **tools**,

- **Input**: whatever we have access to
- **Memory**: as simple as context window, but
- **Planning**: reasoning-based decomposition etc
- **Tools**: similar to our earlier discussion on integrating tool use

Memories

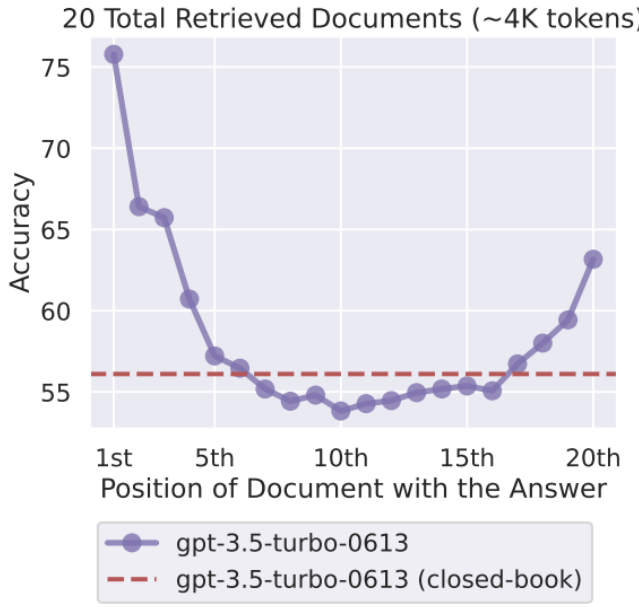
Interactive agents require the use of **memory**

- I.e., we need to store state/other useful information
- Simplest approach: **use LLM context window** as the memory
  - Increasingly tractable with very long-context windows
- Previously, had to create memory systems
  - E.g., MemGPT (Packer et al '24)



Interlude: Long Context Windows

- Simplest approach: **use LLM context window** as the memory
  - Increasingly tractable with very long-context windows
- Note that the advertised long context may not be realistic!
- Liu et al ‘23 (“Lost in the Middle: How Language Models Use Long Contexts”)
- Needle-in a haystack benchmarks
  - Numerous variants



Reasoning and Planning

Classic AI problem (decompose task, plan)

- Addressable via reasoning capabilities of LLMs,
  - Including older methods: chain-of-thought, tree-of-thoughts, self-reflection
  - Example: ReAct (Reason + Act) (Yao et al ‘23)

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software) ', …]

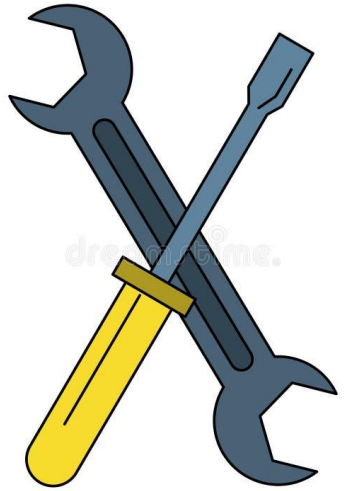
Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys]

Tool Use

Standard component in agentic settings

- Where do we get tools?
  - APIs, web search, etc
  - Or, in some cases use of physical tools
- Most agent frameworks permit the use of web/code-based tools
- Major challenge: tool usage is **noisy**



Tool Noise/Uncertainty

Lots of sources of error,

- Call the wrong tool, interpret output incorrectly, fail to chain together tools in the right way
- Ex: Agentic Insurance Underwriting Benchmark

  "**Tool use errors:** Across models, including top performers, agents made at least one tool call error in 36% of the conversations despite access to the metadata required to use tools properly."

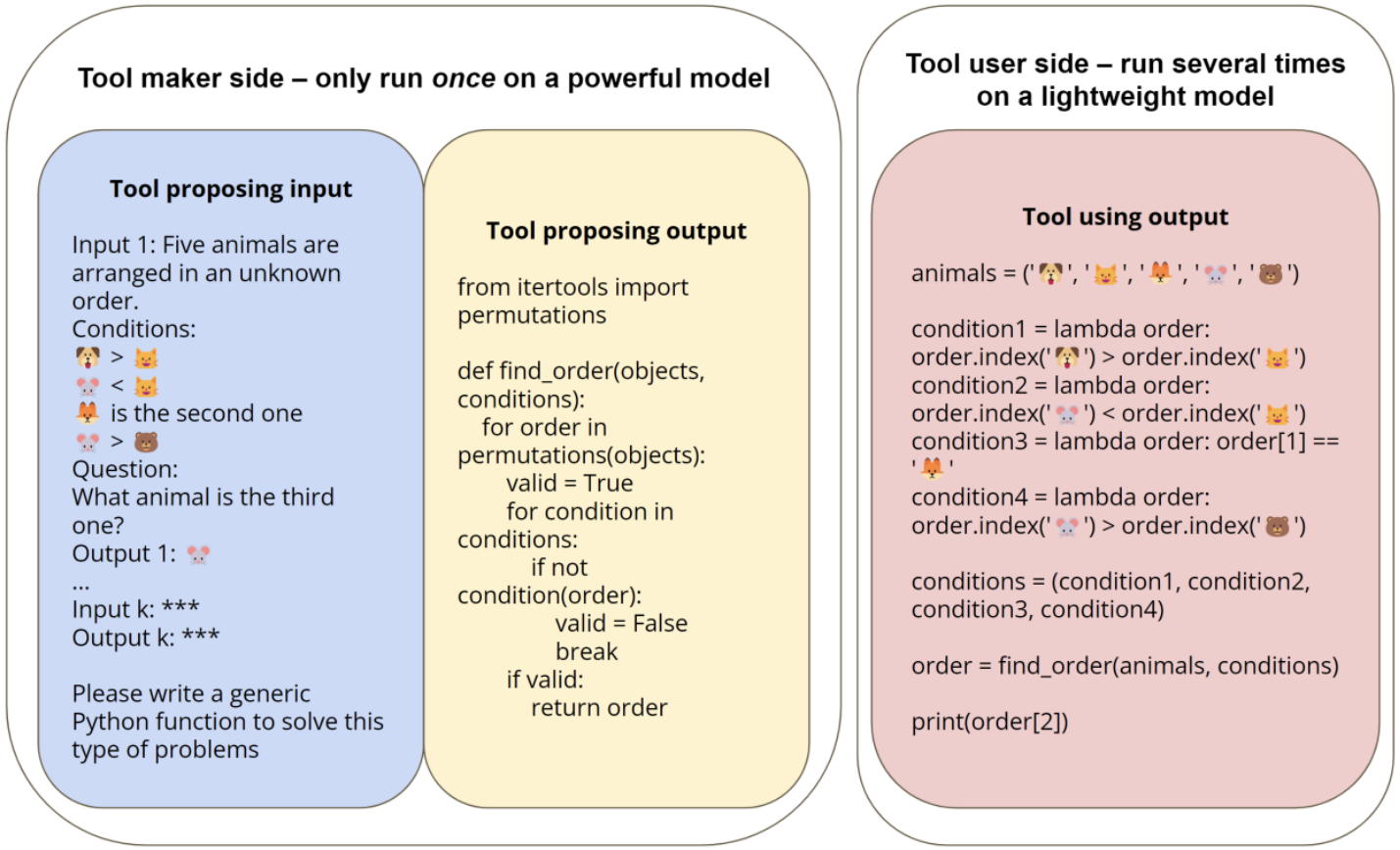
```
================================================ tool_calls =================================================
------------------------------------------------- tool --------------------------------------------------
read_query

------------------------------------------------- input -------------------------------------------------
{"query": "SELECT DISTINCT NAICS_Code, Description FROM naics WHERE Description LIKE '%flour milling%' OR Description
LIKE '%wheat flour%' OR Description LIKE '%corn flour%'"}

================================================ read_query =================================================
Error: ToolException('Error executing tool read_query: SQLite error: no such column: NAICS_Code')  Please fix your
mistakes.
```

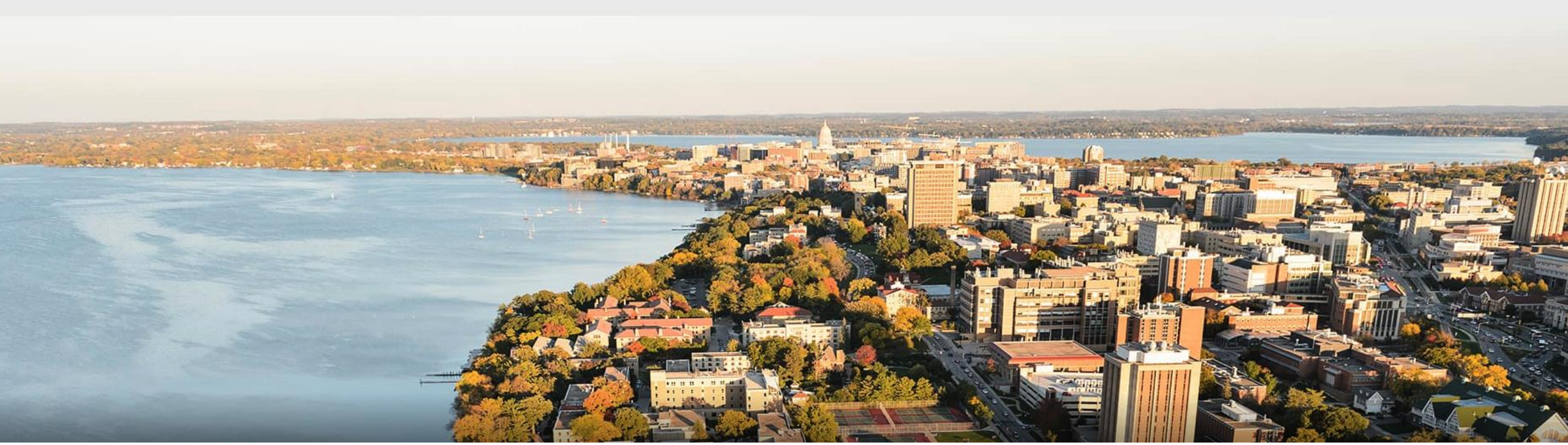
https://huggingface.co/datasets/snorkelai/Multi-Turn-Insurance-Underwriting

Creating New Tools

Note that LLMs can both use & **create** tools

- For example, LATM (Cai '24)
- Powerful model tool creator, less powerful is user





Break & Questions

Outline

- **Introduction to LLM-Powered Agents**
  - Motivation, Goals, Differences vs classical agents / standard LMs, overall architecture and key components
- **Multiagent Systems**
  - Motivation, simulations, architectures and design, communication
- **Evaluation and Challenges**
  - New benchmarks, realism, open problems

Multi-Agent Systems

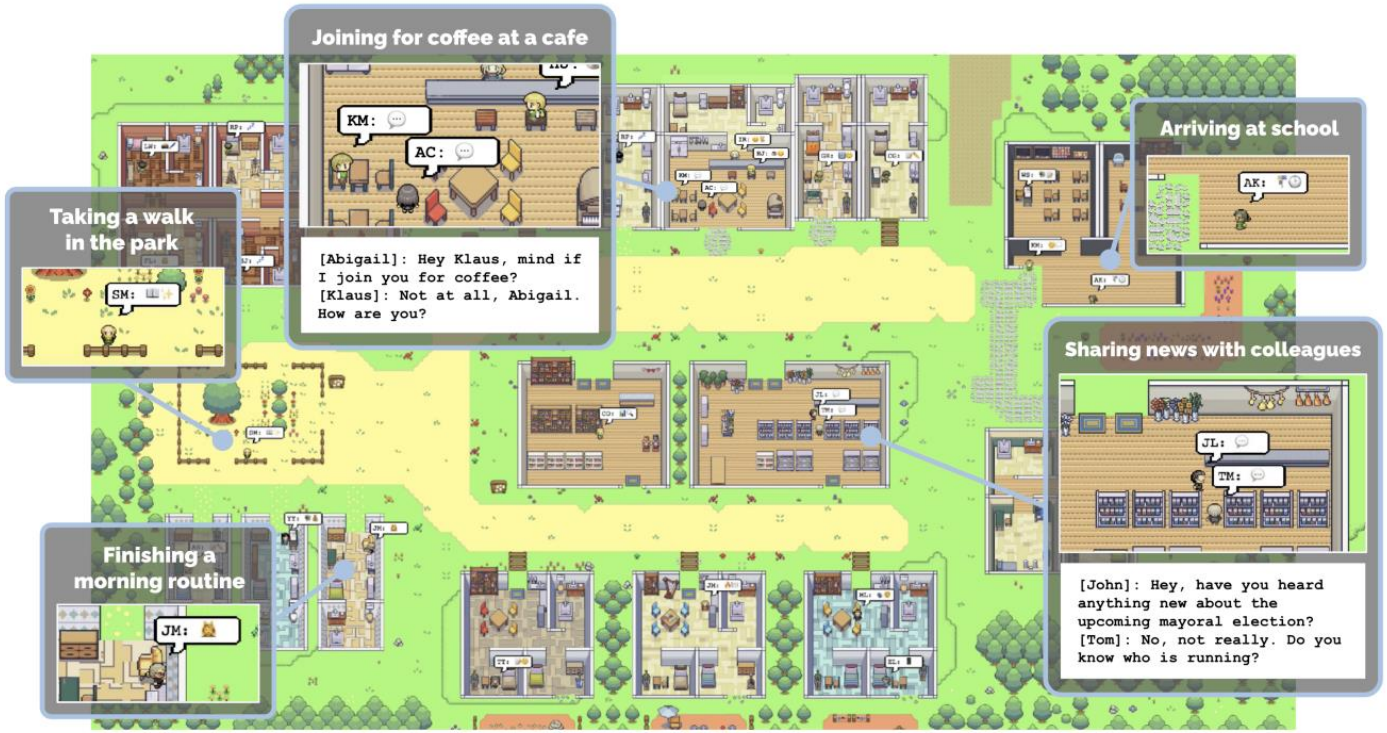
One advantage of agentic approach: can create multiple agents that cooperate to perform tasks. Why?

- **Scale** up workers, permit **specialization**
- Of course, inspired by earlier work,
  - Pre-LLM agent networks called “swarms”,
  - And distributed computing more broadly
- Also useful for simulations
  - Park et al ‘23

Interlude: **Societal Simulations**

Popular new area,

- Difficult to perform large-scale studies of human behavior,
- Try to set up a simulation of how humans interact via agents

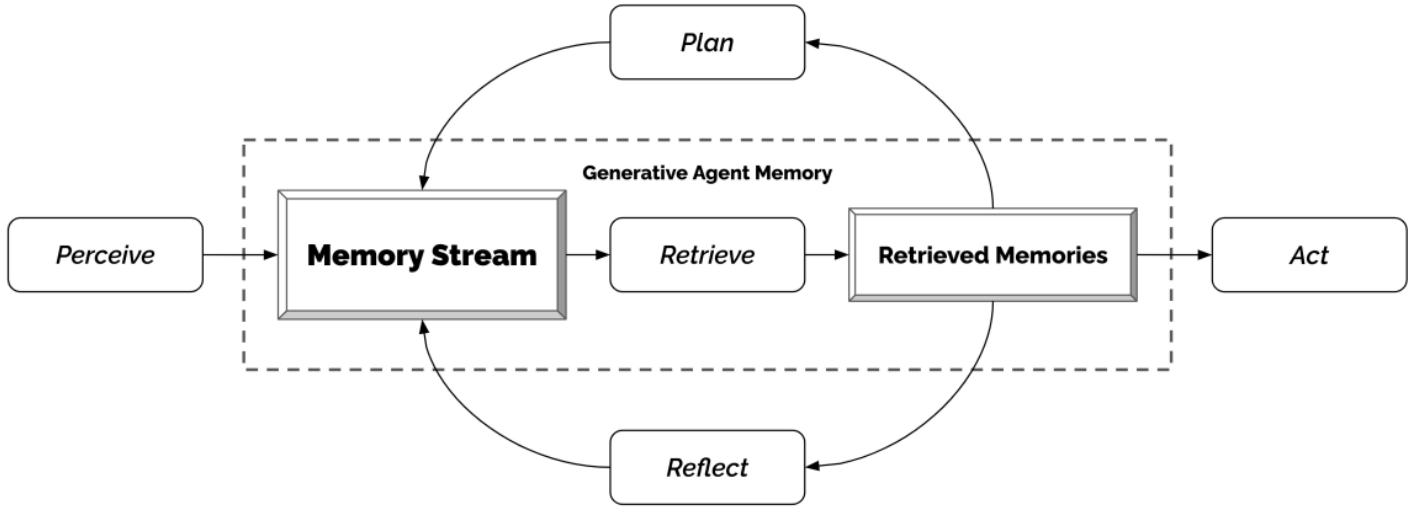


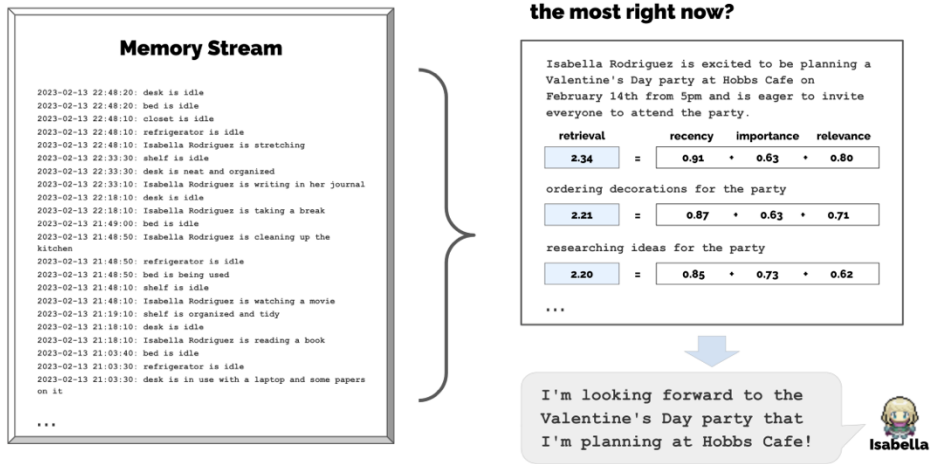
Park et al ‘23

Interlude: **Societal Simulations**

Popular new area,

- Try to set up a simulation of how humans interact via agents
- Itself a large multi-agent system: must tackle each of the design questions

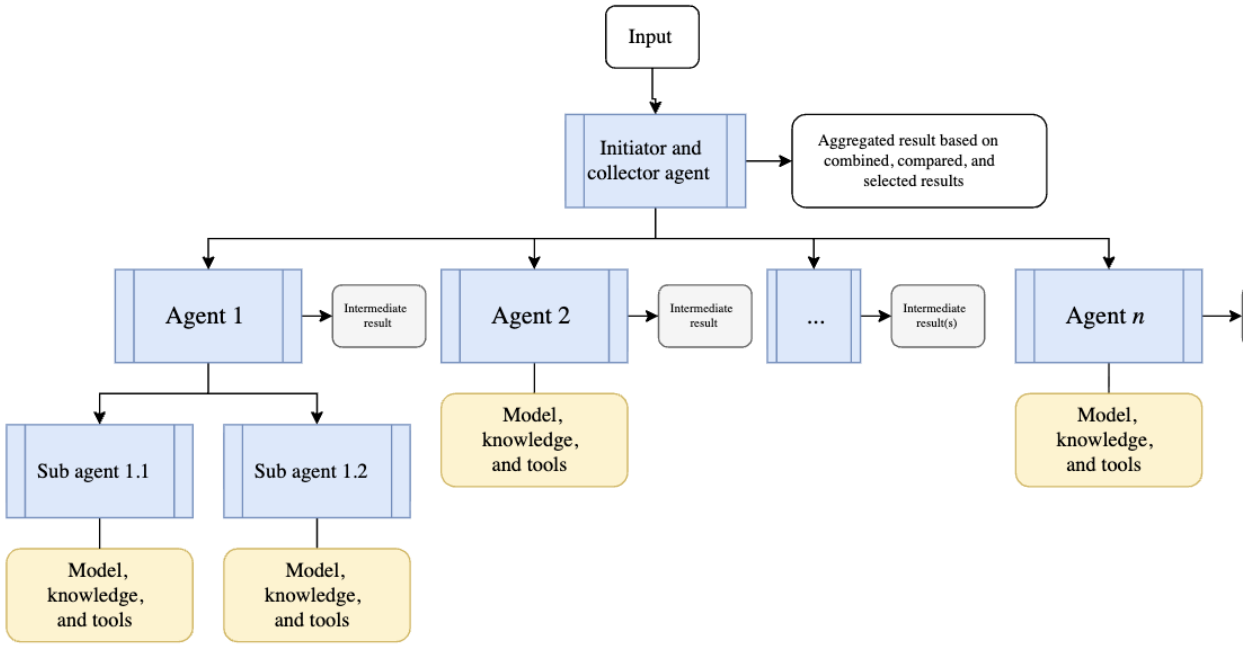




Multi-Agent Systems: Architectures

Lots of design choices to make!

- For example, centralized vs. decentralized
- Centralized: typically an "**orchestrator**" present that can tell other agents what to do.

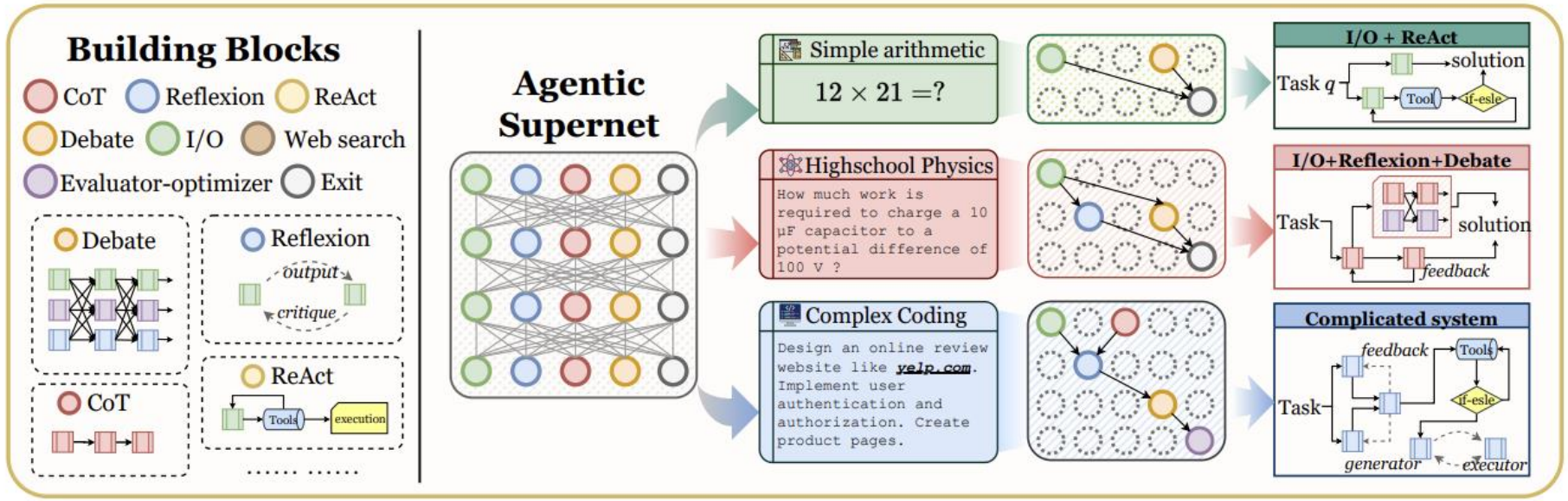


Microsoft

Architecture Search

Lots of design choices to make!

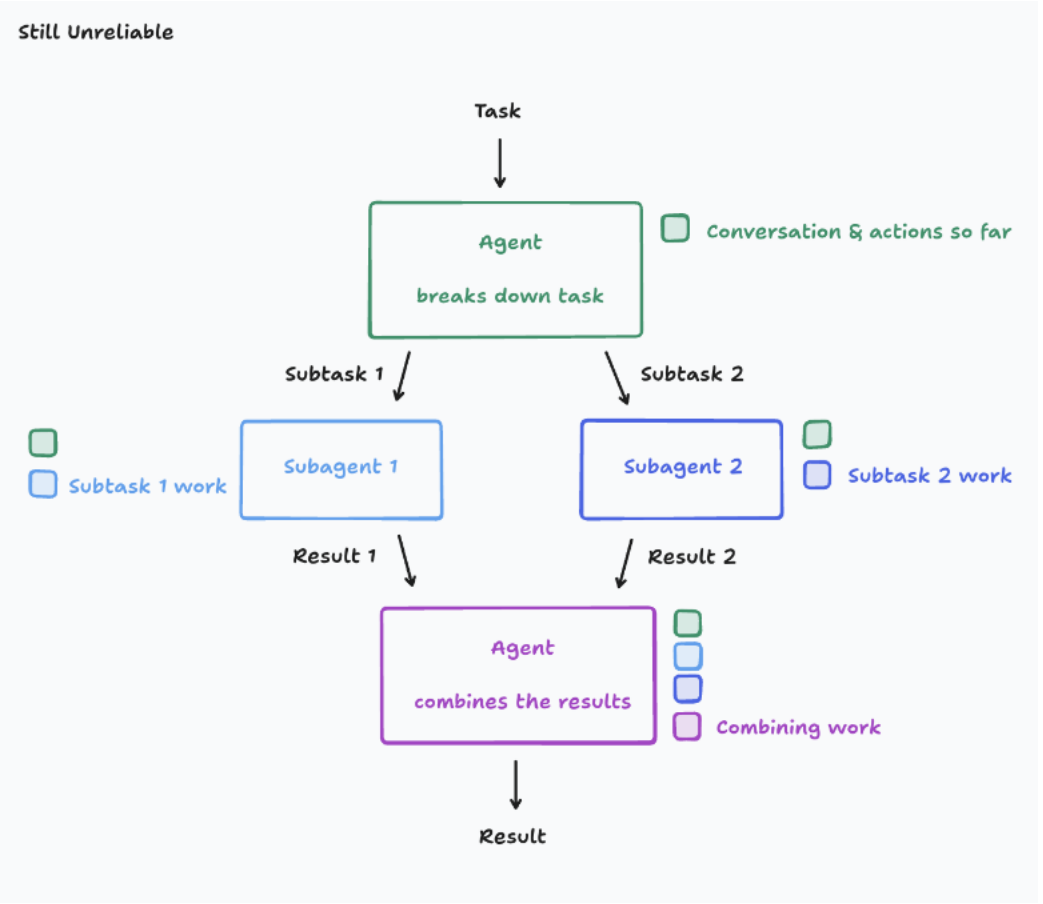
- Can borrow principles from **neural architecture search** to construct an overall agent architecture
- Example: Zhang et al ‘25, “Multi-agent Architecture Search via Agentic Supernet”

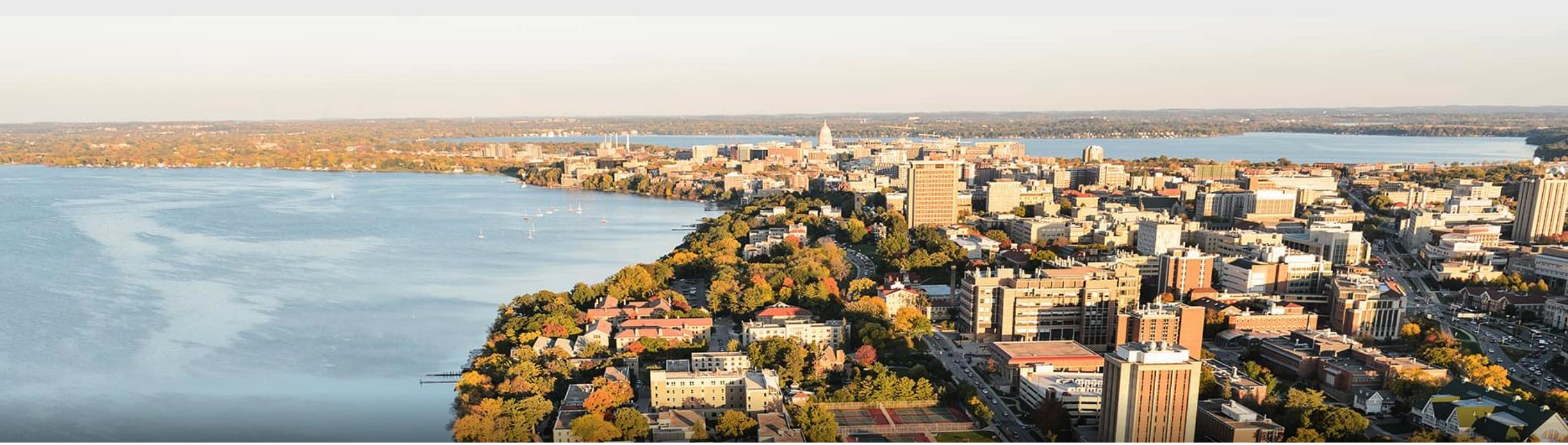


Multi-Agent Systems: Communication

Critical: each agent has access to information that is consistent with others,

- But communicating everything between every pair of users is expensive.
- Nice blog on the challenges: “**Don’t Build Multi-Agents**” (Yan ’25 / Cognition)





Break & Questions

Outline

- **Introduction to LLM-Powered Agents**
  - Motivation, Goals, Differences vs classical agents / standard LMs, overall architecture and key components
- **Multiagent Systems**
  - Motivation, simulations, architectures and design, communication
- **Evaluation and Challenges**
  - New benchmarks, realism, open problems

Evaluation and Benchmarks

As we discussed, lots of complexity to evaluating LLMs/FMs in general, and agentic systems in particular

- Typically cannot just compute accuracy on predictions,
- More broadly, static metrics are not informative enough. Instead, measure several:
  - Task completion,
  - Efficiency,
  - Safety,
  - Cost

Leaderboard

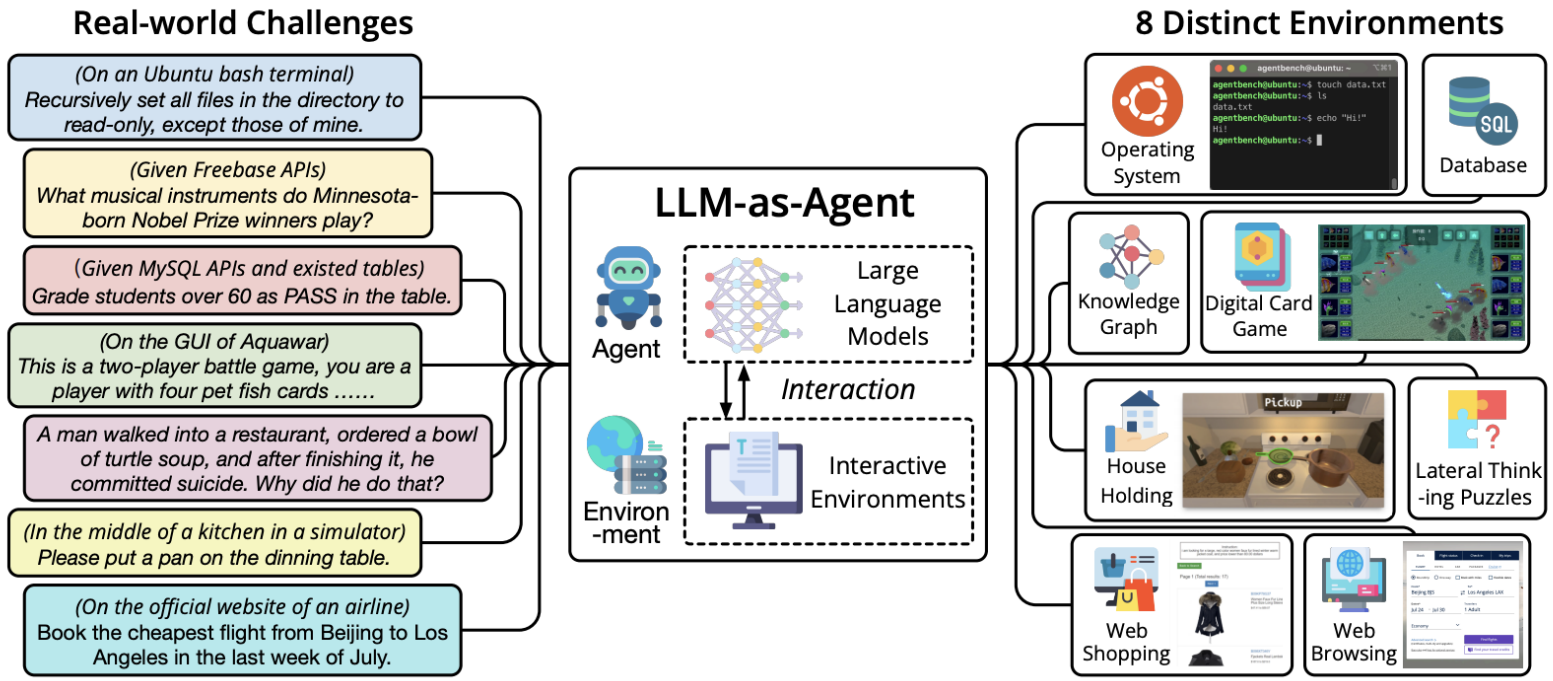
Model	% Resolved	% Score	Avg Steps	Avg Costs ($)	Org	Date	Environment Model	Trajs	Screenshot
MUSE + Gemini 2.5 Flash	41.1%	51.8%	N/A	N/A		2025-10-13	GPT 4o	✓	✗
OpenHands-Versa + Claude Sonnet 4	33.1%	43.2%	46.45	1.63	CMU	2025-06-14	Claude 3.5 Sonnet	✓	✓
OpenHands-Versa + Claude 3.7 Sonnet	30.9%	40.2%	52.73	3.70	CMU	2025-06-14	Claude 3.5 Sonnet	✓	✓

https://the-agent-company.com/#/leaderboard

Evaluation and Benchmarks

Lots of popular benchmarks from 2023 onwards

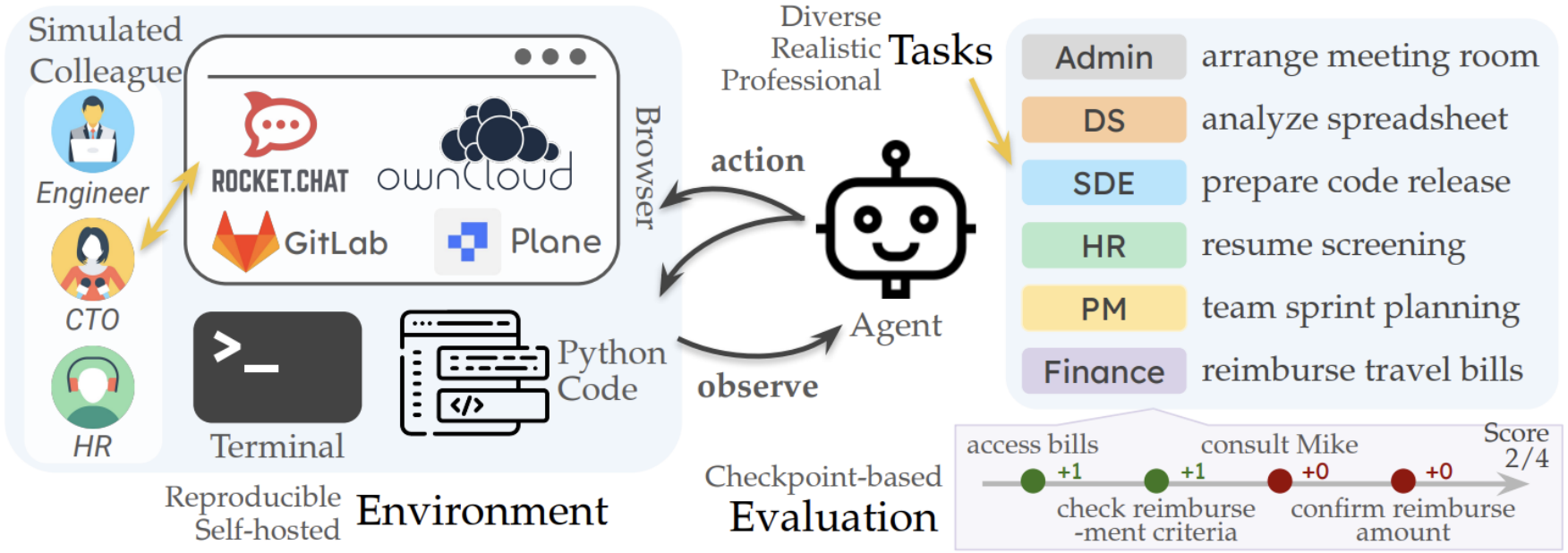
- Already talked about: WebArena, SWE-Bench
- AgentBench (Liu et al ‘24):



Benchmark Realism

Increasingly important: how meaningful are the agentic tasks?

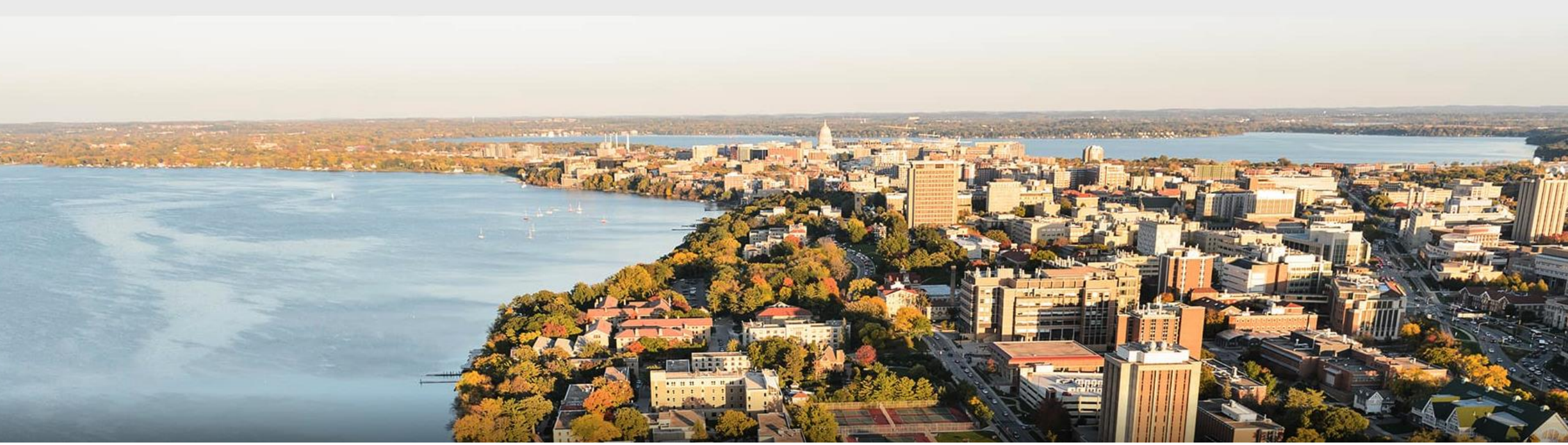
- Good performance on trivial/unrealistic tasks is not useful
- Example: “The Agent Company” (Xu et al ‘24)



Major Challenges

What's left to study? Some examples:

- How do we scale up memories?
- Personalization for agents
- Reliable tool use across environments
- Self-evolving/self-improving systems



Thank You!