From AI Agents to Agentic Systems: Foundations and Edge Perspectives

Saul Urso

ABOUT MYSELF

About me

- **Office:** Room 301
- **E-mail:** saul.urso@phd.unipi.it

Main research Interests

Collaborative AI in edge systems

- Decentralized learning
- Agentic AI

SEMINAR OVERVIEW

01

Language Model based AI Agents

02

Agentic AI systems

03

Agentic AI in edge systems

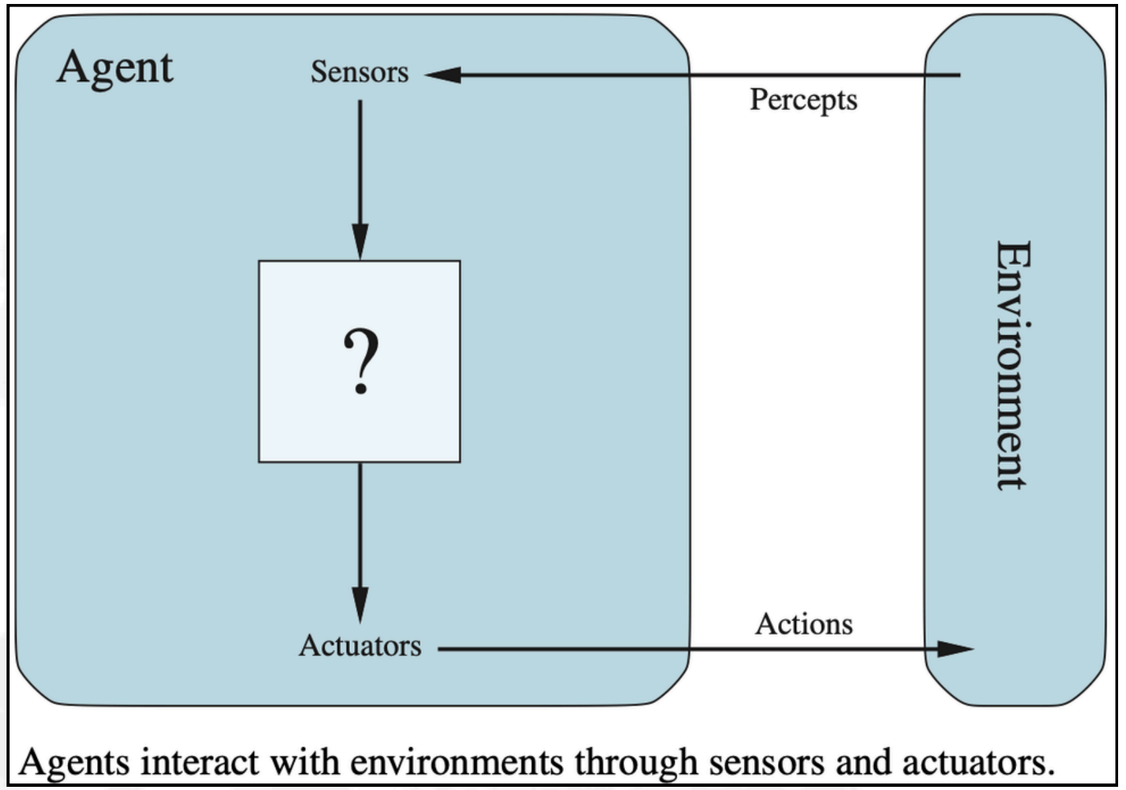
AI AGENTS

WHAT IS AN AGENT?

Definition: An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.

Intuitively:

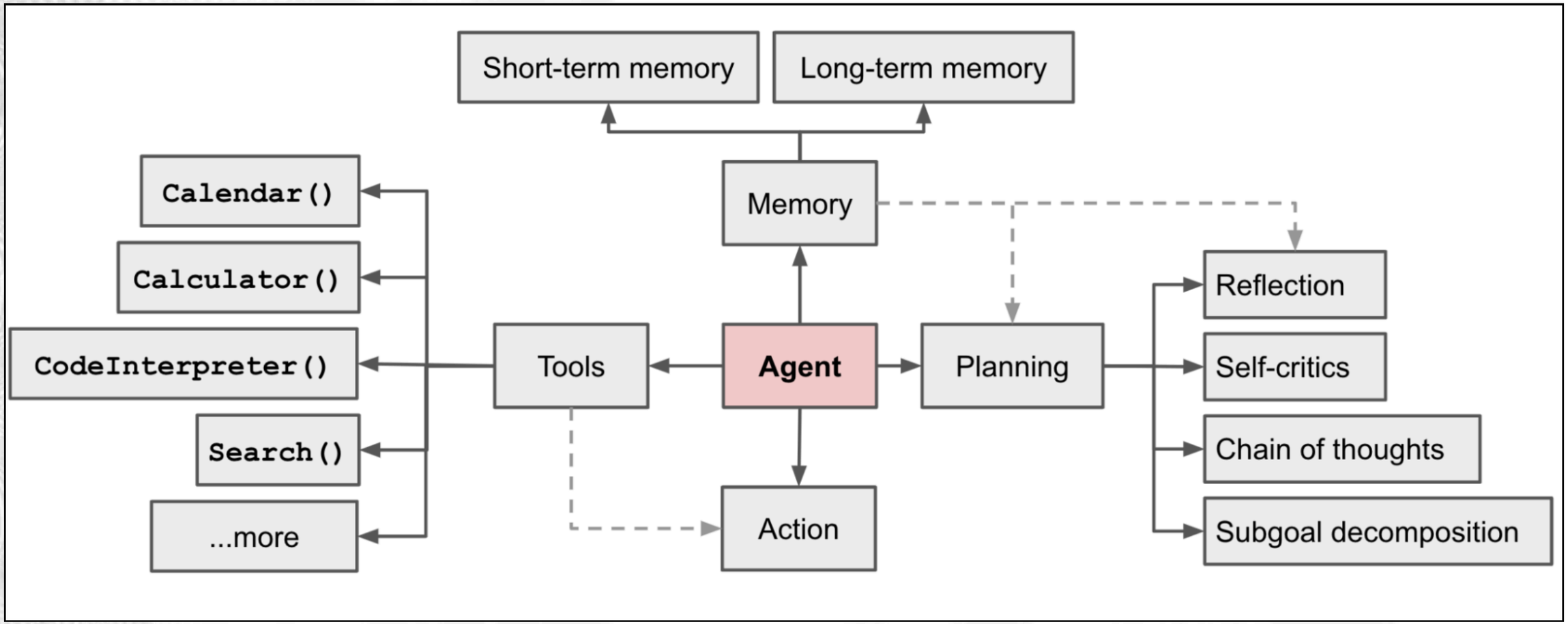
- The sensors parse the environment for the internal logic
- The internal logic (LLM) decides the action
- The actuators execute said action



Agents interact with environments through sensors and actuators.

Russell, S., Norvig, P., & Intelligence, A. (1995). A modern approach. Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs, 25(27), 79-80.

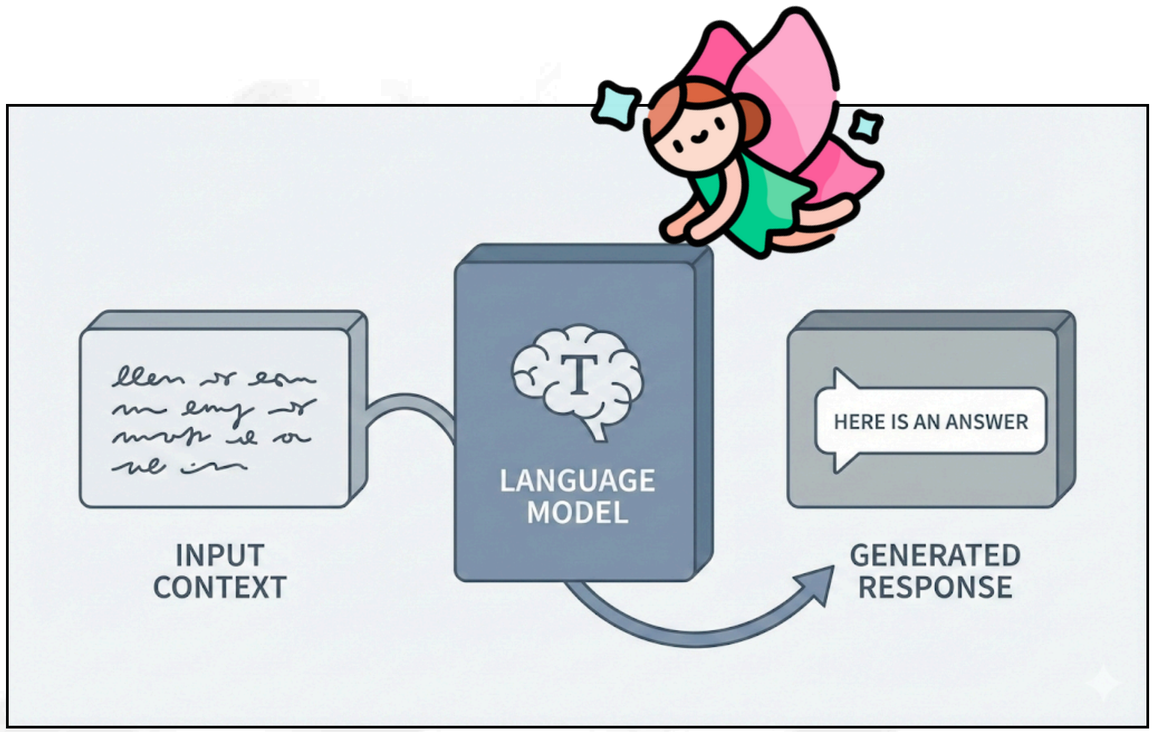
THE TYPICAL STRUCTURE OF LLM-BASED AGENTS



https://lilianweng.github.io/posts/2023-06-23-agent/

LANGUAGE MODELS

For us: LMs are a black box (**magic**) that take some input context, and output some text



Input context:

- Defines the task
- Defines the tools the agent can use
- Usually very structured (relatively to text in general)

Response:

- Contains the action to do/ the answer to the task

PLANNING

Task Decomposition

Use variants of **Chain-of-tought** (CoT) prompting for enhancing model performance on complex tasks.

The model is instructed to “think step by step” to utilize more test-time computation to **decompose hard tasks into smaller and simpler steps**.

Self-Reflection

Make the agent reflect on its past actions explicitly, to

Frameworks like ReAct add a “thinking” step as a possible action for the agent:

$$\hat{\mathcal{A}} = \mathcal{A} \cup \mathcal{L}$$

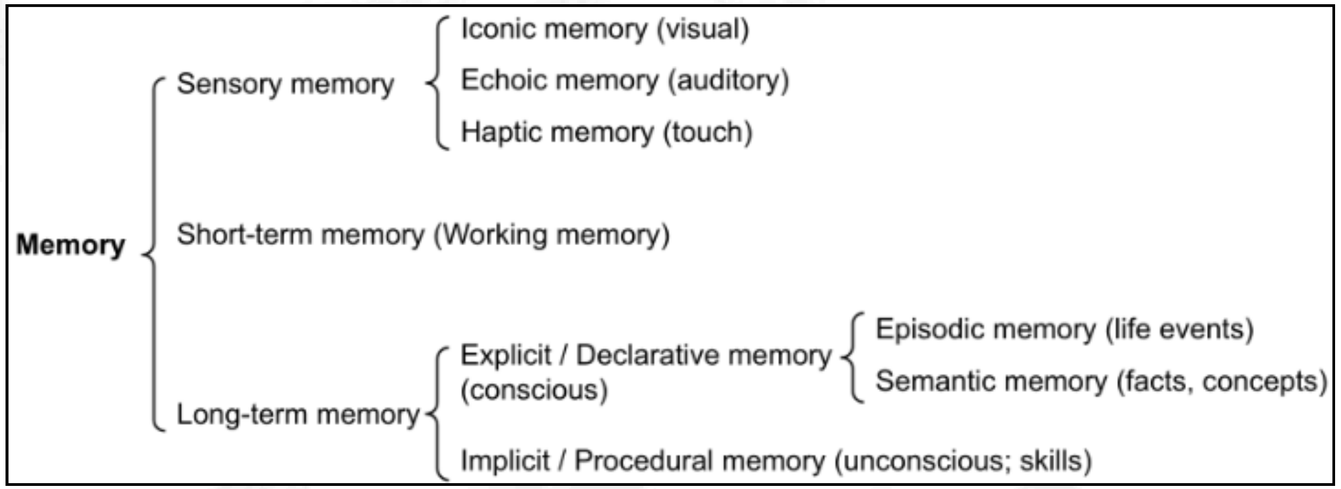
Chen, Q., Qin, L., Liu, J., Peng, D., Guan, J., Wang, P., Hu, M., Zhou, Y., Gao, T., & Che, W. (2025). Towards Reasoning Era: A Survey of Long Chain-of-Thought for Reasoning Large Language Models
Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., & Cao, Y. (2022, September 29). ReAct: Synergizing Reasoning and Acting in Language Models. The Eleventh International Conference on Learning Representations

MEMORY

Sensory memory: learning embedding representations for raw inputs, including text, image or other modalities

Short-term memory: in-context learning. It is short and finite, as it is restricted by the finite context window length of Transformer.

Long-term memory: external vector store that the agent can attend to at query time, accessible via fast retrieval.



Hu, Y., Liu, S., Yue, Y., Zhang, G., Liu, B., Zhu, F., Lin, J., Guo, H., Dou, S., Xi, Z., Jin, S., Tan, J., Yin, Y., Liu, J., Zhang, Z., Sun, Z., Zhu, Y., Sun, H., Peng, B., … Yan, S. (2026). Memory in the Age of AI Agents (arXiv:2512.13564). arXiv. https://doi.org/10.48550/arXiv.2512.13564

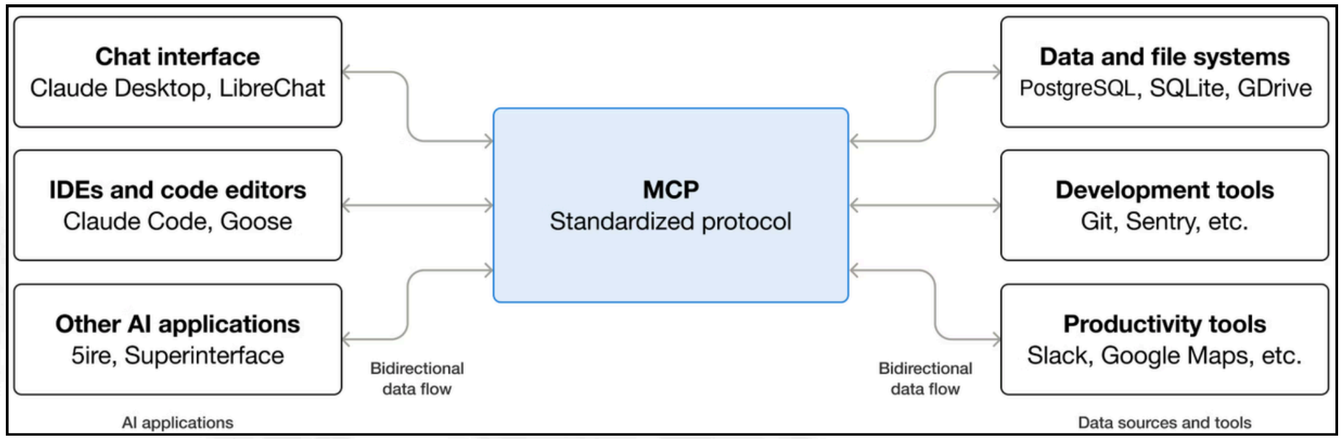
https://www.pinecone.io/

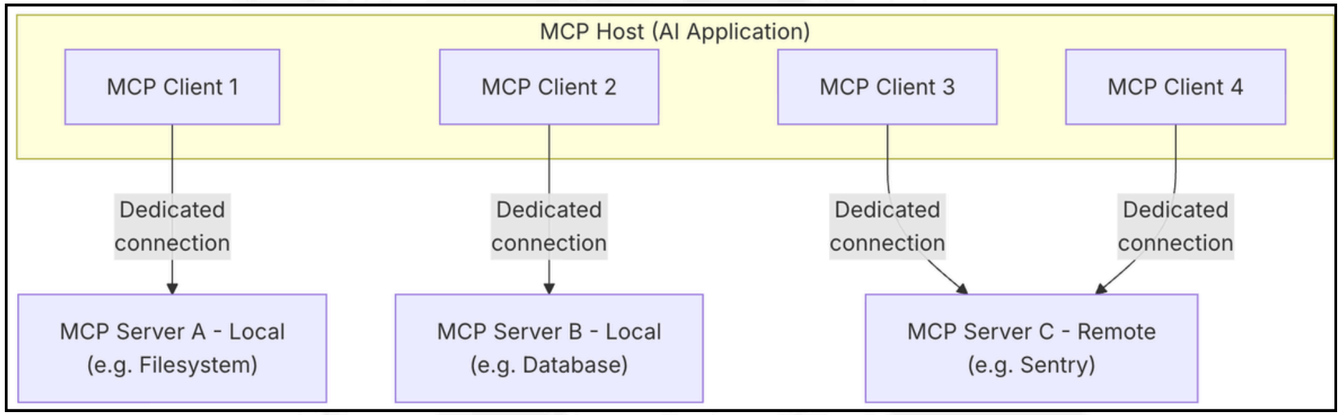
TOOL CALLS

Tools: Executable functions that AI applications (Agents) can invoke to perform actions

Model Context Protocol (MCP): open-source standard for connecting AI applications to external systems.

- **MCP Client:** Maintains a connection to an MCP server and obtains context from an MCP server for the MCP host to use
- **MCP Server:** A program that provides context to MCP clients





https://modelcontextprotocol.io

SWE-AGENT: AN AI AGENT FOR SOFTWARE DEV.

Using the linux terminal is hard due to being fine-grained → give a **subset of high-level instructions**

- **Search and navigation:** *find_file*, *search_file*, *search_dir*
- **File viewer:** *open*, *scroll_down*, *scroll_up* (100 lines window size)
- **File editor:** *edit(start,end,text)*

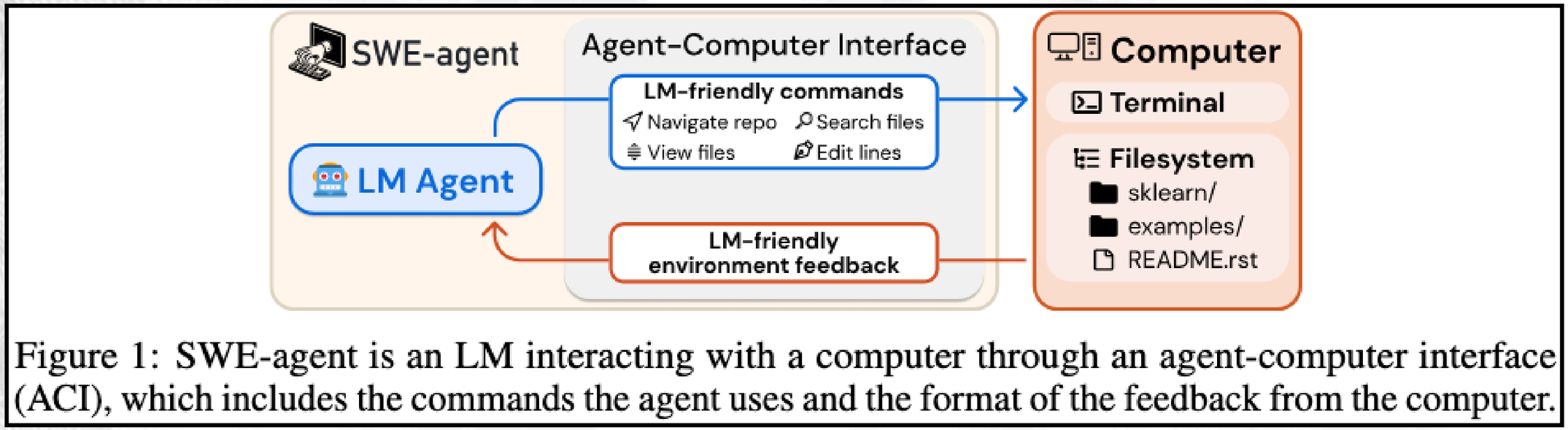


Figure 1: SWE-agent is an LM interacting with a computer through an agent-computer interface (ACI), which includes the commands the agent uses and the format of the feedback from the computer.

Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., & Press, O. (2024). SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering. Advances in Neural Information Processing Systems, 37, 50528–50652. https://doi.org/10.52202/079017-1601

PLANCRAFT: CRAFTING THINGS IN MINECRAFT

Task: Craft a green bed [ ]

1. **Smelt** cactus [ ] into green dye [ ]
2. **Move** green dye [ ] into grid
3. **Move** white bed [ ] into grid
4. **Move** green bed [ ] into inventory

Figure 1: A Plancraft example where the task is to craft a green bed. The agent has to use the observations (text or image) to generate the next action. In this case, the shortest path is: crafting the green dye by smelting the cactus; then moving the items (white bed and green dye) into the correct crafting slots; then moving the resulting green bed into the inventory.

Dagan, G., Keller, F., & Lascarides, A. (2025). Plancraft: An evaluation dataset for planning with LLM agents (arXiv:2412.21033). arXiv. https://doi.org/10.48550/arXiv.2412.21033

AGENTIC AI SYSTEMS

FROM AI AGENTS TO AGENTIC AI SYSTEMS

AI agent

- Typically designed as single-entity systems
- Perform goal-directed tasks by utilizing external tools
- Sequential reasoning
- Real-time information
- **Objective**: to complete well-defined functions

Agentic AI system

- Multiple, specialized agents that coordinate and communicate
- Dynamically allocate sub-tasks within a broader workflow
- **Objective**: to achieve a common goal(s)

CHATDEV: COMMUNICATIVE AGENTS FOR SOFTWARE DEVELOPMENT

Intuition: Multi-agent system for software development starting from high-level application requirements

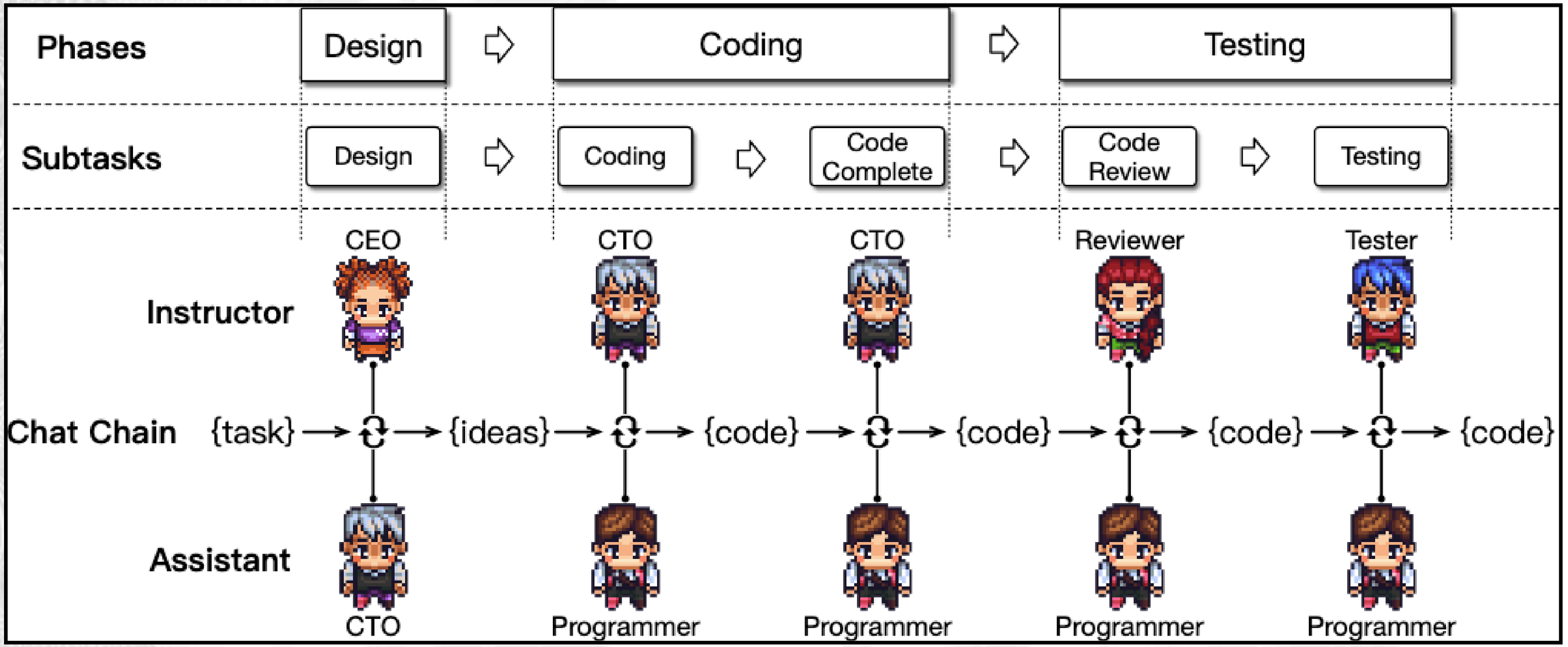
Simulates waterfall development with design, coding and testing phases

Two main parts:

- **Chat chain**: each phase is further divided in sub-tasks, handled with multi-turn debate
- **Communicative dehallucination:** Debate assistant agent asks for more clarifications to the instructor

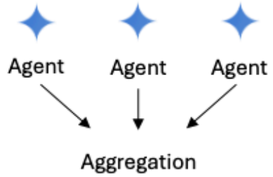
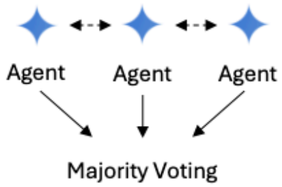
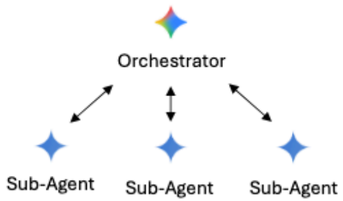
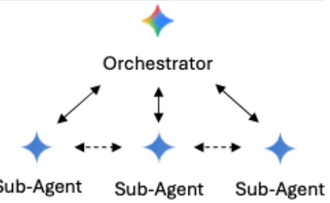
Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., Yang, C., Chen, W., Su, Y., Cong, X., Xu, J., Li, D., Liu, Z., & Sun, M. (2024). ChatDev: Communicative Agents for Software Development. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 15174–15186). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.810

CHATDEV WORKFLOW



SOME SIMPLE AGENTIC SYSTEMS

Table 2 | Architectural comparison of agent methods with objective complexity metrics. Computational complexity measured in terms of LLM calls, coordination overhead, and parallelization potential.

Characteristic	SAS	MAS (Independent)	MAS (Decentralized)	MAS (Centralized)	MAS (Hybrid)
Interaction Type					
LLM Calls	$O(k)$	$O(nk) + O(1)$	$O(dnk) + O(1)$	$O(rnk) + O(r)$	$O(rnk) + O(r) + O(p)$
Sequential Depth	k	k	d	r	r
Comm. Overhead	0	1	$d \cdot n$	$r \cdot n$	$r \cdot n + p \cdot m$
Parallelization Factor	1	n	n	n	n
Memory Complexity	$O(k)$	$O(n \cdot k)$	$O(d \cdot n \cdot k)$	$O(r \cdot n \cdot k)$	$O(r \cdot n \cdot k + p \cdot n)$
Coordination	Sequential	Parallel + Synthesis	Sequential Debate	Hierarchical	Hierarchical + Peer
Consensus	-	Synthesis	Debate	Orchestrator	Orchestrator

* k = max iterations per agent, n = number of agents, r = orchestrator rounds, d = debate rounds, p = peer communication rounds, m = average peer requests per round. Communication overhead counts inter-agent message exchanges. Independent offers maximal parallelization with minimal coordination. Decentralized uses sequential debate rounds. Hybrid combines orchestrator control with directed peer communication.

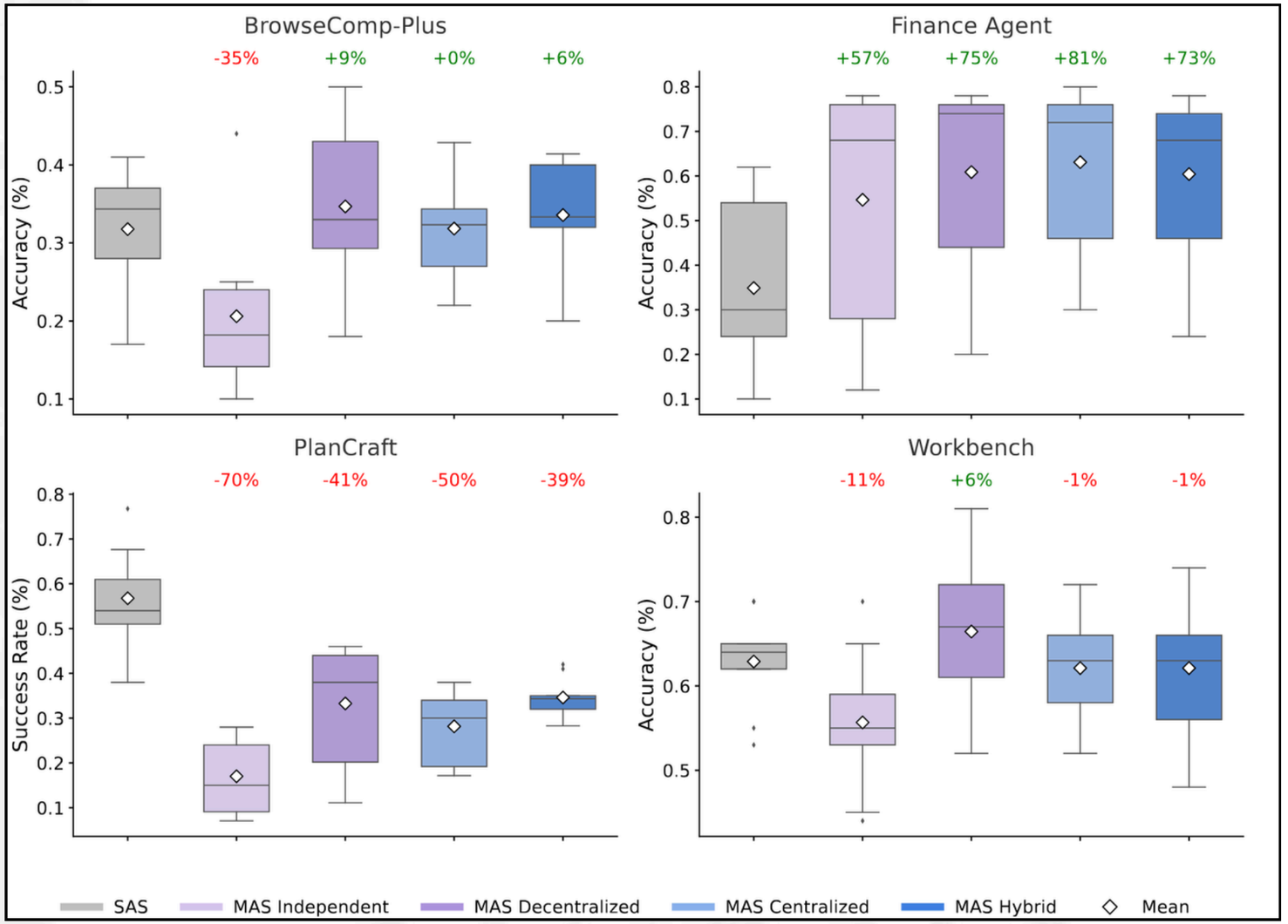
Kim, Y., Gu, K., Park, C., Park, C., Schmidgall, S., Heydari, A. A., Yan, Y., Zhang, Z., Zhuang, Y., Malhotra, M., Liang, P. P., Park, H. W., Yang, Y., Xu, X., Du, Y., Patel, S., Althoff, T., McDuff, D., & Liu, X. (2025). Towards a Science of Scaling Agent Systems (arXiv:2512.08296). arXiv. https://doi.org/10.48550/arXiv.2512.08296

CAN WE STILL SOLVE MINECRAFT?

The **optimal Agentic system** is **task-dependent**

What happens:

- **Plancraft:** due to the high sequential nature of the task, parallelization gives no benefit
- **Searching:** more parallelizable tasks (finance, web search) actually benefit from multiple agents



EDGE SYSTEMS

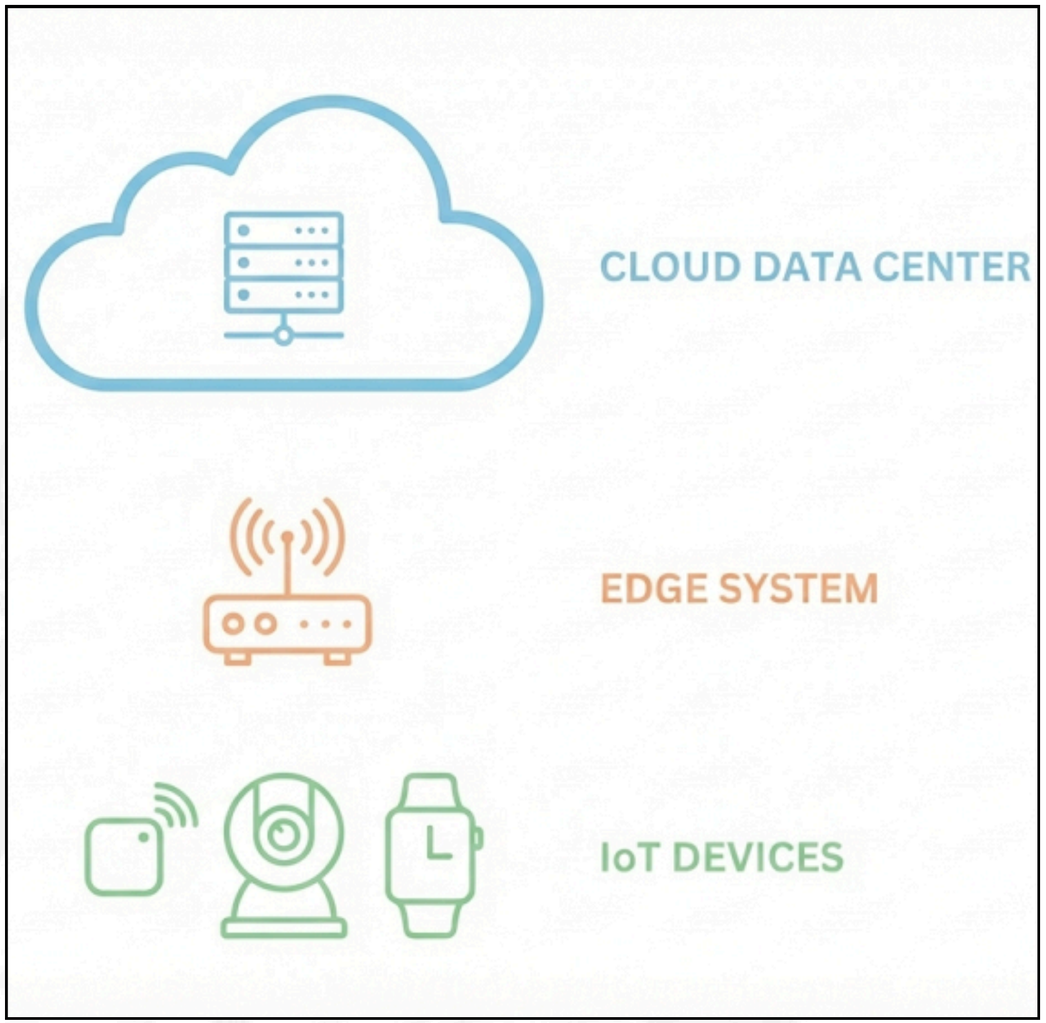
WHAT IS EDGE COMPUTING?

Edge System

Any computing and network resources positioned along the path between the data sources and the cloud data centers

Edge Computing

Approach where data processing and computation happen right at the edge of the network, **near the data source**



Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. IEEE Internet of Things Journal, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

BENEFITS OF EDGE COMPUTING

FASTER RESPONSE TIMES

By processing data locally, edge computing drastically reduces the time it takes for systems to react

BANDWIDTH SAVINGS

Pre-processing and filtering data at the edge means significantly less raw data needs to be transported to the cloud

LOWER ENERGY CONSUMPTION

By having local computations you save the energy used for data transmission

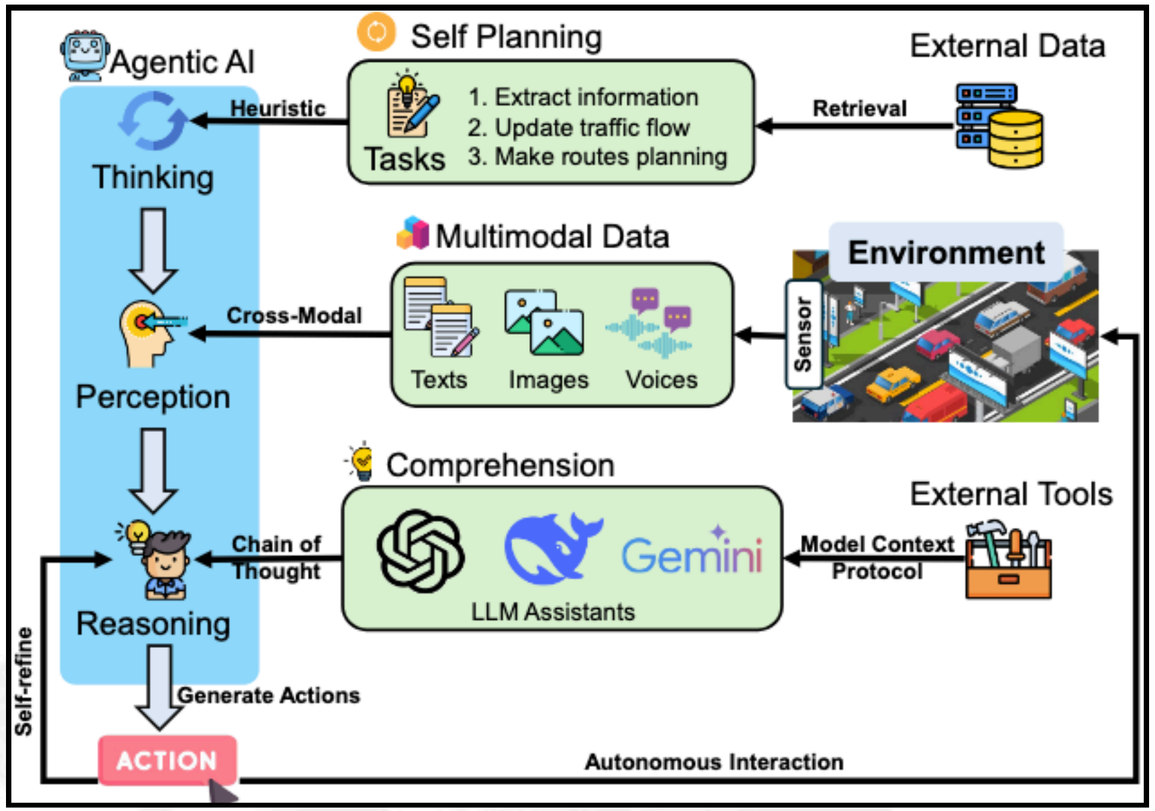
THE MAIN LOOP OF AGENTIC AI AT THE EDGE

Perception Module: captures external multimodal data and environmental context

Comprehension stage: processes and interprets perceptions

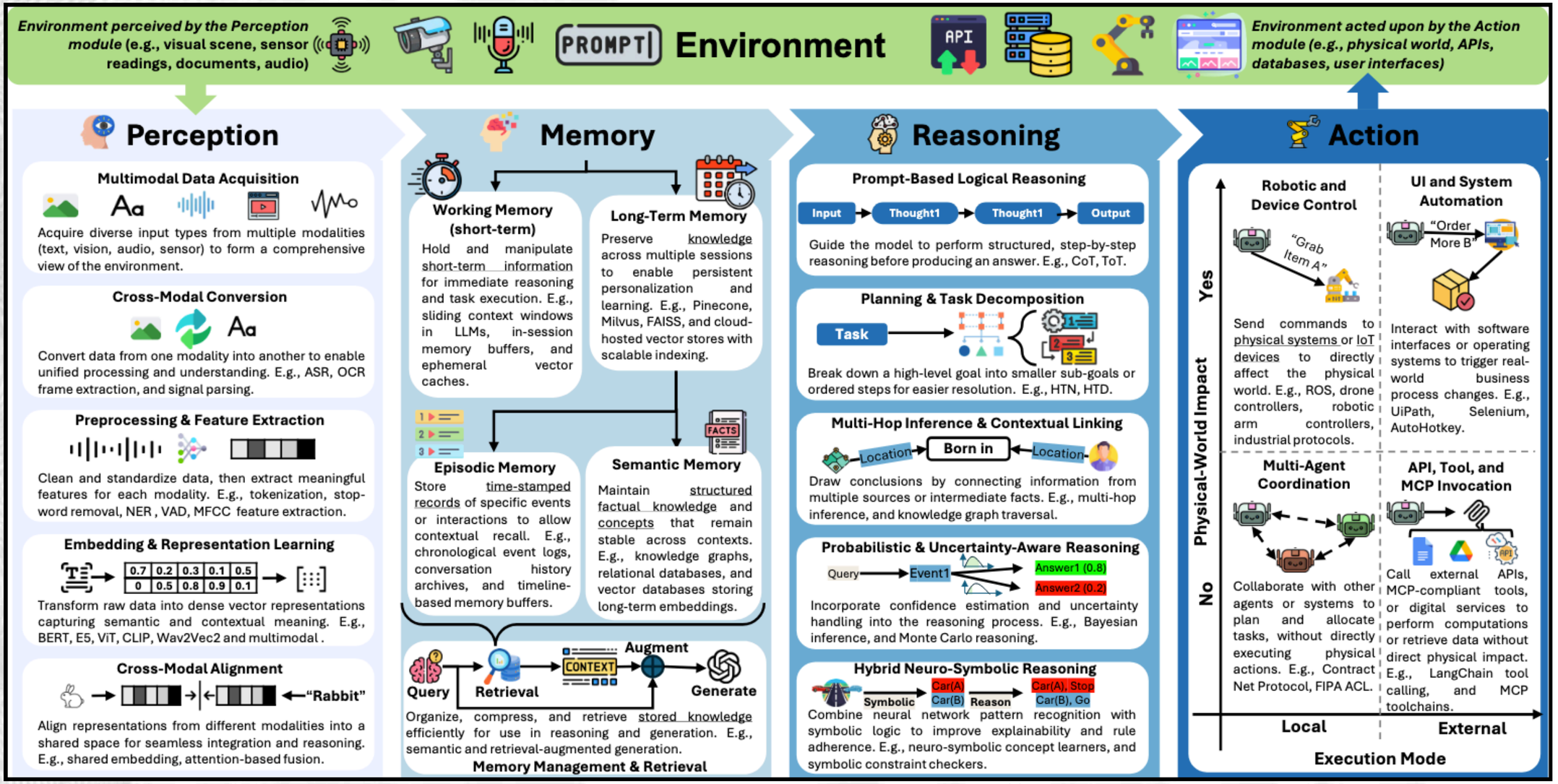
Reasoning stage: explicit planning mechanisms and longterm memory retrieval (e.g., RAG) refine these plans further, incorporating additional contextual insights and prior knowledge.

Finally, **autonomous actions** are executed using External Tools and APIs, with outcomes continuously evaluated and fed back into the loop for iterative self-refinement.



Zhang, R., Liu, G., Liu, Y., Zhao, C., Wang, J., Xu, Y., Niyato, D., Kang, J., Li, Y., Mao, S., Sun, S., Shen, X., & Kim, D. I. (2025). Toward Edge General Intelligence with Agentic AI and Agentification: Concepts, Technologies, and Future Directions (arXiv:2508.18725; Version 1). arXiv. https://doi.org/10.48550/arXiv.2508.18725

AGENTIC AI IN EDGE SYSTEMS: ENVIRONMENT



SUMMARY

AI AGENTS

By using LMs, agents are able to solve different tasks using long-term planning, tools, and memory

AGENTIC AI SYSTEMS

For complex problems, dividing the tasks among different agents can yield better results, but the architecture is task dependent

EDGE SYSTEMS

Edge computing has te potential to allow the large scale employment of agentic systems, without relying only on centralized clouds

OPEN QUESTION(S)

How do we enable the execution of Agentic workflows in Edge systems?

- **Task decomposition:** How can we divide a problem in subtasks according to the current agents deployed at the edge?
- **Task routing:** How do we route the tasks to the optimal agent? When do we use a more complex workflow?
- **Self-organization:** Can we make agents self-organize autonomously instead of manually assigning roles?
- **Perception:** How do we handle the “perception” in edge systems? (modality conversion, feature extraction, etc..)
- **Memory organization:** How can we organize the agentic memory in the edge to ensure fast retrieval?

THANK YOU

CHAIN-OF-TOUGHT PROMPTING

Standard Prompting

Model Input

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?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

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?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

(1) Hotspot QA

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

(1a) Standard

Answer: iPod ✗

(1b) CoT (Reason Only)

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch ✗

(1c) Act-Only

Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control …

Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: …

Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Act 4: Finish[yes] ✗

(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] ✓

ASSIGNING ROLES TO AGENTS

(a) Zero-Shot

Input

User: Xavier was 4 feet tall and grew 3 inches. Cole was 50 inches tall and grew 2 inches over the summer. What is the difference between Cole and Xavier’s height now?

Output

Assistant: Xavier is now 4 feet + 3 inches = 7 inches tall. Cole is now 50 inches + 2 inches = 52 inches tall. The difference between their heights is 52 inches - 7 inches = 45 inches. ✗

(b) Role-Play Prompting

Input

User: From now on, you are an excellent math teacher and always teach your students math problems correctly. And I am one of your students.

Assistant: That's great to hear! As your math teacher, I'll do my best to explain mathematical concepts correctly so that you can understand them easily. Feel free to ask any math problems or questions you have, and I'll be glad to assist you. Let’s dive into the world of mathematics and explore its wonders together!

User: Xavier was 4 feet tall and grew 3 inches. Cole was 50 inches tall and grew 2 inches over the summer. What is the difference between Cole and Xavier’s height now?

Output

Assistant: … Since 1 foot is equal to 12 inches, Xavier's initial height: 4 feet * 12 inches/foot = 48 inches. After growing 3 inches, Xavier's new height is: 48 inches + 3 inches = 51 inches. … Cole's new height after growing 2 inches: 50 inches + 2 inches = 52 inches. … We subtract Xavier's height from Cole's height: 52 inches - 51 inches = 1 inch. Therefore, the difference between Cole and Xavier's height now is 1 inch. ✓

MODEL SIZE AT THE EDGE

Why LARGE?

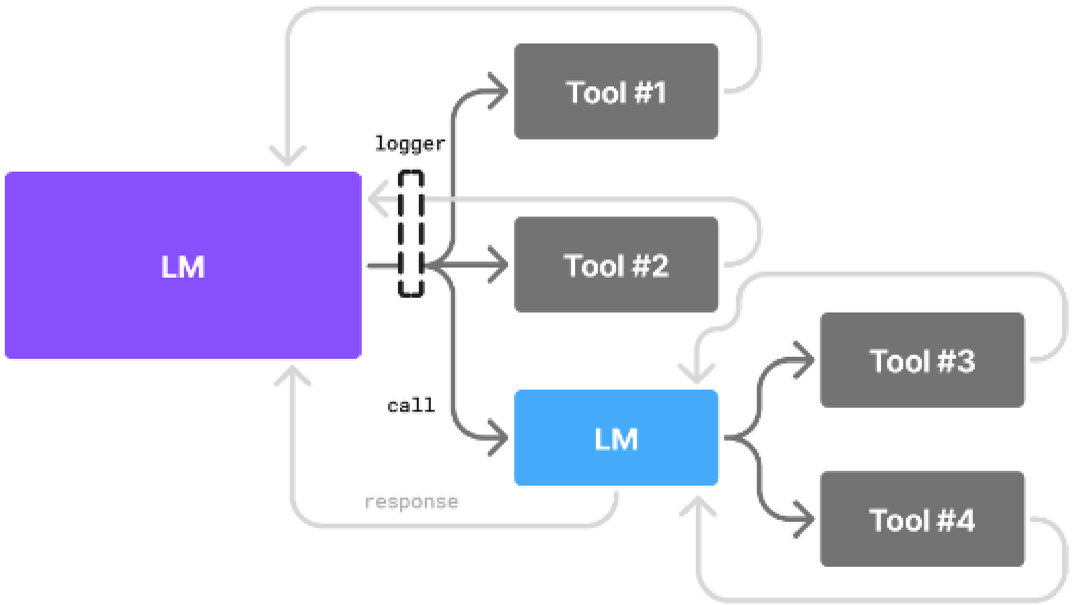
- Large models perform better generally on any task
- Inference is cheaper due to centralization
- Don’t have to pay people to maintain it as the system as a whole is relatively simple
- Very high head start in terms of system development

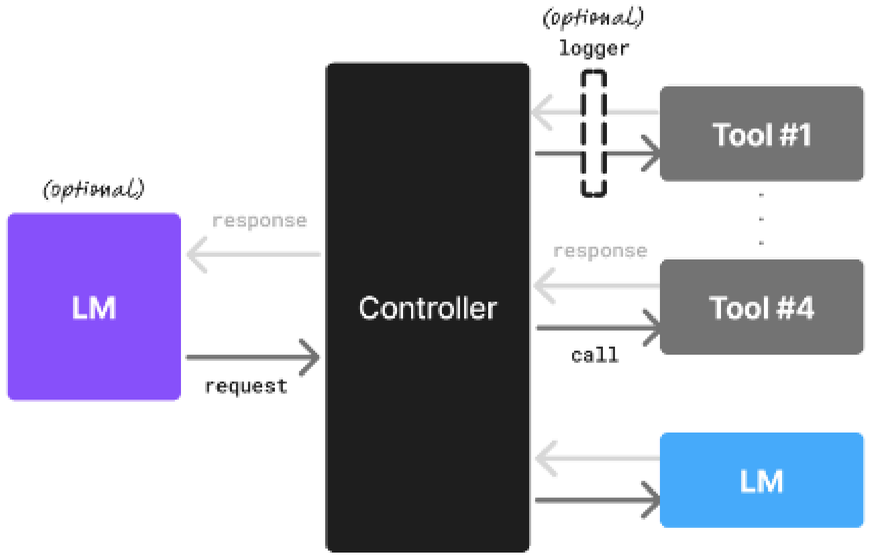
Why Small?

- Can be specialized for different tasks (and agentic tasks tend to be specialized)
- Test-time compute is lower (they are **easier to make adaptive**)
- You can always use a LARGE model to handle the user queries, and then offload everything to small models (See figure)

Belcak, P., Heinrich, G., Diao, S., Fu, Y., Dong, X., Muralidharan, S., Lin, Y. C., & Molchanov, P. (2025). Small Language Models are the Future of Agentic AI (arXiv:2506.02153). arXiv. https://doi.org/10.48550/arXiv.2506.02153

HYBRID ARCHITECTURES





Example Control Flow:

LM T#1 LM T#2 LM LM T#3 LM T#4 LM LM

Example Control Flow:

LM C T#1 C LM C T#4 C LM C LM

Figure 1: An illustration of agentic systems with different modes of agency. *Left: Language model agency.* The language model acts both as the HCI and the orchestrator of tool calls to carry out a task. *Right: Code agency.* The language model fills the role of the HCI (optionally) while a dedicated controller code orchestrates all interactions.