Cognitive Architectures for Language Agents

Theodore Sumers* Shunyu Yao* Karthik Narasimhan Thomas L. Griffiths
Princeton University

Reporter: Xiachong Feng

Authors

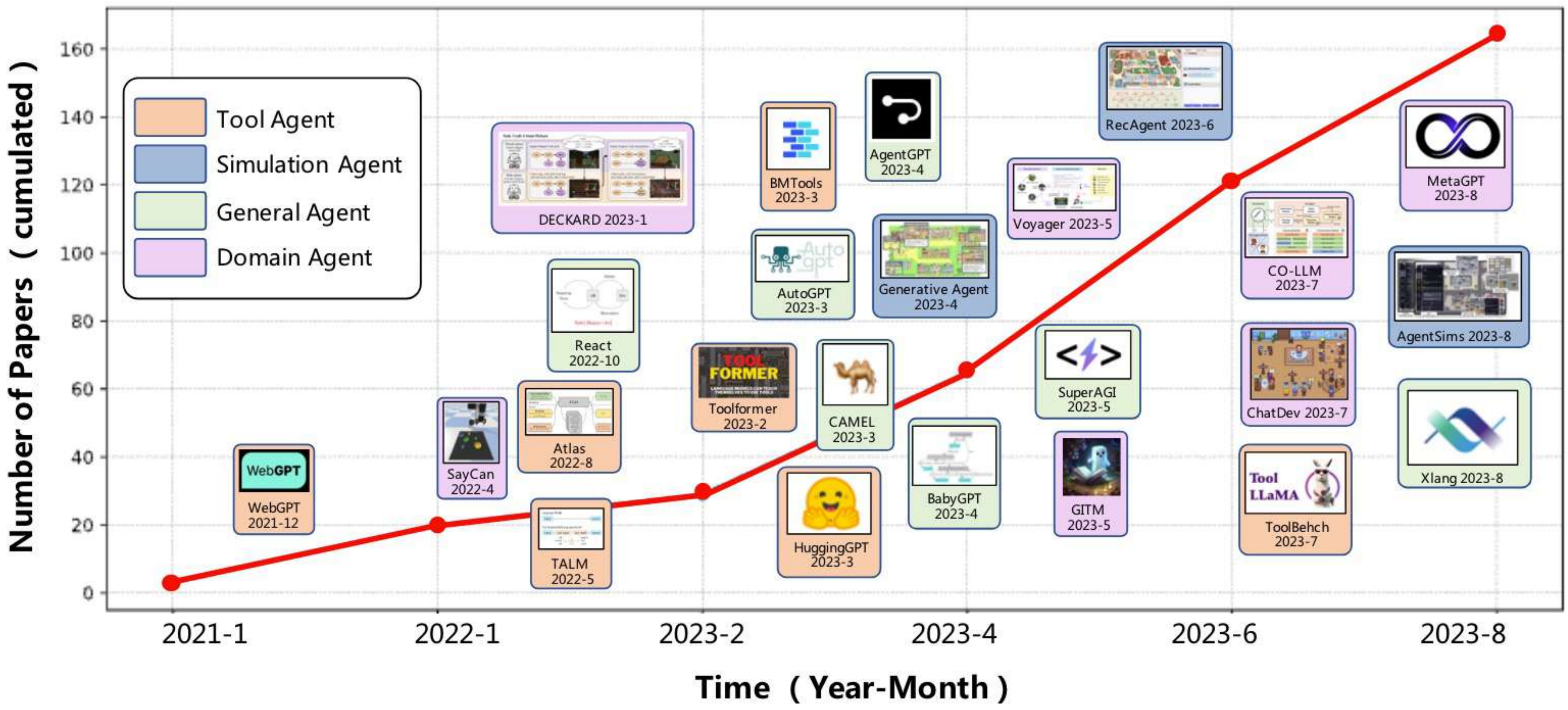
Theodore Sumers
Fifth-year PhD student

Shunyu Yao
PhD student

Karthik Narasimhan
Assistant Professor

Thomas L. Griffiths
Professor of Psychology and Computer Science

Background



A Survey on Large Language Model based Autonomous Agents

Limitation

However, these efforts have largely been piecemeal, **lacking a systematic framework** for constructing a fully-fledged language agent.

Production System

Production systems for string manipulation

Rule: $X\, Y\, Z \to X\, W\, Z$

precondition action

production system

Control flow: From strings to algorithms

Rules

$$* \mid\mid\mid\mid\mid \;\to\; \mid *$$

$$* \;\xrightarrow{\bullet}\; *$$

$$\;\to\; *$$

Input: 11

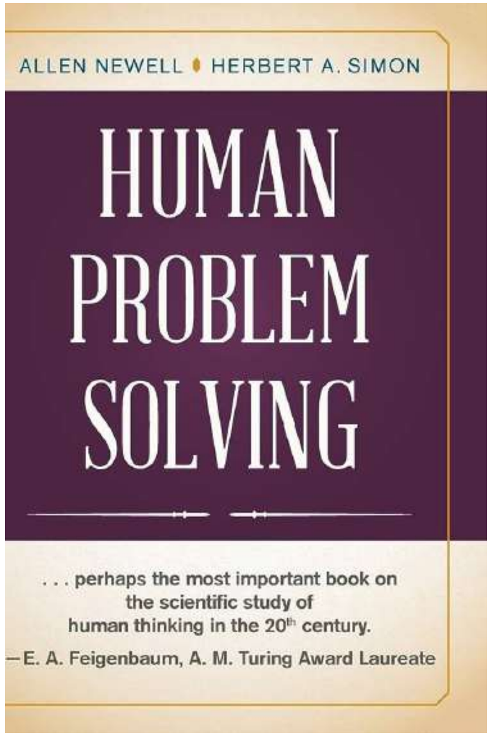
$$* \mid\mid\mid\mid\mid\mid\mid\mid\mid\mid\mid \;\to\; \mid * \mid\mid\mid\mid\mid\mid \;\to\; \mid\mid * \mid \;\xrightarrow{\bullet}\; \mid\mid * \mid$$

2 remainder 1

Simple productions can result in complex behavior

String rewriting to logical operations

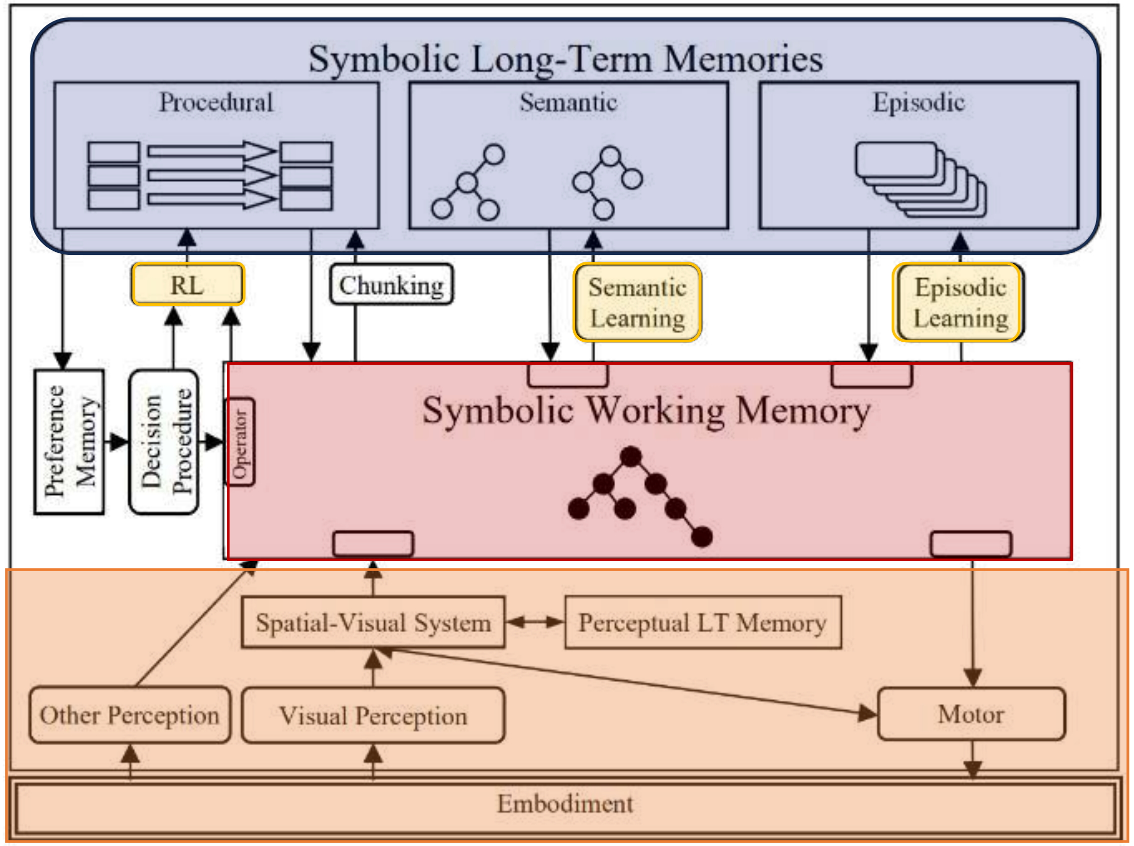
- *preconditions* could be checked against the agent's goals and world state
- *actions* that should be taken if the preconditions were satisfied.



a simple production system to describe the operation of a thermostat

$$
\begin{array}{rcl}
(\text{temperature} > 70^\circ) \wedge (\text{temperature} < 72^\circ) & \rightarrow & \text{stop} \\
\text{temperature} < 32^\circ & \rightarrow & \text{call for repairs; turn on electric heater} \\
(\text{temperature} > 70^\circ) \wedge (\text{furnace off}) & \rightarrow & \text{turn on furnace} \\
(\text{temperature} > 72^\circ) \wedge (\text{furnace on}) & \rightarrow & \text{turn off furnace}
\end{array}
$$

Cognitive architectures: From algorithms to agents (Soar architecture)



Long term memory is divided into three distinct types.

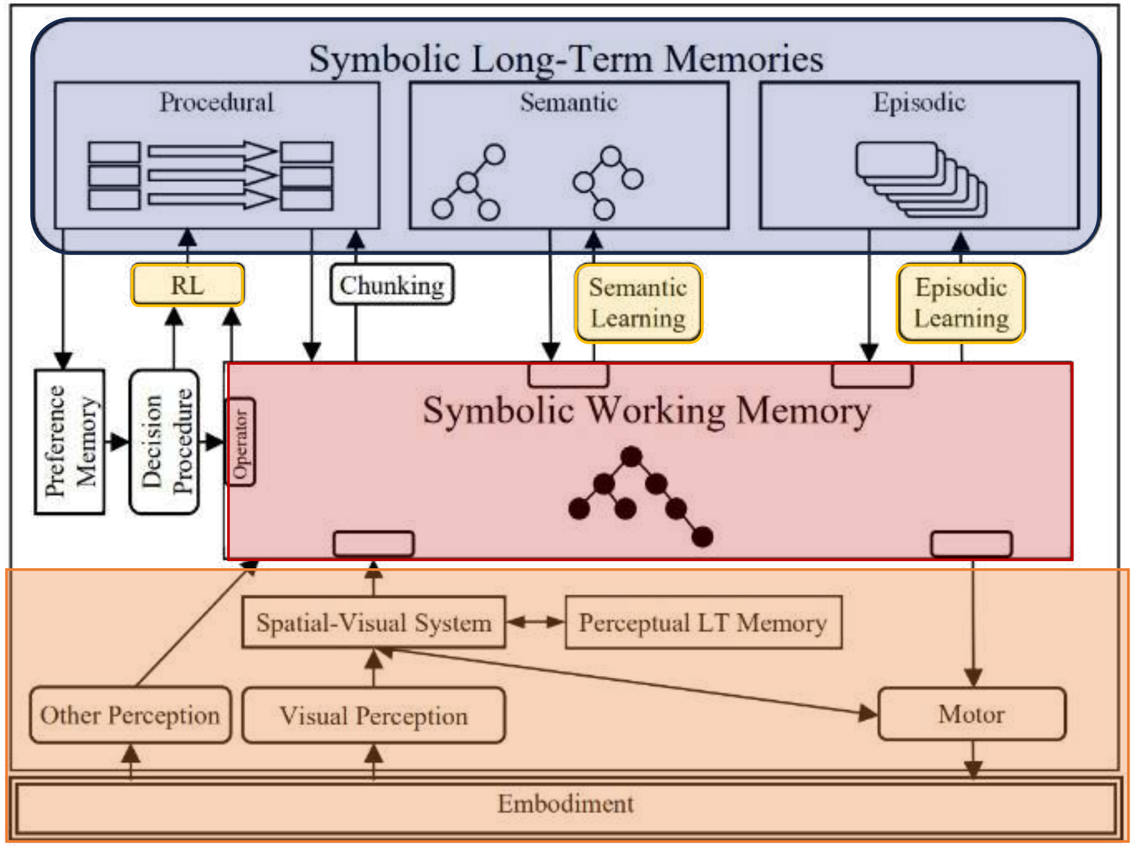
- Procedural memory stores the production system itself: the set of rules that can be applied to working memory to determine the agent's behavior.
- Semantic memory stores facts about the world
- Episodic memory stores sequences of the agent's past behaviors

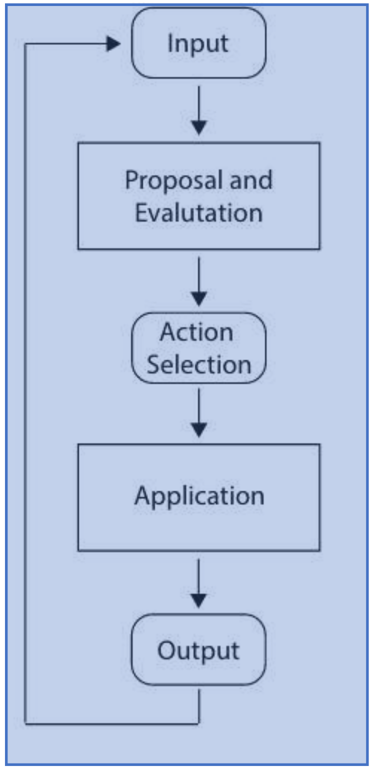
Working memory reflects the agent's current circumstances: it stores the agent's recent perceptual input, goals, and results from intermediate, internal reasoning.

Grounding a variety of sensors stream perceptual input into working memory, where it is available for decision making.

Learning: (1) facts can be written to semantic memory, while experiences can be written to episodic memory (2) Second, behaviors can be modified.

Cognitive architectures: From algorithms to agents (Soar architecture)





Decision making

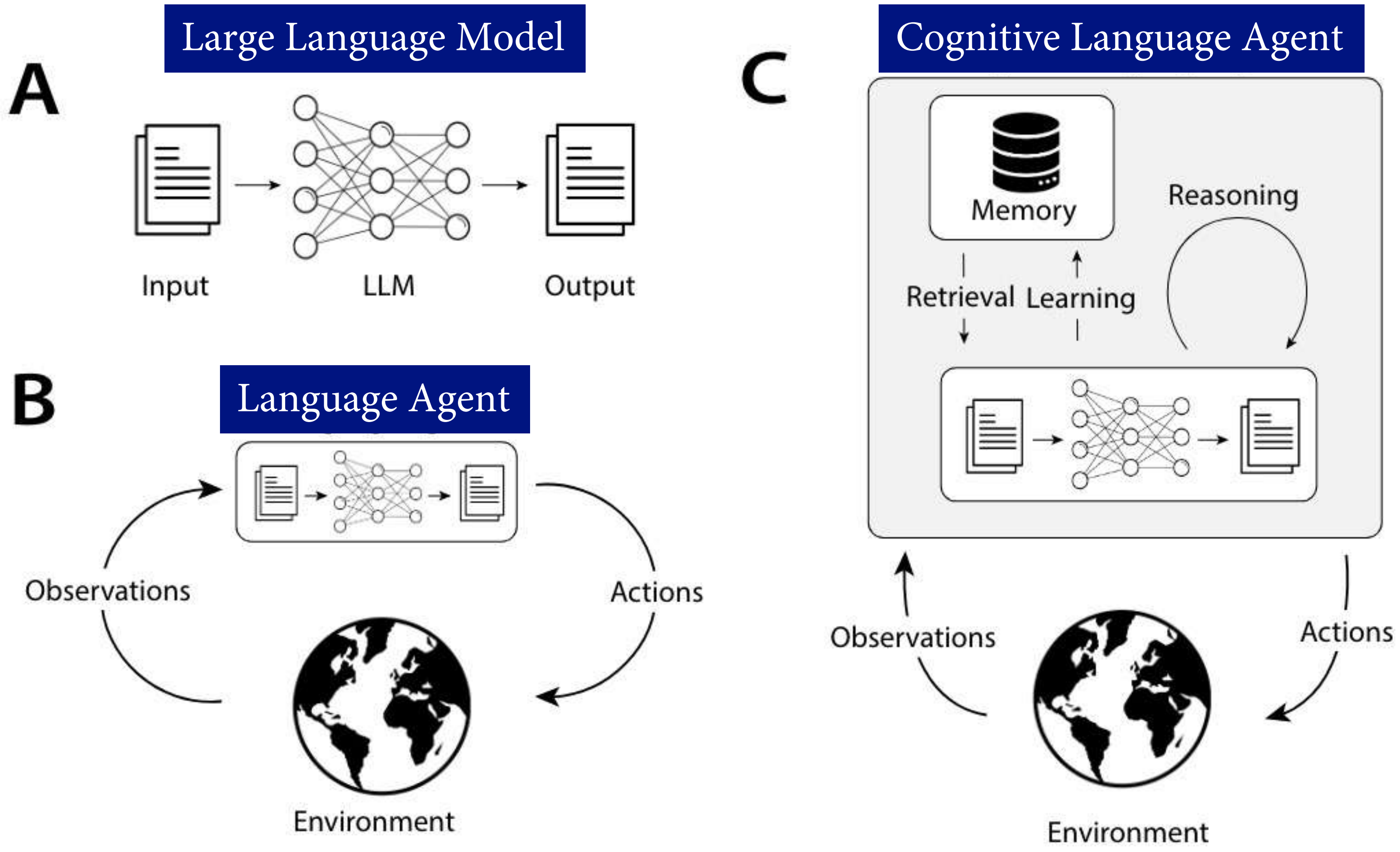
Connections between Language Models and Production Systems

- Language models as probabilistic production systems
  - Language models also define a possible set of expansions or modifications of a string – the prompt provided to the model.
  - LLMs can thus be viewed as probabilistic production systems that sample a possible completion each time they are called, e.g., $X \rightsquigarrow X\ Y$.
- Prompt engineering as control flow

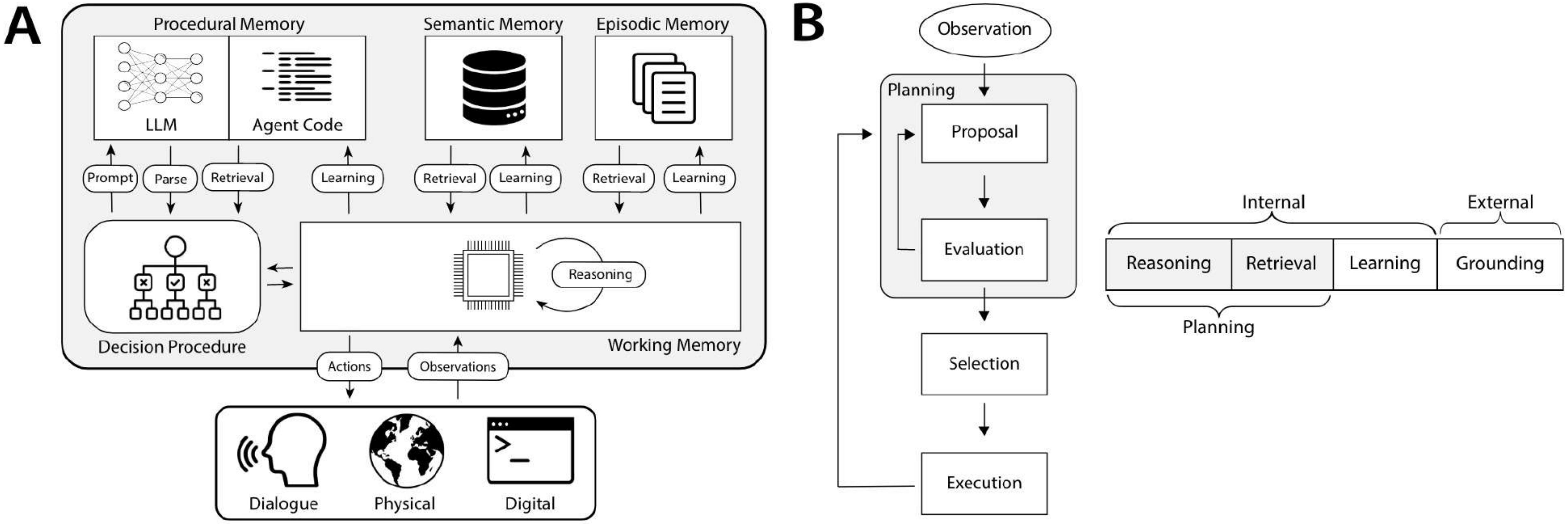
Prompting Method	Production Sequence
Zero-shot	$Q \overset{\text{LLM}}{\rightsquigarrow} Q\ A$
Few-shot (Brown et al., 2020)	$Q \longrightarrow Q_1\ A_1\ Q_2\ A_2\ Q \overset{\text{LLM}}{\rightsquigarrow} Q_1\ A_1\ Q_2\ A_2\ Q\ A$
Zero-shot Chain-of-Thought (Kojima et al., 2022)	$Q \longrightarrow Q_{\text{Step-by-step}} \overset{\text{LLM}}{\rightsquigarrow} Q_{\text{Step-by-step}} A$
Retrieval Augmented Generation (Lewis et al., 2020)	$Q \xrightarrow{\text{Wiki}} Q\ O \overset{\text{LLM}}{\rightsquigarrow} Q\ O\ A$
Socratic Models (Zeng et al., 2022)	$Q \overset{\text{VLM}}{\rightsquigarrow} Q\ O \overset{\text{LLM}}{\rightsquigarrow} Q\ O\ A$
Self-Critique (Saunders et al., 2022)	$Q \overset{\text{LLM}}{\rightsquigarrow} Q\ A \overset{\text{LLM}}{\rightsquigarrow} Q\ A\ C \overset{\text{LLM}}{\rightsquigarrow} Q\ A\ C\ A$

Cognitive Architectures for Language Agents (CoALA):
A Conceptual Framework

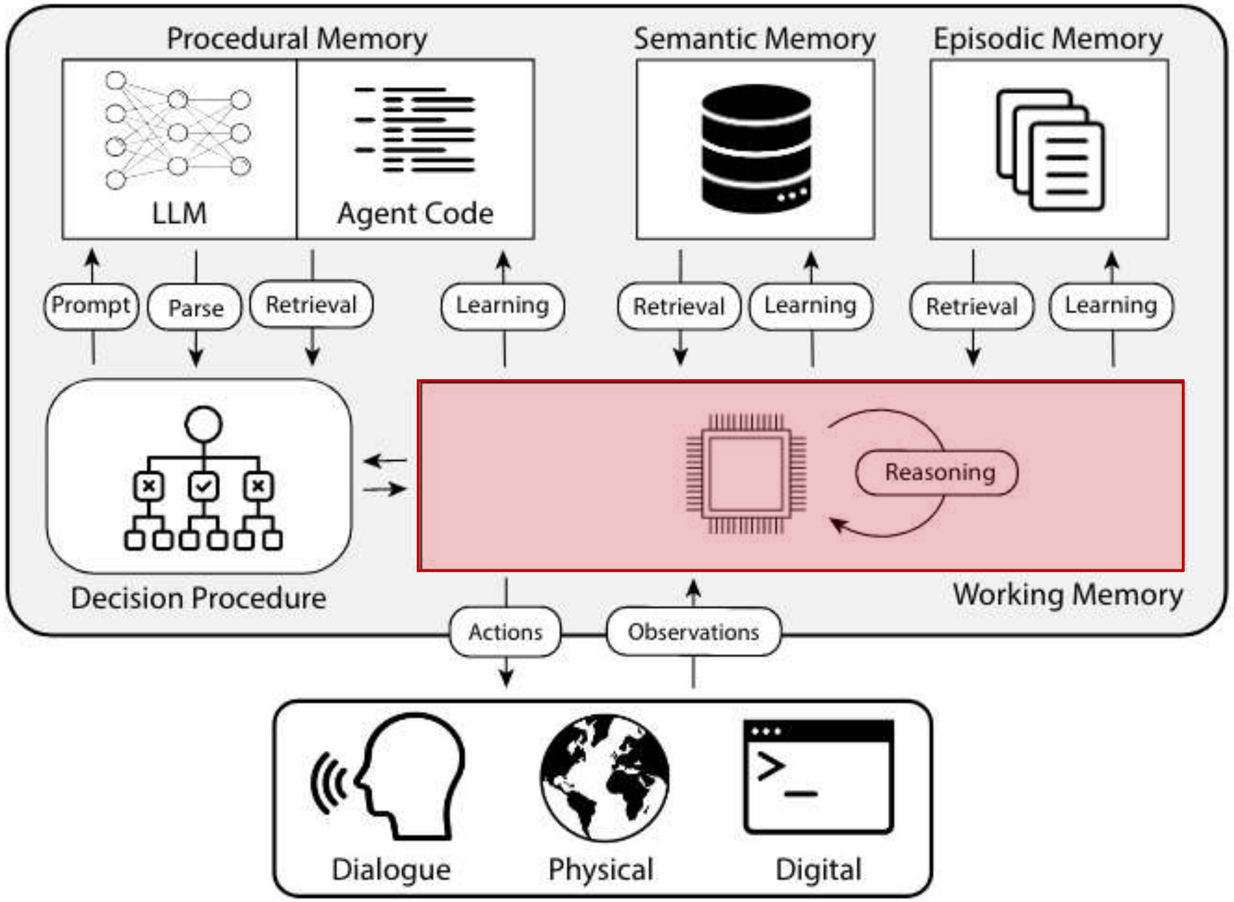
Towards cognitive language agents



Cognitive Architectures for Language Agents (CoALA): A Conceptual Framework



Memory



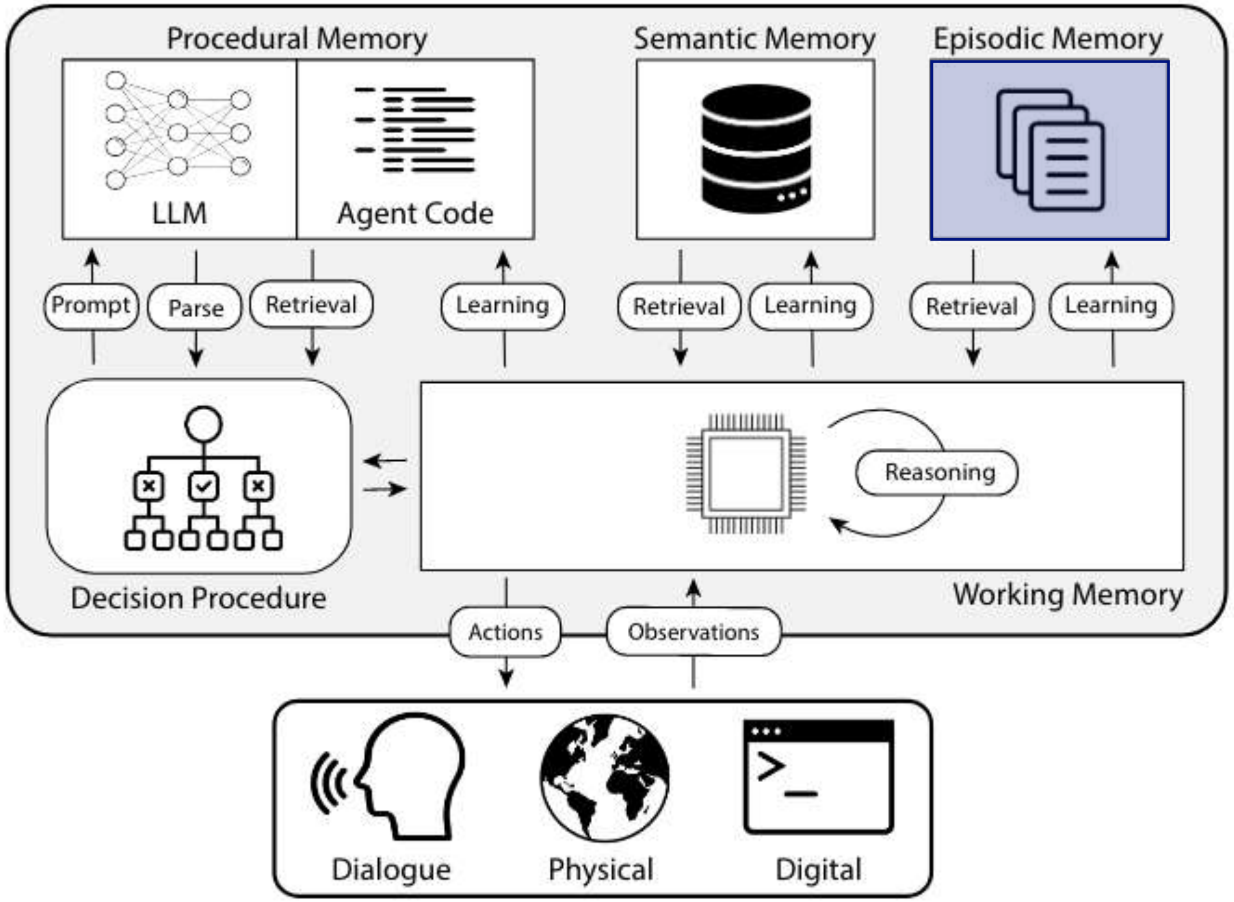
Working Memory.

Working memory maintains *active* and *readily* available information as symbolic variables for the current decision cycle

This includes:

- perceptual inputs from grounding
- knowledge generated by reasoning
- knowledge retrieved from long-term memory
- other core information carried over from the previous decision cycle

Episodic Memory



Episodic Memory.

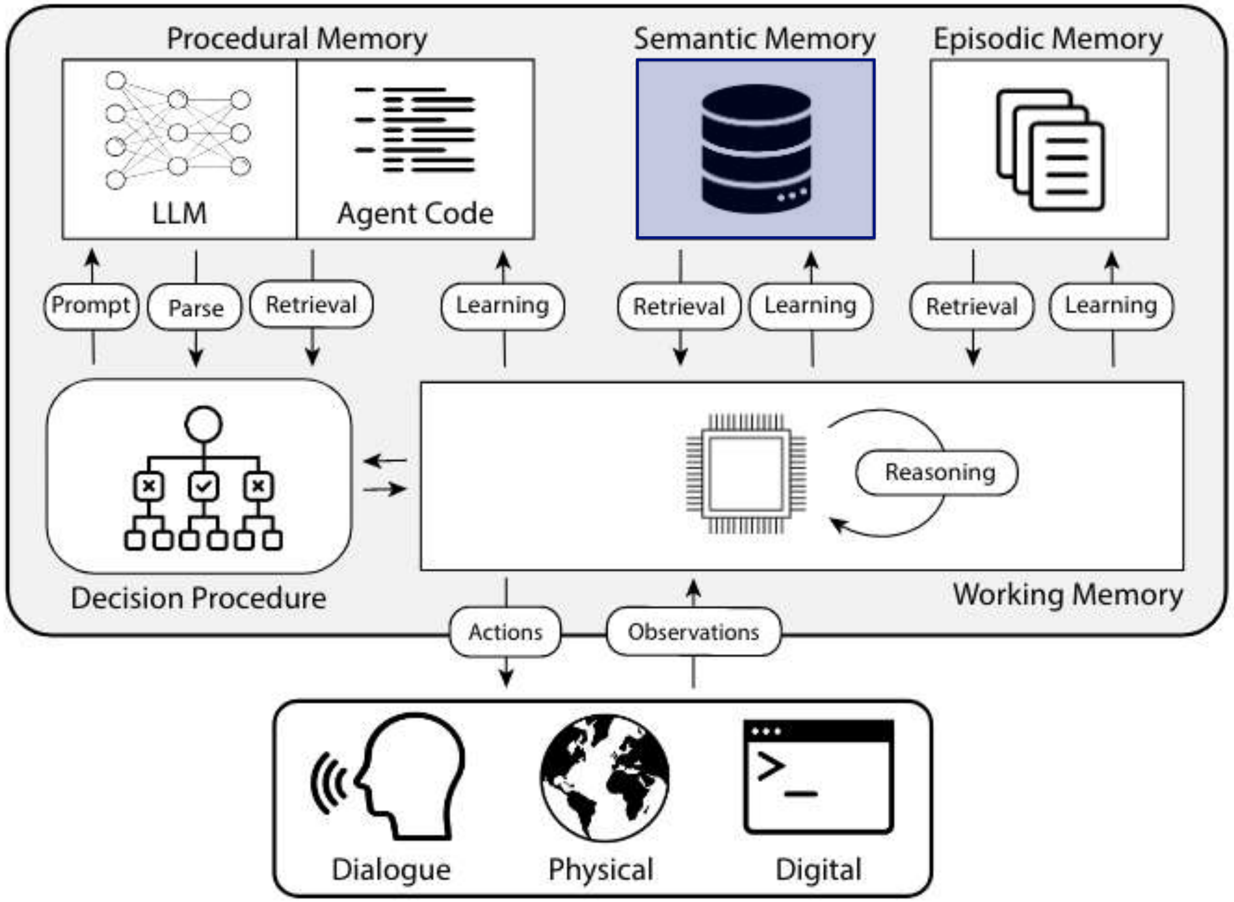
Episodic memory stores experience from *earlier decision cycles*.

This includes:

- training input-output pairs
- history event flows
- game trajectories from previous episodes
- other representations of the agent's experiences.

An agent can also write new experiences from working to episodic memory as a form of learning

Semantic Memory

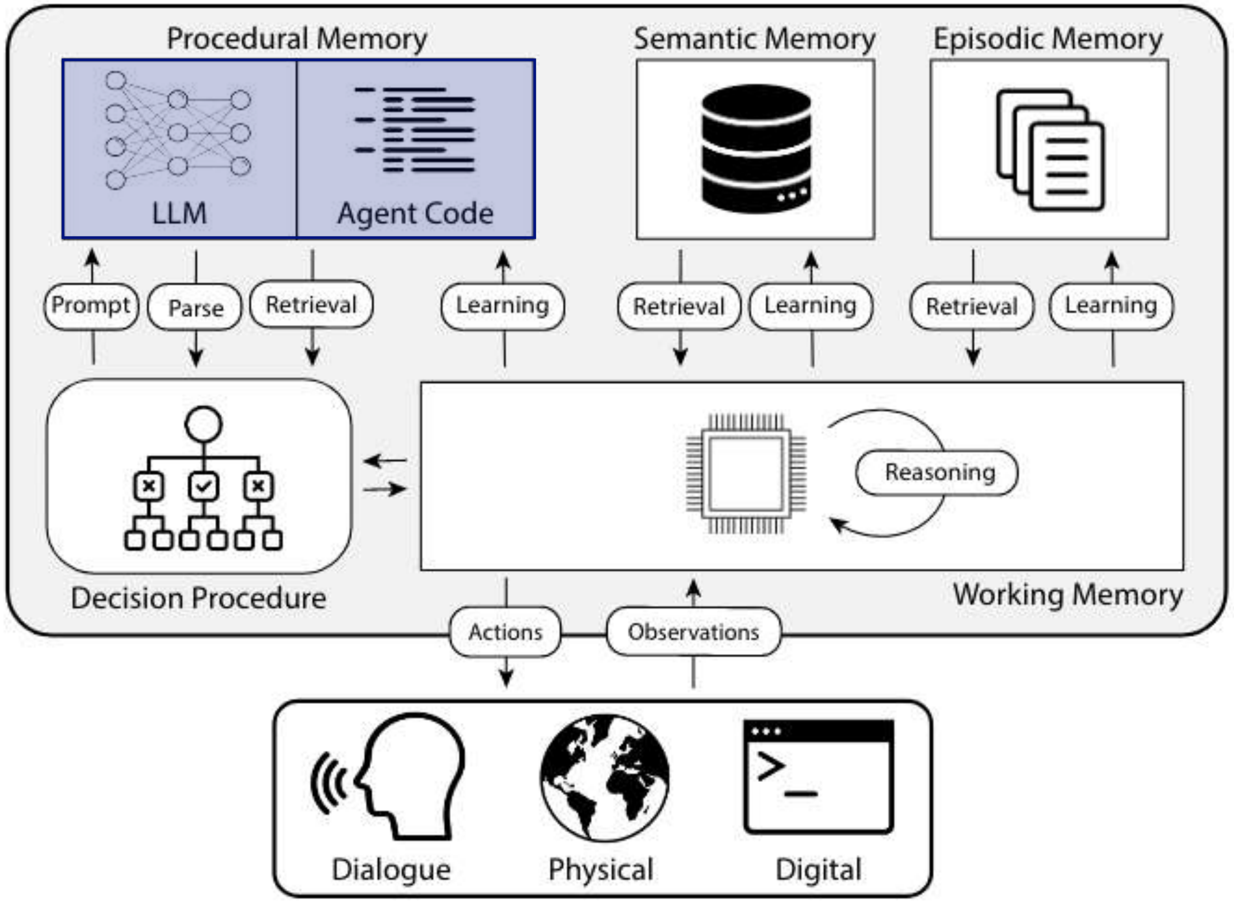


Semantic Memory.

Stores an agent's knowledge about the *world* and *itself*.

Language agents may also write new knowledge obtained from LLM reasoning into semantic memory as a form of learning to incrementally build up world knowledge from experience.

Procedural Memory

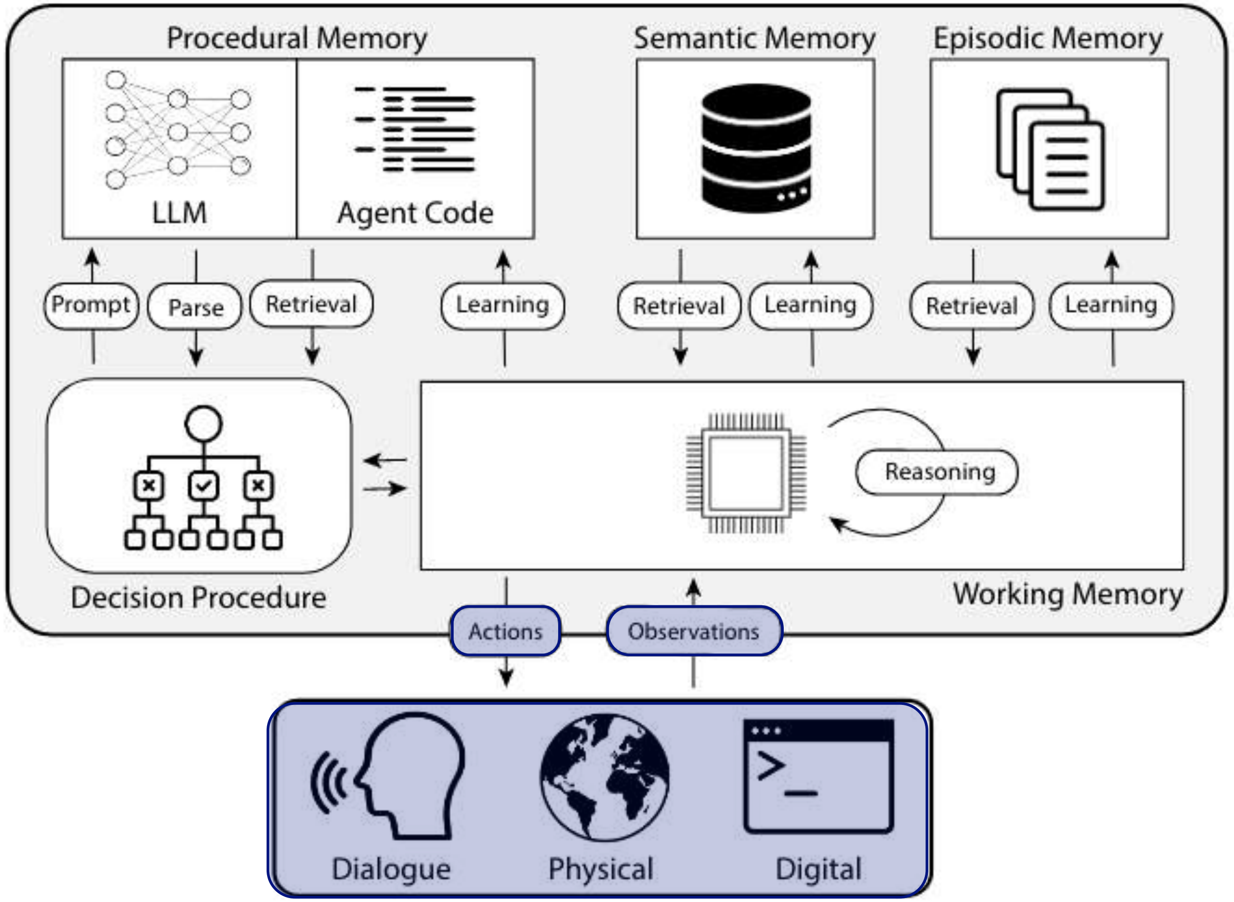


Procedural Memory.

Language agents contain two forms of procedural memory:

- *implicit* knowledge stored in the LLM weights
- *explicit* knowledge written in the agent's code.
  - procedures that implement actions (reasoning, retrieval, grounding, and learning procedures)
  - procedures that implement decision making itself

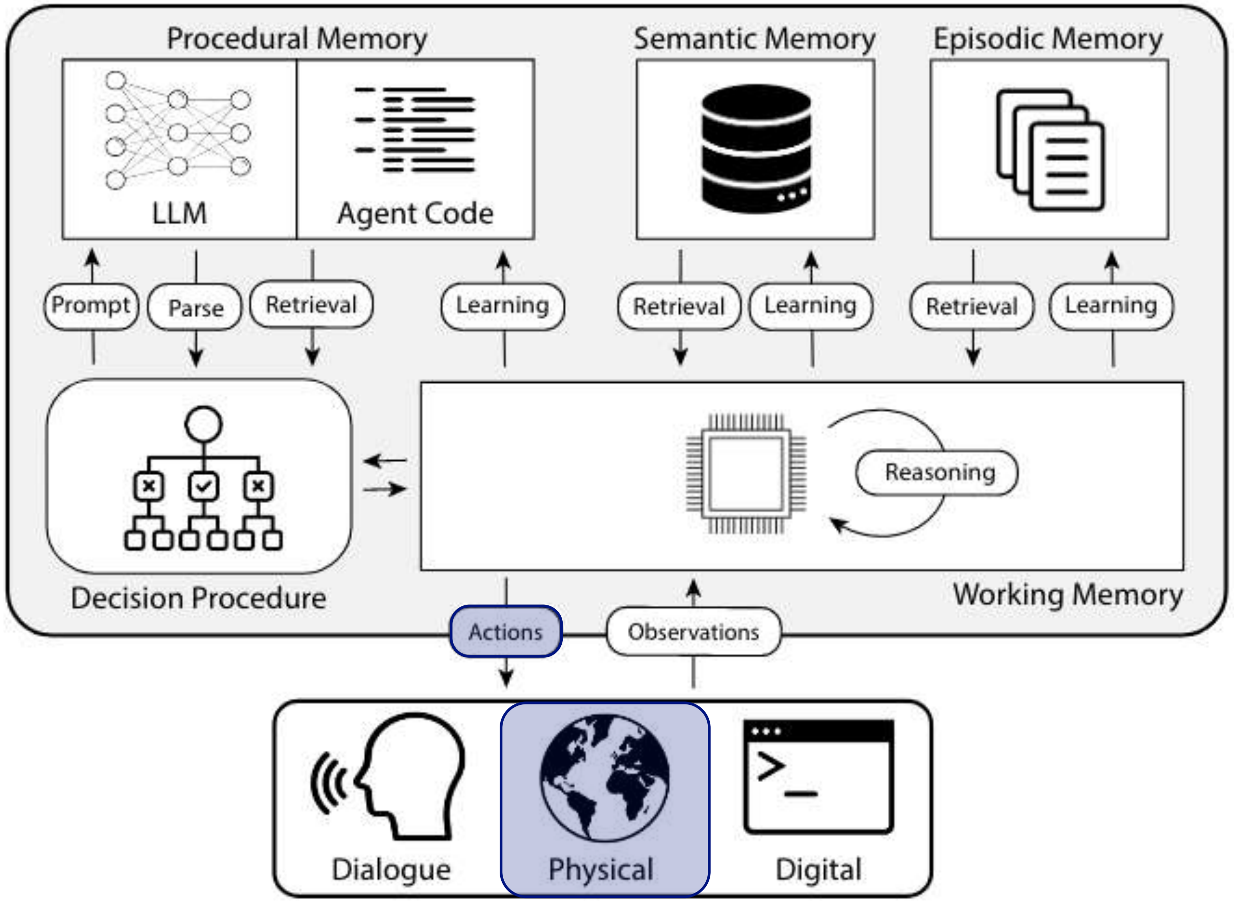
Grounding Actions



Text Game

Grounding procedures execute external actions and process environmental feedback into working memory as text.

Grounding Actions

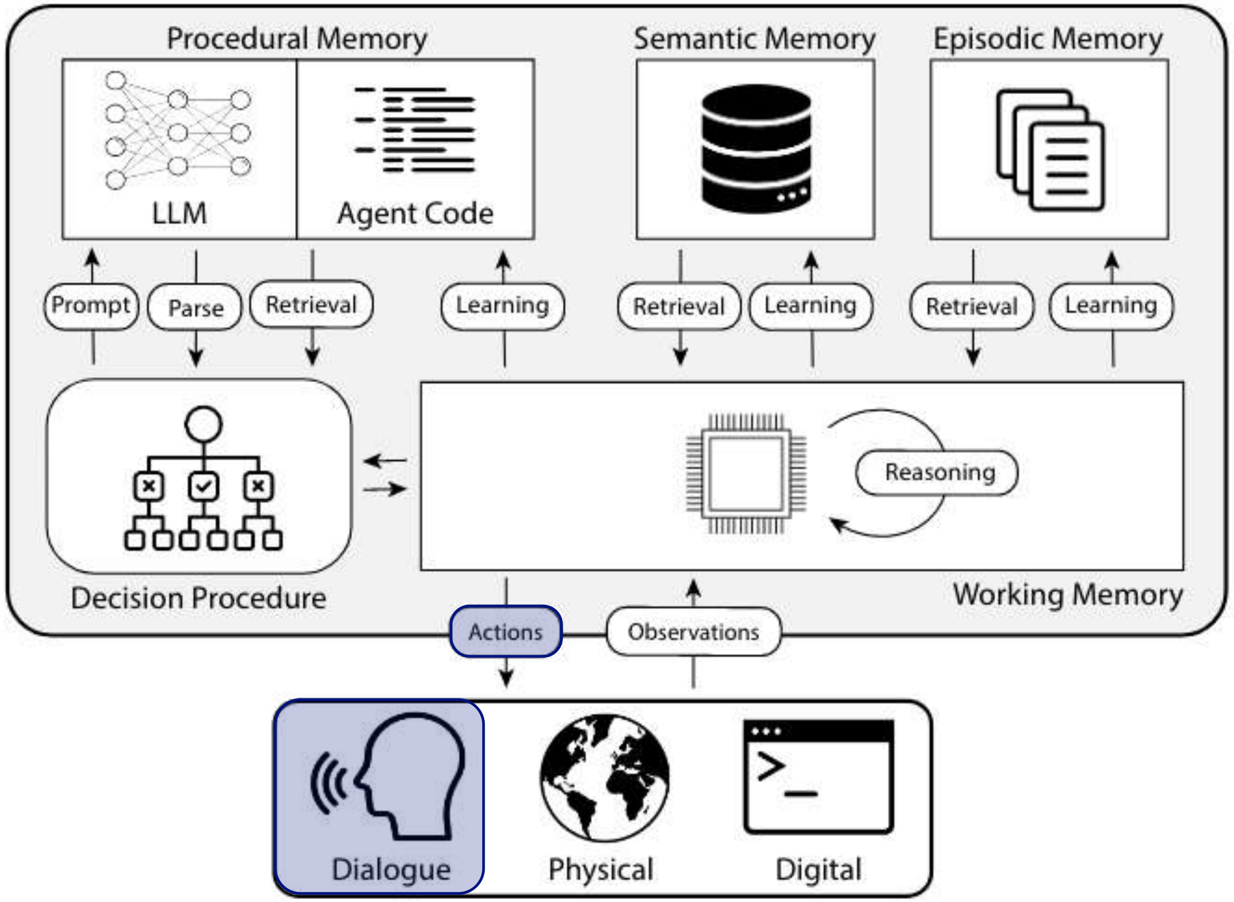


Physical environments

Physical embodiment is the oldest instantiation envisioned for AI agents

It involves processing perceptual inputs (visual, audio, tactile) into textual observations (e.g., via pre-trained captioning models), and affecting the physical environments via robotic planners that take language-based commands.

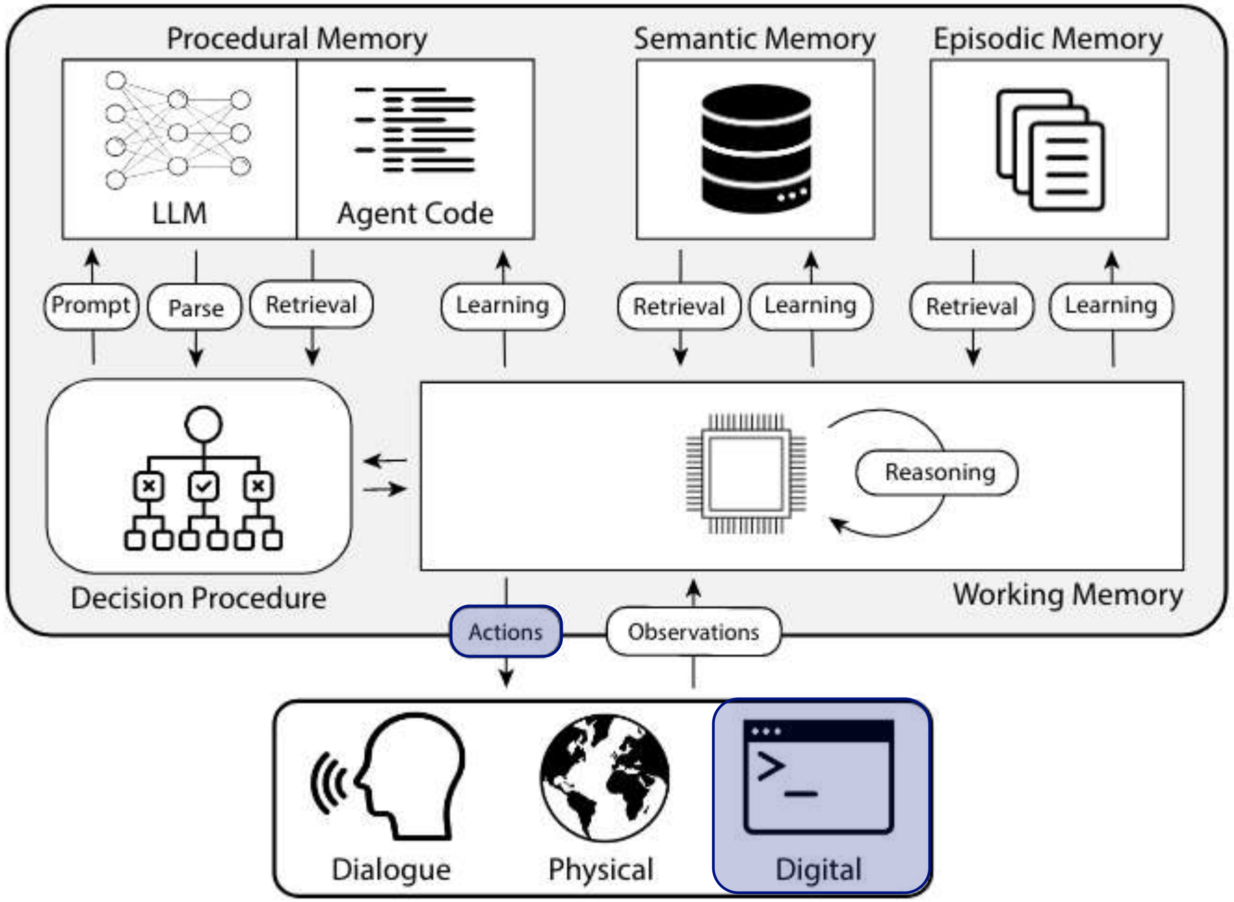
Grounding Actions



Dialogue with humans or other agents

- Interaction among multiple language agents
- Debate
- Collabrative task solving

Grounding Actions

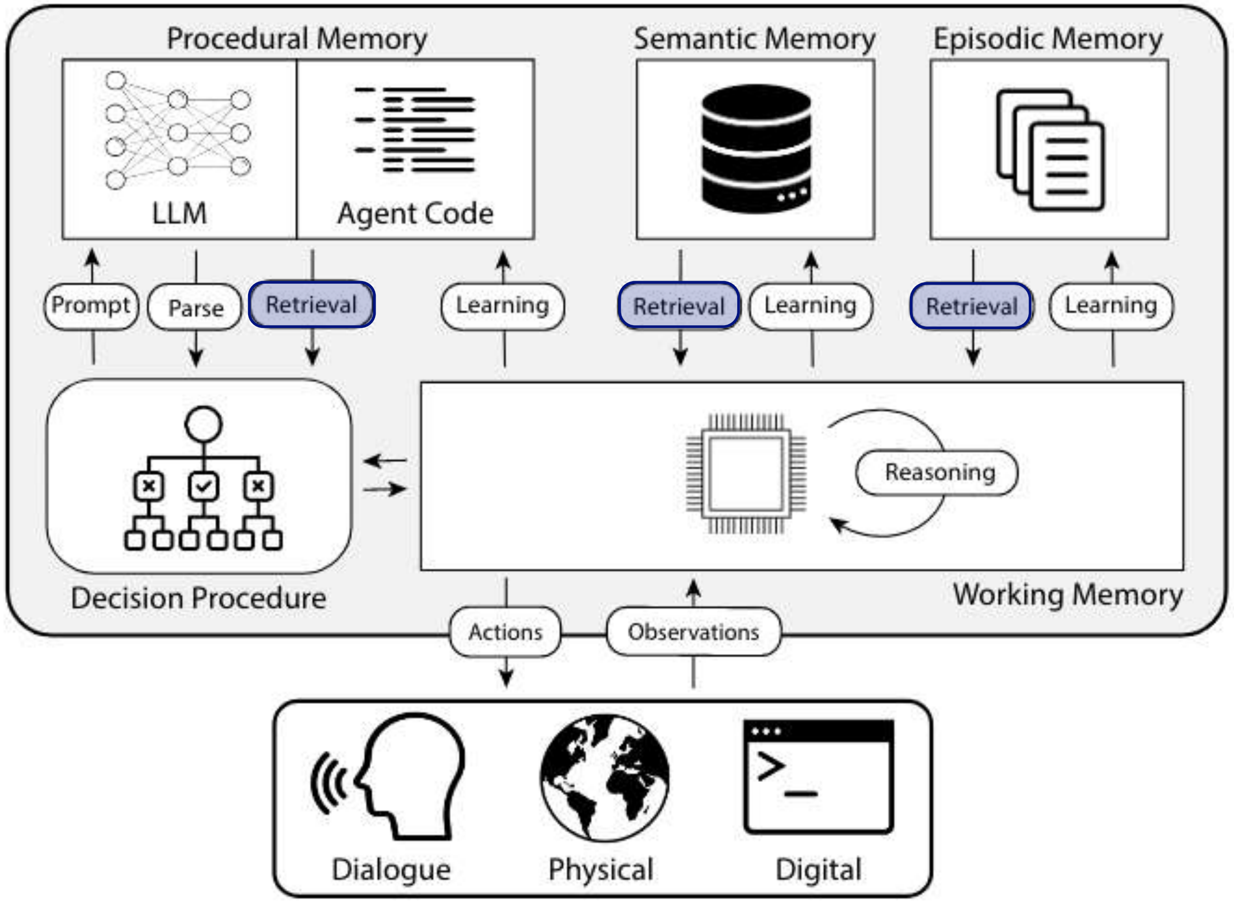


Digital environments

- Interacting with games
- Interacting with APIs
- Interacting with websites
- General code execution

Retrieval Actions

read long-term memory

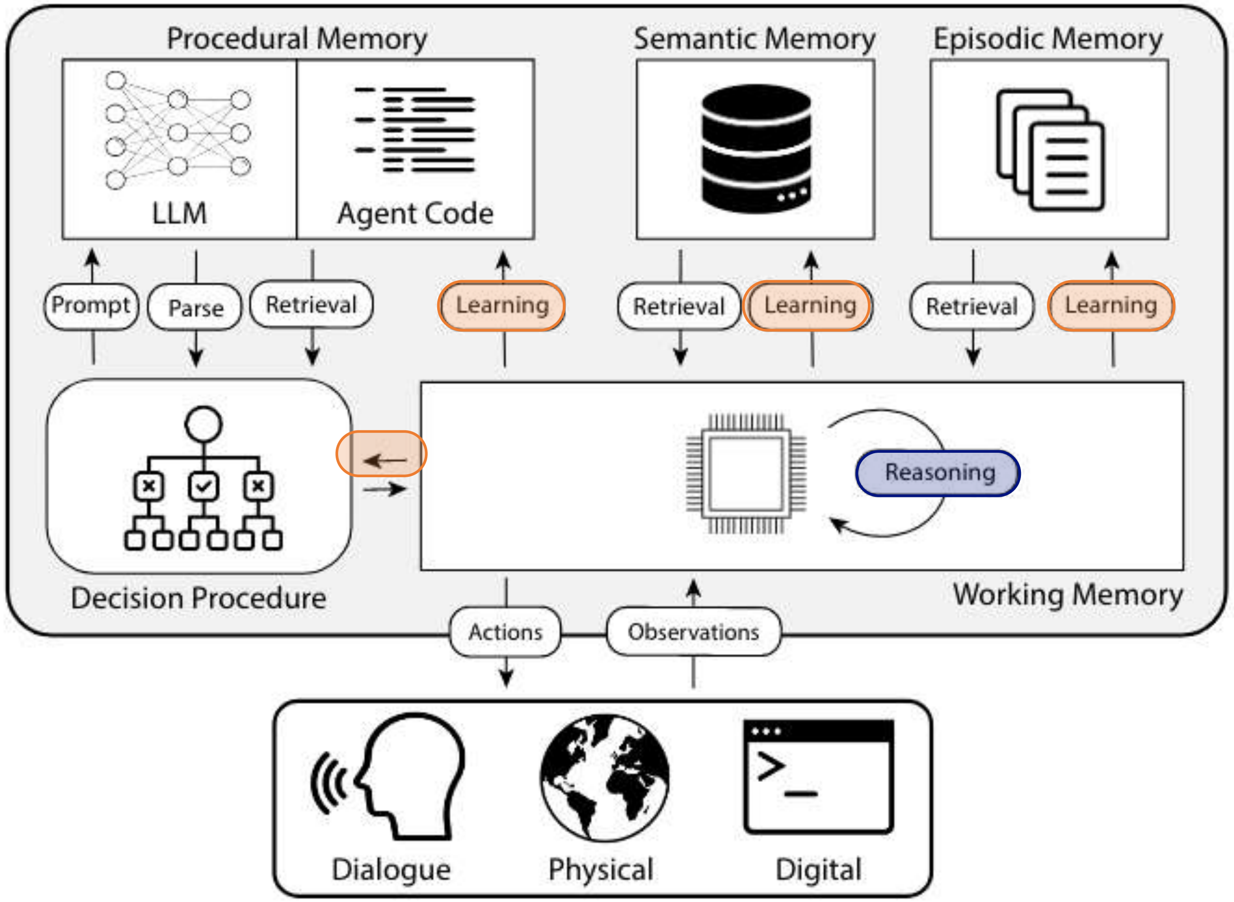


A **retrieval** procedure reads information from long-term memories into working memory

Generative Agents (Park et al., 2023) retrieves relevant events from episodic memory via a combination of recency (rule-based), importance (reasoning-based), and relevance (embedding-based) scores.

Reasoning actions

update working memory



Reasoning allows language agents to process the contents of working memory to generate new information

Reasoning reads from and writes to working memory.

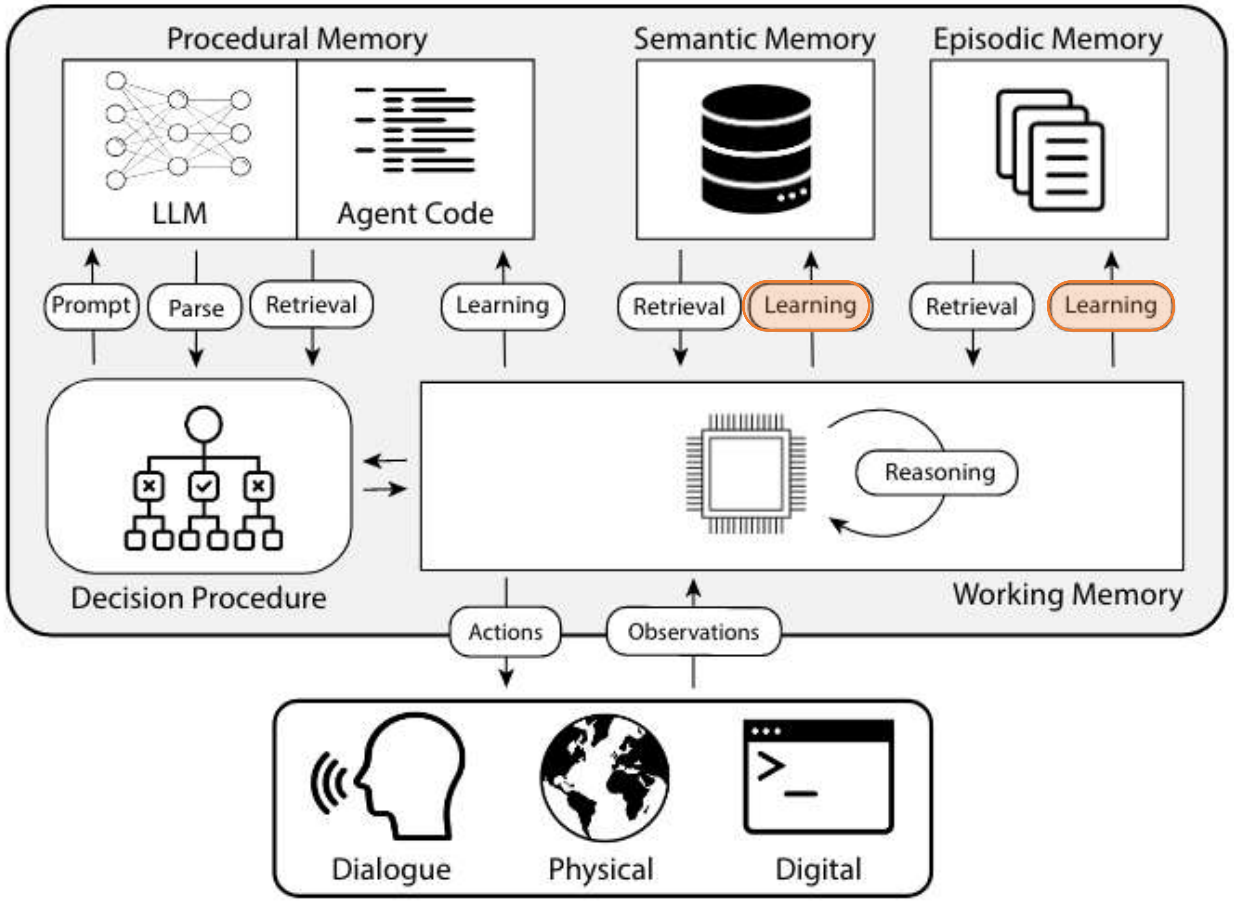
This allows the agent to summarize and distill insights about

- the most recent observation
- the most recent trajectory
- information retrieved from long-term memory

Reasoning can be used to support learning (by writing the results into long-term memory) or decision-making (by using the results as additional context for subsequent LLM calls).

Learning Actions

write long-term memory



Learning occurs by writing information to long-term memory

Updating episodic memory with experience

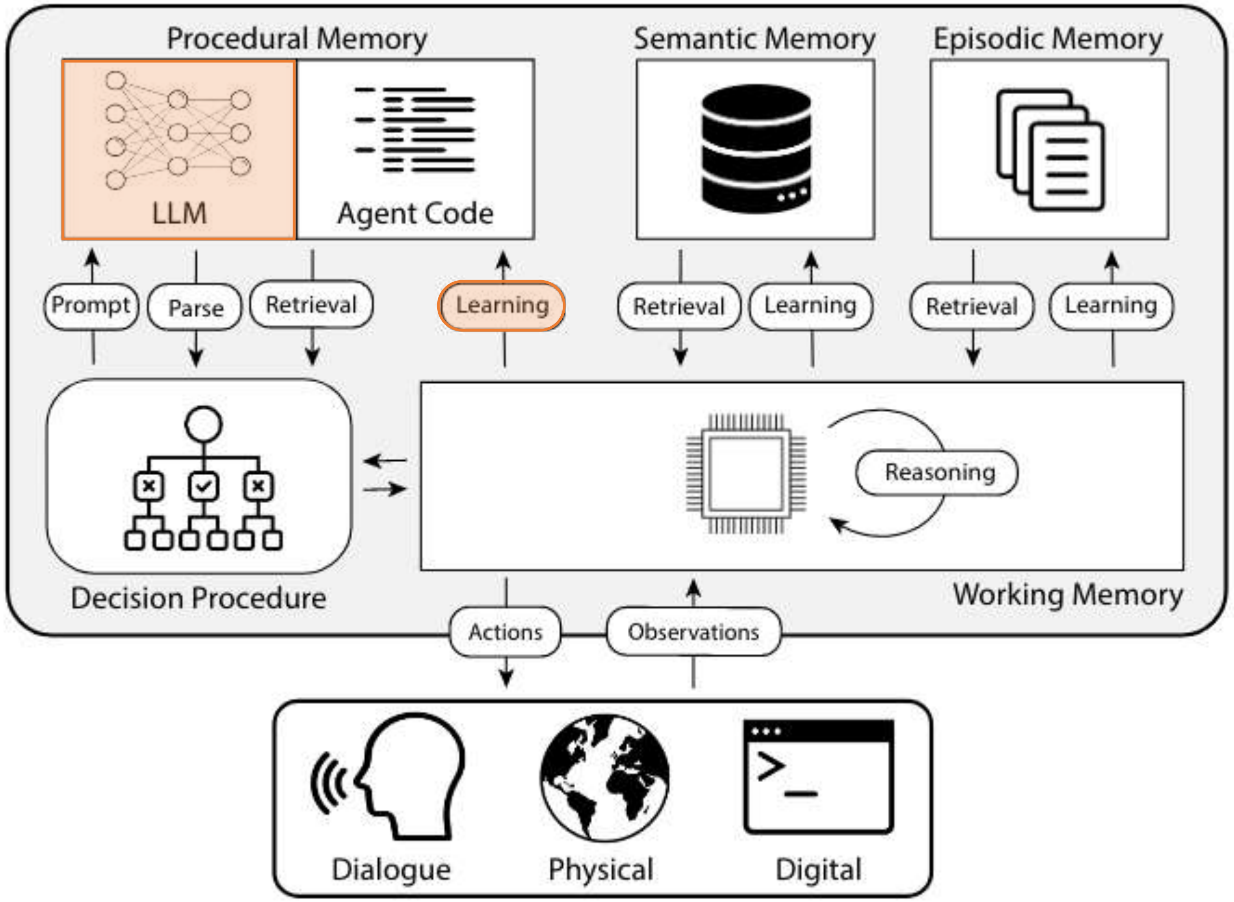
- For language agents, added experiences in episodic memory may be retrieved later as examples and bases for reasoning or decision making

Updating semantic memory with knowledge

- Work in robotics uses vision-language models to build a semantic map of the environment, which can later be queried to execute instructions.

Learning actions

write long-term memory



Learning occurs by writing information to long-term memory

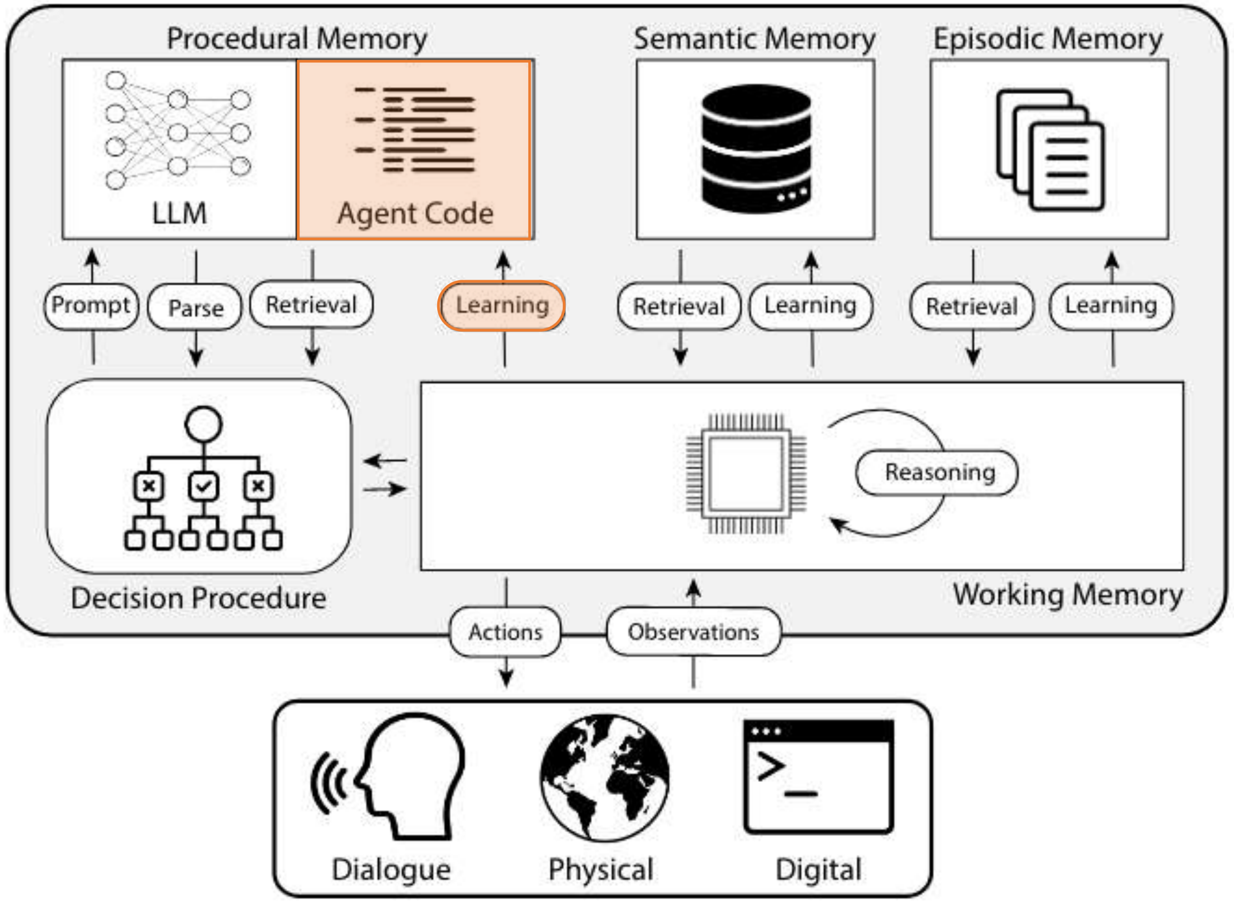
Updating LLM parameters (procedural memory)

The LLM weights represent implicit procedural knowledge.

These can be adjusted to an agent's domain by fine-tuning during the agent's lifetime. Such fine-tuning can be accomplished via supervised or imitation learning (Hussein et al., 2017), reinforcement learning (RL) from environment feedback (Sutton and Barto, 2018), human feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022; Nakano et al., 2021), or AI feedback (Bai et al., 2022).

Learning actions

write long-term memory

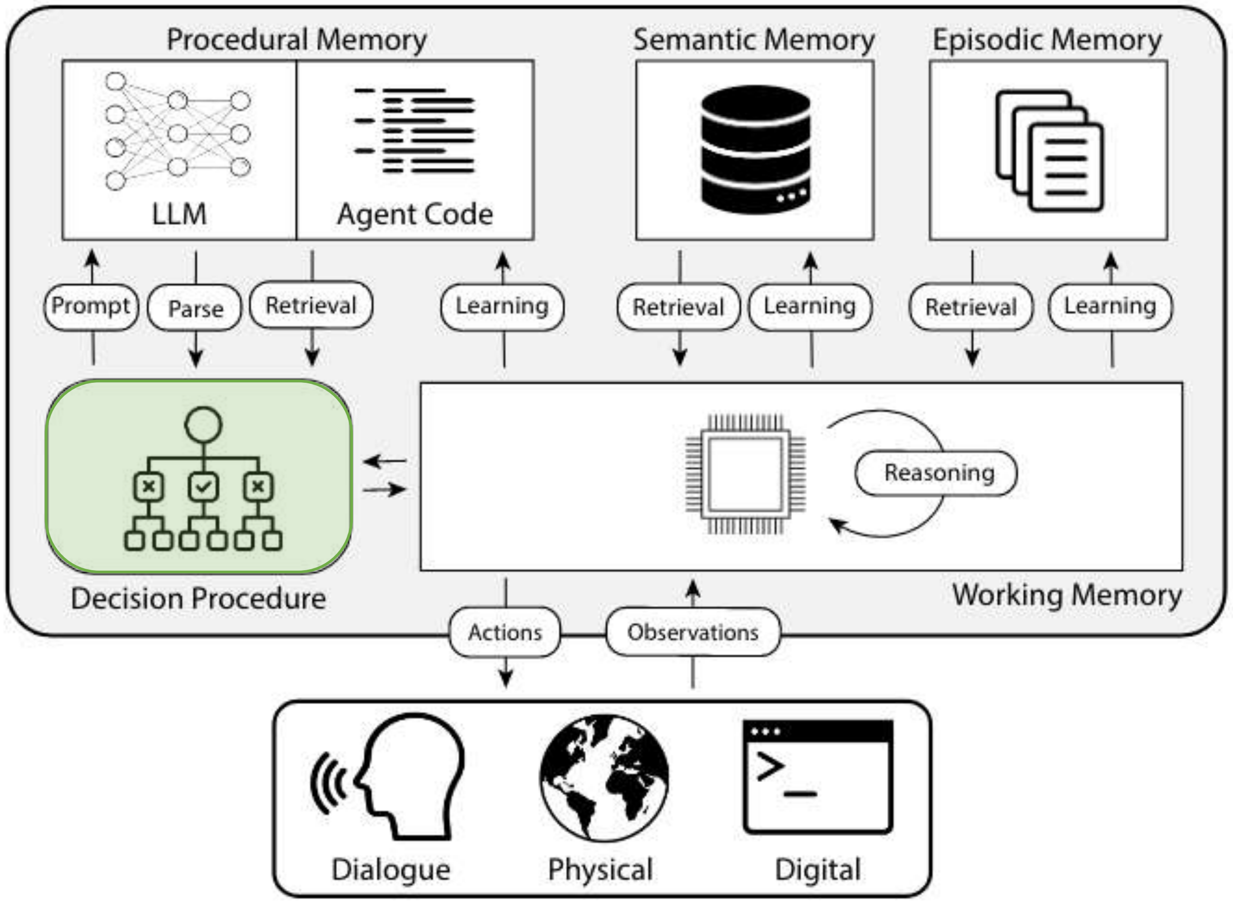


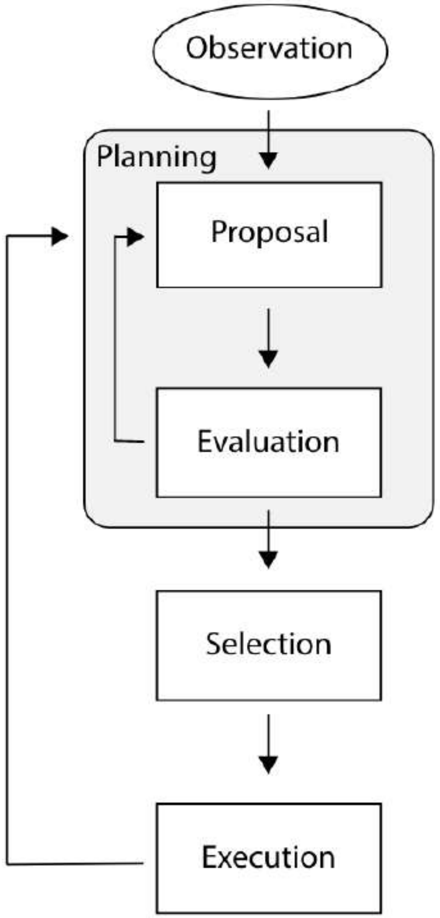
Learning occurs by writing information to long-term memory

Updating agent code (procedural memory).
CoALA allows agents to update their source code.

- Updating reasoning (e.g., prompt templates). Such a prompt update can be seen as a form of learning to reason.
- Updating grounding (e.g., code-based skills) Voyager (Wang et al., 2023a) maintains a curriculum library.
- Updating retrieval.
- Updating learning or decision-making. (updates to these procedures are risky both for the agent's functionality and alignment.)

Decision making





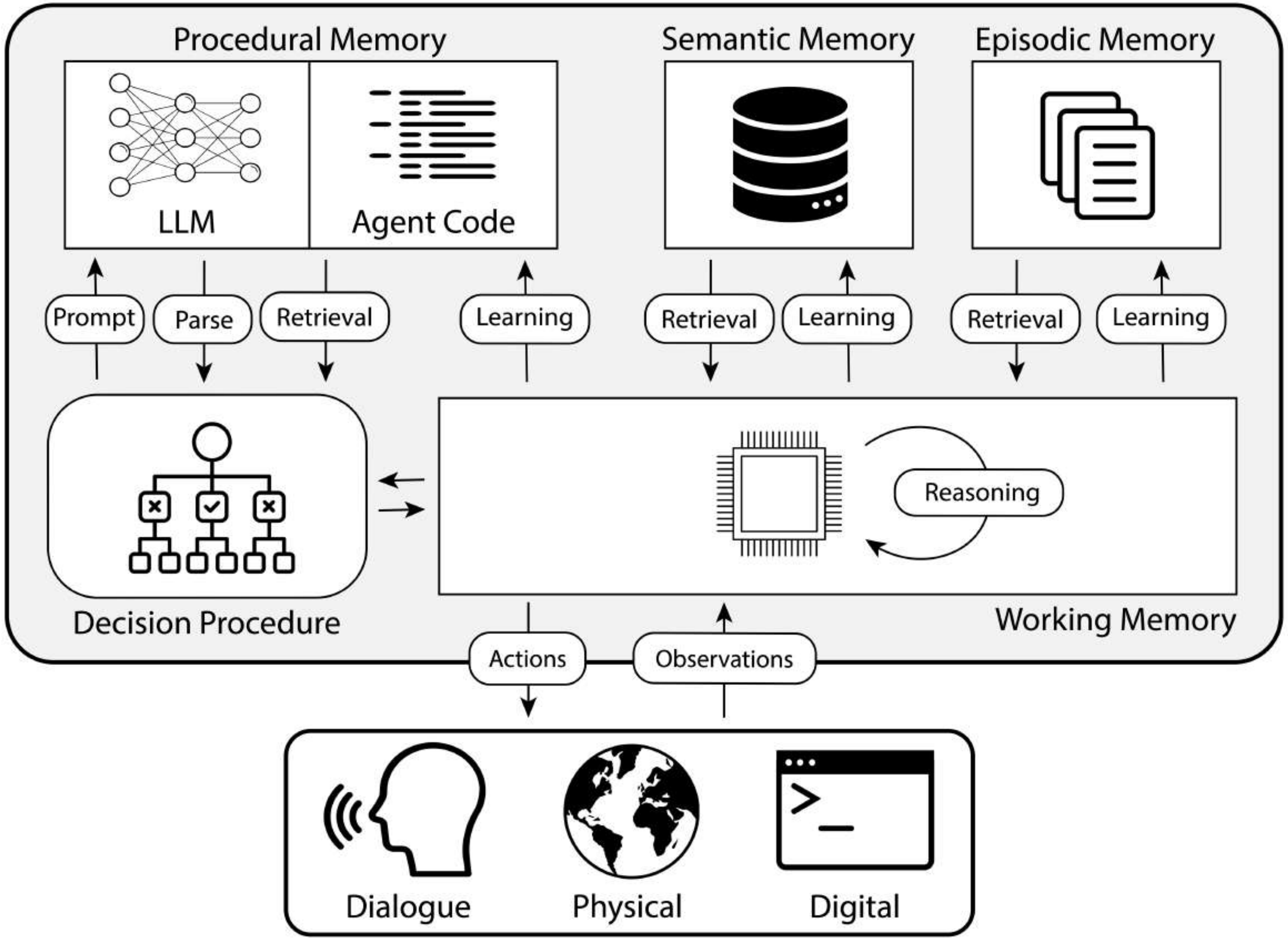
Actionable Insights

- Working memory and reasoning: thinking beyond LLM prompt engineering.
  - The community should think about a structured working memory and systematic "reasoning" actions that update working memory variables.
- Long-term memory: thinking beyond retrieval augmentation.
  - By organically combining existing human knowledge with self-discovered and self-maintained experience, knowledge, and skills in long-term memory, future language agents may more efficiently learn and solve tasks.
- Learning: thinking beyond in-context learning or finetuning.
  - future directions could explore learning smaller models for specific reasoning needs, deleting unneeded memory items for "unlearning", and various ways to combine multiple forms of learning.
- Action space: thinking beyond external tools or actions
  - clear and task-suitable action space, agent safety
- Decision making: thinking beyond action generation.
  - extend such schemes to more complicated tasks, LLM development might be influenced or even shaped by the increased usage of reasoning toward complex decision making

Discussion

- Internal vs. external actions: what is the boundary between an agent and its environment?
  - is a Wikipedia database an internal semantic memory or an external digital environment?
  - Wikipedia is an external environment if constantly modified by other users, but an offline version that only the agent may write to can be considered an internal memory.
- Planning vs. execution: how much should agents plan?
  - Future work should develop mechanisms to estimate the utility of planning and modify the decision procedure accordingly
- Learning vs. acting: how should agents continuously and autonomously learn?
  - Learning could be proposed as a possible action during regular decision-making
- LLMs vs. code: where should agents rely on each?
  - CoALA thus suggests that good design uses agent code primarily to implement classic, generic planning algorithms – and relies heavily on the LLM for action proposal and evaluation.

Conclusion



Procedural Memory
LLM
Agent Code
Semantic Memory
Episodic Memory
Prompt
Parse
Retrieval
Learning
Retrieval
Learning
Retrieval
Learning
Reasoning
Decision Procedure
Working Memory
Actions
Observations
Dialogue
Physical
Digital