Session5: Agentic Memory

[arXiv'25] Mem^p: Exploring Agent Procedural Memory

[SAA'25] Supporting Our AI Overloads: Redesigning Data Systems to be

[EMNLP'25] Memory OS of AI Agent

Presenter: Yitao Wang, Fanglei Shu, Haora Ma, Zhaonian Wang, Zikang Chen (DLM)

2025.9.20

Mem^p: Exploring Agent Procedural Memory

arXiv'25

Authors: Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, Ningyu Zhang

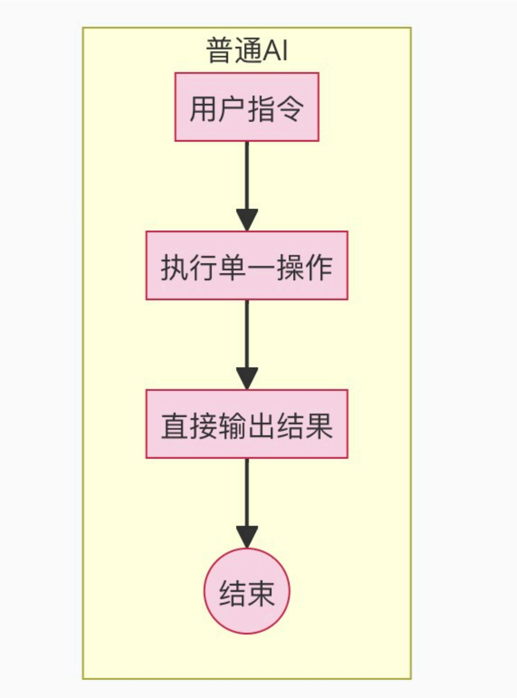
Presenter: Yitao Wang

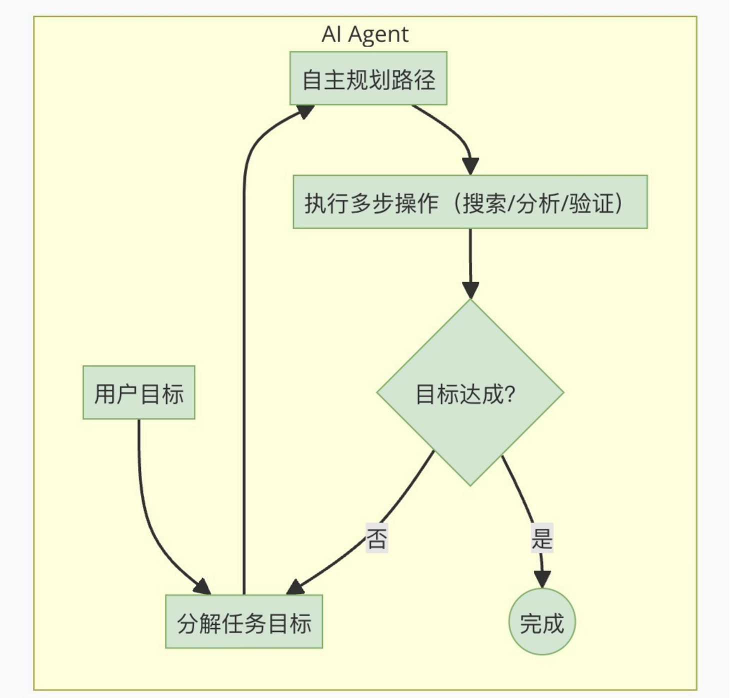
2025.9.20

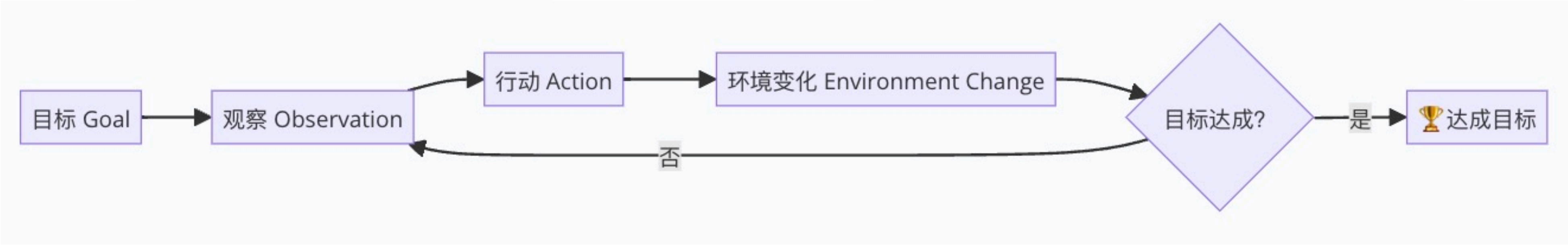
Contents

Background

Agent Workflow

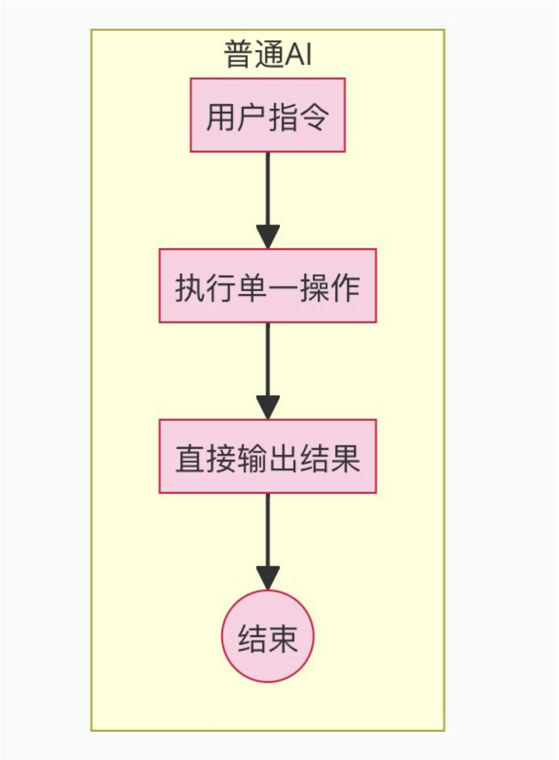


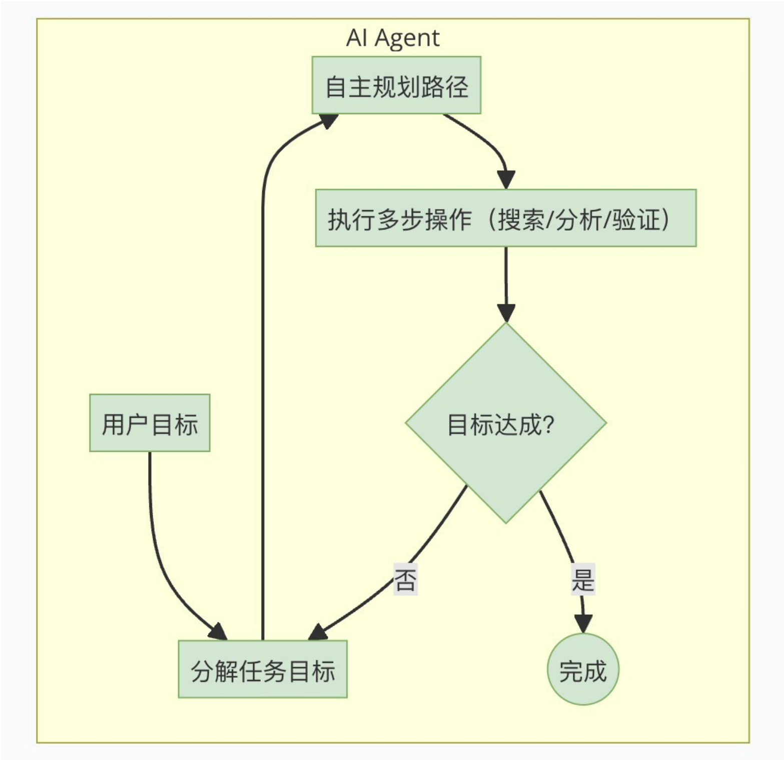


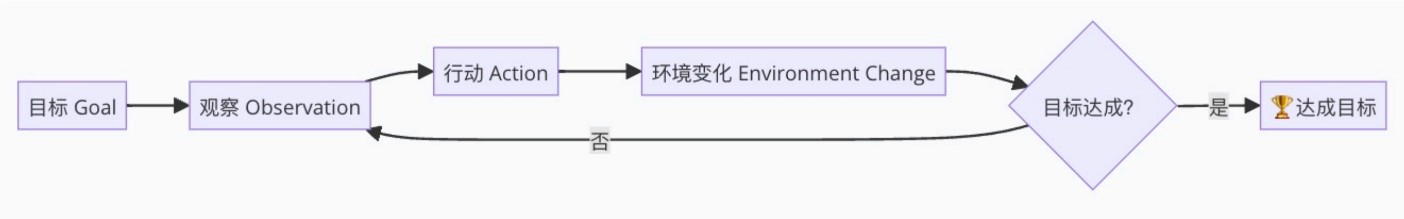


Background

Agent Workflow



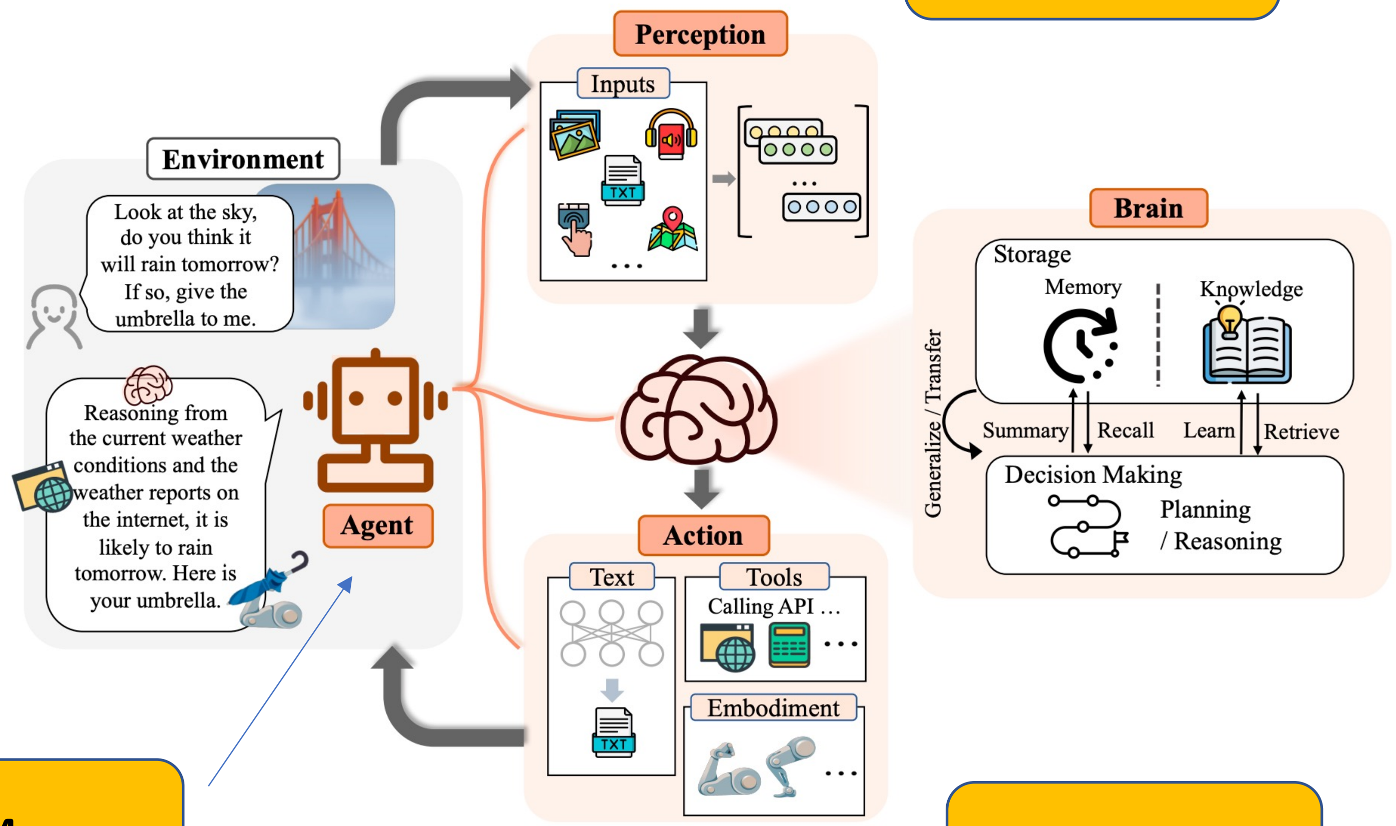




Background

Agent

Tools



Action/CoT

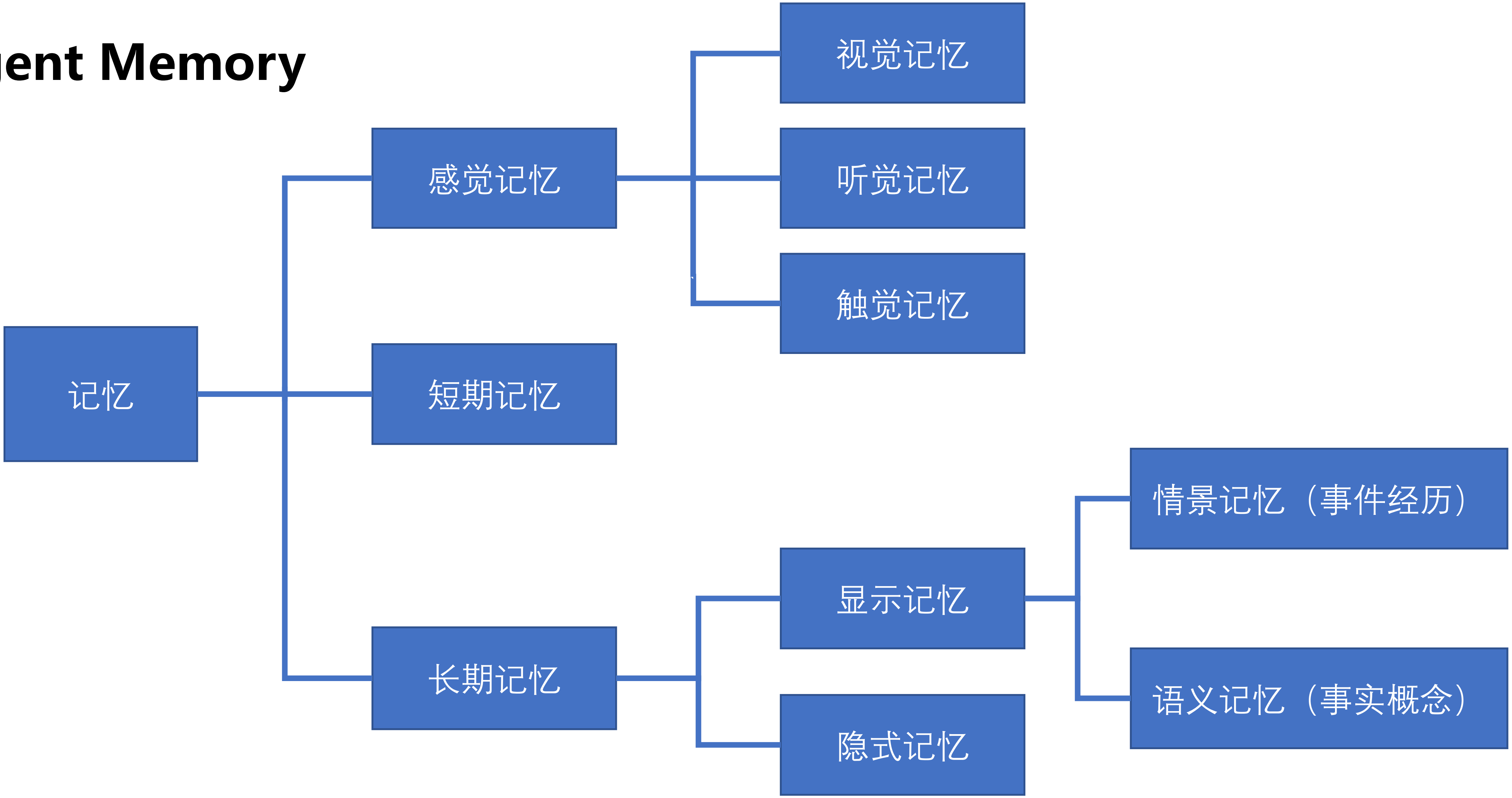
Memory

LLM

Tools

Background

Agent Memory



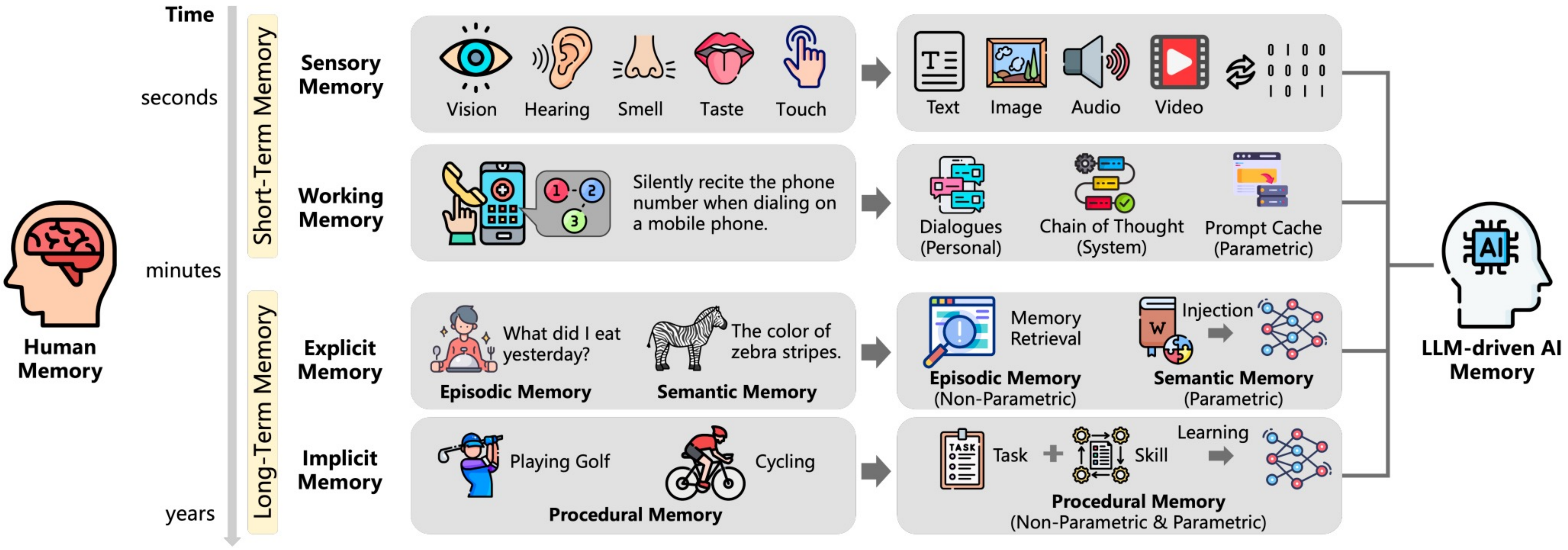
Background

Agent Memory

记忆类型	映射	例子
感觉记忆	学习原始输入的嵌入式表示，包括文本、图像或其他形式，短暂保留感觉印象。	看一张图片，然后在图片消失后能够在脑海中回想起它的视觉印象。
短期记忆	上下文学习（比如直接写入 prompt 中的信息），处理复杂任务的临时存储空间，受有限的上下文长度限制。	在进行心算时记住几个数字，但短期记忆是有限的，只能暂时保持几个项目。
长期记忆	在查询时 Agent 可以关注的外部向量存储，具有快速检索和基本无限的存储容量。	学会骑自行车后，多年后再次骑起来时仍能掌握这项技能，这要归功于长期记忆的持久存储

Background

Agent Memory



From Human Memory to AI Memory: A Survey on Memory Mechanisms in the Era of LLMs, arXiv’25

Background

Agent Memory

Timescale	Consciousness	Mechanism	Example References
Short-term	Explicit	Prompt-Based Context	GPT-2 [22], GPT-3 [23], Prefix-Tuning [24], Prompt-Tuning [25], P-Tuning [26, 27], InstructGPT [28]
	Implicit	Key-Value Cache Mechanism	vLLM [29], StreamingLLM[30], H2O[31], LESS [32], KVQuant [33], RetrievalAttention [34], Memory^3 [1]
		Hidden State Steering	Steer [35], ICV [36], ActAdd [37], StyleVec [38], CAA [39], FreeCtrl [40], EasyEdit2 [41]
		Activation Circuit Modulation	SAC [42], DESTEIN [43], LM-Steer [44]
Long-term	Explicit	Non-parametric Retrieval-Augmented Generation	kNN-LMs [45, 46], MEMWALKER [9], Graph RAG [10], LightRAG [11], NodeRAG [47, 48], HyperGraphRAG [49], HippoRAG [50, 51], PGRAG[52], Zep [53], A-MEM [54], Mem0[55]
	Implicit	Parametric Knowledge	BERT [56], RLHF [57], CTRL [58], SLayer [59]
		Modular Parameter Adaptation	LoRA [60], PRAG [61], DyPRAG [62], SERAC [63], CaliNet [64], DPM [65], GRACE [66]
		Parametric Memory Editing	ROME [67], MEMIT [68], AlphaEdit [69], AnyEdit [70], EasyEdit [71], AdaPLE [72], MEMAT [73]

Background

Agent Memory — Implementation

编码：获取和处理信息，将其转化为可以存储的形式。

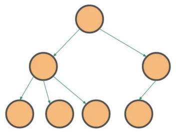
0	0	0	0	1
1	0	0	0	0
0	0	0	1	0

…

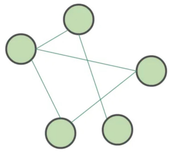
存储：在短期记忆或长期记忆中保留编码信息的过程。

0	0	0	0	1
1	0	0	0	0
0	0	0	1	0

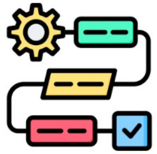
…



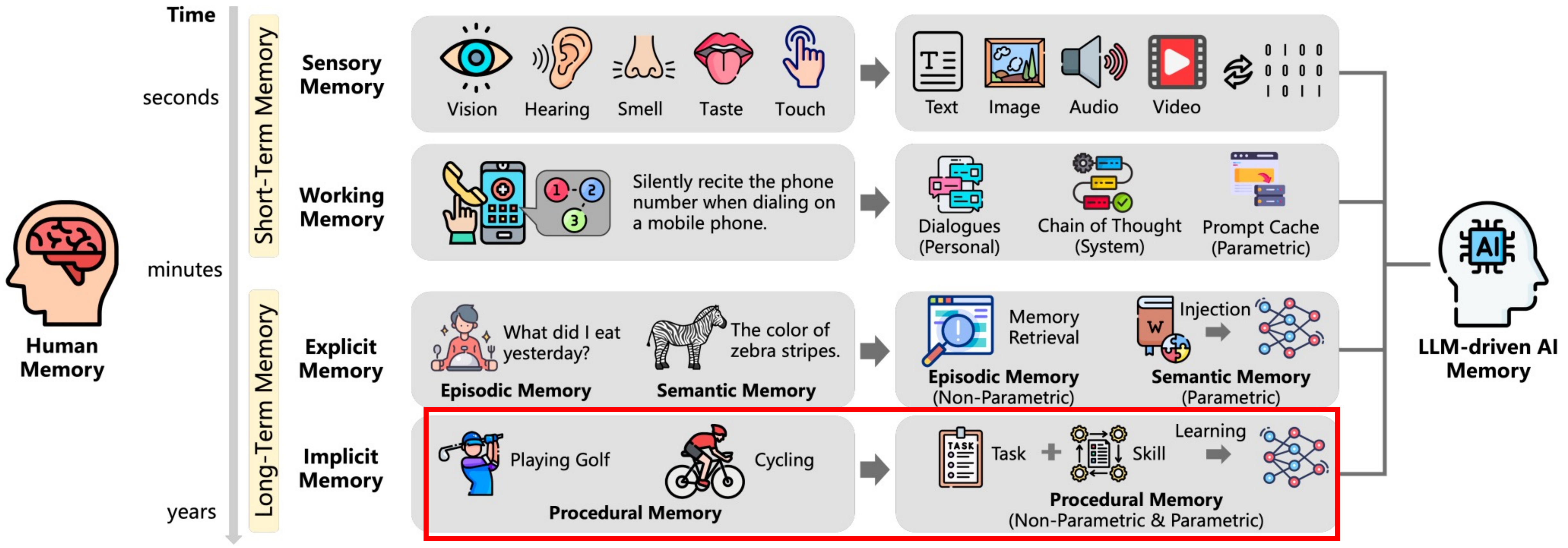
&



检索：也可称为回忆，即在需要时访问并使存储的信息重新进入意识的过程。

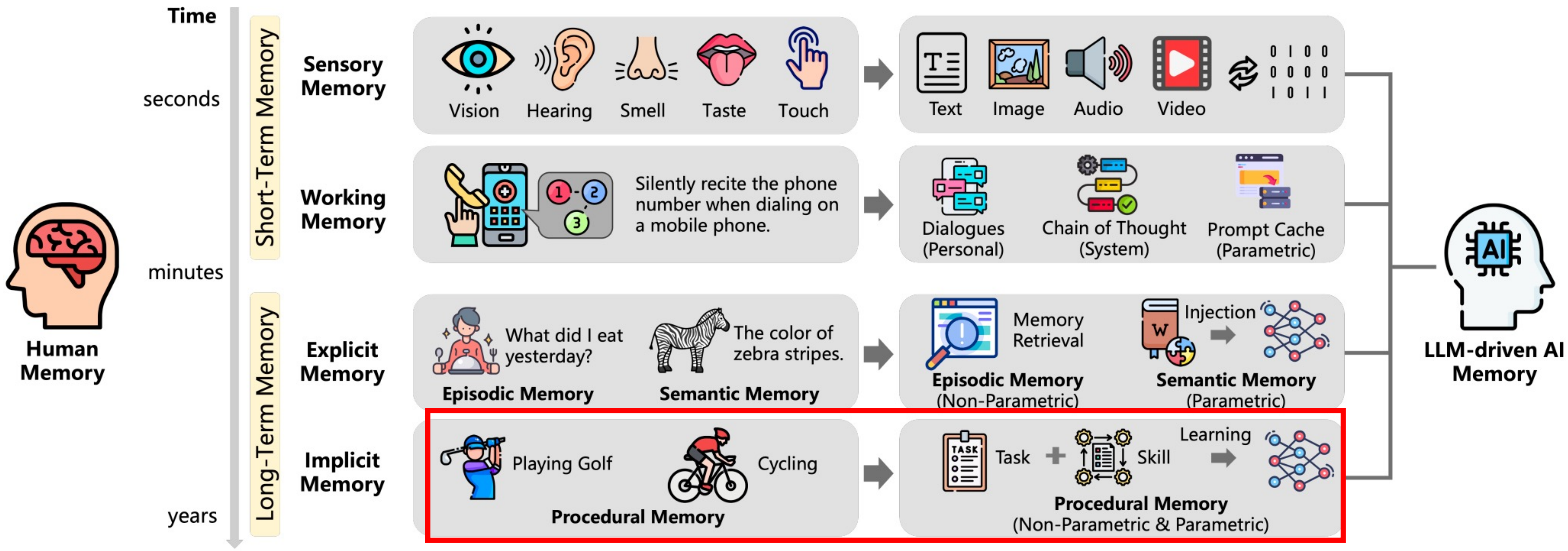


Background

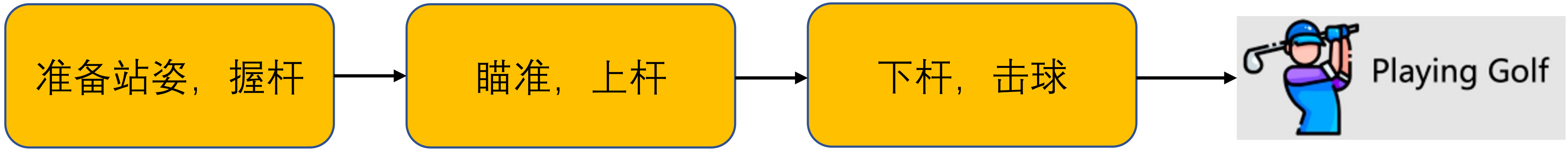


Time
seconds
minutes
years
Short-Term Memory
Long-Term Memory
Human Memory
Sensory Memory
Vision
Hearing
Smell
Taste
Touch
Text
Image
Audio
Video
Working Memory
Silently recite the phone number when dialing on a mobile phone.
Dialogues (Personal)
Chain of Thought (System)
Prompt Cache (Parametric)
Explicit Memory
What did I eat yesterday?
Episodic Memory
The color of zebra stripes.
Semantic Memory
Memory Retrieval
Episodic Memory (Non-Parametric)
Injection
Semantic Memory (Parametric)
Implicit Memory
Playing Golf
Cycling
Procedural Memory
Task
Skill
Learning
Procedural Memory (Non-Parametric & Parametric)
LLM-driven AI Memory

Background



Time
seconds
minutes
years
Short-Term Memory
Long-Term Memory
Human Memory
Sensory Memory
Vision
Hearing
Smell
Taste
Touch
Text
Image
Audio
Video
0 1 0 0
0 0 0 0
1 0 1 1
Working Memory
Silently recite the phone number when dialing on a mobile phone.
Dialogues (Personal)
Chain of Thought (System)
Prompt Cache (Parametric)
Explicit Memory
What did I eat yesterday?
The color of zebra stripes.
Episodic Memory
Semantic Memory
Memory Retrieval
Injection
Episodic Memory (Non-Parametric)
Semantic Memory (Parametric)
Implicit Memory
Playing Golf
Cycling
Procedural Memory
TASK
Task
Skill
Learning
Procedural Memory (Non-Parametric & Parametric)
LLM-driven AI Memory
AI



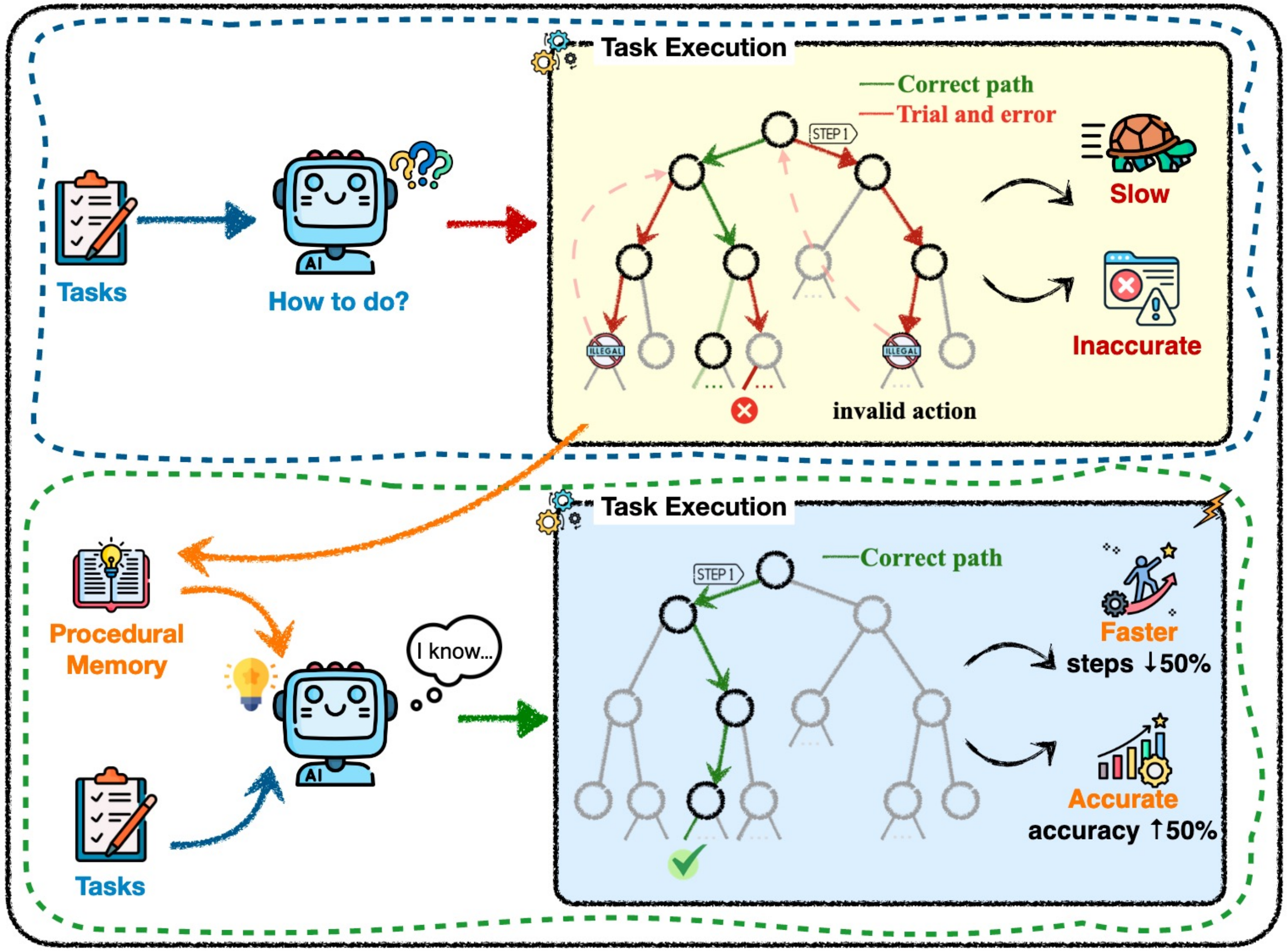
准备站姿，握杆
瞄准，上杆
下杆，击球
Playing Golf

Contents

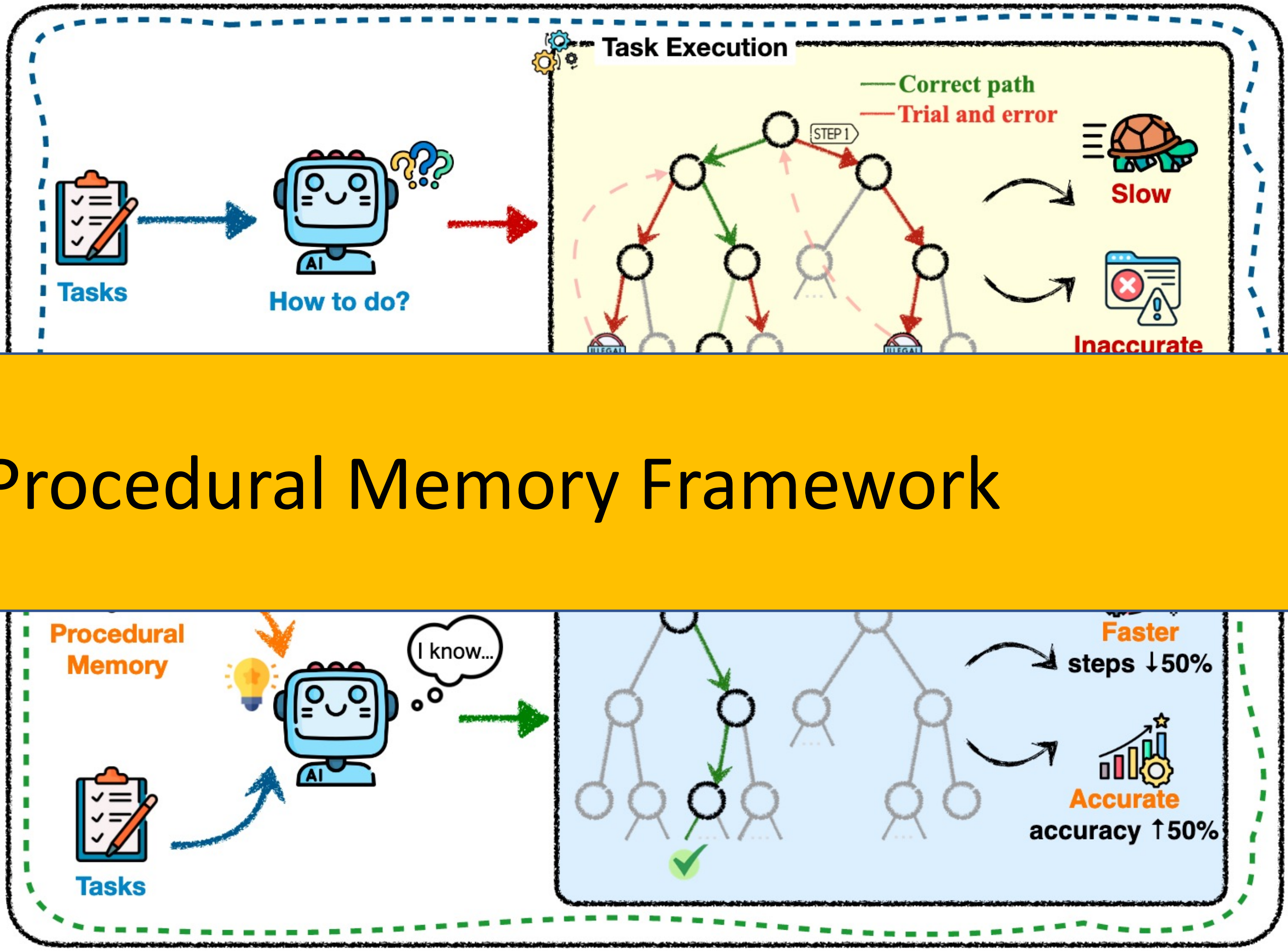
- Background
- Motivation
- Method
- Evaluation
- Thinking

Motivation

现有的 Agent 的程序性记忆非常薄弱。
即使有着很强大的语义记忆和情节记忆，
每次执行程序流时也还是需要从头开始
一边探索（试错）一边执行（低效）



Motivation



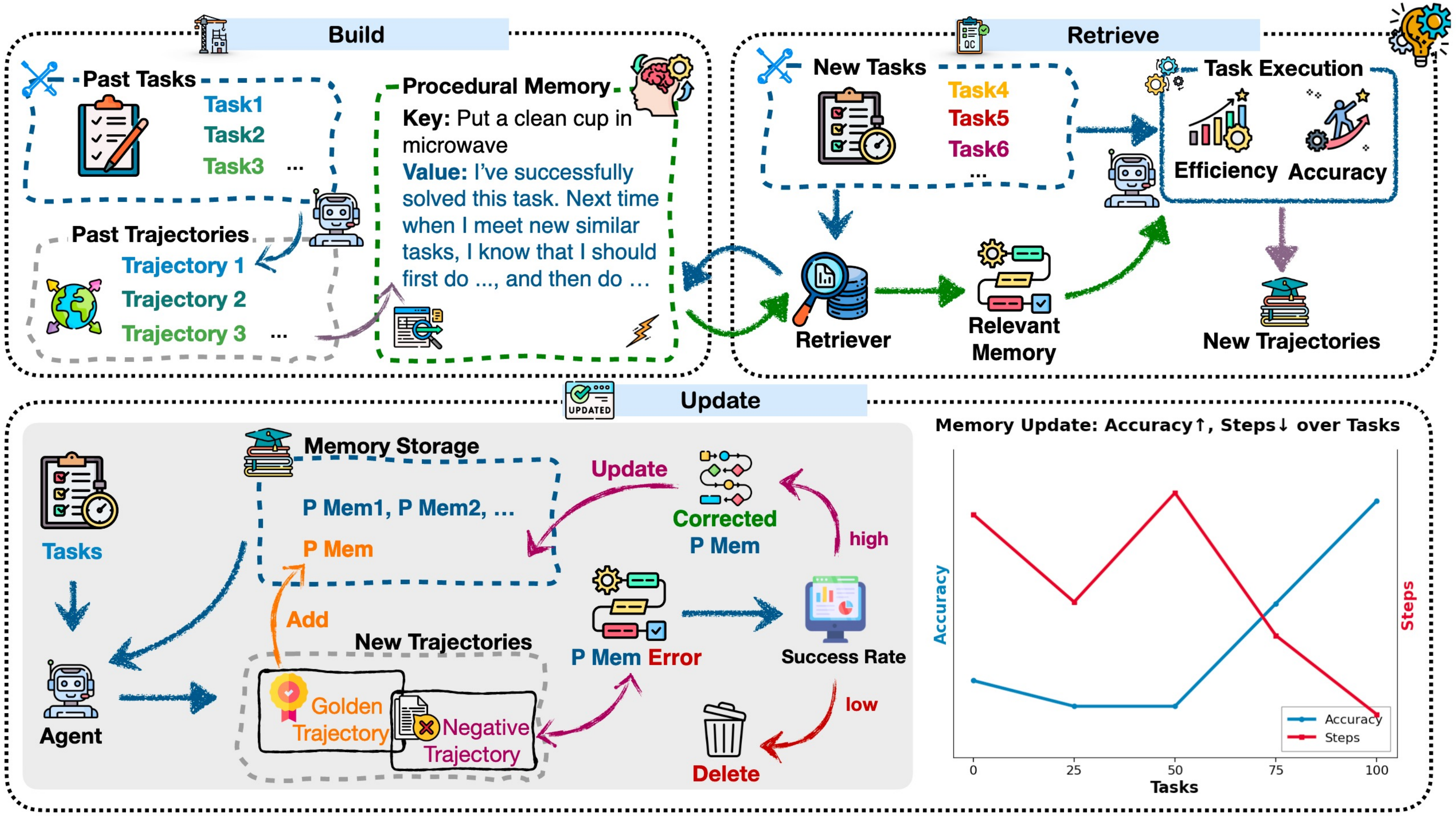
Mem^p — An Agent Procedural Memory Framework

Contents

- Background
- Motivation
- **Method**
- Evaluation
- Thinking

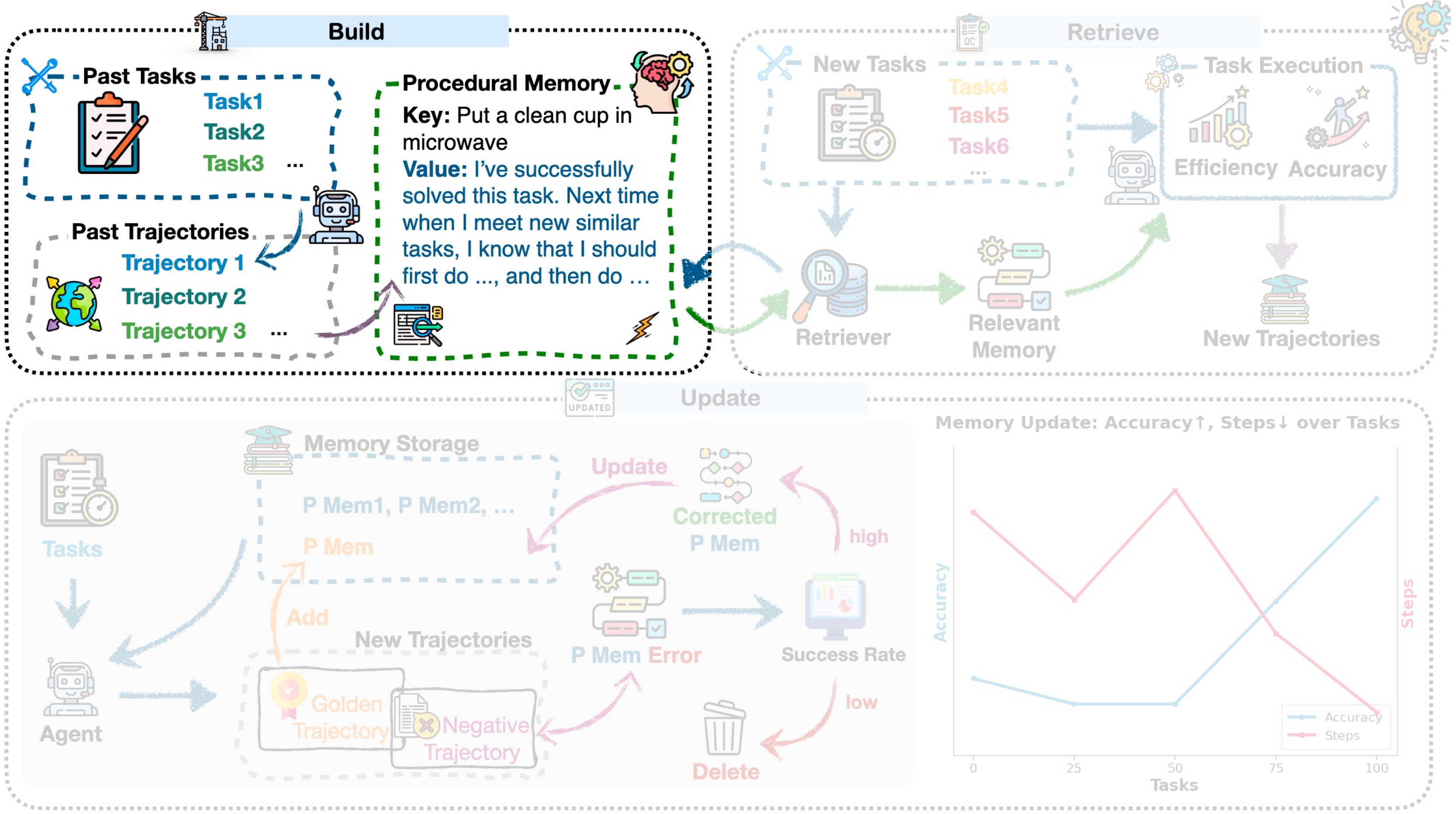
Method

Agent Memory



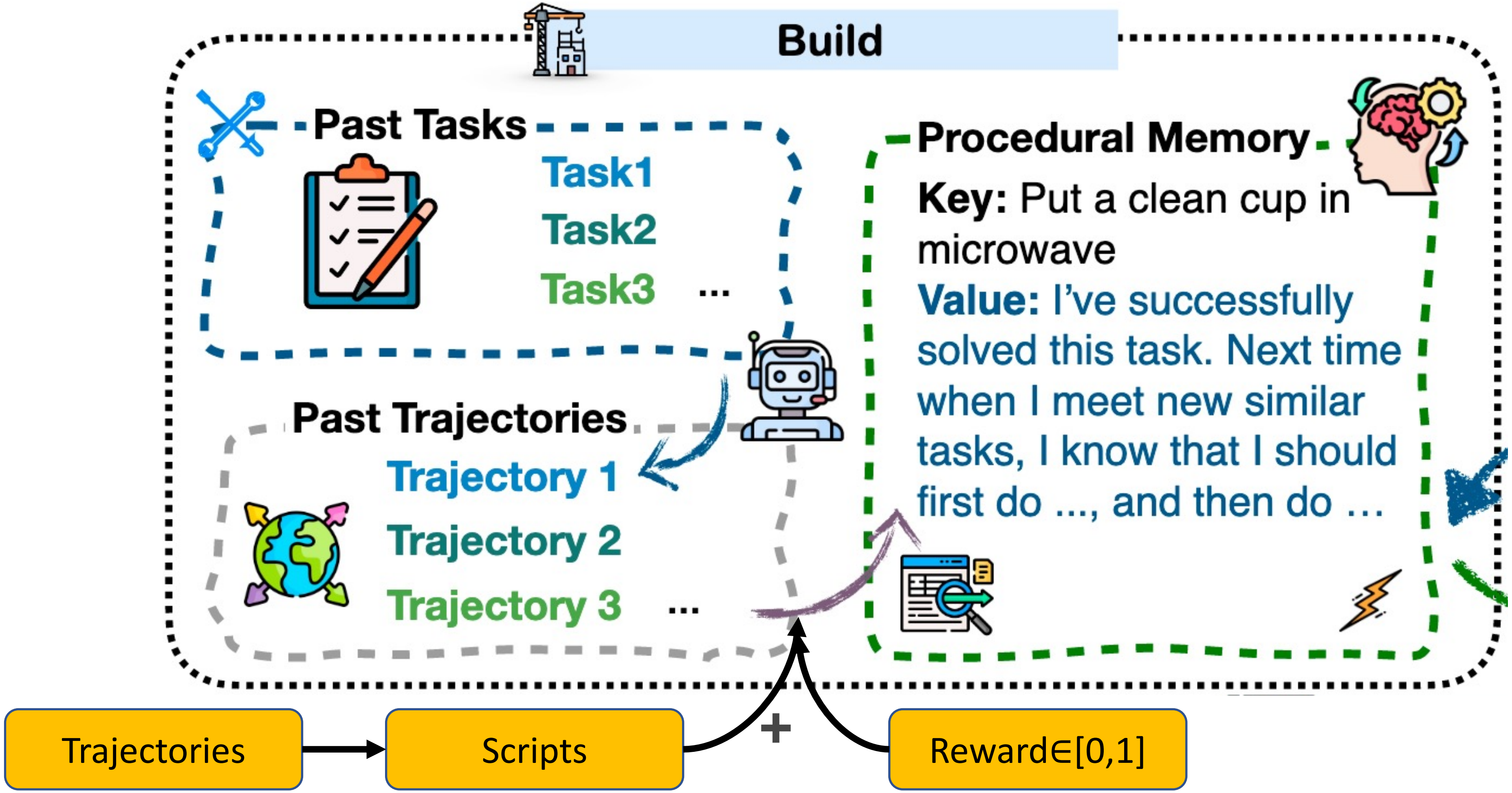
Method

Agent Memory



Method

Agent Memory



Method

Agent Trajectories

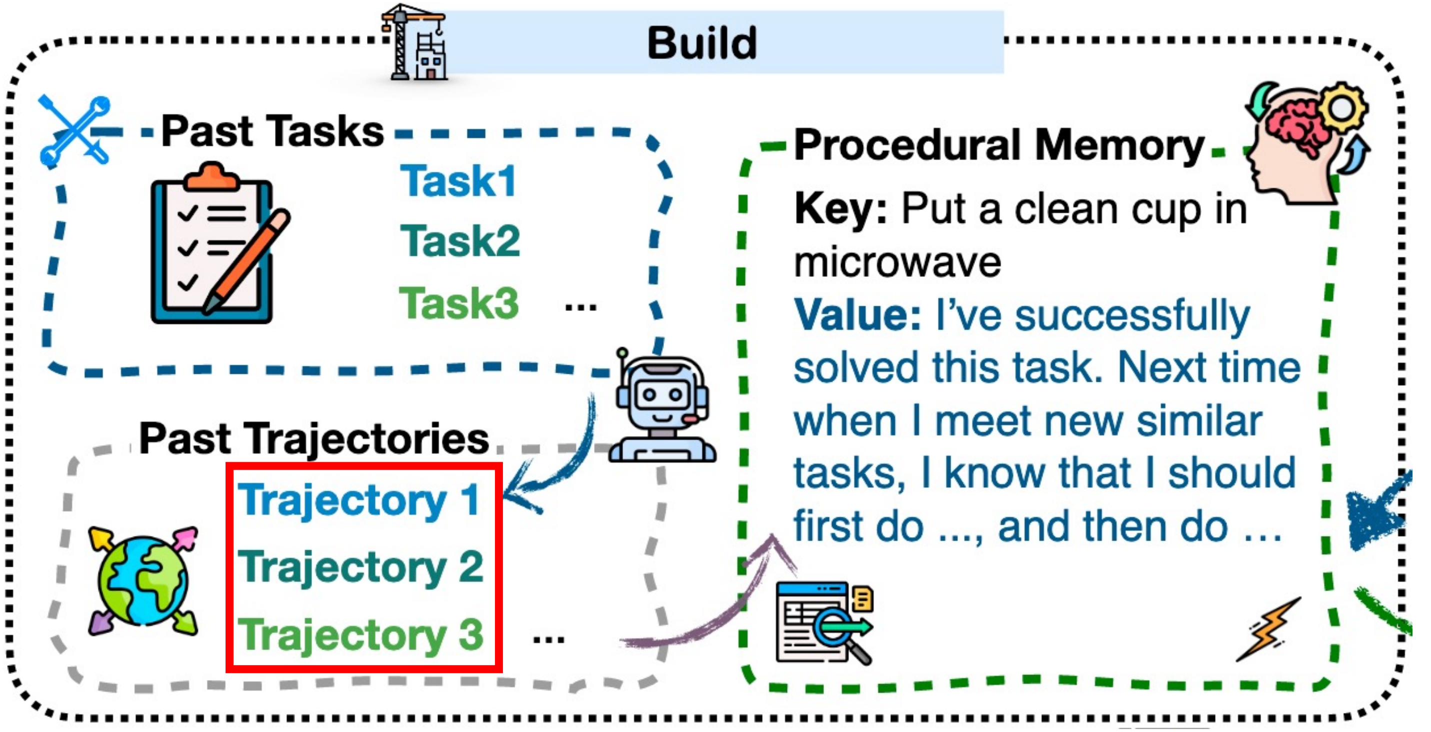
- Key (任务): “Heat an egg”
- Value（记忆）：

执行序列： “状态-观察-动作” 组成的详细日志，记录了智能体从开始到结束的每一步操作。

  - 状态：智能体在某个时刻所处的环境
  - 动作：智能体执行的操作，例如 go to fridge
  - 观察：执行动作后环境返回的反馈，例如

In the fridge 1, you see an egg 1, a cup 1…

最终结果：任务是成功还是失败

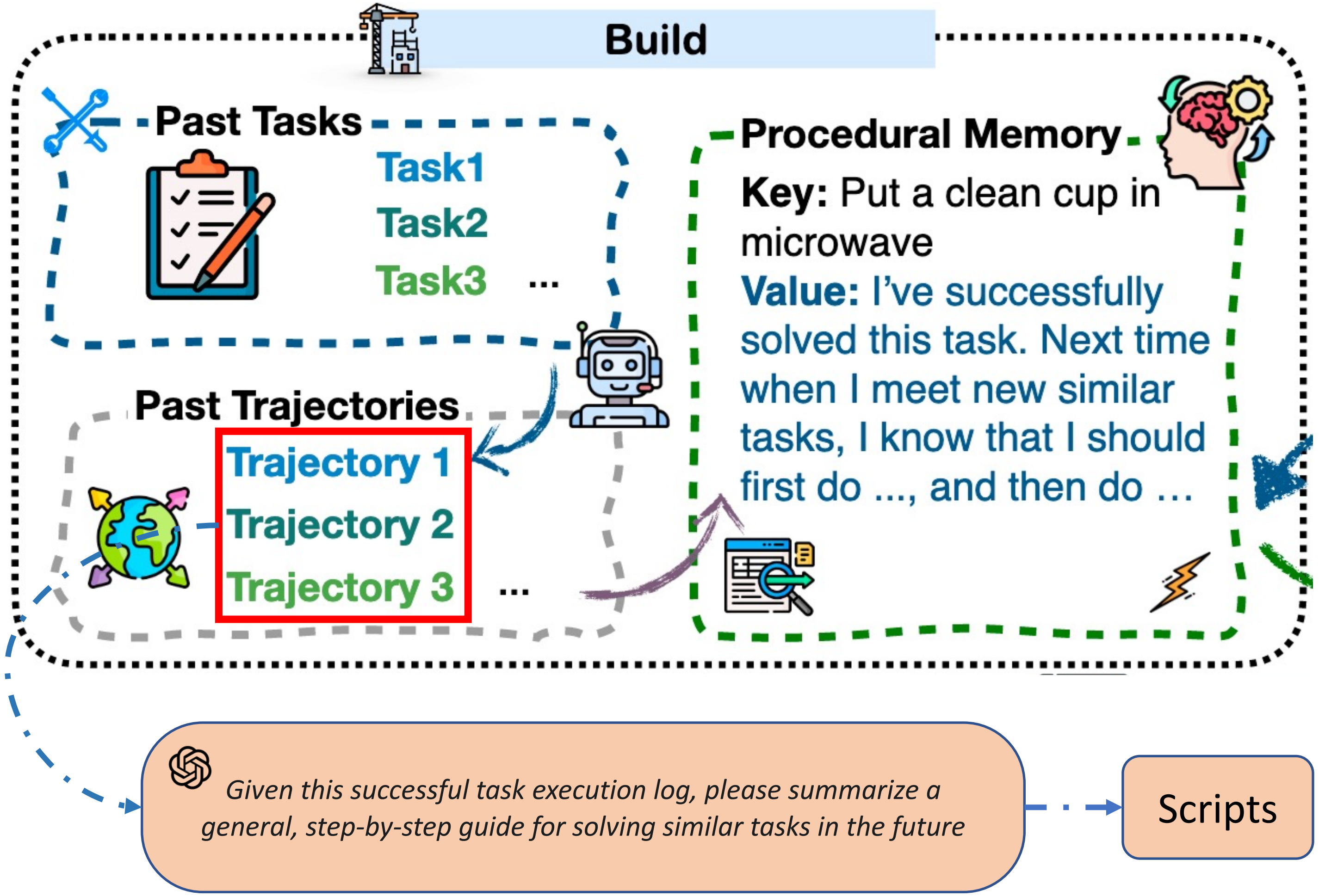


在构建记忆库时，通常只使用 **成功完成任务** 的 Trajectory

优势：细节丰富；实现简单 劣势：泛化能力差；冗余信息多

Method

Agent Scripts



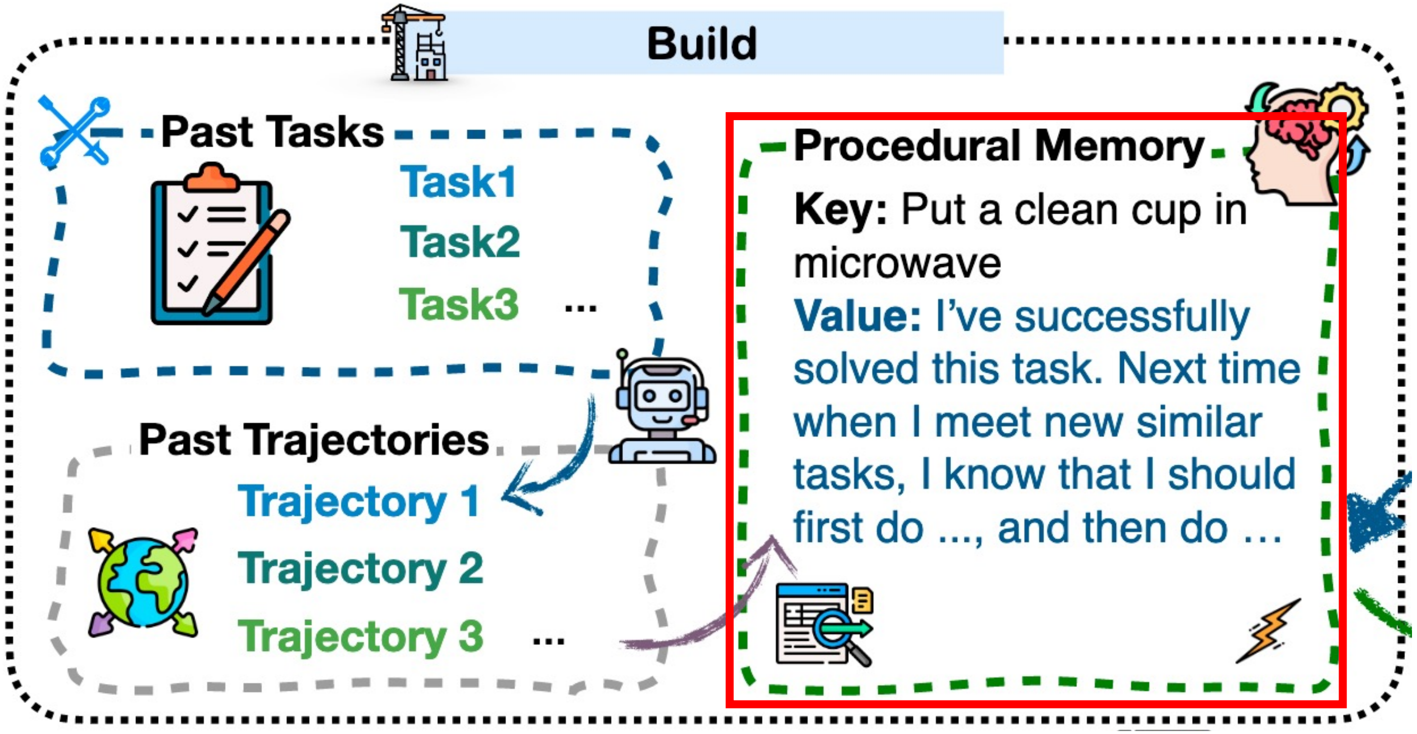
- Key (任务): “Heat an egg”
- Value (记忆):

要加热一个物体（如鸡蛋），首先你需要在环境中定位并拾取它，它通常位于储存容器（如冰箱）中。然后，你需要找到一个合适的加热设备（如微波炉）。将物体放入设备并启动加热程序。最后，根据任务要求处理加热后的物体。

优势：泛化能力强；简洁高效
劣势：可能丢失重要细节

Method

Agent Procedural Memory



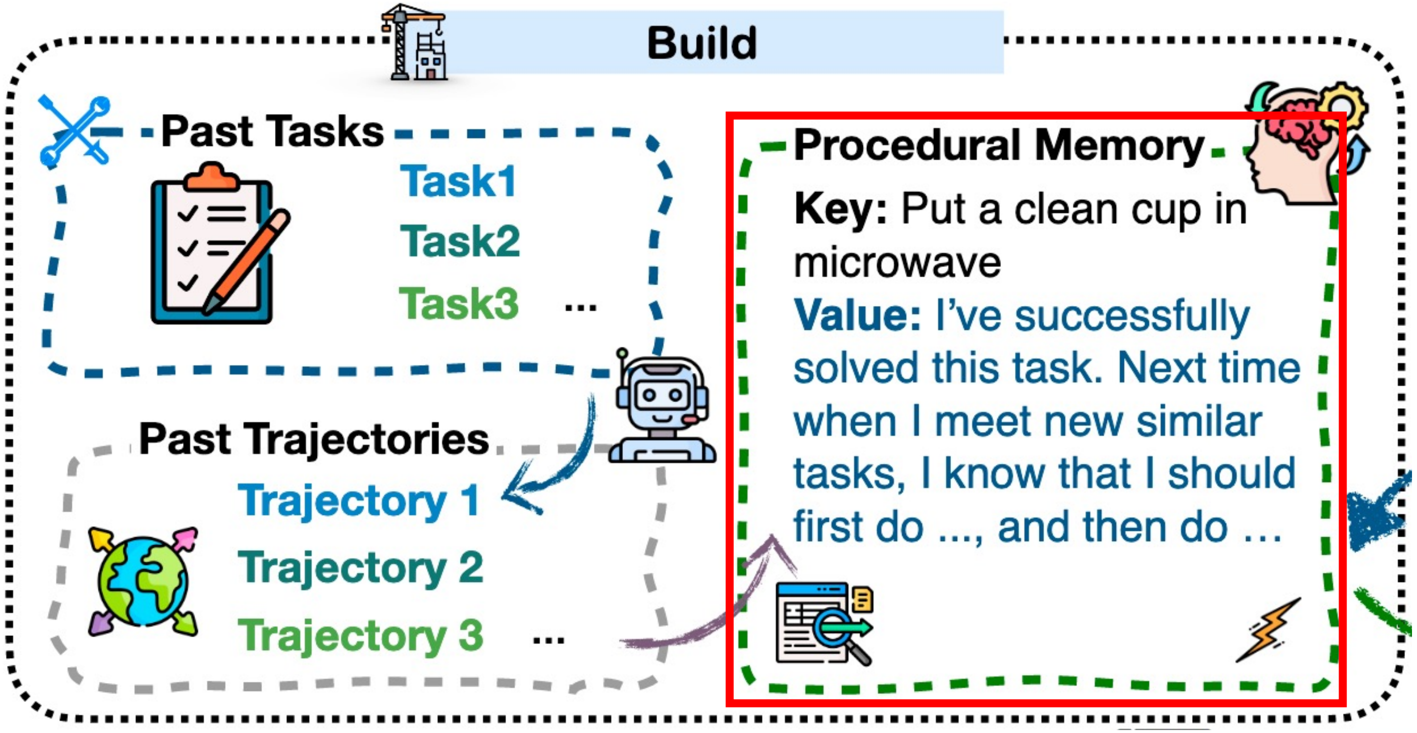
$$\tau = (s_0, a_0, o_1, s_1, a_1, o_2, \ldots, s_T),$$

$$r = R(env, s_T, \tau) \in [0, 1]$$

$$Mem = \sum_{t=1}^{T} m^{p_t}, where\ m^{p_t} = B(\tau_t, r_t)$$

Method

Agent Procedural Memory



[任务指导 - Script]: "要加热一个物体，首先定位并拾取它..."

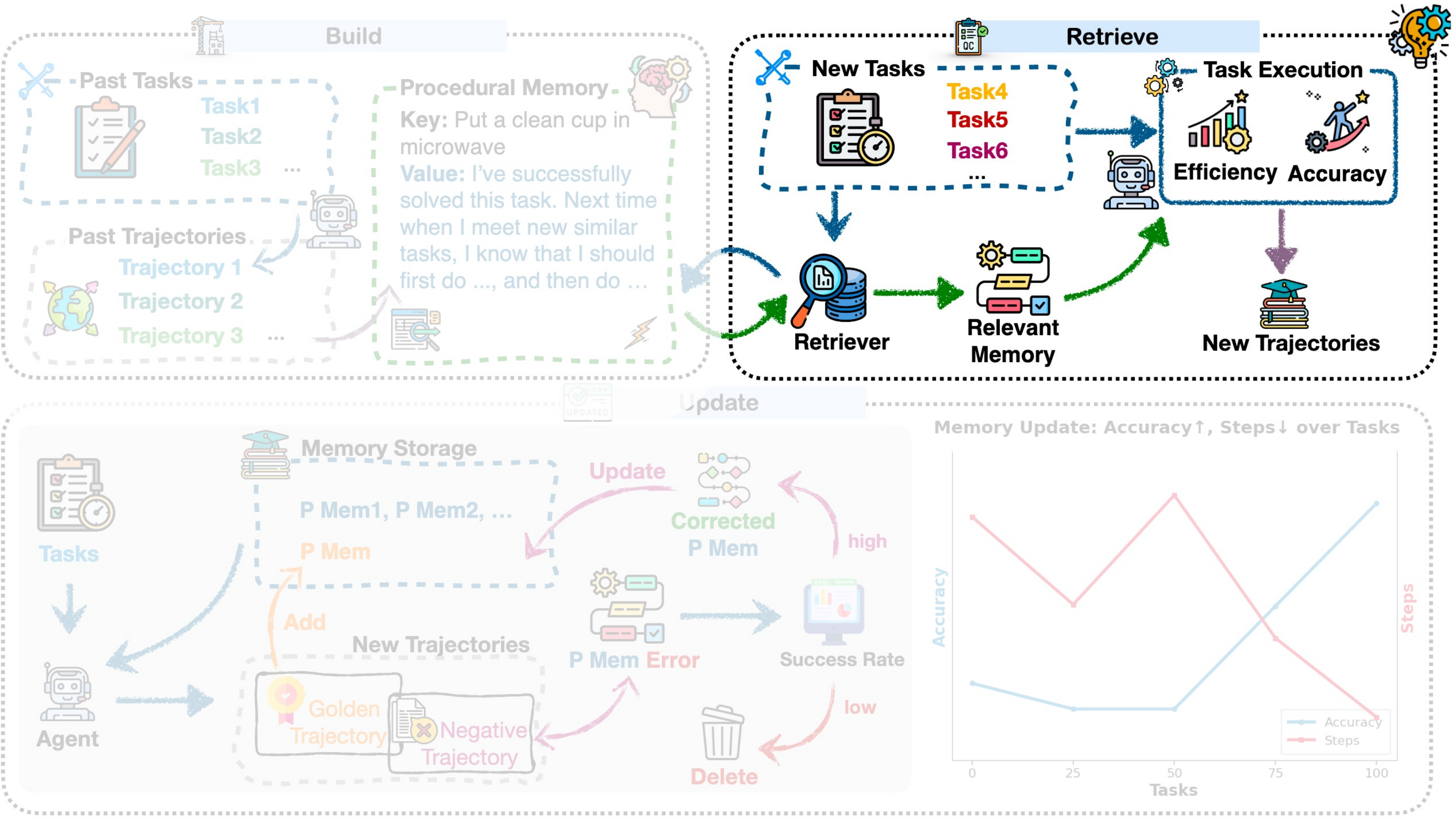
[相似任务范例 - Trajectory for heating an egg]: "1. go to fridge 1 -> 2. take egg 1 from fridge 1..."

优势：兼具泛化能力和细节
劣势：上下文成本过高

$$Mem = \sum_{t=1}^{T} m^{p_t}, where\ m^{p_t} = B(\tau_t, r_t)$$

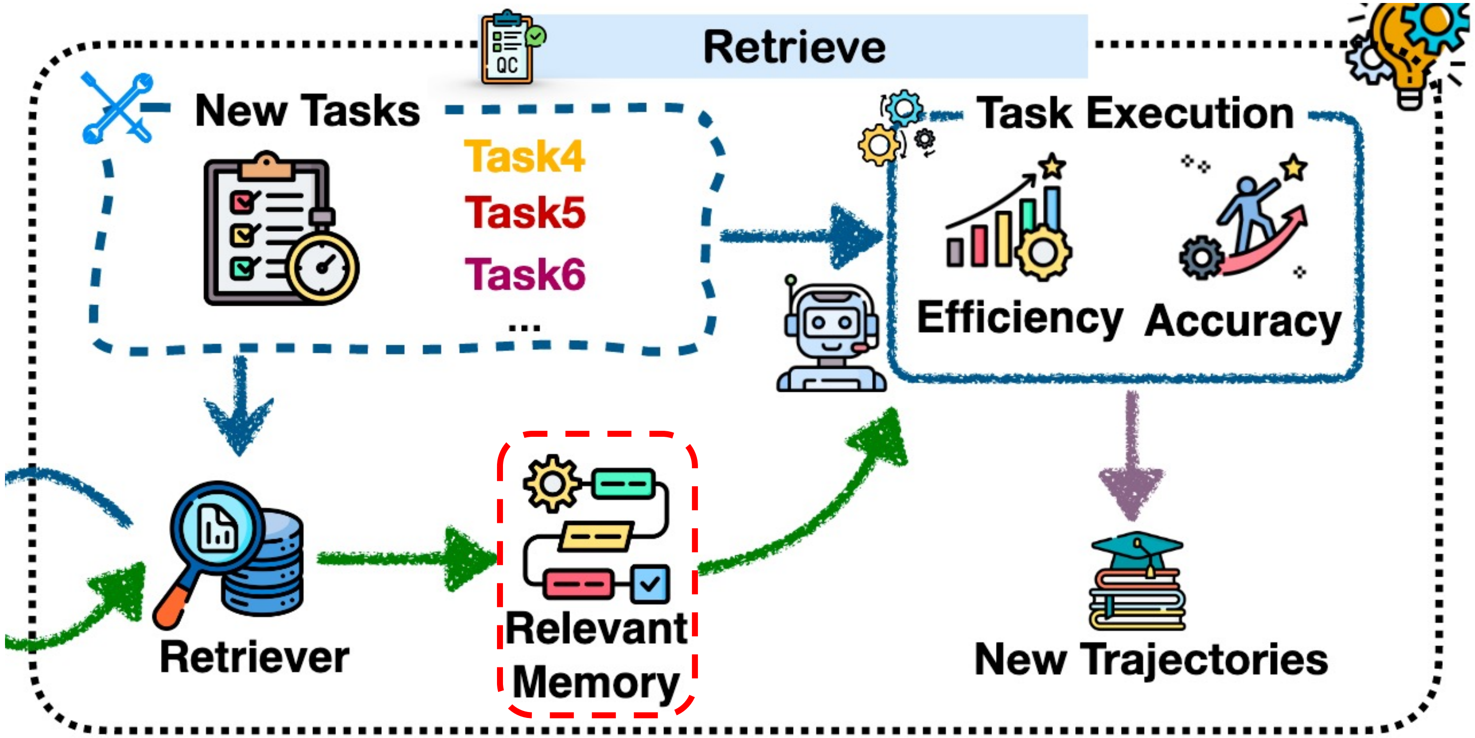
Method

Retrieve



Method

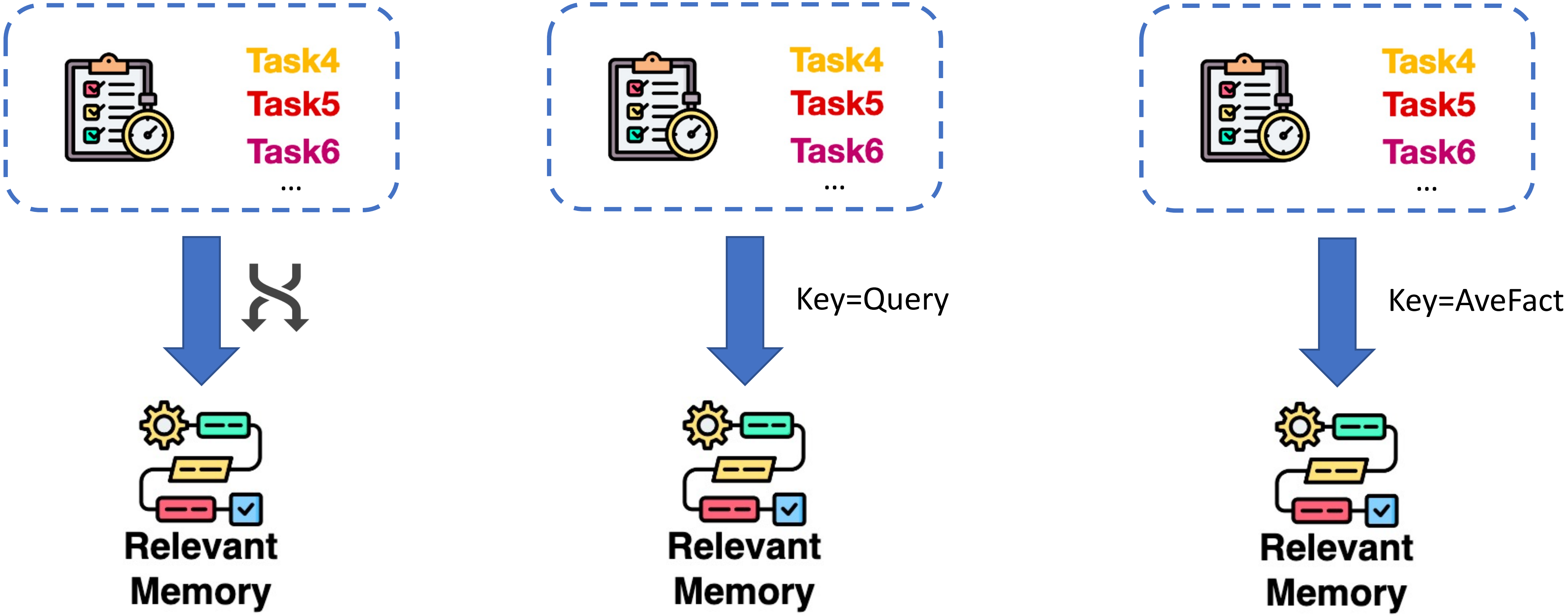
Retrieve



$$m_{retrieved} = \arg\max_{m^{p_i} \in Mem} S(t_{new}, t_i) \Rightarrow m_{retrieved} = \arg\max_{m^{p_i} \in Mem} \frac{\phi(t_{new}) \cdot \phi(t_i)}{\|\phi(t_{new})\| \|\phi(t_i)\|}$$

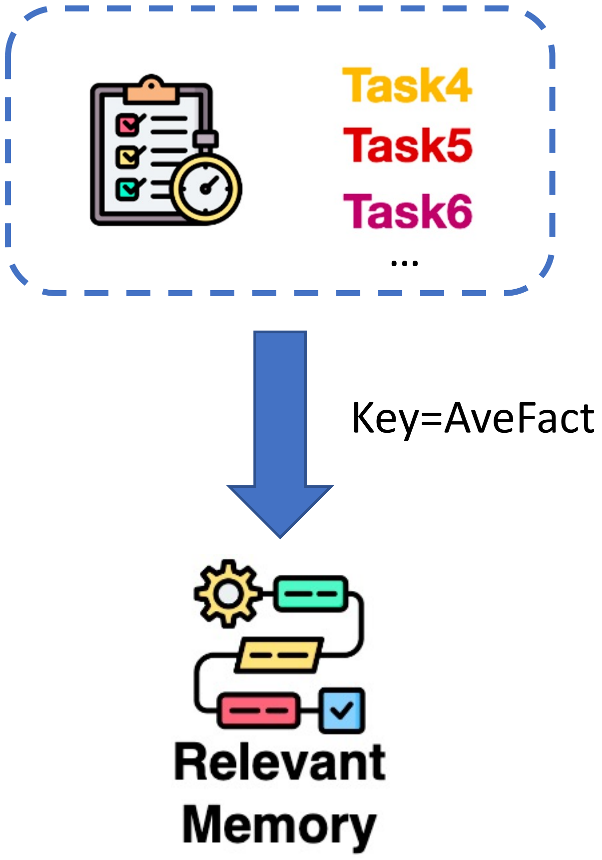
Method

Retrieve



Method

Retrieve — Key=AveFact



Step1: 事实提取

- 调用 LLM，对该记忆的**原始任务描述(Key)**进行分析，提取**核心事实(动词-名词)**
- 输入：假设记忆库里有一条旧记忆，其任务描述为：
  `Task: cook an egg and place it in the trash bin`
- LLM 指令 (类似): "Extract the core keywords (actions and objects) from this sentence."
- **输出 (核心事实列表):** `["cook", "egg", "place", "trash bin"]`

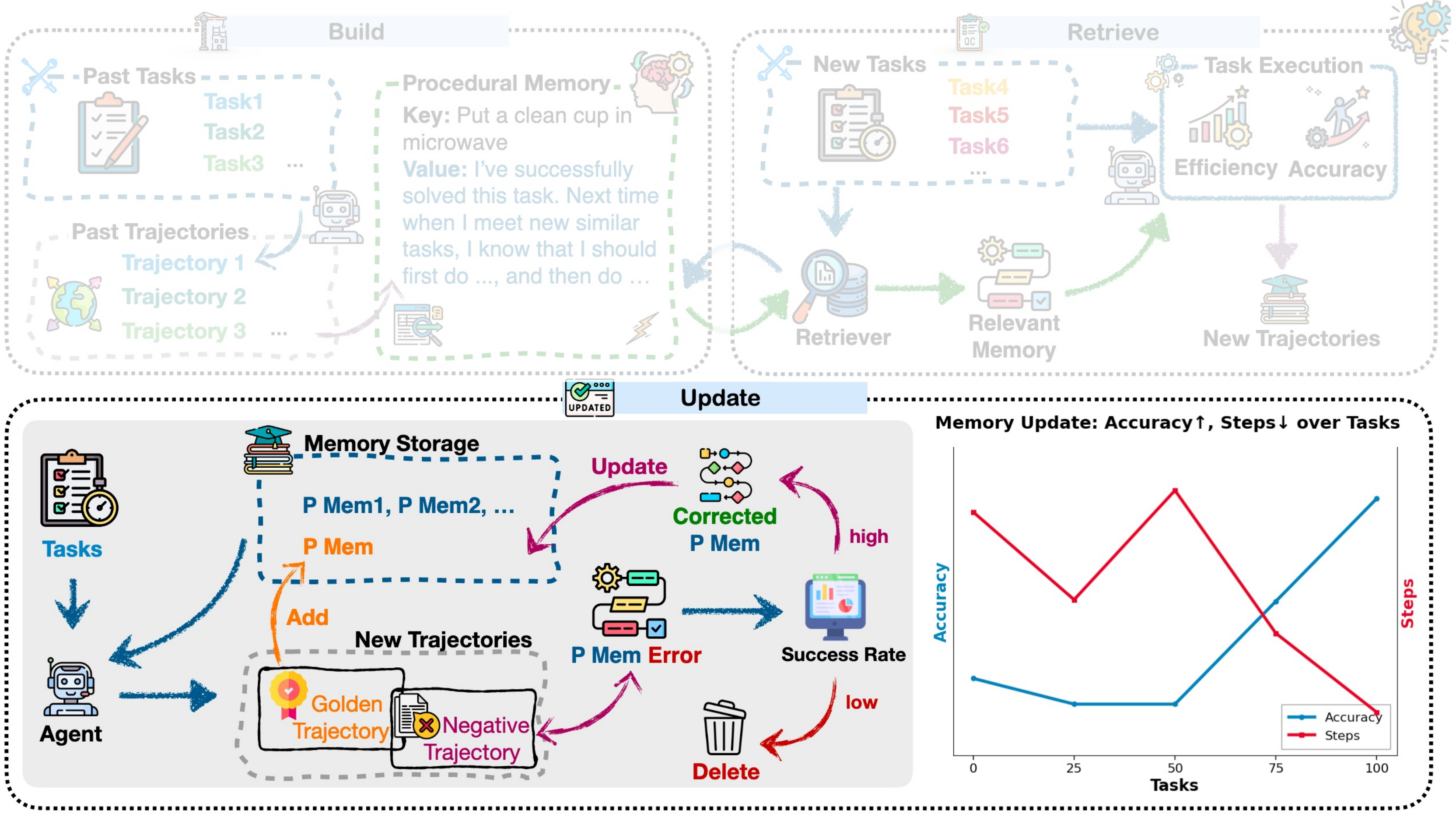
Step2: 事实向量化与平均

- `V_cook = model.embed("cook") -> [0.8, 0.1, …]`
- `V_egg = model.embed("egg") -> [0.3, 0.9, …]`
- `V_place = model.embed("place") -> [0.6, 0.5, …]`
- `V_trash_bin = model.embed("trash_bin") -> [0.2, 0.7, …]`
- `V_avg_fact_mem = (V_cook+V_egg+V_place+V_trash_bin)/4`

Step3: 使用 `V_avg_fact_mem` 向量化匹配

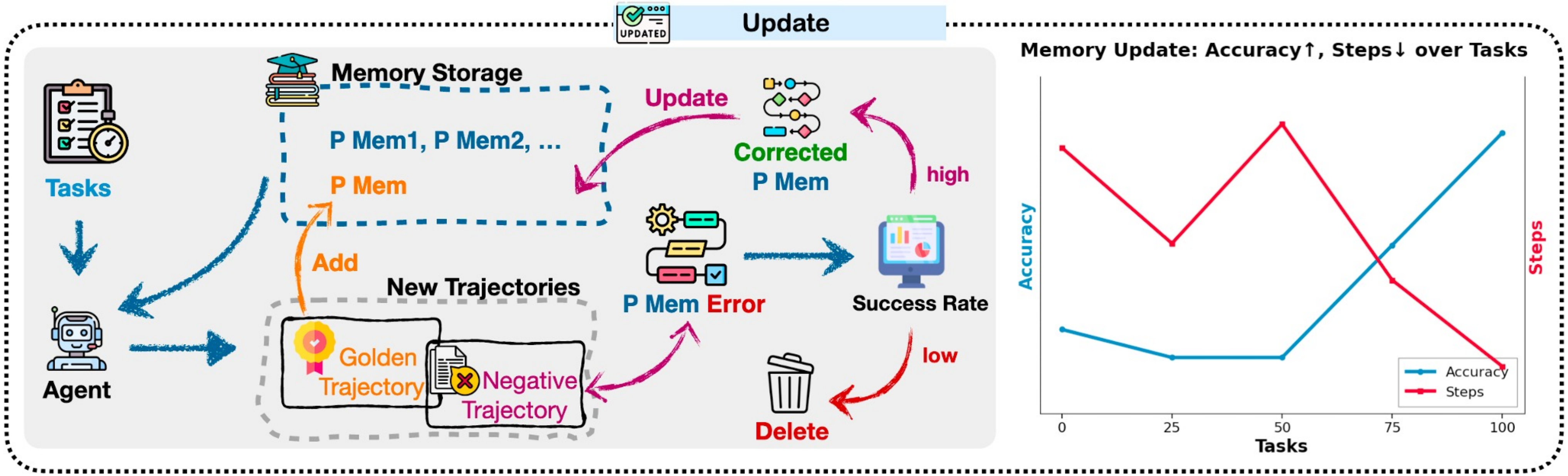
Method

Agent Memory



Method

Update



- 简单追加：直接 merge 记忆，无论对错
- 验证过滤：只保留成功的记忆，丢弃失败的记忆
- 反思调整：当一个检索的记忆导致任务失败时，将原始记忆和失败记忆结合起来进行反思，给定提示词，由 LLM 生成一个修正后的记忆，取代旧的错误版本

Contents

- Background
- Motivation
- Method
- **Evaluation**
- Thinking

Evaluation

Setting

Model

- GPT-4o
- Claude-3.5-sonnet
- Qwen2.5-72b

Benchmark

- TravelPlanner (ICML'24)
- ALFWorld (ICLR'21)

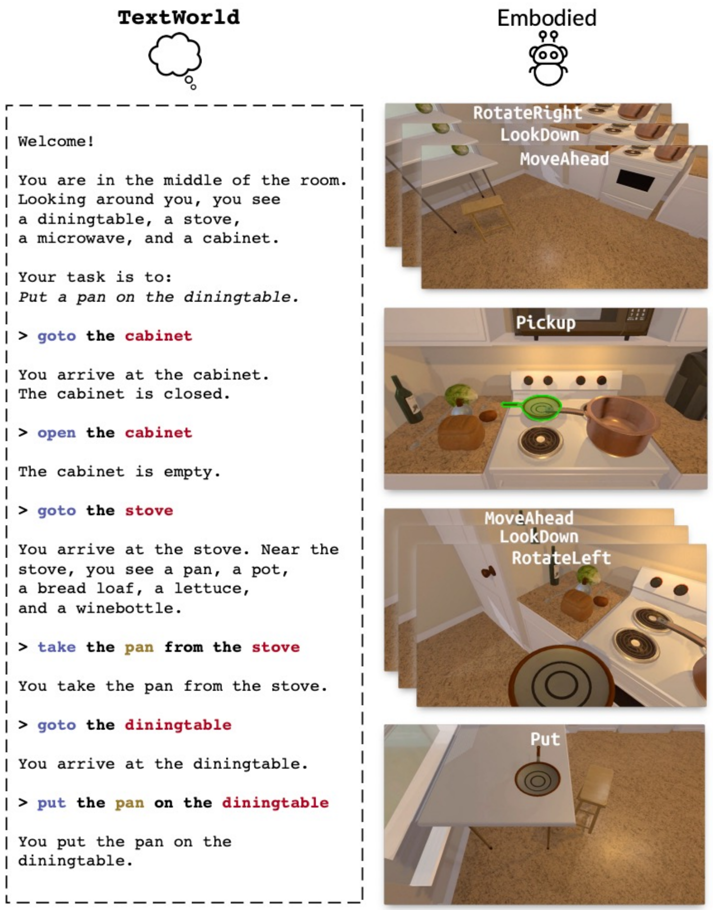


Figure 1: ALFWorld: Interactive aligned text and embodied worlds. An example with high-level text actions (left) and low-level physical actions (right).

Evaluation

Setting

Model
- GPT-4o
- Claude-3.5-sonnet
- Qwen2.5-72b

Metric
- Commensense (#CS)
- Hard Constraint (#HC)
- Steps

用于表示 AI Agent 常识推理能力的指标，测试 AI Agent 是否具有物理常识、社会常识、事件因果常识

- Dev
- Test

Benchmark
- TravelPlanner (ICML’24)
- ALFWorld (ICLR’21)

Evaluation

Setting

Model
- GPT-4o
- Claude-3.5-sonnet
- Qwen2.5-72b

Metric
- Commensense (#CS)
- Hard Constraint (#HC)
- Steps

表明在任务中明确给出、严格遵守的条件；旅行必须在8月10日出发，8月15日返回（如果8月11日出发，就违反了硬性约束）

衡量 AI Agent 的精确性和可靠性，越高代表 Agent 越 “可靠” ；越低意味着 Agent 幻觉越严重

- Dev
- Test

Benchmark
- TravelPlanner (ICML’24)
- ALFWorld (ICLR’21)

Evaluation

Setting

Model
- GPT-4o
- Claude-3.5-sonnet
- Qwen2.5-72b

Benchmark
- TravelPlanner (ICML’24)
- ALFWorld (ICLR’21)

Metric
- Commensense (#CS)
- Hard Constraint (#HC)
- Steps

AI Agent 执行步数，用于衡量效率

- Dev
- Test

Evaluation

Setting

Model
- GPT-4o
- Claude-3.5-sonnet
- Qwen2.5-72b

Metric
- Commensense (#CS)
- Hard Constraint (#HC)
- Steps

- Dev
- Test

Seen Split，环境布局已知，任务指令未知

Benchmark
- TravelPlanner (ICML’24)
- ALFWorld (ICLR’21)

Evaluation

Setting

Model
- GPT-4o
- Claude-3.5-sonnet
- Qwen2.5-72b

Metric
- Commensense (#CS)
- Hard Constraint (#HC)
- Steps

Benchmark
- TravelPlanner (ICML'24)
- ALFWorld (ICLR'21)

- Dev
- Test

Unseen Split，环境布局未知，任务指令未知

Evaluation

Build

Model	Granularity	TravelPlanner #CS ↑	TravelPlanner #HC ↑	TravelPlanner Steps ↓	ALFWorld Dev ↑	ALFWorld Test ↑	ALFWorld Steps ↓
GPT-4o	No Memory	71.93	**12.88**	17.84	39.28	42.14	23.76
	Script	72.08	5.50	15.79	66.67	56.43	18.52
	Trajectory	76.02	8.25	14.64	67.17	74.29	16.49
	Proceduralization	**79.94**	9.76	**14.62**	**87.14**	**77.86**	**15.01**
Claude-3.5-sonnet	No Memory	63.49	**33.06**	18.84	39.20	34.97	24.12
	Script	62.08	29.61	19.21	56.13	53.59	19.38
	Trajectory	65.76	29.61	17.72	69.28	71.78	15.97
	Proceduralization	**65.46**	30.14	**15.29**	**82.50**	**74.72**	**15.79**
Qwen2.5-72b	No Memory	56.57	7.34	18.32	44.91	41.25	21.38
	Script	58.59	7.34	18.53	66.24	61.88	17.13
	Trajectory	63.41	12.66	18.12	64.49	69.57	16.40
	Proceduralization	**63.82**	**14.19**	**17.94**	**85.71**	**77.19**	**15.32**

结论：
- 任何形式的记忆都比无记忆要好
- 程序化记忆综合表现要更好
- Script 和 Trajectory 各有侧重

在 ALFWorld 中，Trajectory > Script;
原因在于 ALFWorld 是一个与物理环境交互的任务，操作空间固定且离散(go to, take, open)，因此，正确的操作序列非常重要

Evaluation

Build

Model	Granularity	TravelPlanner #CS ↑	TravelPlanner #HC ↑	TravelPlanner Steps ↓	ALFWorld Dev ↑	ALFWorld Test ↑	ALFWorld Steps ↓
GPT-4o	No Memory	71.93	**12.88**	17.84	39.28	42.14	23.76
	Script	72.08	5.50	15.79	66.67	56.43	18.52
	Trajectory	76.02	8.25	14.64	67.17	74.29	16.49
	Proceduralization	**79.94**	9.76	**14.62**	**87.14**	**77.86**	**15.01**
Claude-3.5-sonnet	No Memory	63.49	**33.06**	18.84	39.20	34.97	24.12
	Script	62.08	29.61	19.21	56.13	53.59	19.38
	Trajectory	65.76	29.61	17.72	69.28	71.78	15.97
	Proceduralization	**65.46**	30.14	**15.29**	**82.50**	**74.72**	**15.79**
Qwen2.5-72b	No Memory	56.57	7.34	18.32	44.91	41.25	21.38
	Script	58.59	7.34	18.53	66.24	61.88	17.13
	Trajectory	63.41	12.66	18.12	64.49	69.57	16.40
	Proceduralization	**63.82**	**14.19**	**17.94**	**85.71**	**77.19**	**15.32**

结论：
- 任何形式的记忆都比无记忆要好
- 程序化记忆综合表现要更好
- Script 和 Trajectory 各有侧重

在 ALFWorld 中，Trajectory > Script;
原因在于 ALFWorld 是一个与物理环境交互的任务，操作空间固定且离散(go to, take, open)，因此，正确的操作序列非常重要

Evaluation

Retrieve

Model	Policy	#*CS* ↑	#*HC* ↑	Steps ↓
GPT-4o	No Memory	71.93	**12.88**	17.84
	Random Sample	<u>74.59</u>	6.72	<u>15.12</u>
	Key=Query	73.38	<u>8.95</u>	15.44
	Key=AveFact	**76.02**	8.25	**14.64**
Claude-3.5-sonnet	No Memory	63.49	**33.06**	18.84
	Random Sample	63.99	<u>29.91</u>	17.93
	Key=Query	<u>64.93</u>	28.56	**17.60**
	Key=AveFact	**65.76**	29.61	<u>17.72</u>
Qwen2.5-72b	No Memory	56.57	7.34	18.32
	Random Sample	59.76	8.43	<u>18.31</u>
	Key=Query	<u>61.71</u>	<u>11.97</u>	18.54
	Key=AveFact	**63.41**	**12.66**	**18.12**

Table 2: Results on **Retrieve Policy** on TravelPlanner.

结论：
- 语义检索比随机检索效果要更好
- Key=AveFact 综合表现优于 Key=Query
- 在 HC 指标上 Key=AveFact 有时表现不佳

原因在于 Key=AveFact 本质上是一种**抽象化，**关键词之间的精确关系可能会丢失。而 HC 硬性约束过于严格

Evaluation

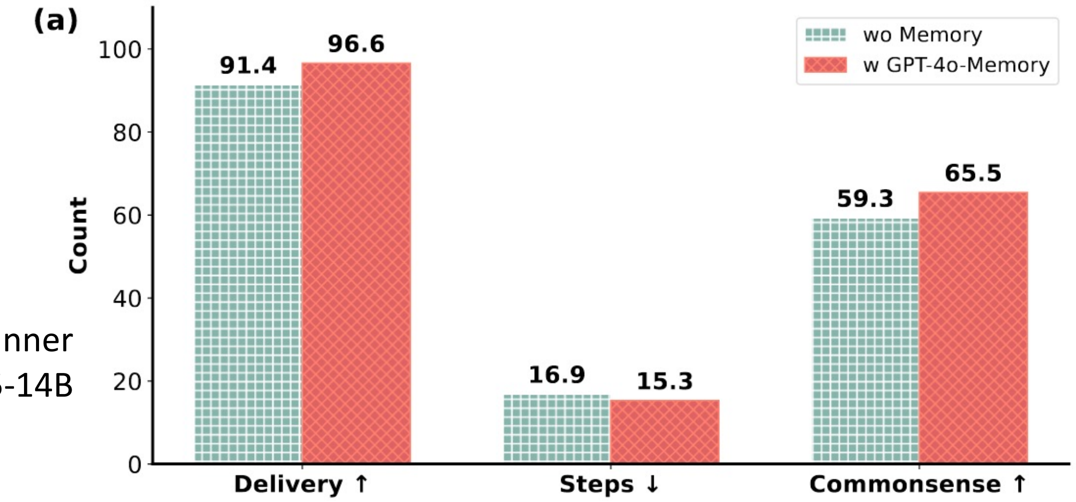
Update

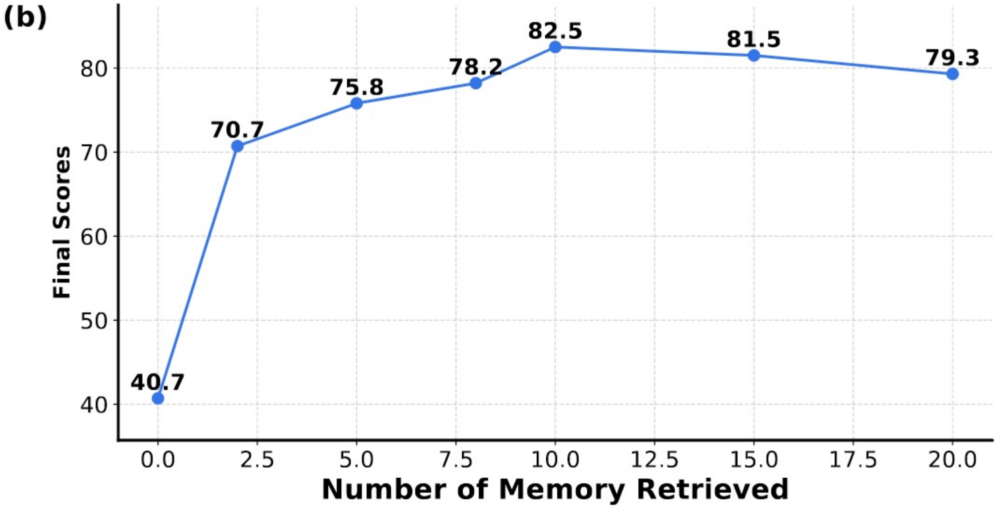
Figure 3: Reward gain and steps reduction vs. trajectory group index with **procedural memory**.

Evaluation

Extensibility

TravelPlanner
Qwen2.5-14B





交付率（完整、有效）

Evaluation

Case Study

Contents

- Background
- Motivation
- Method
- Evaluation
- **Thinking**

Thinking

能不能进一步提高？

- 当前的检索主要依赖查询向量的相似度匹配；可以结合上下文感知来优化检索方式，比如检索不应只看当前指令，还应该考虑智能体当前的上下文状态，比如优先检索当前停留的 App 的记忆；
- 当前记忆存储形式主要以非结构化的文本形式存在，可以考虑用知识图谱的形式存储程序性记忆，将动作流程用节点图表示，动作用节点表示，条件用边表示，这样可解释性更强，也容易修改和推理

能不能用到我们的场景？

- 效仿图 4，可以考虑云端结合，用云端产生的程序性记忆让端侧小模型来使用
- 文中提到的 Adjustment 更新策略，也可以考虑云端结合，端侧执行轻量化操作，例如简单的 merge 和 validation 云侧进行失败记忆的反思和更新，将结果传回端侧
- 存储资源限制，考虑记忆压缩方式来减少记忆占用

泛化性？

- 论文中的任务（AFLWorld, TravelPlanner）目标是明确且封闭的，而日常使用场景是开放、动态且模糊的，需要考虑更强的意图理解而不仅是停留在操作序列上

Q & A

Presenter: Yitao Wang
2025.9.20