



VillagerAgent: 减少幻觉提高任务分解效率的多智能协作体框架

Yubo Dong, Xukun Zhu, Zhengzhe Pan, Linchao Zhu, Yi Yang
ReLER Lab, CCAI, Zhejiang University

01

多智能体的研究
概述

02

经典多智能体框
架解析

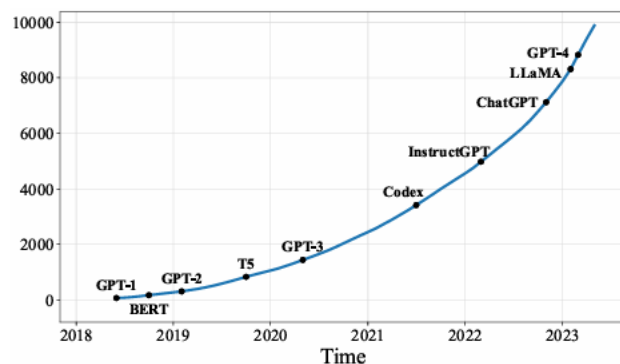
03

基于有向无环图
的多智能体框架
VilagerAgent

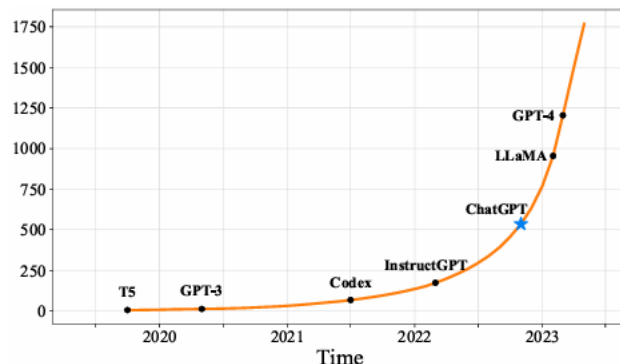
04

基准测试对比及
在多智能体协作
应用探讨

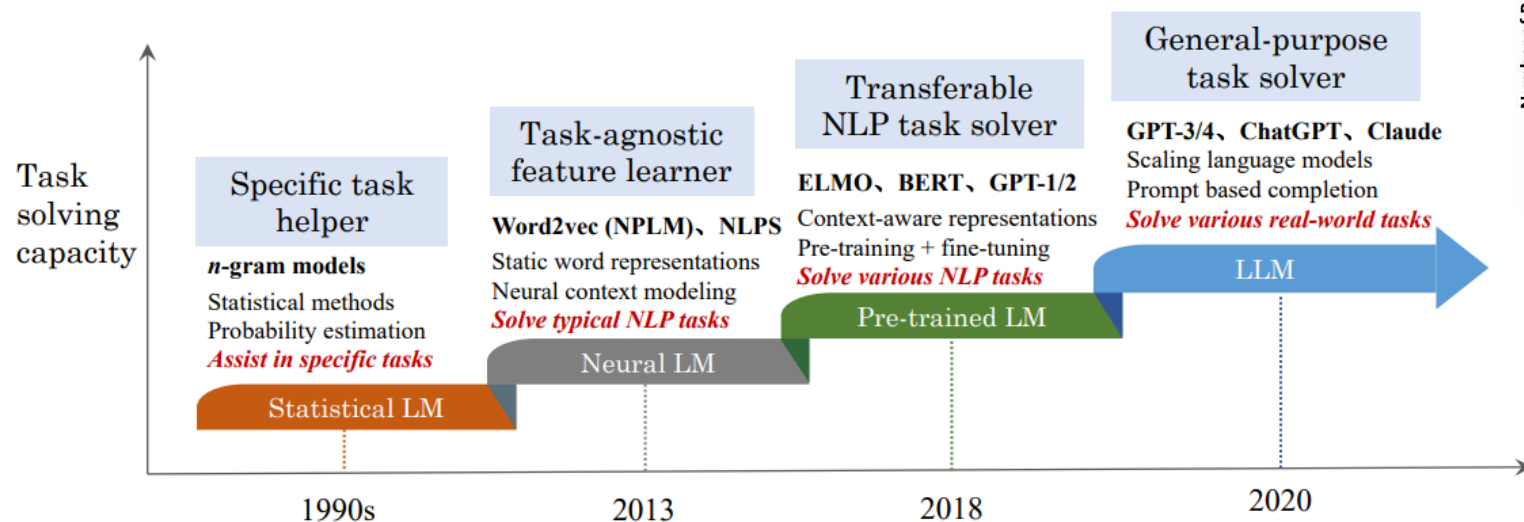
多智能体的研究概述



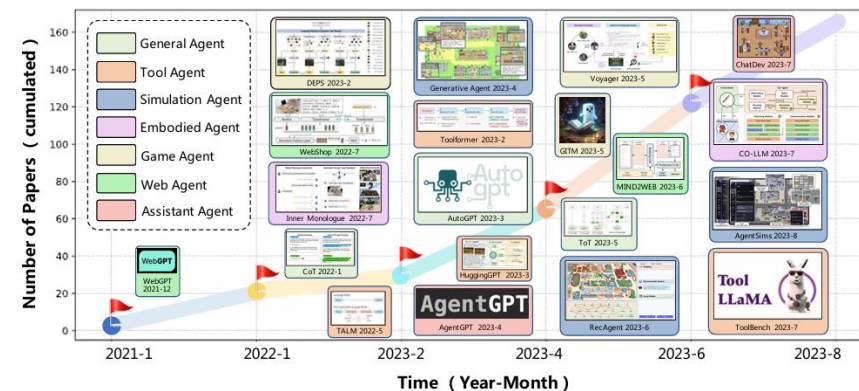
(a) Query="Language Model"



(b) Query="Large Language Model"



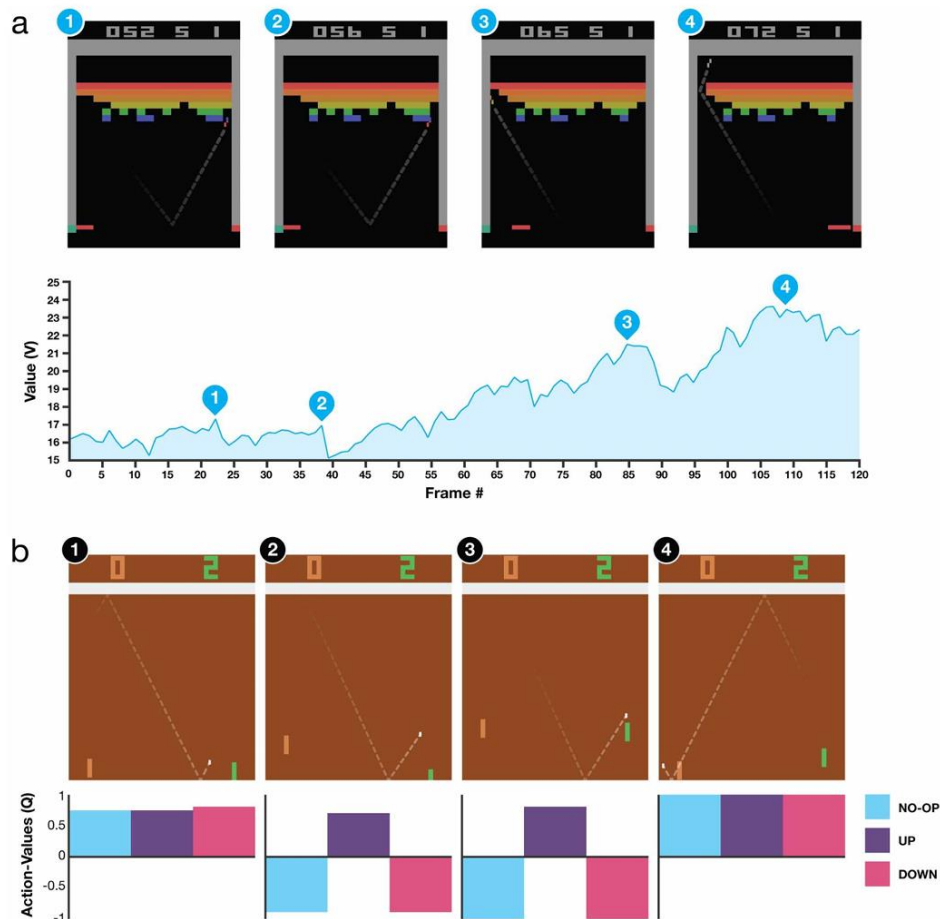
LLM: 理解语义 世界模型



LLM As Agents

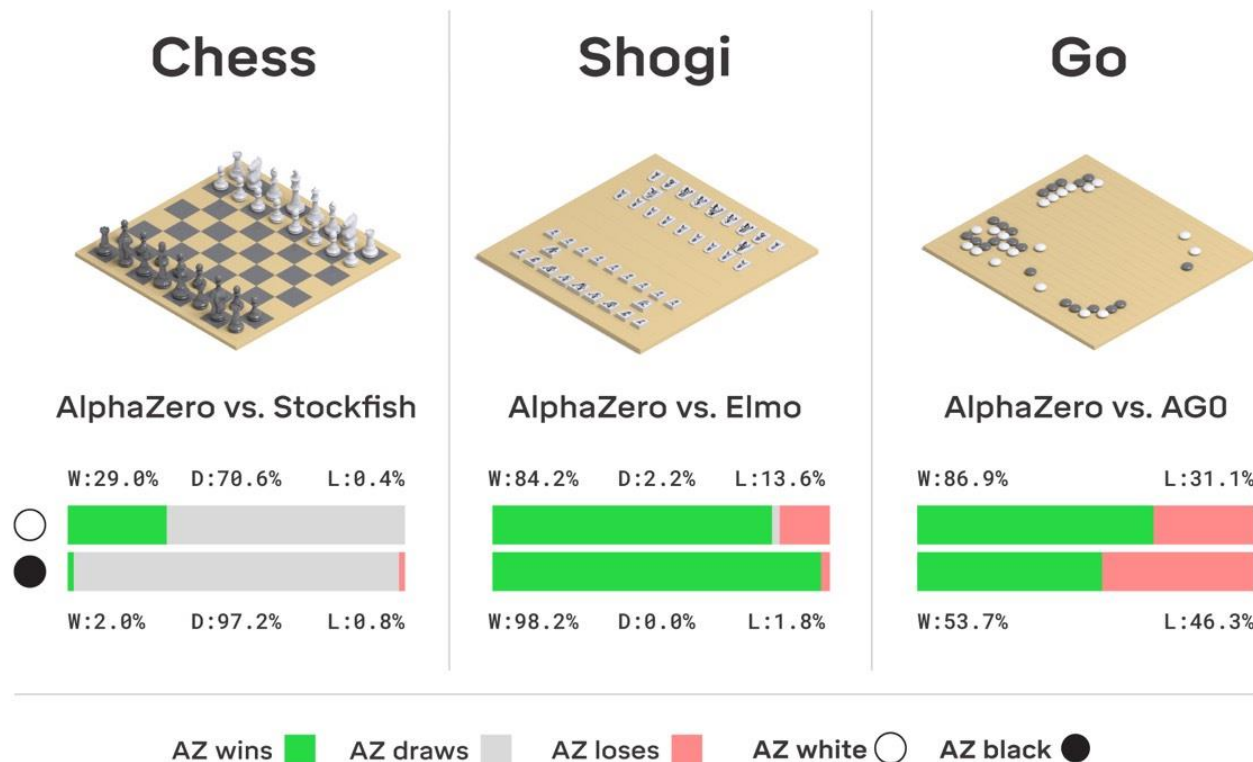
2021-2024

RL 的局限性

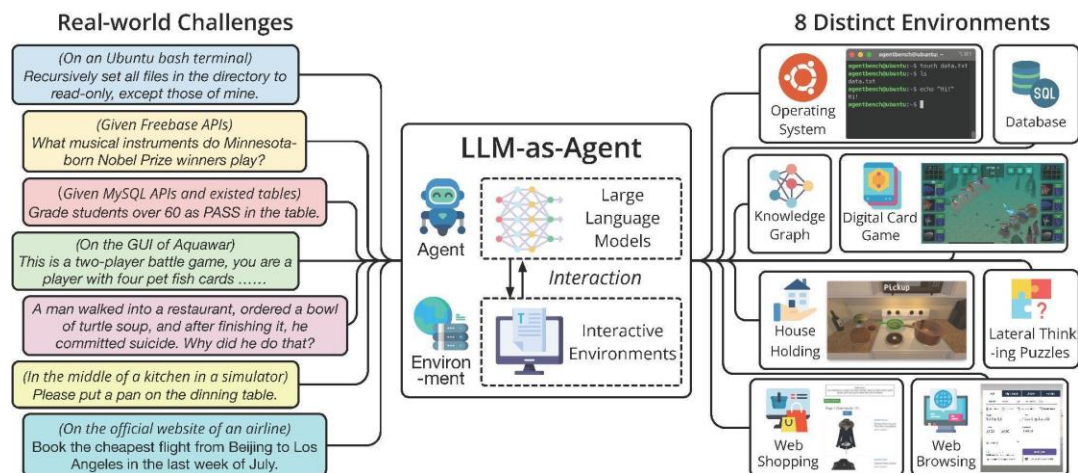


DQN Visualization of learned value functions on two games, Breakout and Pong.

求是创新

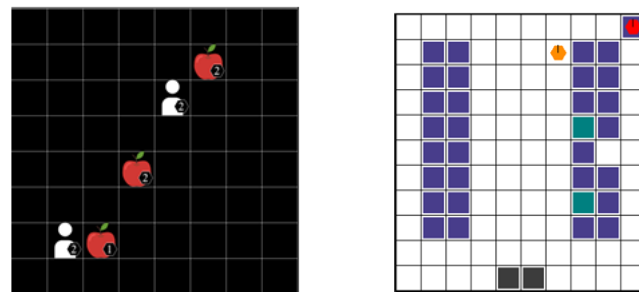


Enlarge / Starting from random play, knowing just the basic rules, AlphaZero defeated a world champion program in the games of Go, chess, and shogi.

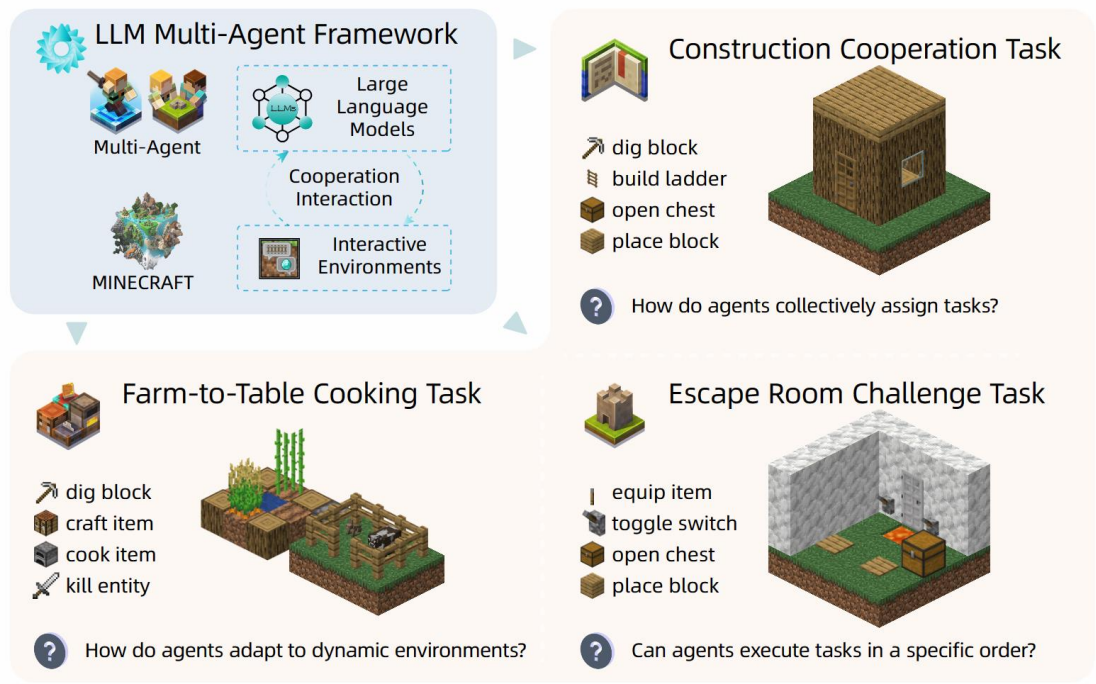


AgentBench (Single Agent)

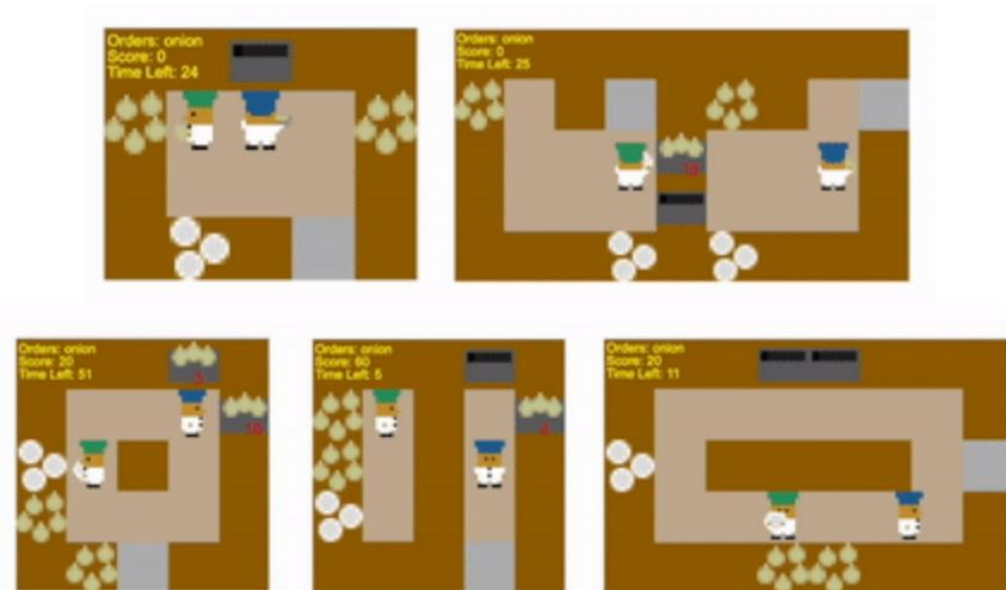
LLM的通用性展现



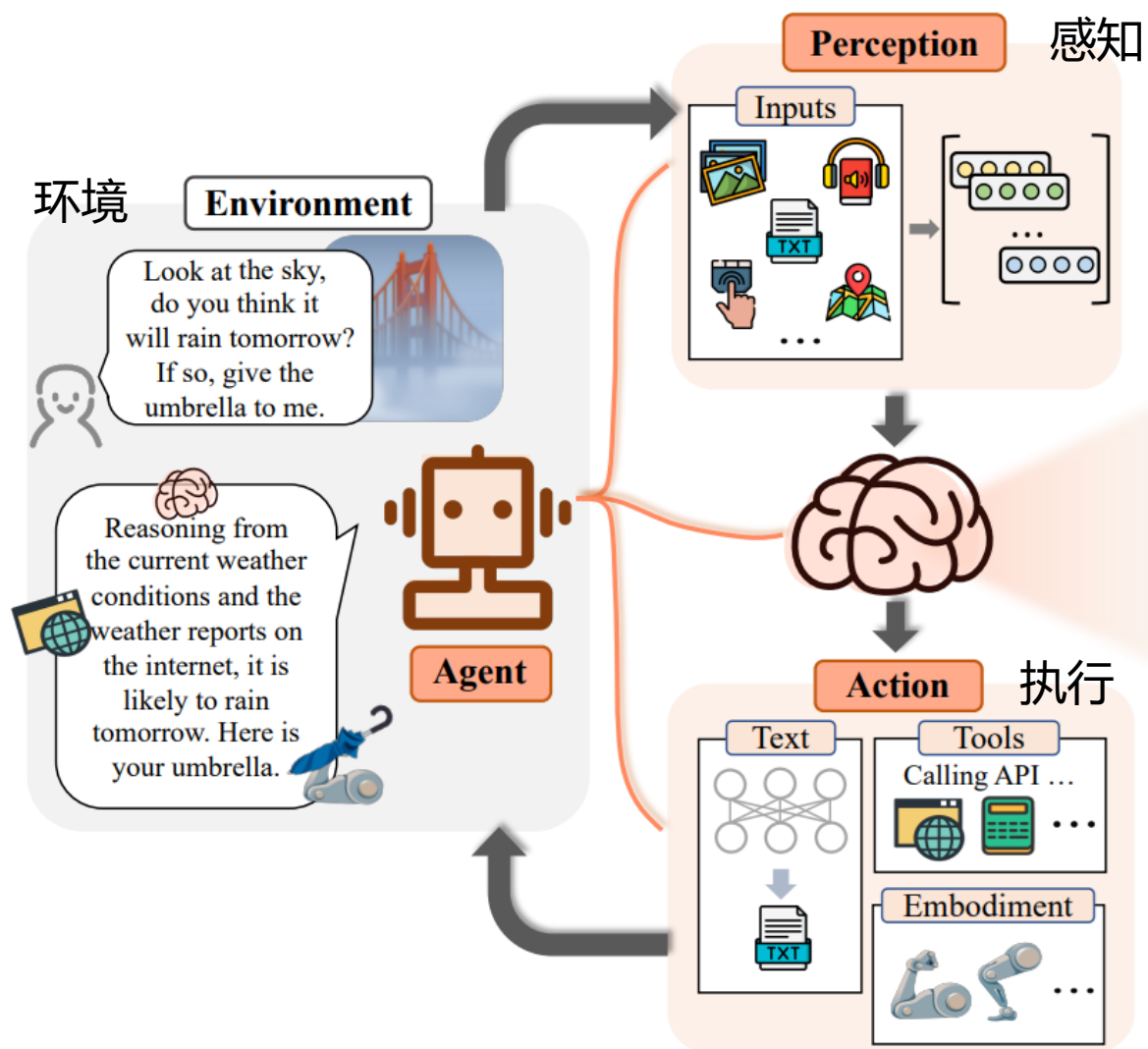
MultiAgent RL Benchmark



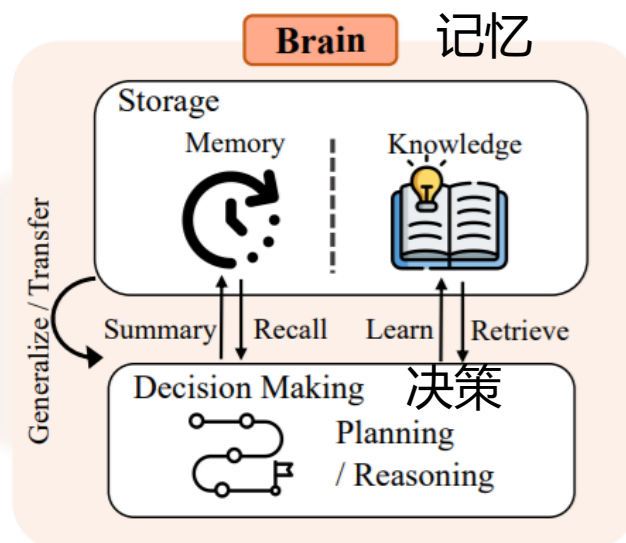
VillagerBench (Ours)



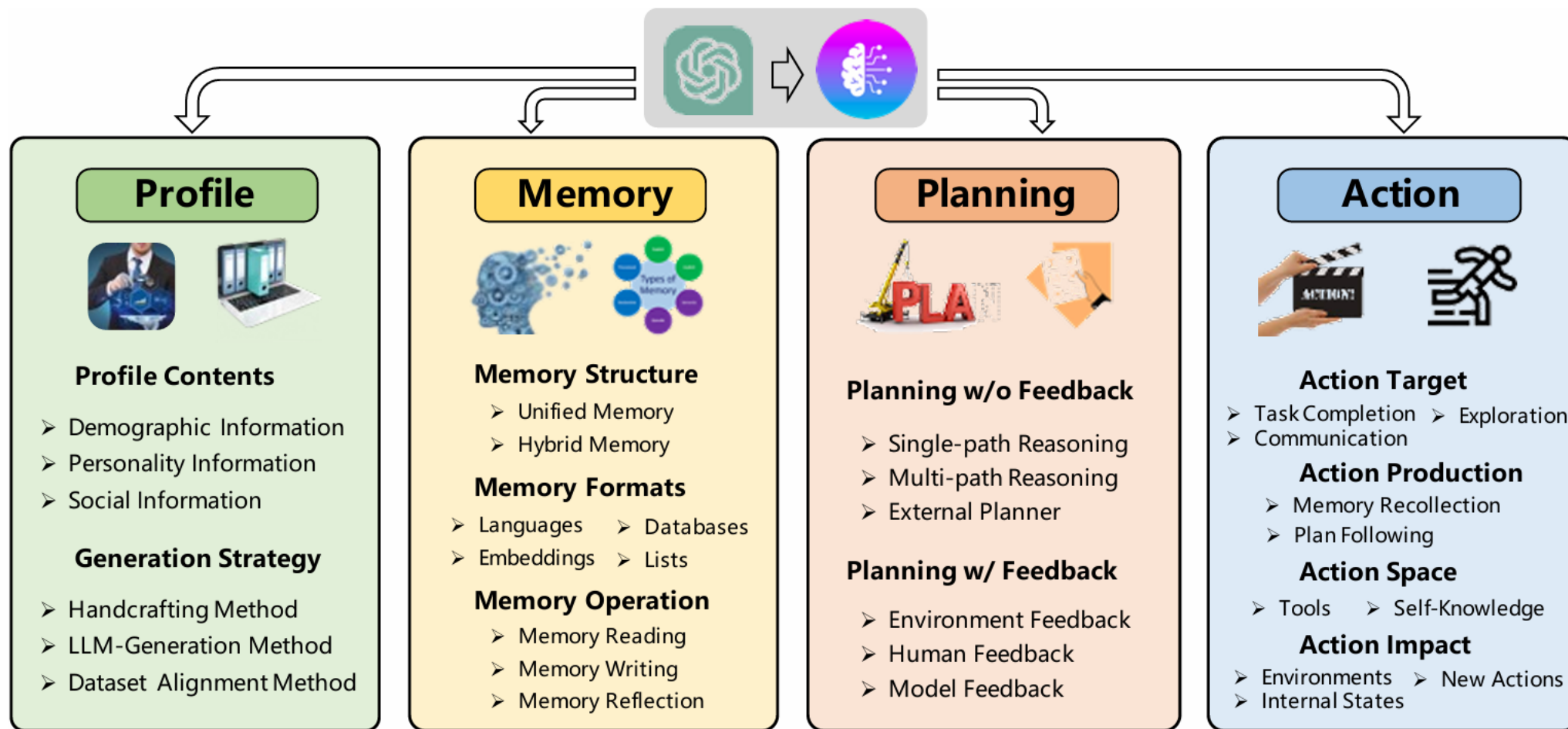
Overcooked-ai Benchmark



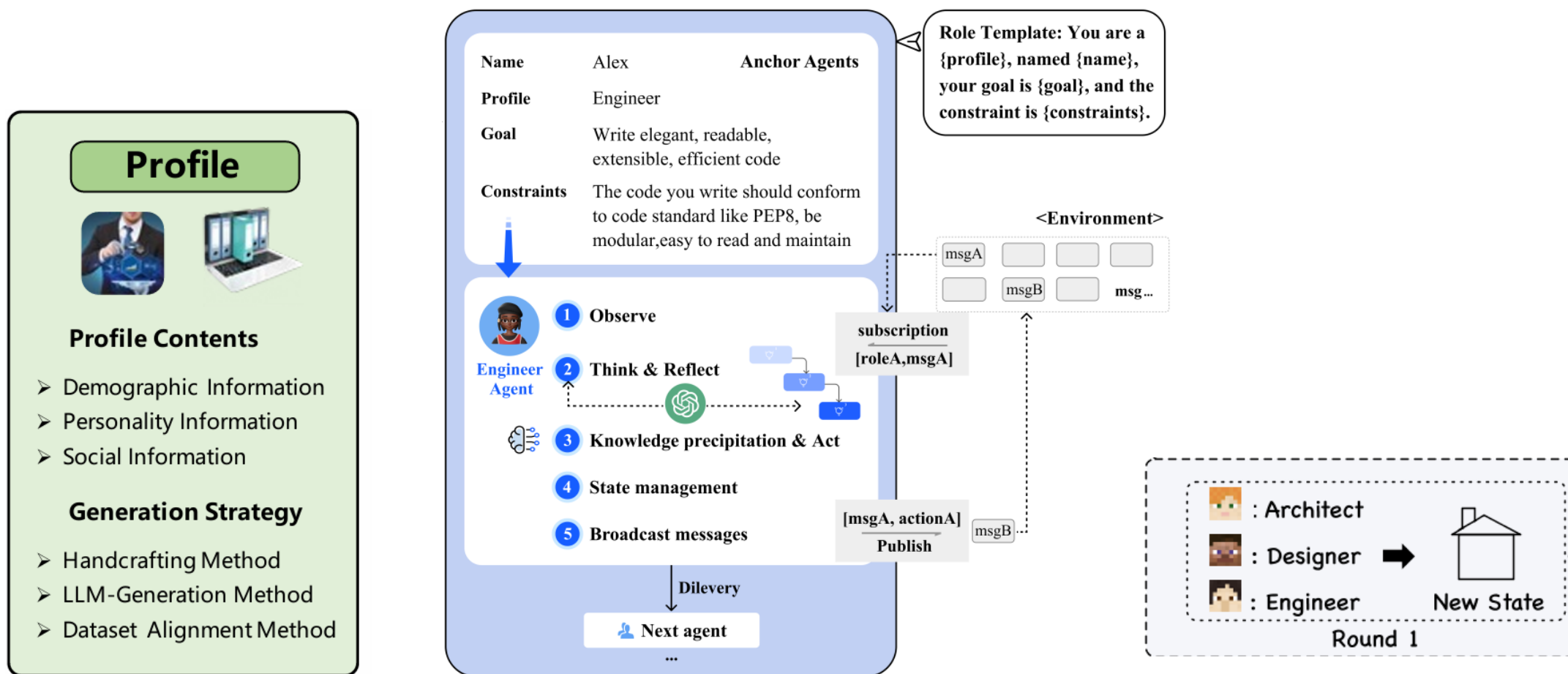
如何定义LLM Agent



"An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."

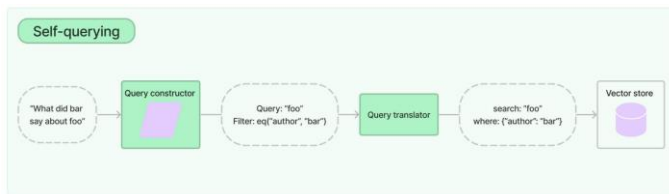


A unified framework for the architecture design of LLM-based autonomous agent.



A unified framework for the architecture design of LLM-based autonomous agent.

Memory retrieval



RAG



Memory



Memory Structure

- Unified Memory
- Hybrid Memory

Memory Formats

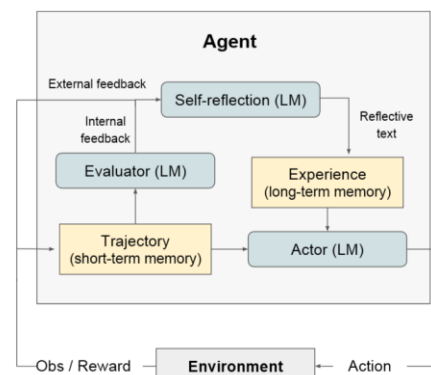
- Languages
- Databases
- Embeddings
- Lists

Memory Operation

- Memory Reading
- Memory Writing
- Memory Reflection

Memory capability

Reflexion 通过反思增强



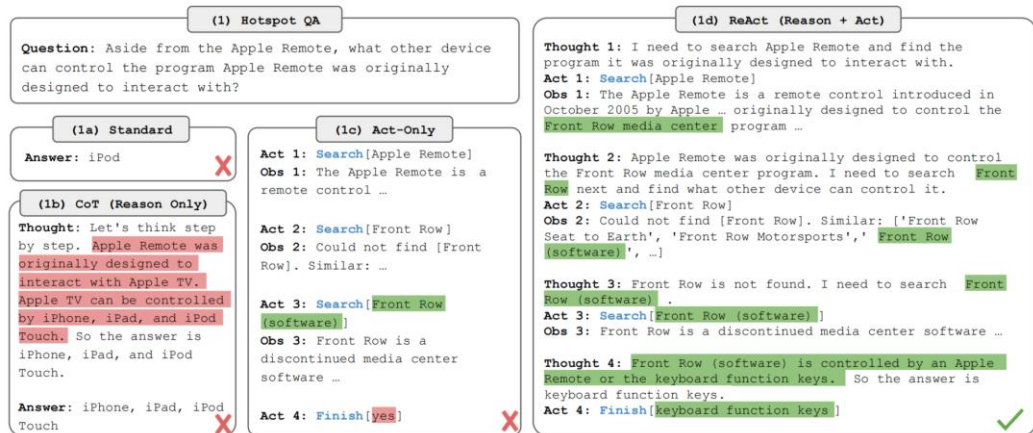
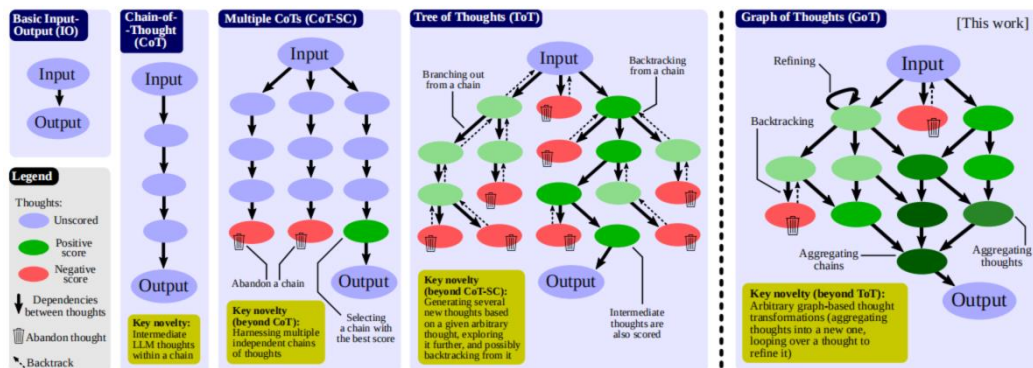
Algorithm 1 Reinforcement via self-reflection

```
Initialize Actor, Evaluator, Self-Reflection:  
 $M_a, M_e, M_{sr}$   
Initialize policy  $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$   
Generate initial trajectory using  $\pi_\theta$   
Evaluate  $\tau_0$  using  $M_e$   
Generate initial self-reflection  $sr_0$  using  $M_{sr}$   
Set  $mem \leftarrow [sr_0]$   
Set  $t = 0$   
while  $M_e$  not pass or  $t < \max \text{ trials}$  do  
    Generate  $\tau_t = [a_0, o_0, \dots, a_i, o_i]$  using  $\pi_\theta$   
    Evaluate  $\tau_t$  using  $M_e$   
    Generate self-reflection  $sr_t$  using  $M_{sr}$   
    Append  $sr_t$  to  $mem$   
    Increment  $t$   
end while  
return
```

Figure 2: (a) Diagram of Reflexion. (b) Reflexion reinforcement algorithm

A unified framework for the architecture design of LLM-based autonomous agent.

LLM+P: 通过PDDL求解任务并转换成自然语言描述



Planning

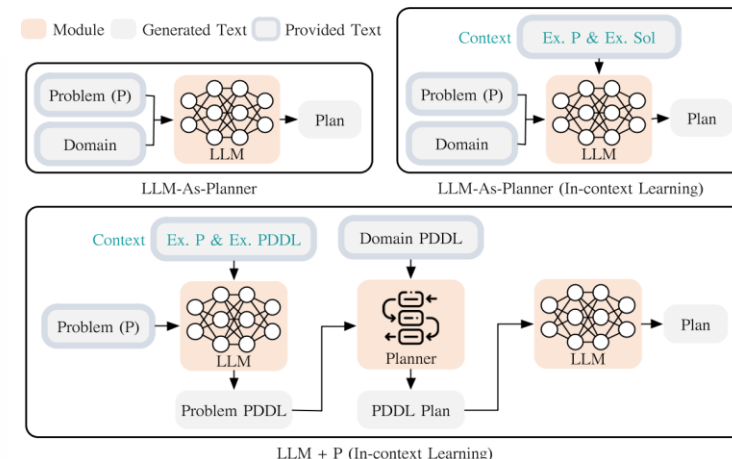


Planning w/o Feedback

- Single-path Reasoning
- Multi-path Reasoning
- External Planner

Planning w/ Feedback

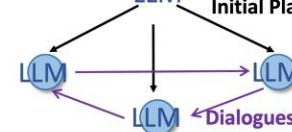
- Environment Feedback
- Human Feedback
- Model Feedback



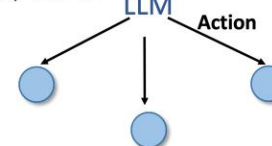
(a) DMAS



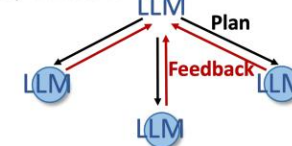
(b) HMAS-1



(c) CMAS

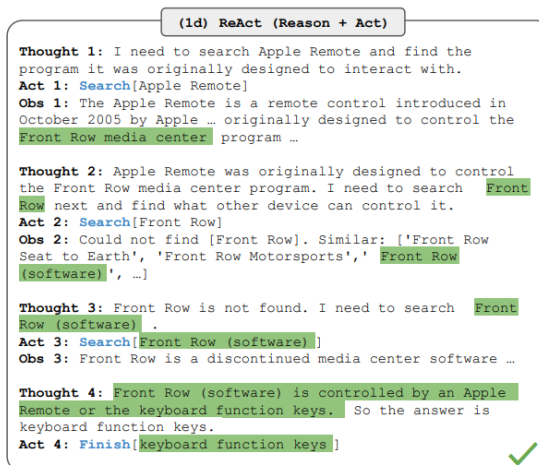
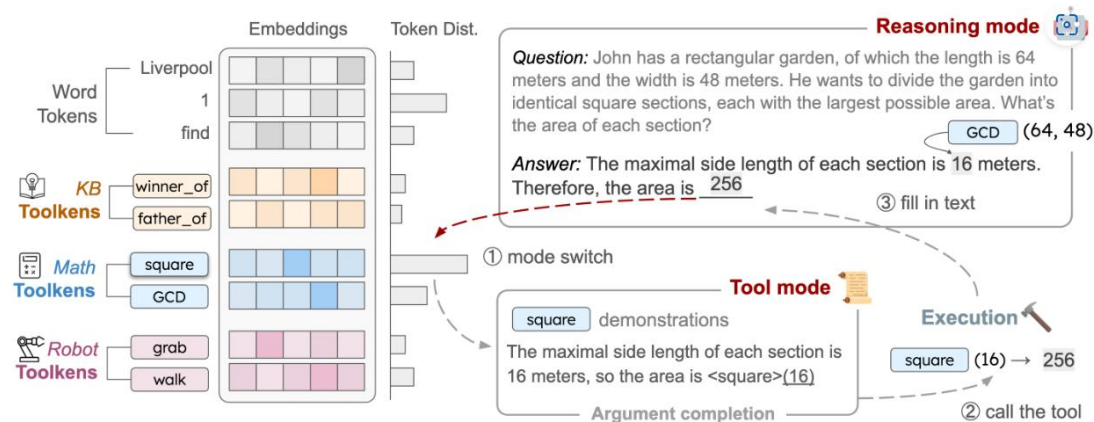
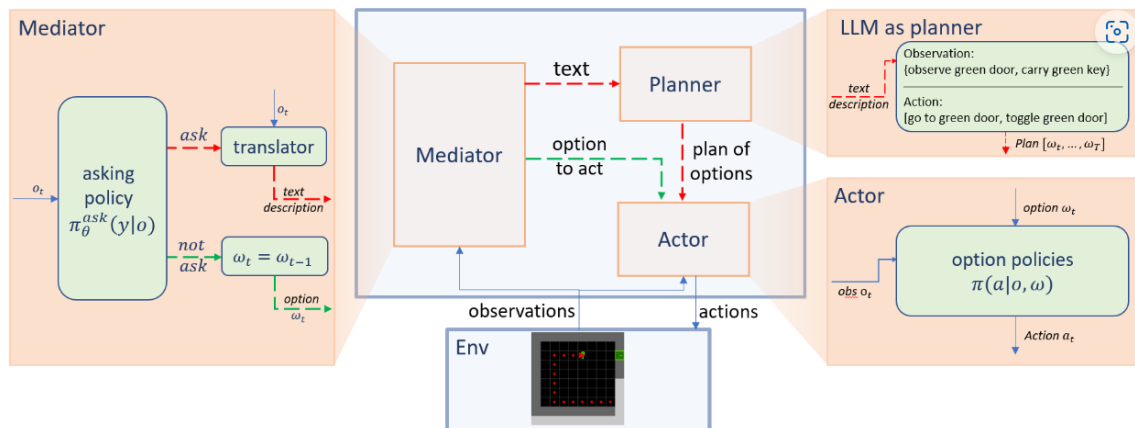


(d) HMAS-2



A unified framework for the architecture design of LLM-based autonomous agent.

LLM4RL



A unified framework for the architecture design of LLM-based autonomous agent.

01

多智能体的研究
概述

02

经典多智能体框
架解析

03

基于有向无环图
的多智能体框架
VilagerAgent

04

基准测试对比及
在多智能体协作
应用探讨

Stanford Altown Generative Agents: Interactive Simulacra of Human Behavior



Stanford Altown Generative Agents: Interactive Simulacra of Human Behavior

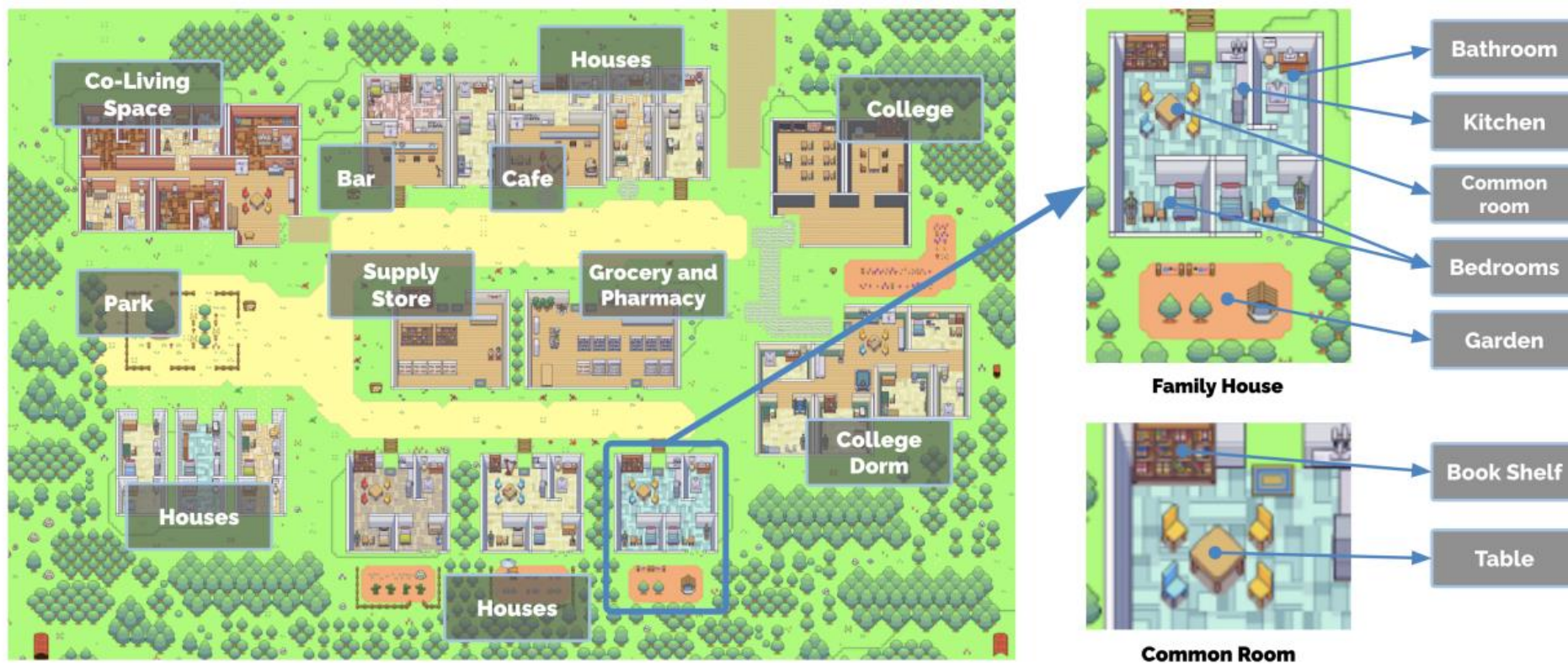
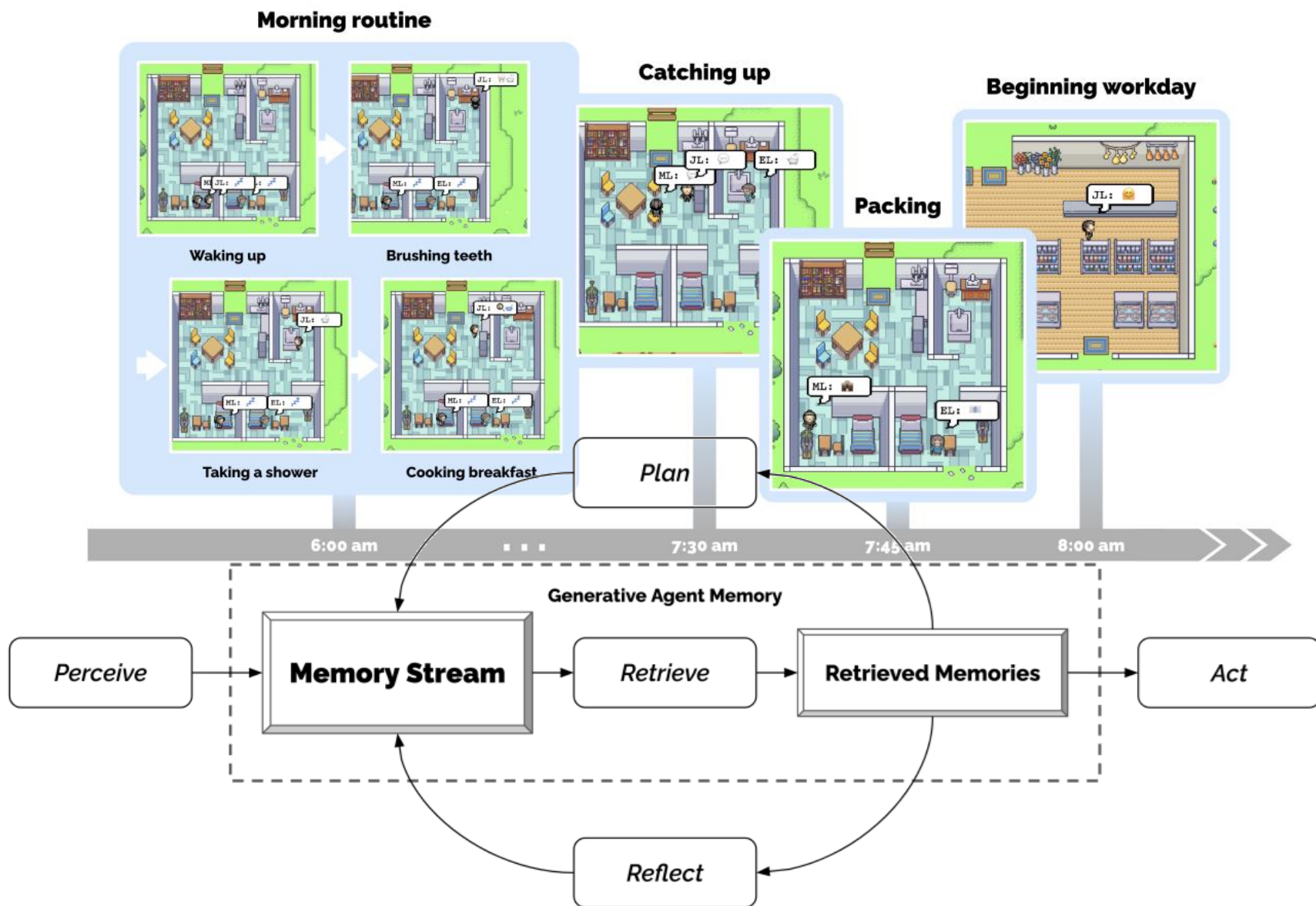
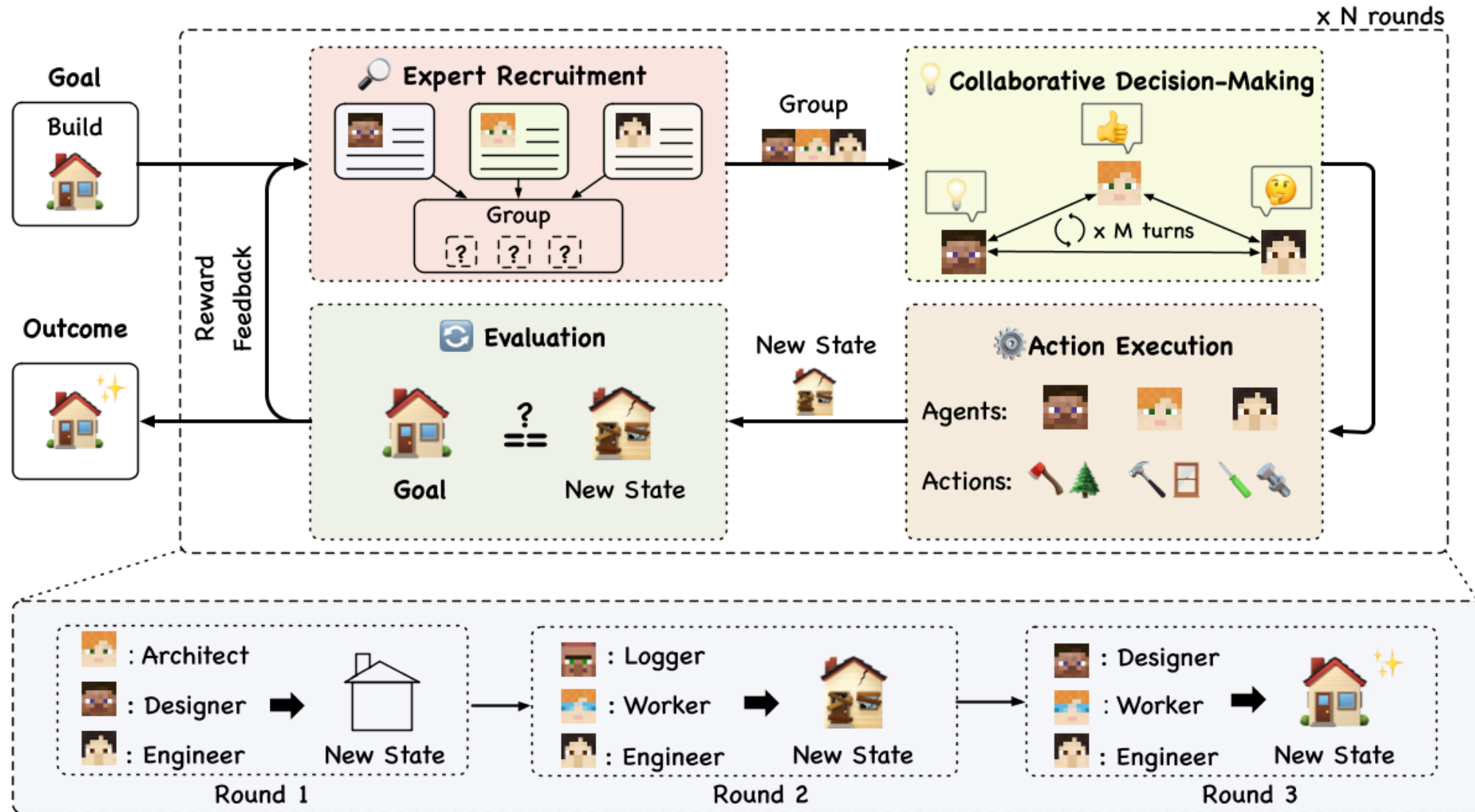


Figure 2: The Smallville sandbox world, with areas labeled. The root node describes the entire world, children describe areas (e.g., houses, cafe, stores), and leaf nodes describe objects (e.g., table, bookshelf). Agents remember a subgraph that reflects the parts of the world they have seen, maintaining the state of those parts as they observed them.

Stanford Altoun Generative Agents: Interactive Simulacra of Human Behavior



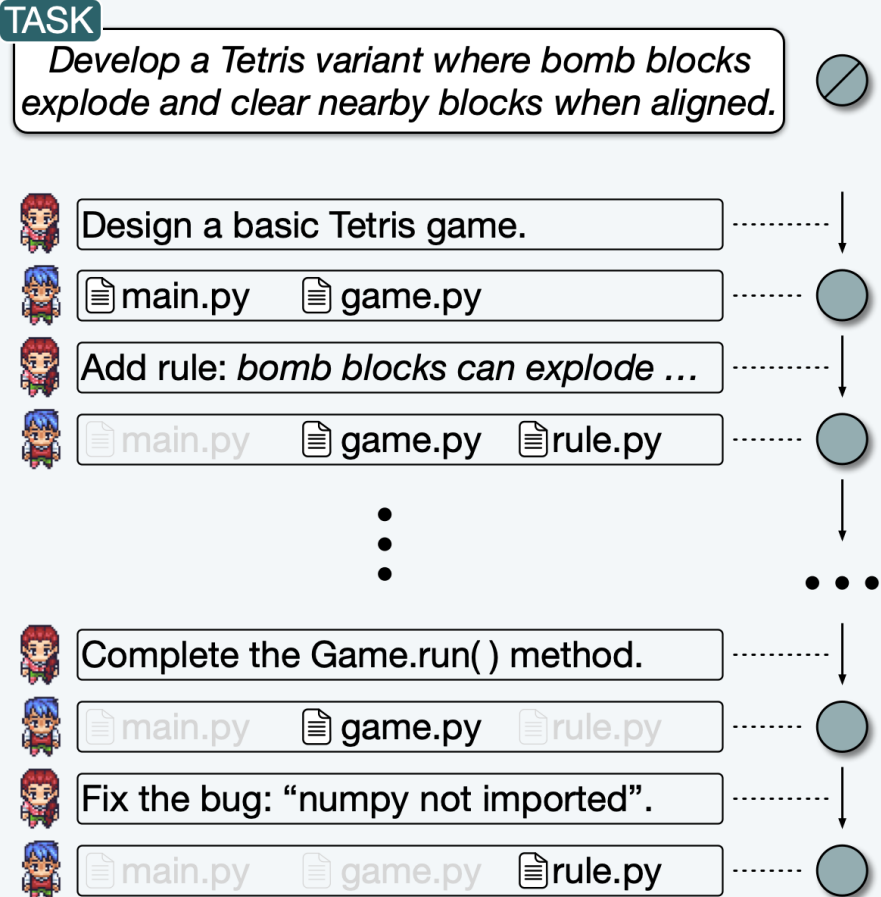
AgentVerse



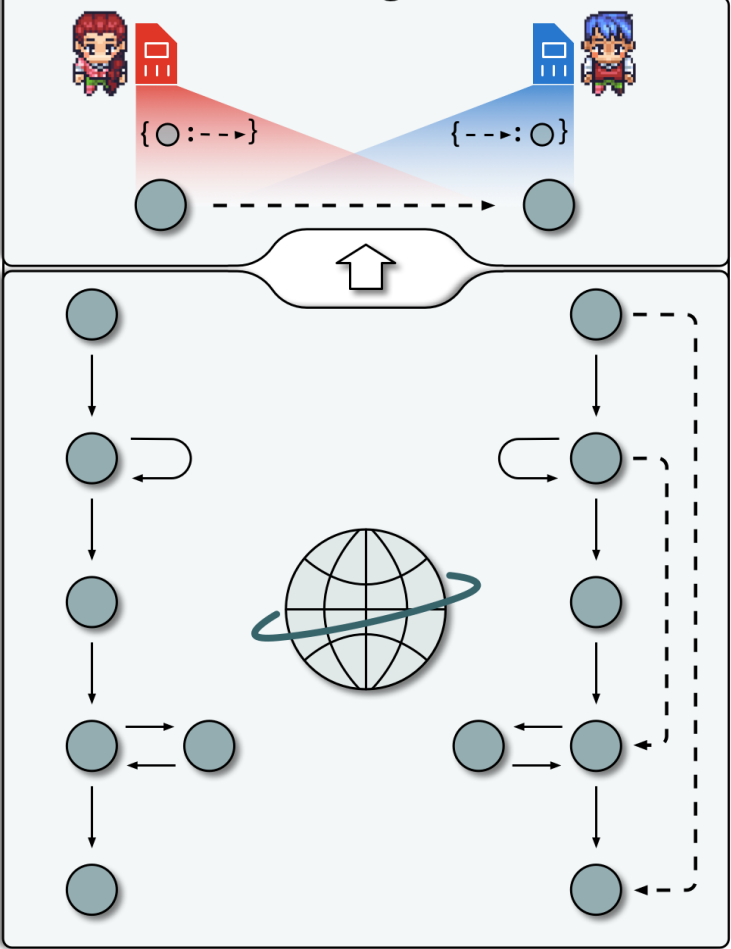
ChatDev 多角色 开发流程



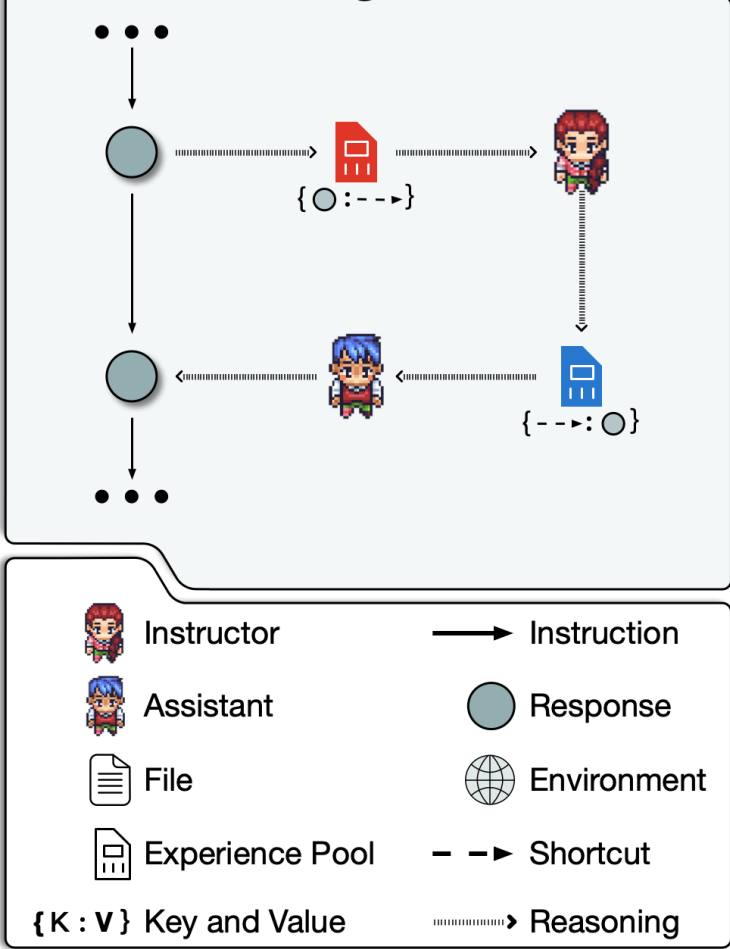
Co-Tracking



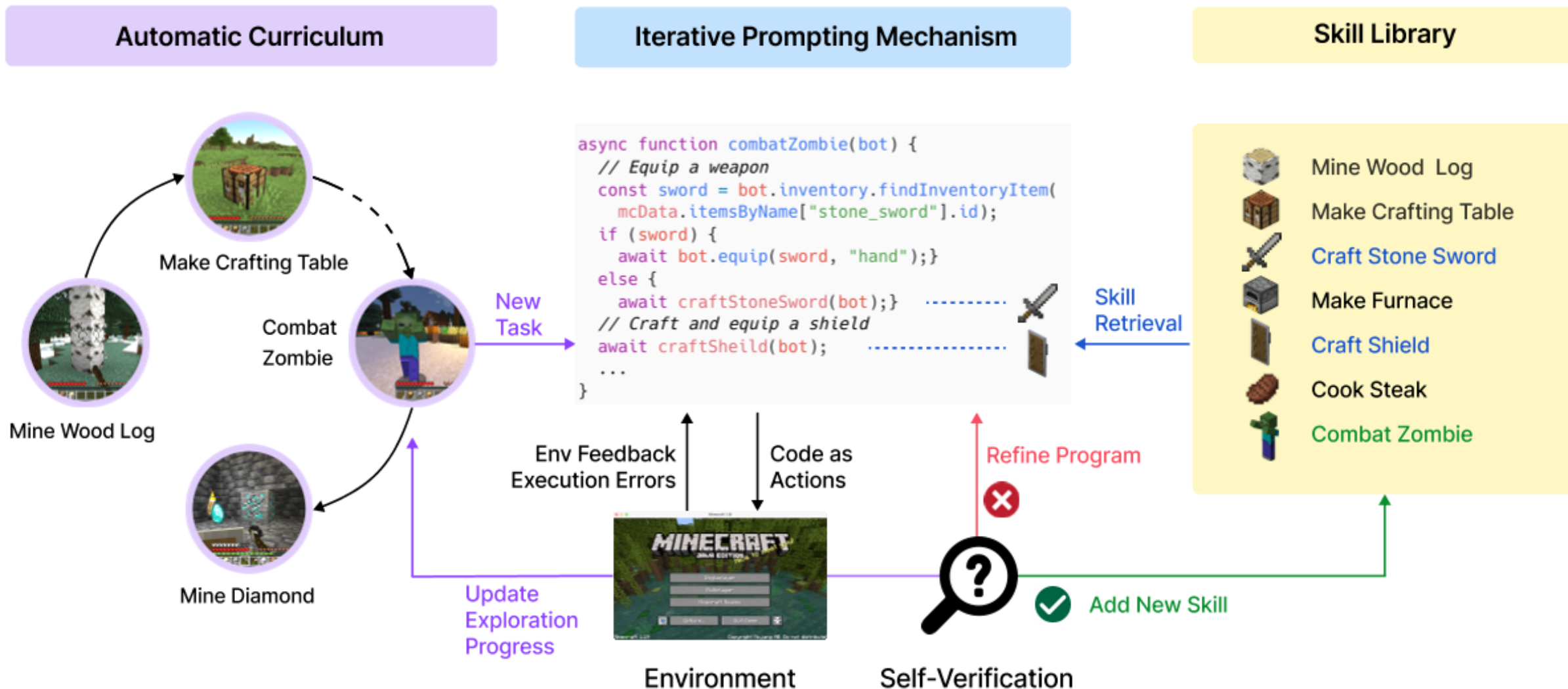
Co-Memorizing



Co-Reasoning



Voyager



01

多智能体的研究
概述

02

经典多智能体框
架解析

03

基于有向无环图
的多智能体框架
VilagerAgent

04

基准测试对比及
在多智能体协作
应用探讨

基于有向无环图的多智能体框架 VilagerAgent

目标:

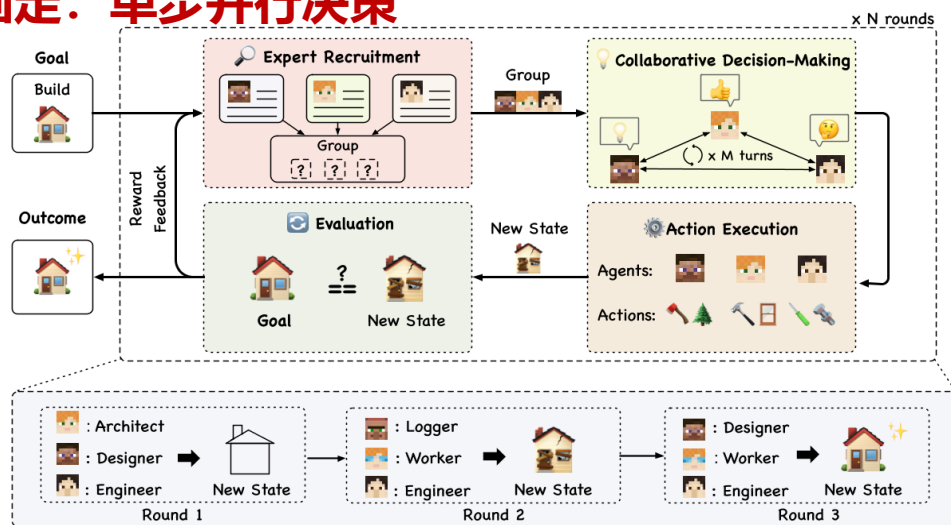
- 提升多智能体系统在**复杂任务**中的**协作能力**, 开发解决复杂、动态**依赖关系**的多智能体框架

挑战:

- 传统的多智能体系统在处理复杂的空间、因果和时间依赖关系受框架局限, 且**幻觉问题**导致决策可控性不强。

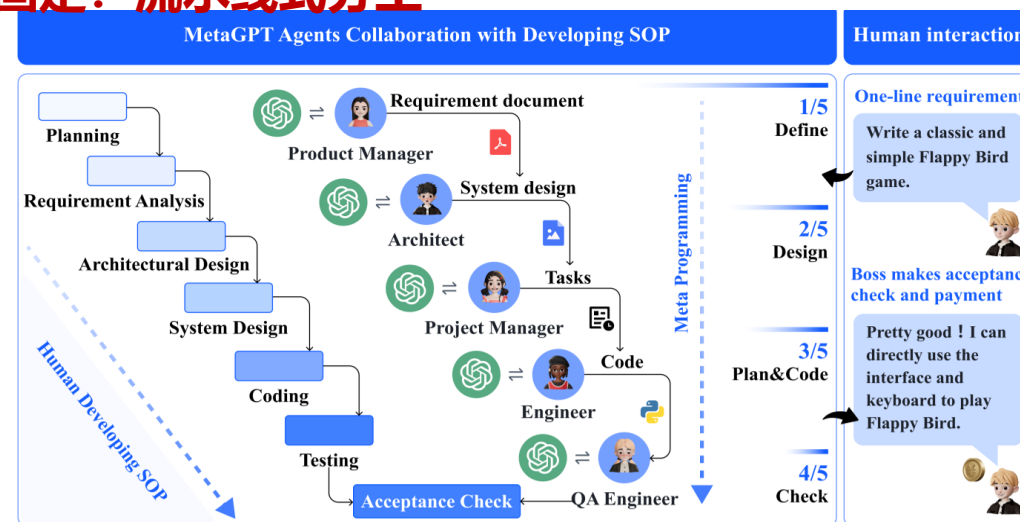
幻觉问题导致结果不可控 分工方式固定导致任务局限性

固定: 单步并行决策



AgentVerse多智能体框架

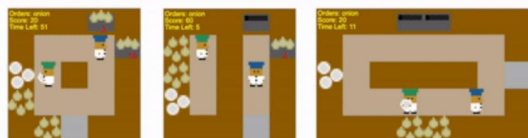
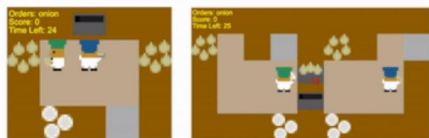
固定: 流水线式分工



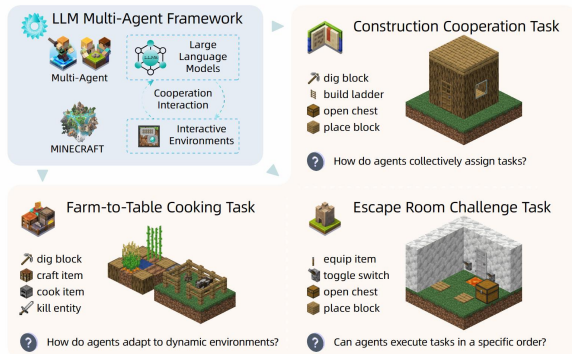
MetaGPT多智能体框架

基于有向无环图的多智能体框架 *VillagerAgent*

方法：实现基于有向图的多智能体协作框架，在Minecraft测试环境进行多种复杂能力测试



Overcooked-ai 测试环境



VillagerBench 测试环境



Construction Cooperation

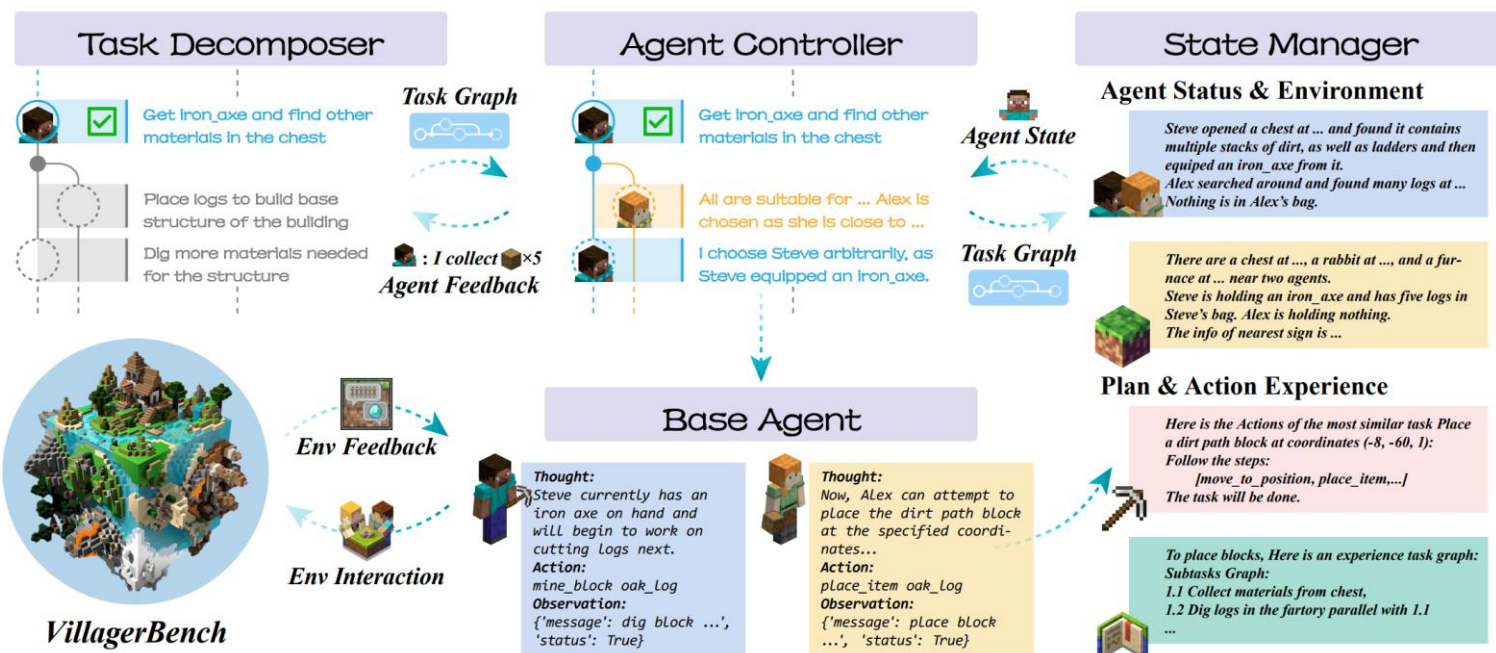


Farm-to-Table Cooking



Escape Room Challenge

Scenario 1: Using the provided blueprint, please collaborate to build in Minecraft.



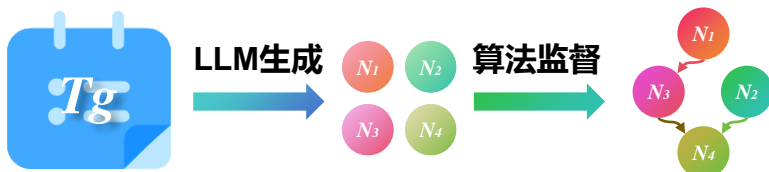
VillagerBench

VillagerAgent 多智能体决策框架

基于有向无环图的多智能体框架 *VilagerAgent*

方法：实现基于有向图的多智能体协作框架，在Minecraft测试环境进行多种复杂能力测试

任务分解&更新



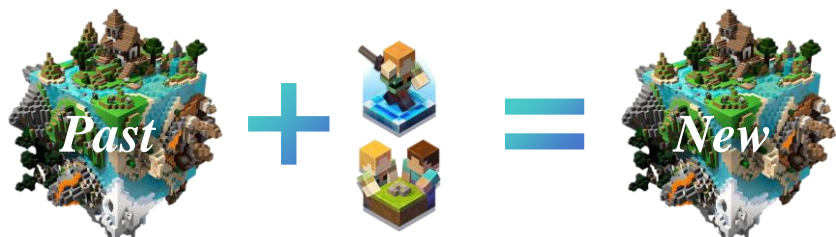
基于LLM自回归特性，提出任务分解和子任务图更新算法

子任务分配



提出显示的可控子任务分配算法，将子任务异步分配给多智能体

全局状态管理



实现管理器，更新智能体状态和全局环境信息



Construction Cooperation



Farm-to-Table Cooking



Escape Room Challenge

VillagerBench 测试环境

Algorithm 1 Convert Task List to Graph

```
1:  $G \leftarrow (V, E)$  with  $V \leftarrow \emptyset, E \leftarrow \emptyset$ 
2:  $L \leftarrow [N_1, N_2, \dots, N_n]$   $\triangleright$  Input list
3: for  $i \leftarrow 1$  to  $n$  do
4:    $V \leftarrow V \cup \{N_i\}$   $\triangleright$  Add element as a node
5:   if  $P(N_i) \neq \emptyset$  then
6:     for all  $p_j \in P(N_i)$  do
7:        $E \leftarrow E \cup \{(p_j, N_i)\}$   $\triangleright$  Add edges
      from predecessors
8:     end for
9:   else if  $i > 1$  then
10:    for all  $p_k \in P(N_{i-1})$  do
11:       $E \leftarrow E \cup \{(p_k, N_i)\}$   $\triangleright$  Share
      predecessors with previous element
12:    end for
13:  end if
14: end for
```

Algorithm 2 Find Ready-to-Execute Tasks

```
Require:  $G = (V, E)$   $\triangleright$  Task graph with nodes
and edges
Require:  $S \subseteq V$   $\triangleright$  Set of successfully executed
tasks
Require:  $U \subseteq V$   $\triangleright$  Set of unexecuted tasks
1:  $R \leftarrow \emptyset$   $\triangleright$  Result set of ready-to-execute tasks
2: for all  $N_i \in U$  do
3:    $P(N_i) \leftarrow \{p_j \mid (p_j, N_i) \in E\}$   $\triangleright$  Find
predecessors of  $N_i$ 
4:   if  $P(N_i) = \emptyset$  or  $P(N_i) \subseteq S$  then
5:      $R \leftarrow R \cup \{N_i\}$   $\triangleright$  Add if no
predecessors or all predecessors executed
6:   end if
7: end for
8: return  $R$ 
```


基础智能体的行为空间：40+ API



1 LEADER

getInfos->
readPlanList ->
assignPlan->
updatePlanState->
...



14 AGENTS

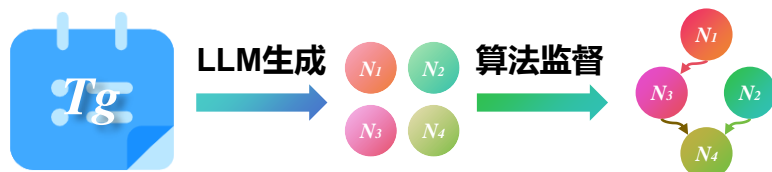
getInfos ->
receivePlan ->
searchChest ->
gotoChest ->
interactChest ->
equipWood->
gotoTargetPlace->
placeBlock
...



基于有向无环图的多智能体框架 *VilagerAgent*

方法：实现基于有向图的多智能体协作框架，在Minecraft测试环境进行多种复杂能力测试

任务分解&更新



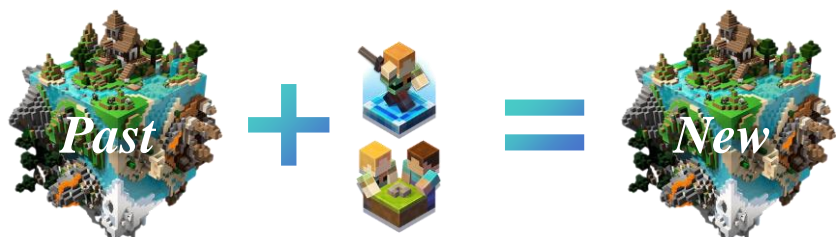
基于LLM自回归特性，提出任务分解和子任务图更新算法

子任务分配



提出显示的可控子任务分配算法，将子任务异步分配给多智能体

全局状态管理



实现管理器，更新智能体状态和全局环境信息



VillagerBench 测试环境

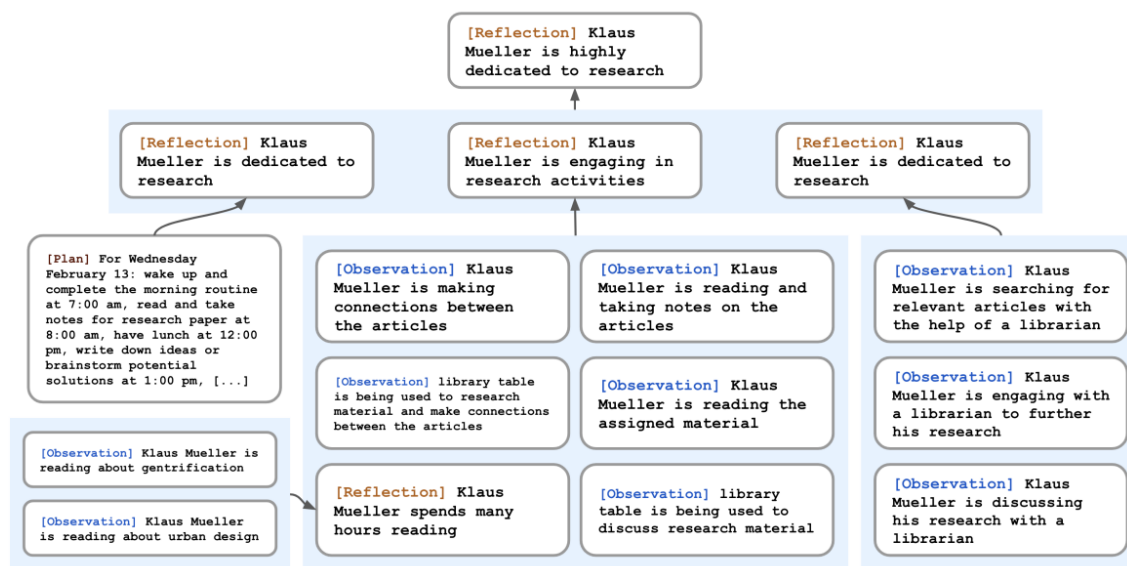


Figure 7: A reflection tree for Klaus Mueller. The agent's observations of the world, represented in the leaf nodes, are recursively synthesized to derive Klaus's self-notion that he is highly dedicated to his research.

基于有向无环图的多智能体框架 VilagerAgent

分析幻觉成因：多轮上下文对话、大范围规划

Task: Farming_20
Environment: bucket and egg in chest, wheat and sugarcane in farm
Agent num: 2

VillagerAgent

我们方法

```
Plan:
[
  {
    "id": 1,
    "description": "Harvest wheat and craft into wheat blocks if necessary",
    "milestones": [
      "Navigate to wheat at [45, -59, 129] and [45, -59, 131]",
      "Harvest a total of 3 wheat",
      "Craft wheat into wheat blocks if less than 3 wheat is harvested"
    ],
    "retrieval paths": [
      "~/meta-data/ingredients/3"
    ],
    "required subtasks": [],
    "assigned agents": ["Alice"]
  },
  {
    "id": 2,
    "description": "Find sugarcane or honey bottles to craft sugar",
    "milestones": [
      "Scan for sugarcane or honey bottles in the environment or chests",
      "Navigate to the location of sugarcane or honey bottles",
      "Collect or withdraw 2 sugar canes or honey bottles",
      "Craft 2 sugars from the collected items"
    ],
    "retrieval paths": [
      "~/meta-data/ingredients/1",
      "~/meta-data/ingredients/2"
    ],
    "required subtasks": [],
    "assigned agents": ["Bob"]
  }
]

Action:
[...]
```

正确

AgentVerse

对比方法

```
Discussion:
[Alice]: Hey Bob, to make a cake we need 3 buckets of milk, 2 sugar, 1 egg, and 3 wheat. Let's check the chests for these ingredients first. If we don't find them, we'll need to gather them. How about I focus on getting the milk and wheat, and you focus on finding sugar and an egg? If we find extra ingredients in the chests, we can adjust our plan accordingly.
[Bob]: Sounds good, Alice. I'll start by checking the chests for sugar and an egg. If I don't find them there, I'll go look for sugarcane to make sugar and search for chickens to get an egg. Let's meet back here with the ingredients we've gathered. If we both end up finding some of the same ingredients, we can share them to make sure we have everything we need. Let's get to it!
[Alice]: Bob, I've managed to get the 3 buckets of milk and found 2 wheat in the chests. How did your search go? Do you have the sugar and egg? If not, we should split up to find the remaining wheat, sugar, and egg. I can go harvest or find another wheat, and if you haven't found sugar and egg yet, you could continue searching for those. Let's update each other on what we still need.
[Bob]: ...

Action:
[...]
```

hallucination: Alice hasn't done it

幻觉：Alice并未执行相关指令

```
Discussion:
[Alice]: Bob, great work on finding the sugar cane and the egg. Since you've got the sugar cane, could you please craft the 2 additional sugar we need? Also, since you found iron ingots, please craft the third bucket. I'll go ahead and collect the 3 wheat from the farm now. Once we have all the ingredients, we can start making the cake. Let me know once you've crafted the sugar and the bucket.
[Bob]: Alice, I'm on it. I'll craft the 2 additional sugar from the sugar cane and then use the iron ingots to craft the third bucket. After that, I'll have all the items we need from my side. I'll wait for you to collect the wheat, and then we can combine our ingredients to make the cake. I'll let you know as soon as I've finished crafting. Let's keep up the good work!
[Alice]: ...
[Bob]: ...

Action:
[...]
```

hallucination: no iron ingots in chest

幻觉：铁锭不在箱子中

```
Observation: {message: "don't have iron ingots in chest", status: False}
```

失败：幻觉导致错误决策

我们基于有向图的LLM决策结构规避多轮对话决策模式，同时能够有效提取出子任务划分和执行结构

我们发现幻觉往往出现在多轮对话后期。LLM在模仿前文信息，生成下文的过程中会增添不存在的事实。

幻觉在接下来的多轮对话中被进一步扩大，新的虚假信息生成。

错误的信息最终导致了错误的决策方案，并导致执行失败。

01

多智能体的研究
概述

02

经典多智能体框
架解析

03

基于有向无环图
的多智能体框架
VilagerAgent

04

基准测试对比及
在多智能体协作
应用探讨

基于有向无环图的多智能体框架 VilagerAgent

复杂测试：基于有向图的多智能体协作框架，GPT-4 优于其他模型效果

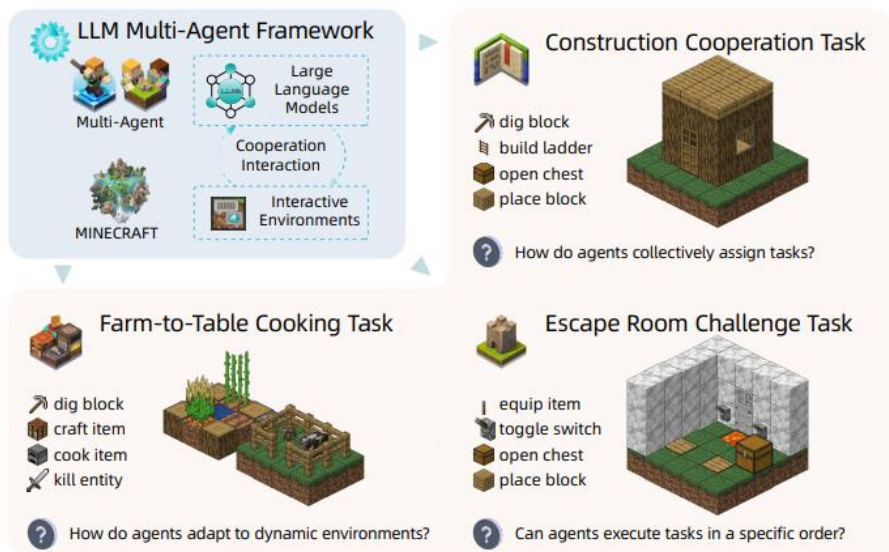


Figure 1: Minecraft Multi-Agent Benchmark (VillagerBench) is the first multi-scenario benchmark designed to evaluate the cooperative capabilities of multi-agent systems within the real-world context of Minecraft.

$$\text{Completion (C)} = \frac{\# \text{ Indicators Detected}}{\# \text{ Total Indicators Expected}}$$

完成度指标 C

$$\text{Efficiency (E)} = \frac{\# \text{ Task Completion Rate}}{\# \text{ Total Execution Time}}$$

执行效率 E

$$t' = \frac{t - \min(t)}{\max(t) - \min(t)}$$

$$\text{Balance(B)} = 1 - \sigma(t')$$

各智能体负载平衡度 B

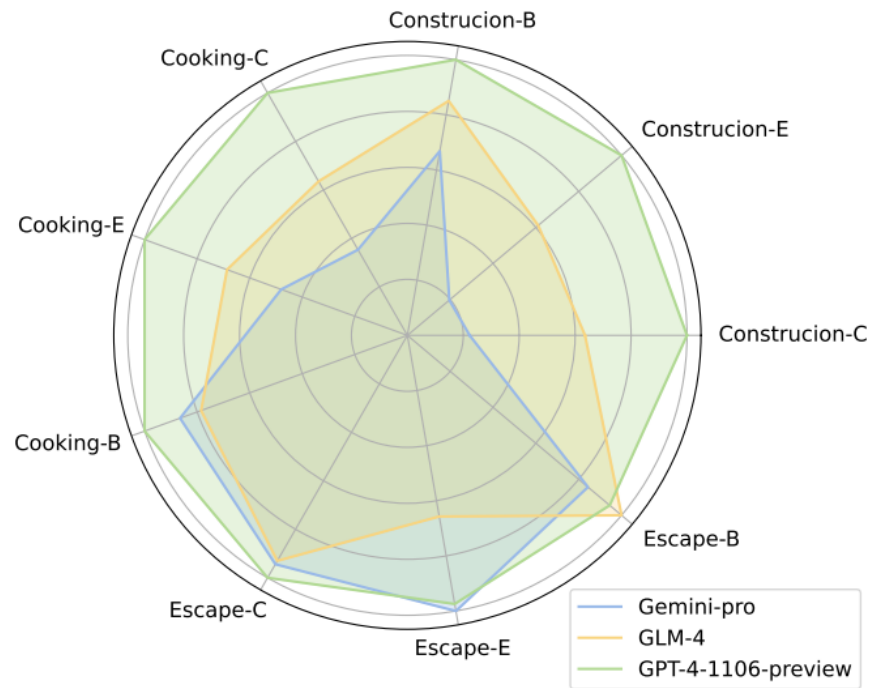


Figure 4: Comparison of LLMs on VillagerBench. We show the relative performance gap against the best in each scenario. GPT-4-1106-preview achieves higher scores across most metrics, whereas Gemini-Pro demonstrates better efficiency in the Escape Room Challenge.

基于有向无环图的多智能体框架 *VilagerAgent*



复杂测试：基于有向图的多智能体协作框架，在Minecraft测试环境中表现优于AgentVerse方法

Task: Farming_20

Environment: bucket and egg in chest, wheat and sugarcane in farm

Agent num: 2

DAGENT 我们方法

Plan:

```
[
  {
    "id": 1,
    "description": "Harvest wheat and craft into wheat blocks if necessary",
    ...
    "assigned agents": ["Alice"]
  },
  {
    "id": 2,
    "description": "Find sugar cane or honey bottles to craft sugar",
    ...
    "assigned agents": ["Bob"]
  }
]
```

正确

AgentVerse 对比方法

Discussion:

[Alice]: Hey Bob, to make a cake we need 3 buckets of milk, 2 sugar, 1 egg, and 3 wheat. Let's check the chests for these ingredients first. If we don't find them, we'll need to gather them. How about I focus on getting the milk and wheat, and you focus on finding sugar and an egg? If we find extra ingredients in the chests, we can adjust our plan accordingly. What do you think? ✓

[Bob]: Sounds good, Alice. I'll start by checking the chests for sugar and an egg. If I don't find them there, I'll go look for sugarcane to make sugar and search for chickens to get an egg. Let's meet back here with the ingredients we've gathered. If we both end up finding some of the same ingredients, we can share them to make sure we have everything we need. Let's get to it! ✓

[Alice]: Bob, I've managed to get the 3 buckets of milk and found 2 wheat in the chests. How did your search go? Do you have the sugar and egg? If not, we should split up to find the remaining wheat, sugar, and egg. I can go harvest or find another wheat, and if you haven't found sugar and egg yet, you could continue searching for those. Let's update each other on what we still need. hallucination: Alice hasn't done it

[Bob]: ...

幻觉：Alice并未执行相关指令

Models	Cooking Task Avg. Score			
	C (%)	ACR	E (%/min)	B (%)
AgentVerse gpt	29.75	48.64	3.54	87.13
VillagerAgent	26.05	32.92	3.35	83.15
gemini				
VillagerAgent	46.84	54.07	4.79	75.46
glm				
VillagerAgent gpt (2-agents)	73.75	58.11	6.98	96.13
VillagerAgent gpt (3-agents)	85.26	55.60	21.90	84.38

Table 3: Performance comparison between AgentVerse(Chen et al., 2023) and VillagerAgent on the Farm-to-Table Task. Note that gpt refers to GPT-4-1106-preview, gemini to Gemini-Pro, and glm to GLM-4

扩展评估：在Overcooked-AI 中，我们的方法表现**优于ProAgent**

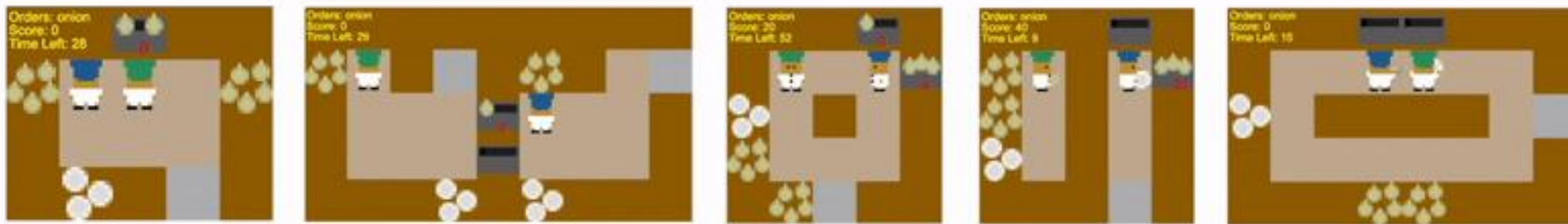


Figure 6: Overcooked-AI Scenarios

Layout	PBT	FCP	MEP	COLE	ProAgent	VillagerAgent (ours)
Cramped Room	178.8 ± 16.5	196.3 ± 16.8	185 ± 15	163.8 ± 24.1	197.3 ± 6.1	213.3 ± 9.43
Asymmetric Advantages	182.2 ± 27.9	185.7 ± 22.7	155.7 ± 63.9	201.3 ± 34.5	228.7 ± 23	304 ± 8.76
Coordination Ring	141.3 ± 28	148.8 ± 19.4	167.2 ± 22.4	168.8 ± 26.1	175.3 ± 29	226.7 ± 18.9
Forced Coordination	15.3 ± 17.1	44.7 ± 36.4	23.3 ± 19.8	24 ± 21.8	49.7 ± 33.1	120 ± 16.97
Counter Circuit	64.7 ± 45.9	58.3 ± 37.5	74.3 ± 39.1	95.5 ± 25.2	126.3 ± 32.3	148 ± 4.38

Table 5: Performance comparison of VillagerAgent and ProAgent across different scenarios in Overcooked-AI.

[build_judge: 已用[蓝色带釉陶瓦]替换了位于142, 64, -20的物品格]
[build_judge: 已用[云杉木门]替换了位于142, 64, -20的物品格]
[build_judge: 已用[砖块墙]替换了位于142, 64, -20的物品格]
[build_judge: 已用[白色带釉陶瓦]替换了位于142, 64, -20的物品格]
[build_judge: 已用[龙首]替换了位于142, 64, -20的物品格]
[build_judge: 已用[苦力怕的头]替换了位于142, 64, -20的物品格]
[build_judge: 已用[僵尸的头]替换了位于142, 64, -20的物品格]
[build_judge: 已用[铁块]替换了位于142, 64, -20的物品格]
[build_judge: 已用[灯笼]替换了位于142, 64, -20的物品格]
[build_judge: 已用[红色旗帜]替换了位于142, 64, -20的物品格]
[build_judge: 已用[红色旗帜]替换了位于142, 64, -20的物品格]
[build_judge: 已用[荧石]替换了位于142, 64, -20的物品格]
[build_judge: 已用[淡蓝色染色玻璃]替换了位于142, 64, -20的物品格]
[build_judge: 已用[白色染色玻璃]替换了位于142, 64, -20的物品格]

<Dorothy> What should I do next?

<build_judge> Time is 1207

<Mabel> I am thinking

<Glenda> What should I do next?

<Margaret> Well .. Let me think about it

<build_judge> Time is 1427

Dorothy talk_to I want to build a house executing book task 01 to task 50, can you help me?_



VillagerAgent: 减少幻觉提高任务分解效率的多智能协作体框架

Yubo Dong, Xukun Zhu, Zhengzhe Pan, Linchao Zhu, Yi Yang
ReLER Lab, CCAI, Zhejiang University

<https://cnsdqd-dyb.github.io/VillagerAgent.github.io/2024/04/01/VillagerAgent.html>