MindAgent: Emergent Gaming Interaction

Motivation

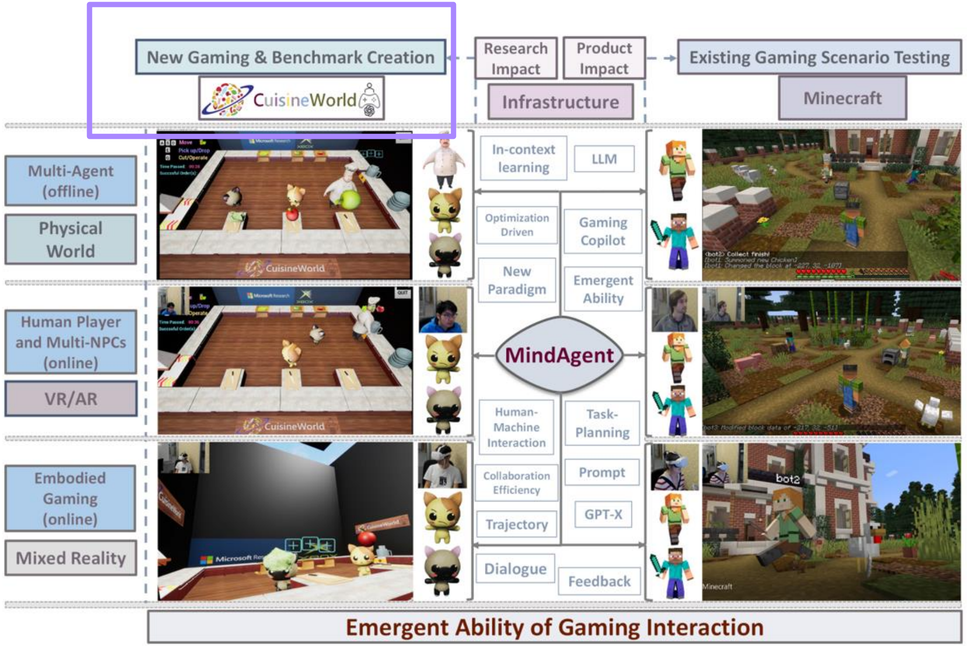
1. LLMs have demonstrated **strong planning abilities** in various domains (e.g., robotics, reasoning, task automation).
2. However, their capabilities in multi-agent planning and coordination remain largely **unexplored**.
3. The authors want to **evaluate how well LLMs can handle multi-agent planning**, specifically in gaming environments **where multiple agents must collaborate (LLM and human NPCs)**

Do you agree with these statements?

- "LLMs can perform zero-shot multi-agent planning, scheduling multiple agents into completing tasks without explicit training."
  → Do you believe the LLM really understands coordination, or is it just pattern-matching from prompts?
- "Without environmental feedback, LLMs make repeated errors, indicating that structured prompts are crucial for efficient planning."
  → If the LLM relies heavily on feedback, does it mean it lacks true adaptability?
- "LLMs exhibit emergent reasoning capabilities, successfully dispatching more agents than than the number seen in its prompt examples."
  → Do you believe this prove generalization ability can be extended to larger scale such as 10 or 100?

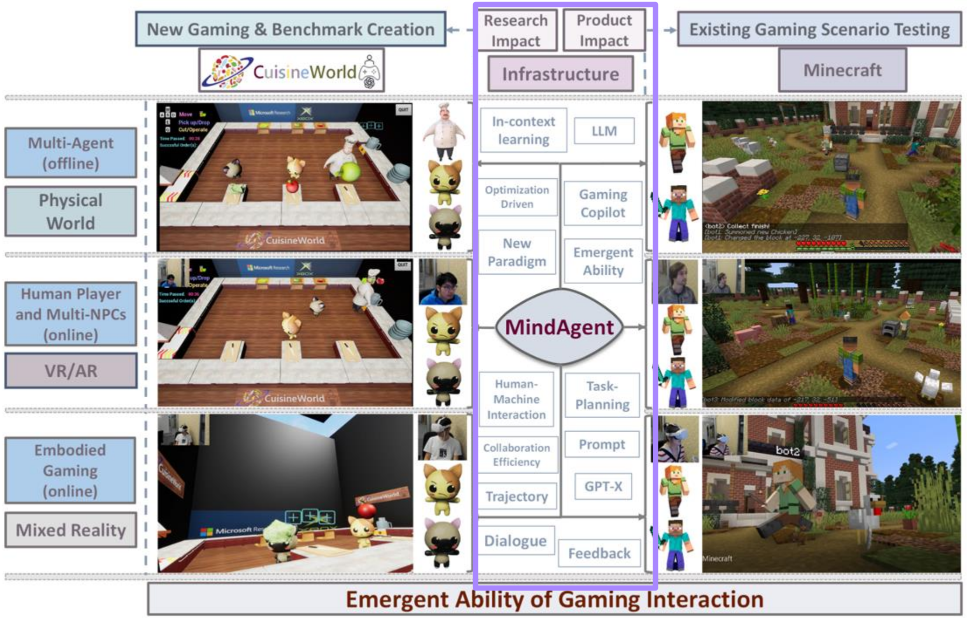
Contribution Overview

1. Created a New Gaming Benchmark
   – CUISINEWORLD



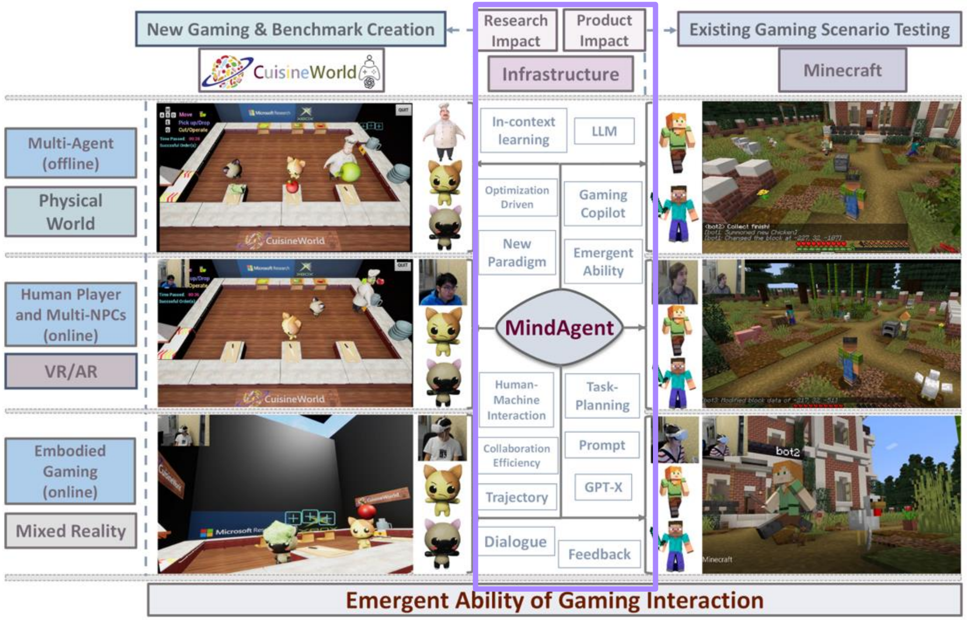
Contribution Overview

1. Created a New Gaming Benchmark
   – CUISINEWORLD
2. **Developed MINDAGENT**



Contribution Overview

1. Created a New Gaming Benchmark – CUISINEWORLD
2. Developed MINDAGENT
3. Tested LLMs on the Benchmark



Contribution Overview

1. Created a New Gaming Benchmark – CUISINEWORLD
2. Developed MINDAGENT
3. Tested LLMs on the Benchmark
4. Applied AI in Real-World Gaming Scenarios

New Gaming Benchmark – CUISINEWORLD

Benchmark	Multi-task	Object Interaction	Tool Use	Maximum Agents	Collabo-ration	Human in-the-loop	Procedural Level Generation
ALFWorld (Shridhar et al., 2020)	✓	✓	✓	1	✗	✗	✗
WAH (Puig et al., 2020)	✓	✓	✗	2	✓	✓	✗
TextWorld (Côté et al., 2019)	✓	✓	✓	1	✗	✗	✓
Generative Agents (Park et al., 2023)	✓	✓	✓	25	✗	✗	✓
EMATP (Liu et al., 2022)	✓	✓	✓	2	✓	✗	✗
Overcooked-AI (Carroll et al., 2019)	✗	✓	✓	2	✓	✓	✗
HandMeThat (Wan et al., 2022)	✓	✓	✓	2	✓	✗	✗
DialFRED (Gao et al., 2022)	✓	✓	✓	2	✓*	✗	✗
TEACH (Padmakumar et al., 2022)	✓	✓	✓	2	✓*	✗	✗
CerealBar (Suhr et al., 2019)	✗	✗	✗	2	✓	✗	✗
LIGHT (Urbanek et al., 2019)	✓	✗	✗	1369	✗	✓	✓
Diplomacy (Bakhtin et al., 2022)	✗	✗	✗	7	✓	✓	✗
CUISINEWORLD (Ours)	✓	✓	✓	4+	✓	✓	✓

New Gaming Benchmark – CUISINEWORLD

1. Replace API call with text-based action command
2. Implement in-context learning

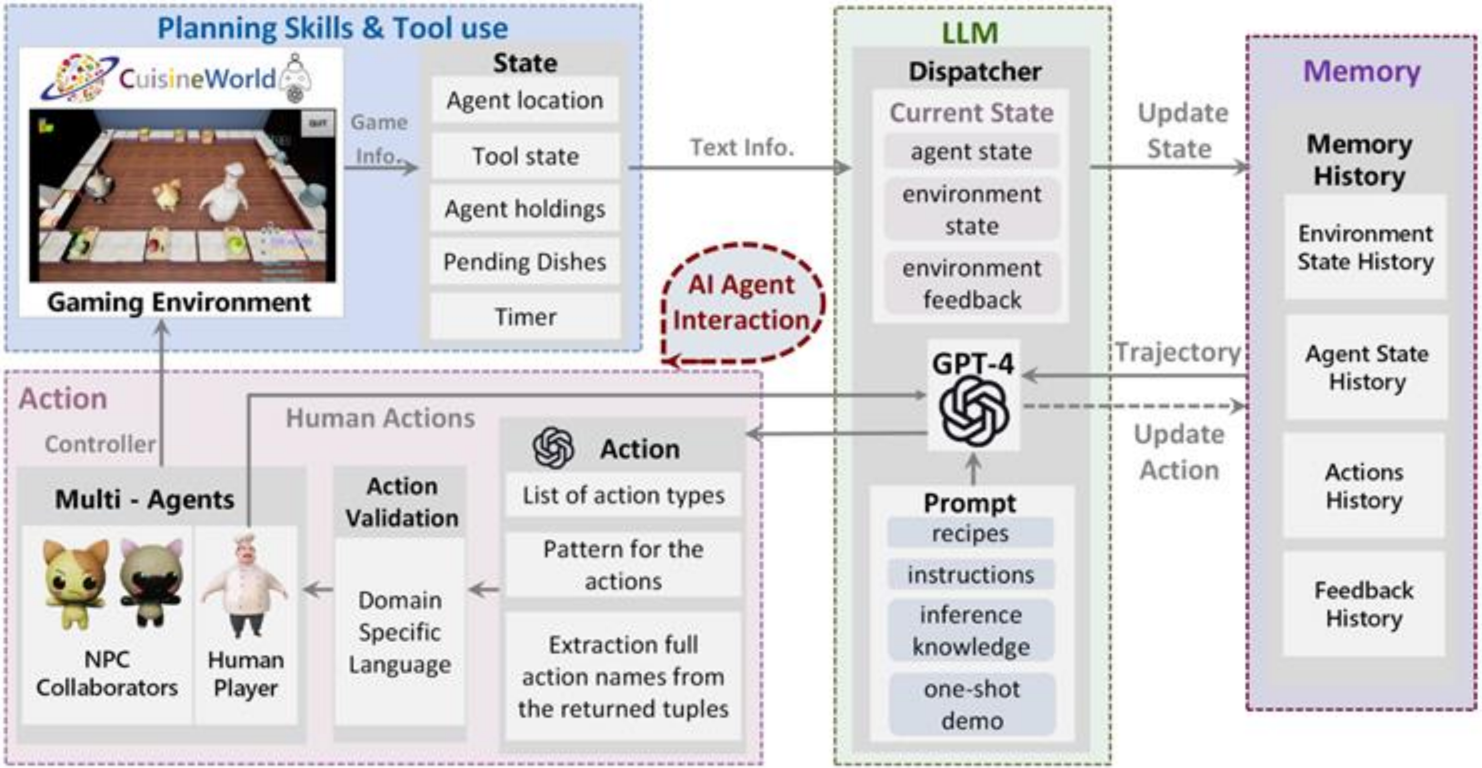
◆ **Example: Traditional API vs. LLM-Controlled Agent**

Action	Overcooked/Minecraft API Call	LLM-Generated Command
Move to storage	`agent.move_to(storage)`	`"goto(agent1, storage)"`
Pick up ingredient	`agent.pick("onion")`	`"get(agent1, storage, onion)"`
Cook food	`agent.use_tool("oven")`	`"activate(agent1, oven)"`
Serve food	`agent.place("dish, table")`	`"put(agent1, serving_table)"`

🔄 **How LLM Thinks:**

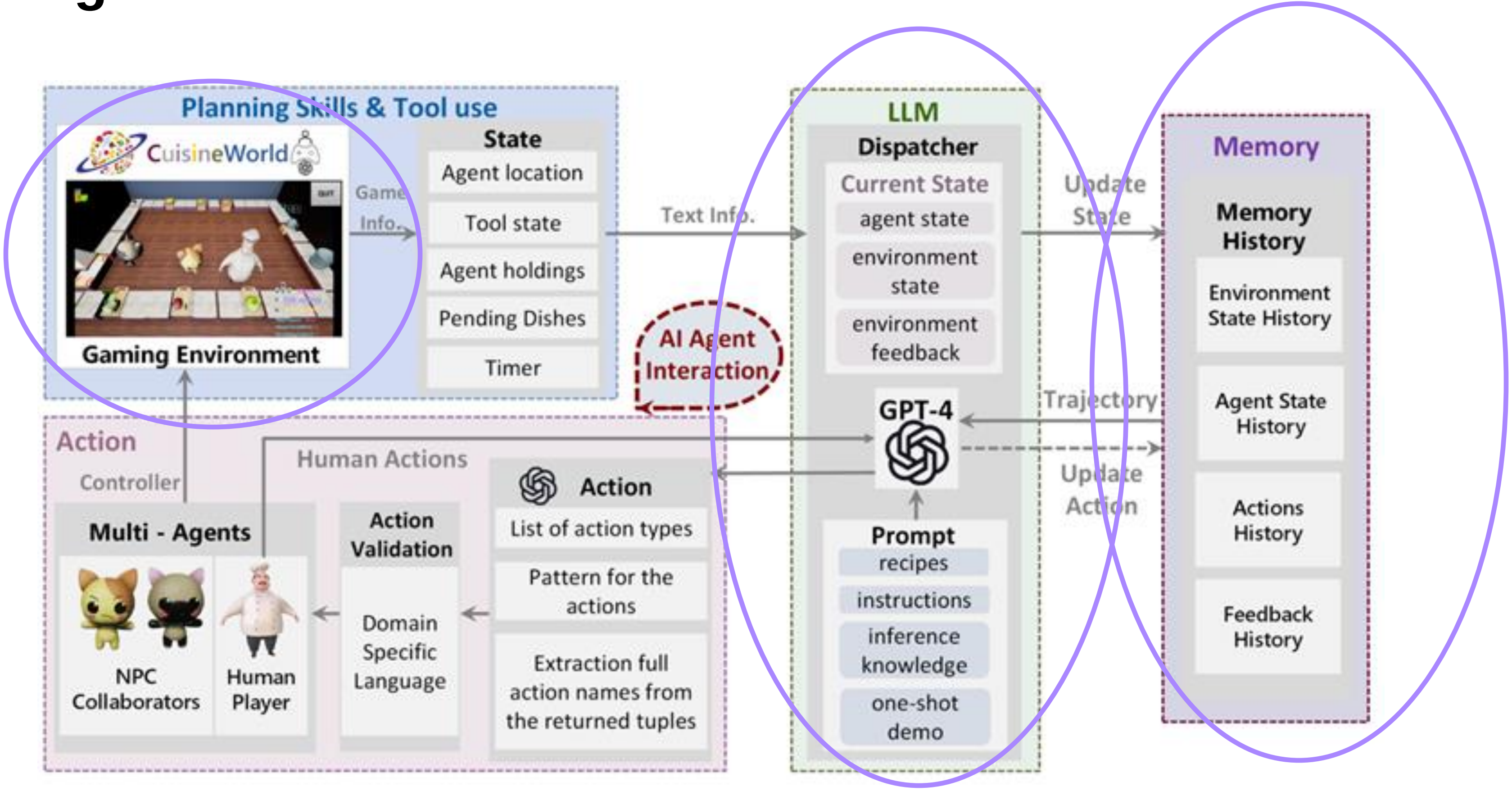
- Reads `"Agent 1 is at storage. There is an onion."`
- Determines **what step to take next**.
- Generates `"get(agent1, storage, onion)"`.

MindAgent



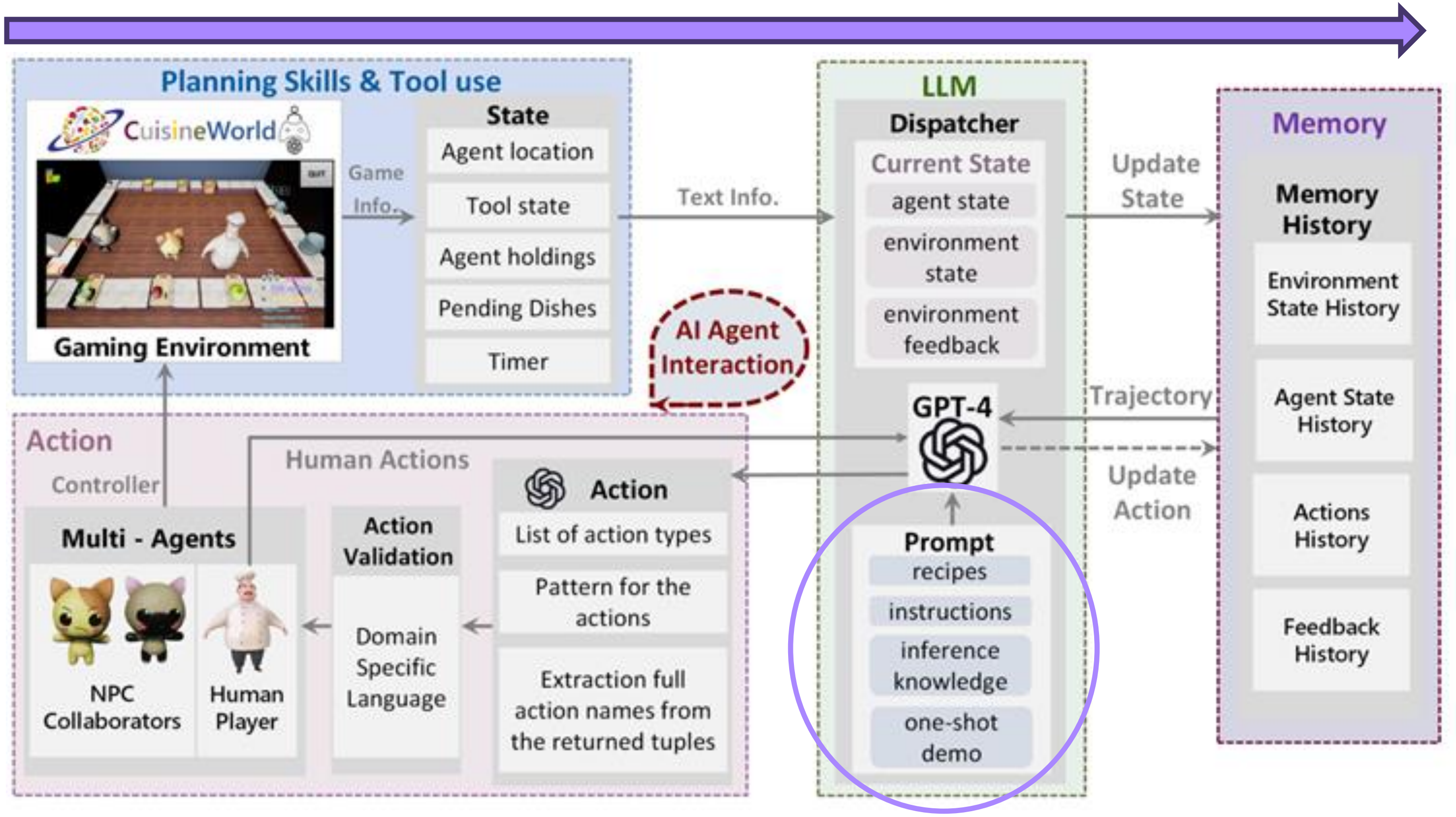
Planning Skills & Tool use
CuisineWorld
Gaming Environment
Game Info.
State
Agent location
Tool state
Agent holdings
Pending Dishes
Timer
Text Info.
AI Agent Interaction
LLM
Dispatcher
Current State
agent state
environment state
environment feedback
GPT-4
Prompt
recipes
instructions
inference knowledge
one-shot demo
Update State
Trajectory
Update Action
Memory
Memory History
Environment State History
Agent State History
Actions History
Feedback History
Action
Controller
Human Actions
Multi - Agents
NPC Collaborators
Human Player
Action Validation
Domain Specific Language
Action
List of action types
Pattern for the actions
Extraction full action names from the returned tuples

MindAgent

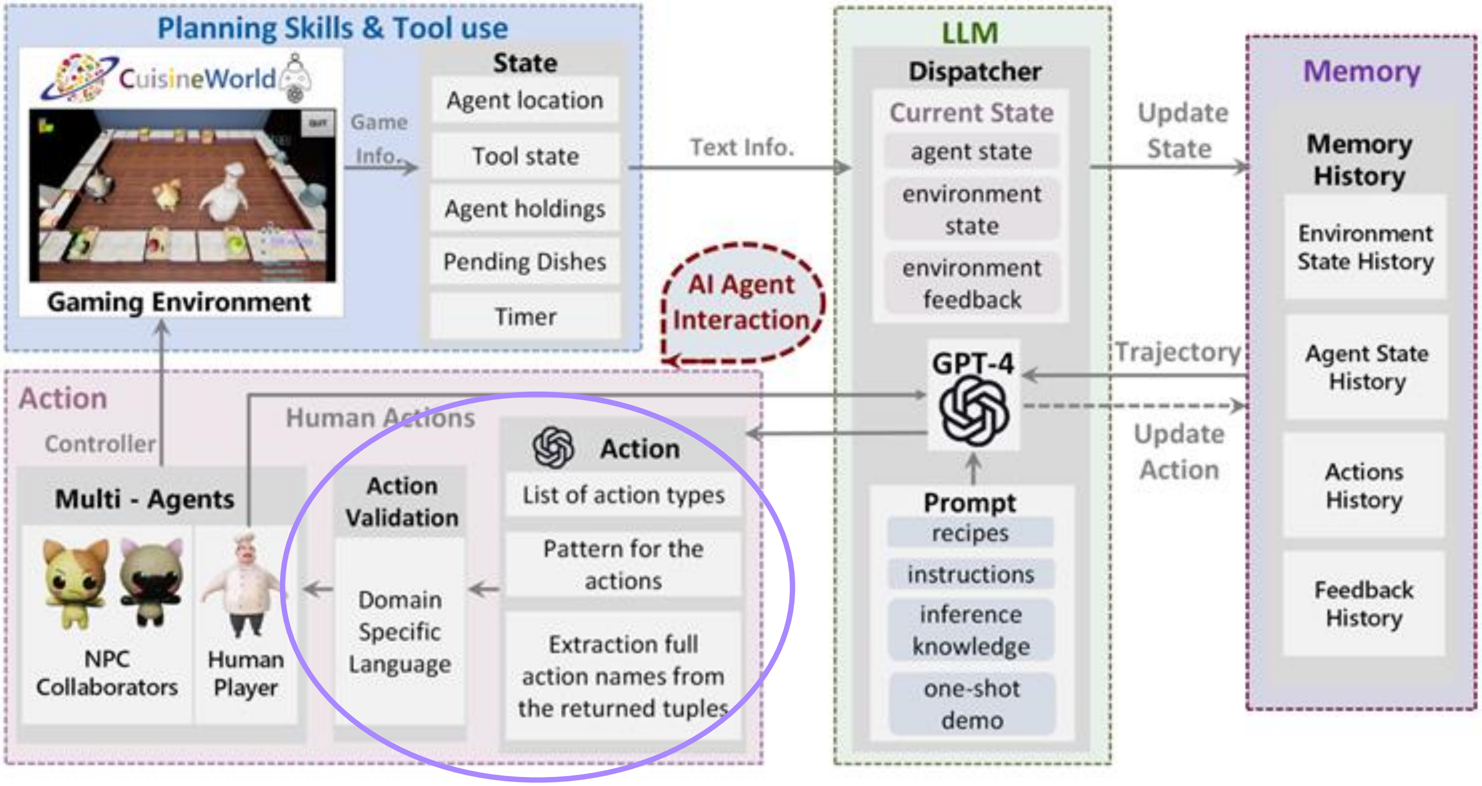


Planning Skills & Tool use
CuisineWorld
Gaming Environment
Game Info.
State
Agent location
Tool state
Agent holdings
Pending Dishes
Timer
Text Info.
AI Agent Interaction
LLM
Dispatcher
Current State
agent state
environment state
environment feedback
GPT-4
Prompt
recipes
instructions
inference knowledge
one-shot demo
Update State
Trajectory
Update Action
Memory
Memory History
Environment State History
Agent State History
Actions History
Feedback History
Action
Controller
Human Actions
Multi - Agents
NPC Collaborators
Human Player
Action Validation
Domain Specific Language
Action
List of action types
Pattern for the actions
Extraction full action names from the returned tuples

MindAgent

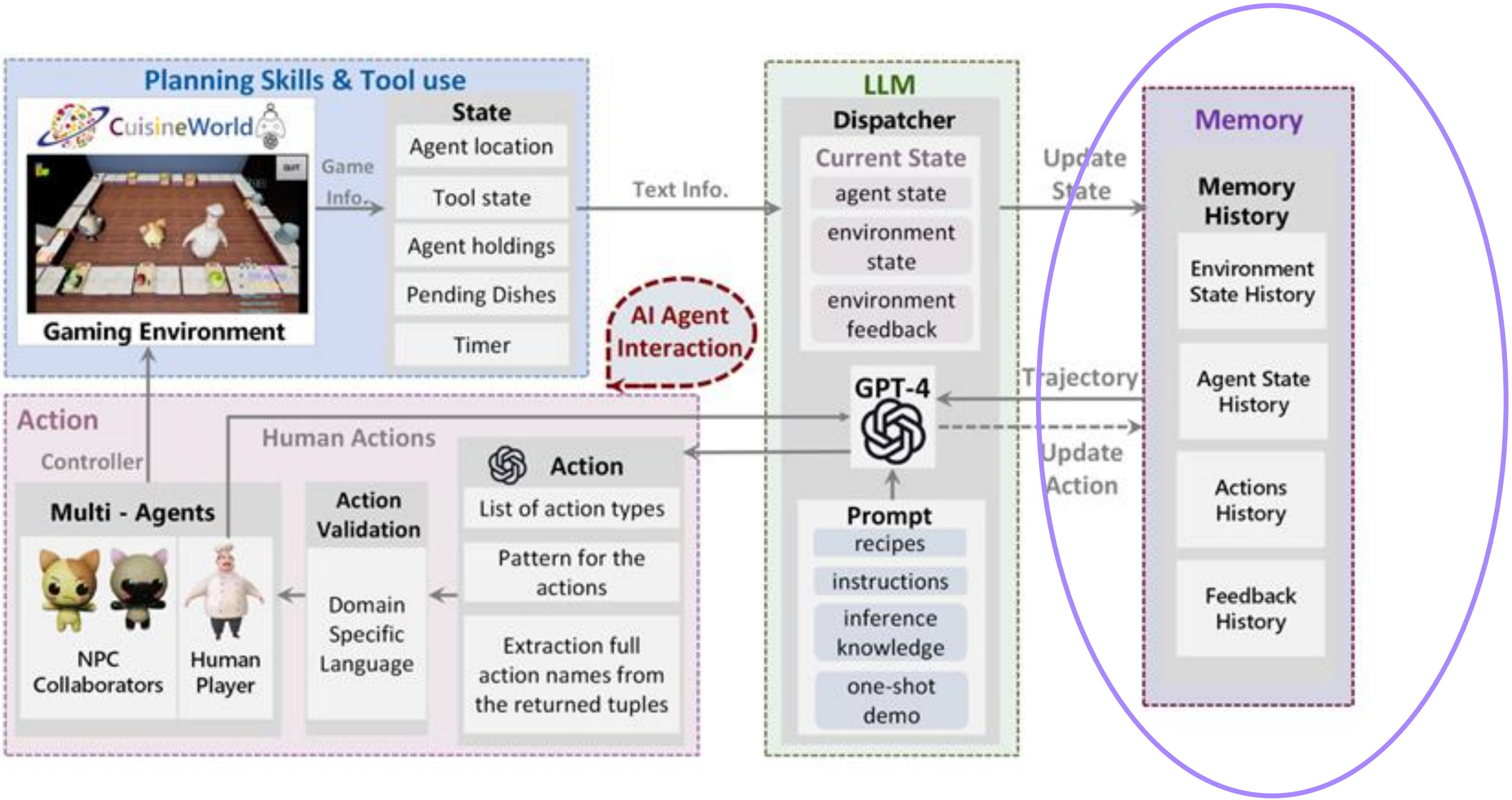

Planning Skills & Tool use
CuisineWorld
Gaming Environment
Game Info.
State
Agent location
Tool state
Agent holdings
Pending Dishes
Timer
Text Info.
AI Agent Interaction
LLM
Dispatcher
Current State
agent state
environment state
environment feedback
GPT-4
Prompt
recipes
instructions
inference knowledge
one-shot demo
Update State
Trajectory
Update Action
Memory
Memory History
Environment State History
Agent State History
Actions History
Feedback History
Action
Controller
Human Actions
Multi - Agents
NPC Collaborators
Human Player
Action Validation
Domain Specific Language
Action
List of action types
Pattern for the actions
Extraction full action names from the returned tuples

MindAgent



Planning Skills & Tool use
CuisineWorld
Gaming Environment
Game Info.
State
Agent location
Tool state
Agent holdings
Pending Dishes
Timer
Text Info.
LLM
Dispatcher
Current State
agent state
environment state
environment feedback
GPT-4
Prompt
recipes
instructions
inference knowledge
one-shot demo
Update State
Trajectory
Update Action
Memory
Memory History
Environment State History
Agent State History
Actions History
Feedback History
AI Agent Interaction
Action
Controller
Human Actions
Multi - Agents
NPC Collaborators
Human Player
Action Validation
Domain Specific Language
Action
List of action types
Pattern for the actions
Extraction full action names from the returned tuples

MindAgent



Planning Skills & Tool use
CuisineWorld
Gaming Environment
Game Info.
State
Agent location
Tool state
Agent holdings
Pending Dishes
Timer
Text Info.
AI Agent Interaction
LLM
Dispatcher
Current State
agent state
environment state
environment feedback
GPT-4
Prompt
recipes
instructions
inference knowledge
one-shot demo
Update State
Trajectory
Update Action
Memory
Memory History
Environment State History
Agent State History
Actions History
Feedback History
Action
Controller
Human Actions
Multi - Agents
NPC Collaborators
Human Player
Action Validation
Domain Specific Language
Action
List of action types
Pattern for the actions
Extraction full action names from the returned tuples

MindAgent Mechanism

Multi-Agent Task Management

- The system handles **N agents**, each responsible for completing **P tasks**.
- Each task is further divided into **Mp sub-tasks**.
- The number and type of tasks are **unknown at the start**, and tasks arrive dynamically over time.

Task Scheduling and Expiry

- Tasks must be completed within a **time limit**.
- If a task **exceeds the time limit**, it is marked as **failed**.
- The system aims to **maximize completed tasks** while minimizing failures.

MindAgent Mechanism – Mathematical Model

We aim to find valid and optimal task planning, scheduling, and allocations. We define q_{pim} and c_{pim} as quality and cost, respectively, for allocating agent i to work on the sub-task m for the p th task in the episode. Then the combined utility for the sub-task is:

$$u_{pim} = \begin{cases} q_{pim} - c_{pim}, & \text{if agent } i \text{ can execute sub-task m for the } p \text{ th task in the episode} \\ -\infty. & \text{otherwise} \end{cases}$$

We define the assignment of sub-task m to agent i as

$$v_{pim} = \begin{cases} 1, & \text{agent } i \text{ is assigned to sub-task m for the } p \text{ th task in the episode} \\ 0. & \text{otherwise} \end{cases}$$

MindAgent Mechanism – Mathematical Model

$$\arg\max_{v} \sum_{p=1}^{P} \sum_{i=1}^{N} \sum_{m=1}^{M_p} u_{pim} v_{pim} \tag{2}$$

Subject to:

$$\begin{array}{rll} \sum_p \sum_i \sum_m \tau_{pim} v_{pim} & \leq T_{max} & \\ \sum_i v_{pim} & \leq 1 & \forall m \in M, \forall p \in P \\ v_{pim} & \in \{0,1\} & \forall i \in N, \forall m \in M, \forall p \in P \end{array}$$

- Constraints:
  - The total execution time does not exceed the maximum limit.
  - Each sub-task is assigned to at most one agent.
  - The assignment can only be 1 or 0.
- NP-hard

- The LLM receives **natural language feedback from the environment**, such as,
  - “Collect finish” → when an agent successfully picks up an item.
  - “Agent IDs cannot be the same” → when the same agent is assigned multiple tasks incorrectly.

Results – Collaboration Efficiency of LLMs in Multi-Agent Planning

Metric Used: Collaboration Score (CoS)

$$\text{CoS} = \frac{\#\text{Completed Tasks}}{\#\text{Completed Tasks} + \#\text{Failed Tasks}}$$

- the system performance is generally better when there are more agents -> LLM dispatcher can coordinate more agents to execute tasks more efficiently
- the system performance degrades with more agents in less demanding conditions -> LLM dispatcher struggles when there are fewer tasks

2-agent	very simple			simple			intermediate			advanced			Avg.
	level 0	level 1	level 7	level 2	level 4	level 8	level 3	level 9	level 10	level 5	level 11	level 12	
GPT4 $\tau_{\text{int},(1)}$	18/54	18/56	12/31	14/34	12/30	3/30	10/26	7/20	7/23	6/23	6/21	10/36	0.318
GPT4 $\tau_{\text{int},(2)}$	18/31	17/34	10/23	13/26	12/22	9/22	10/17	8/11	6/12	5/13	4/14	8/21	0.486
GPT4 $\tau_{\text{int},(3)}$	18/25	19/25	10/17	16/18	11/18	6/16	11/13	6/8	7/10	8/10	9/9	8/17	0.709
GPT4 $\tau_{\text{int},(4)}$	18/18	18/19	12/12	11/14	11/12	7/11	12/12	8/8	9/9	6/7	8/9	11/12	**0.912**
GPT4 $\tau_{\text{int},(5)}$	18/18	17/17	12/12	11/13	11/13	9/9	11/11	4/5	7/7	8/8	8/8	9/12	**0.937**
CoS	0.727	0.706	0.682	**0.687**	**0.664**	0.504	0.764	0.725	0.701	0.661	0.692	0.559	0.673

Table 3: 2 agents performance on different tasks

3-agent	very simple			simple			intermediate			advanced			Average
	level 0	level 1	level 7	level 2	level 4	level 8	level 3	level 9	level 10	level 5	level 11	level 12	
GPT4 $\tau_{\text{int},(1)}$	21/55	24/55	16/33	17/33	9/28	6/32	12/25	5/20	8/21	7/22	7/22	9/26	**0.368**
GPT4 $\tau_{\text{int},(2)}$	20/31	25/33	11/22	4/24	13/24	7/21	14/20	9/12	9/13	7/14	8/14	10/23	0.549
GPT4 $\tau_{\text{int},(3)}$	22/25	21/26	17/17	11/20	9/17	4/15	13/14	8/8	12/12	7/7	9/10	10/16	**0.791**
GPT4 $\tau_{\text{int},(4)}$	22/22	20/21	14/14	9/13	7/10	6/10	10/10	6/7	10/10	5/8	7/8	11/13	0.846
GPT4 $\tau_{\text{int},(5)}$	20/20	15/16	11/12	10/14	10/11	8/9	12/12	6/6	8/8	5/5	8/8	6/10	0.914
CoS	**0.781**	**0.778**	**0.780**	0.528	0.600	0.455	0.822	**0.771**	**0.815**	0.689	**0.733**	**0.570**	**0.694**

Table 4: 3 agents performance on different tasks

4-agent	very simple			simple			intermediate			advanced			Average
	level 0	level 1	level 7	level 2	level 4	level 8	level 3	level 9	level 10	level 5	level 11	level 12	
GPT4 $\tau_{\text{int},(1)}$	22/54	18/55	17/34	13/34	8/28	9/33	16/27	5/20	8/23	5/22	8/22	8/35	0.349
GPT4 $\tau_{\text{int},(2)}$	24/32	21/33	14/24	14/25	12/24	11/22	16/19	7/12	9/15	7/14	6/12	12/23	**0.590**
GPT4 $\tau_{\text{int},(3)}$	23/25	23/26	13/18	11/19	10/17	11/17	15/17	8/9	11/11	7/8	10/11	9/17	0.785
GPT4 $\tau_{\text{int},(4)}$	22/22	21/22	14/14	7/15	10/13	10/12	12/13	9/9	10/10	6/7	8/8	9/13	0.875
GPT4 $\tau_{\text{int},(5)}$	14/18	20/20	14/14	7/13	9/11	7/8	12/12	5/5	7/7	6/6	3/5	7/10	0.859
CoS	0.771	0.761	0.761	0.505	0.592	**0.626**	**0.848**	0.744	0.790	**0.692**	0.675	0.534	0.692

Table 5: 4 agents performance on different tasks

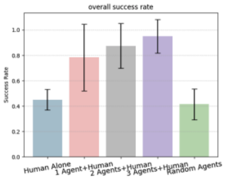
Results – Human-Agent Collaboration Study

Hypotheses Tested:

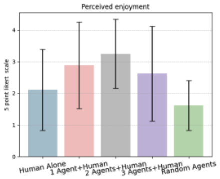
- H1: Humans perform better when collaborating with AI agents.
- H2: More AI agents improve human task productivity.
- H3: Players find the game more enjoyable when collaborating with AI agents.

Key Results:

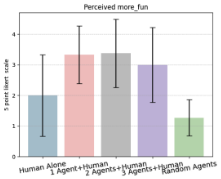
- Humans felt more productive with AI assistance (p = 0.0104).
- Collaboration led to higher engagement and enjoyment (p = 0.0126).
- The success rate of human-AI teams was significantly higher than humans alone.



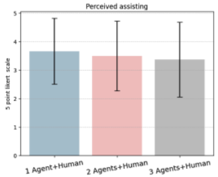
(a) **Collaboration score** We can tell that the collaboration score is higher if more agents are collaborating with human players, even though the difference is not significant.



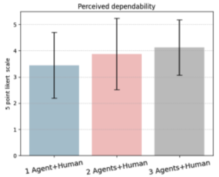
(b) **Perceived Enjoyment** Humans enjoy the game more if they collaborate with the right number of agents



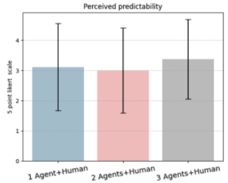
(c) **Perceived more fun** due to collaboration. Players enjoy the game more because of collaborating with competent agents.



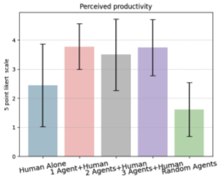
(d) **Perceived Assisting**. There is no significant difference in terms of human perceptions of helpfulness when collaborating with more agents, even though the task success rate is higher.



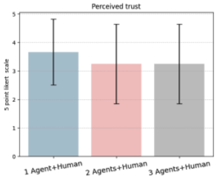
(e) **Perceived dependability**. When collaborating with more agents, players depend on the agents more.



(f) **Perceived Predictability**. There is no difference in terms of the predictability of agents' behaviors when collaborating with more agents.



(g) **Perceived productivity**. Players think collaborating with AI agents will improve productivity.



(h) **Perceived Trust**. There is no difference in terms of trust when collaborating with more agents.

Figure 5: Human Evaluations

Results – Emerging capabilities

Capability	What It Means	Example Behavior
Zero-shot multi-agent planning	Can handle unseen tasks with minimal examples	GPT-4 completes new game levels without explicit training
Scaling across different agent numbers	Learns from small settings and applies to larger ones	Trained with 2 agents, generalizes to 3 or 4 agents
Dynamic task adaptation	Adjusts plans in real time as new tasks arrive	Prioritizes urgent tasks, delays others
Self-correction	Learns from mistakes and avoids repeated failures	Stops assigning multiple agents to the same task

level_3	4agent using 4agent module	4agent using 2agent module	3agent using 3agent module	3agent using 2agent module
GPT4 $\tau_{\text{int},(1)}$	16/27	14/27	12/25	11/25
GPT4 $\tau_{\text{int},(2)}$	16/19	16/20	14/20	11/19
GPT4 $\tau_{\text{int},(3)}$	15/17	15/16	13/14	12/14
GPT4 $\tau_{\text{int},(4)}$	12/13	13/13	10/10	12/12
GPT4 $\tau_{\text{int},(5)}$	12/12	12/12	12/12	11/11
CoS	0.848	0.851	0.822	0.775

Table 8: Using different numbers of agent demos

Do you agree with these statements?

- "LLMs can perform zero-shot multi-agent planning, scheduling multiple agents into completing tasks without explicit training."
  → Do you believe the LLM really understands coordination, or is it just pattern-matching from prompts?
- "Without environmental feedback, LLMs make repeated errors, indicating that structured prompts are crucial for efficient planning."
  → If the LLM relies heavily on feedback, does it mean it lacks true adaptability?
- "LLMs exhibit emergent reasoning capabilities, successfully dispatching more agents than than the number seen in its prompt examples."
  → Do you believe this prove generalization ability can be extended to larger scale such as 10 or 100?