

Efficient Human-AI Coordination via Preparatory Language-Based Convention

Under review (ICLR 2024)

Yudi Zhang

- [1] Anonymous. Agent Instructs Large Language Models to be General Zero-Shot Reasoners. Submitted to ICLR 2023. <https://openreview.net/forum?id=zIJFG7wW2d>.
- [2] Code: https://anonymous.4open.science/r/AgentInstruct_ICLR2024

Motivation

Existing methods for human-AI coordination typically train an agent to coordinate with

- a diverse set of policies —— AI systems with constrained capacity
- with human models fitted from real human data. —— High-quality data may be unavailable

Observation:

Prior to coordination, humans engage in **communication** to establish **conventions** that specify individual roles and actions, making their coordination proceed in an orderly manner.

Outline

Employing the LLM to develop an action plan (or equivalently, a convention) that effectively guides both human and AI.

1. A naive solution : Using LLM to generate convention:

- Input: task requirements, human preferences, the number of agents, and other pertinent information
- Output: a comprehensive convention that facilitates a clear understanding of tasks and responsibilities for all parties involved.

2. Decomposing the convention formulation problem into **sub-problems with multiple new sessions** being sequentially employed and **human feedback**, will yield a more efficient coordination convention

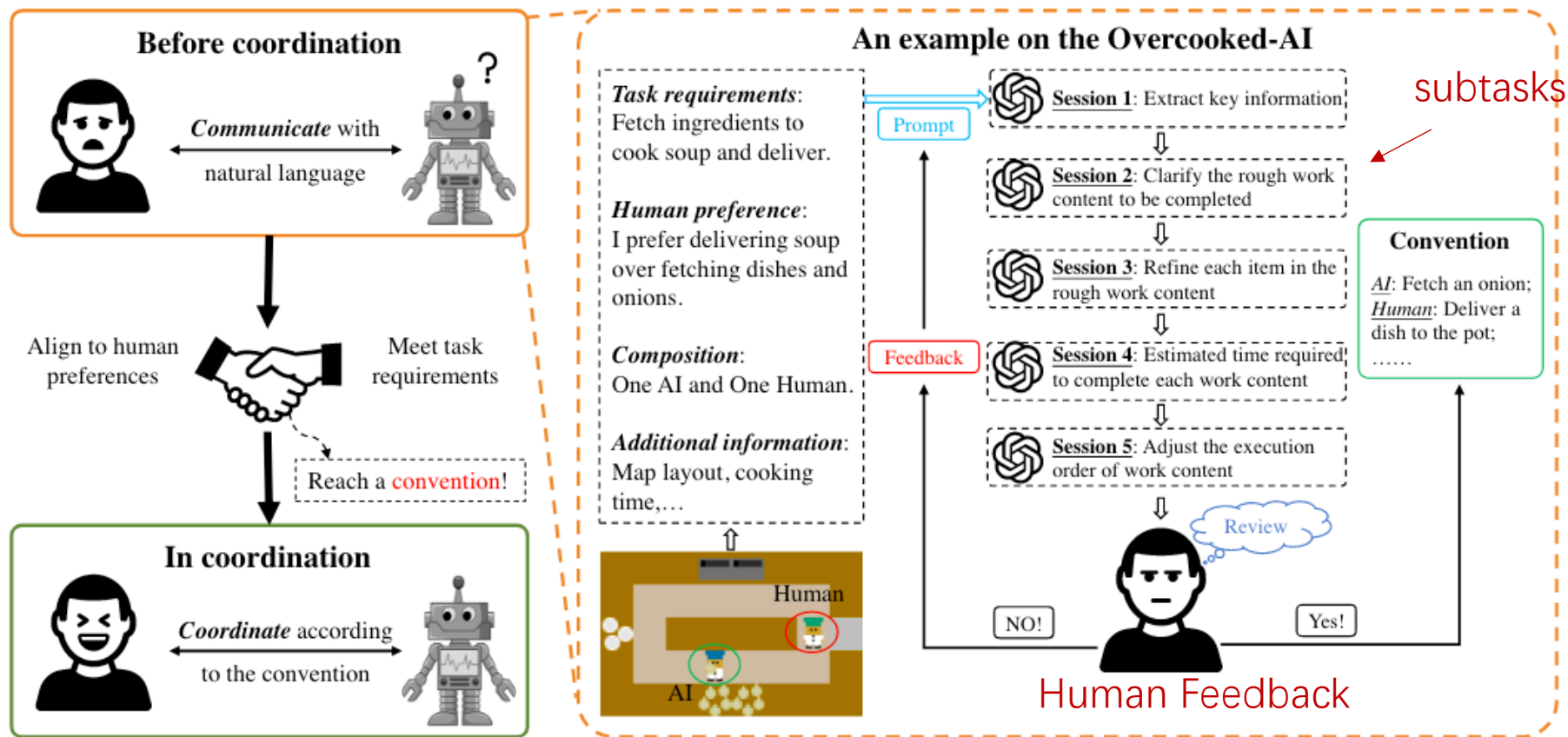


Figure 1: Overview of our proposed HAPLAN on the Overcooked-AI.

Task Planning with Multiple Sessions

- decompose a complex problem into multiple sub-problems and assign them sequentially to a new session.
- The solution provided by one session serves as part of the prompt for the subsequent session.

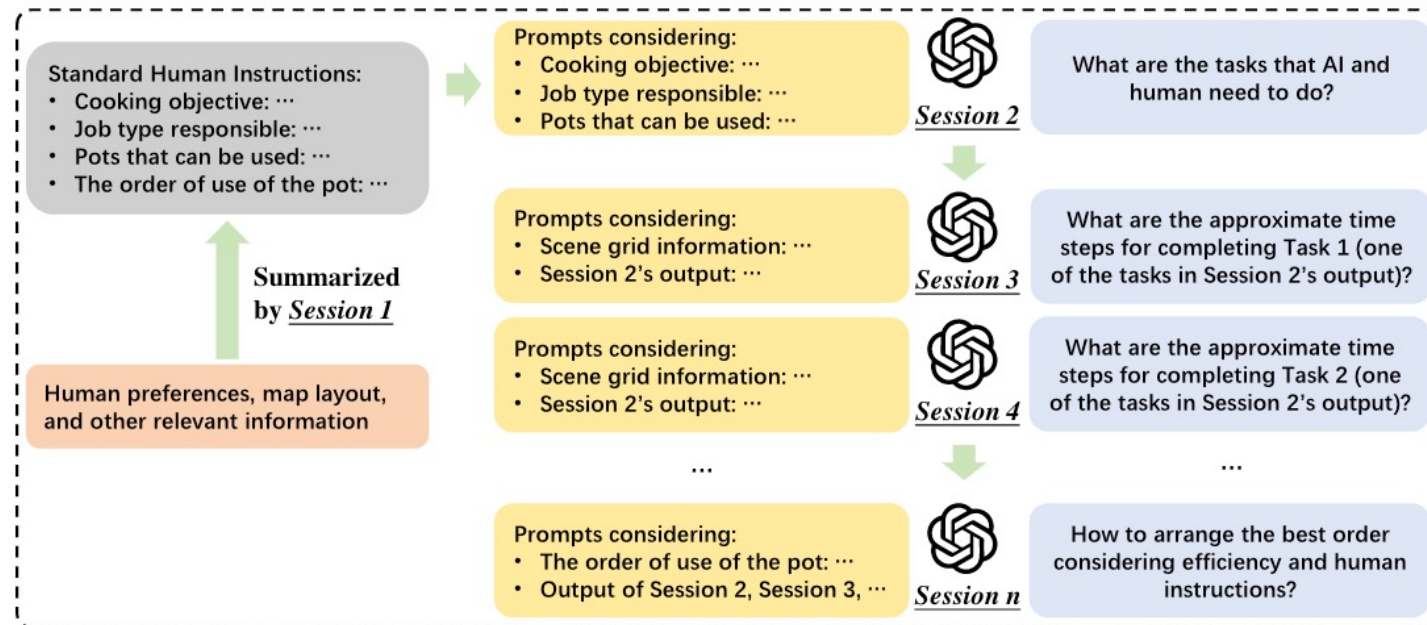
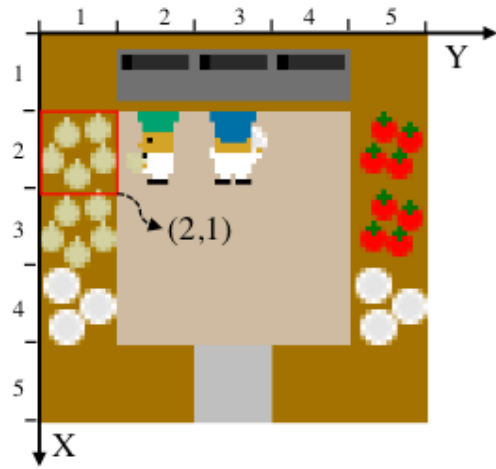


Figure 2: An example of task planning via multiple sessions.

A convention by the multiple sessions



A Convention Developed by Our Method on the *Many Orders* Layout of Overcooked-AI



 <ol style="list-style-type: none">1. <u>Fetch</u> an onion at (2,1);2. <u>Deliver</u> the onion to (1,2);3. <u>Fetch</u> a dish at (4,1);4. <u>Deliver</u> the dish to (1,2);5. <u>Fetch</u> the onion soup at (1,2);6. <u>Deliver</u> the soup to (5,3);.....	 <ol style="list-style-type: none">1. <u>Fetch</u> a tomato at (2,5);2. <u>Deliver</u> the tomato to (1,3);3. <u>Fetch</u> a dish at (4,5);4. <u>Deliver</u> the dish to (1,3);5. <u>Fetch</u> the tomato soup at (1,3);6. <u>Deliver</u> the soup to (5,3);.....
--	--

Figure 3: An example of conventions on Overcooked-AI. **Left:** Layout of the *Many Orders* map; **Right:** A convention for human and AI, where the left part is action plans for human and the right part is action plans for AI. (x, y) in the plans denotes the region in the layout whose coordinate on the X-axis is x and coordinate on the Y-axis is y .

Human Validation to Re-plan: Review generated content and provide suggestions as part of the prompts for the first session to re-plan the convention.

Execution with Pre-trained Skills

- Two skills
 - Fetch A at B
 - Deliver A to B
- Behavior Cloning

Experiments

Table 1: Experimental results on Overcooked-AI environment of HAPLAN and baselines when coordinating with human proxy policies. The best values have been **bolded**.

Layout	Partner	FCP	MEP	HSP	HAPLAN
Counter Circle	Onion Placement Delivery	104.38±9.66	133.75±20.27	135.38±15.19	140.00±26.92
		86.88±9.49	83.12±7.26	96.25±7.81	103.75±10.53
Asymmetric Advantages	Onion Placement & Delivery (Pot1) Delivery (Pot2)	233.13±17.75	256.25±18.66	282.88±17.03	260.63±18.36
		215.00±16.58	250.00±19.36	258.13±21.71	268.00±9.79
Soup Coordination	Onion Placement & Delivery Tomato Place & Delivery	199.38±6.09	105.00 ± 32.78	198.75±4.84	219.38±3.47
		44.38±29.04	192.50±9.68	128.12±30.76	220.63±3.47
Distant Tomato	Tomato Placement Tomato Place & Delivery	38.75±30.79	27.50±27.27	148.75±68.36	210.00±15.00
		175.62±24.35	180.00±22.36	198.12±37.20	251.25±23.41
Many Orders	Tomato Placement Delivery	140.62±32.59	170.00±33.91	248.75±29.55	256.36±35.99
		194.38±12.48	175.63±35.61	208.13±25.42	241.21±12.97

Baselines: Fictitious Co-Play (FCP) (Heinrich et al., 2015), Maximum Entropy Population-based training (MEP) (Zhao et al., 2023a) and Hidden-utility Self-Play (HSP) (Yu et al., 2023)

Partner: scripted policies in HSP

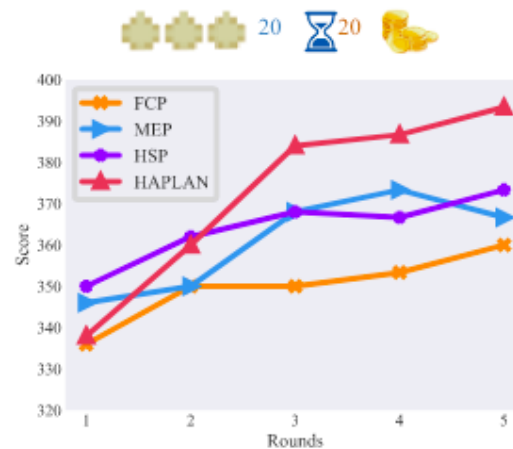
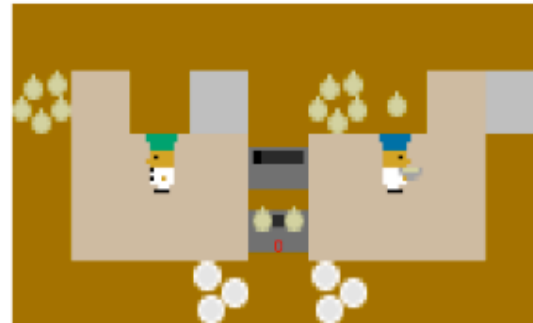
Experiments

Table 2: Experimental results on Overcooked-AI environment of HAPLAN and baselines when coordinating with real humans. The best values in each round of coordination have been **bolded**.

		Counter Circle	Asymmetric Advantages	Soup Coordination	Distant Tomato	Many Orders
First Round	FCP	120.00±12.64	336.00±24.97	192.00±20.39	314.00±25.37	329.00±32.38
	MEP	140.00±21.91	346.00±25.37	184.00±14.96	310.00±22.36	318.00±31.55
	HSP	140.00±15.49	350.00±34.92	184.00±8.01	330.00±24.08	340.00±43.81
	HAPLAN	138.00±20.88	338.00±27.49	192.00±18.33	324.00±29.39	349.00±63.01
Second Round	FCP	138.00±10.77	350.00±18.43	194.00±18.01	338.00±18.86	340.00±29.66
	MEP	154.00±12.81	350.00±27.21	186.00±12.81	332.00±20.39	342.00±36.27
	HSP	154.00±15.62	362.00±18.86	196.00±14.96	348.00±18.33	372.00±37.09
	HAPLAN	160.00±15.49	360.00±25.29	204.00±21.54	356.00±17.43	382.00±58.95
Third Round	FCP	136.00±17.43	350.00±25.69	198.00±28.91	336.00±34.41	349.00±23.01
	MEP	158.00±16.61	368.00±20.39	196.00±12.00	340.00±21.91	350.00±36.05
	HSP	160.00±12.64	368.00±27.12	198.00±10.77	352.00±25.61	376.00±33.22
	HAPLAN	168.00±13.26	384.00±21.54	214.00±15.62	370.00±22.36	414.00±56.61

Analysis of LLMs in Human-AI Coordination

- Explainable AI



Human Reflection

Human says: Join me in making onion soup. You use the pot at bottom, while I use the pot on top.
Human does: Fetch onion at pot 1, and deliver the cooked soup.
Human finds: *Delivery costs less time than placement for him/her, while it is the opposite for AI.*

Human says: Join me in making onion soup. You fetch onion and deliver it to the pot, while I deliver the soup.
Human does: Fetch the dish and deliver the soup when it is ready.
Human finds: *Time spent waiting for the onions to cook with dish seems to be wasted.*

Human says: Join me in making onion soup. You fetch onion and deliver it to the pot, while I deliver the soup.
Human does: Before fetching dish and delivering the soup, fetch one onion and deliver it to the pot.
Human finds: *It works well. I follow this practice of placing a few onions before going to deliver the soup.*

.....

Figure 4: Details of results on the *Asymmetric Advantages* layout.

Analysis of LLMs in Human-AI Coordination

- Incorporating human domain knowledge
- Taking Many Orders layout as an example, humans intuitively tend to believe that actively utilizing all three pots is essential for completing the task efficiently.

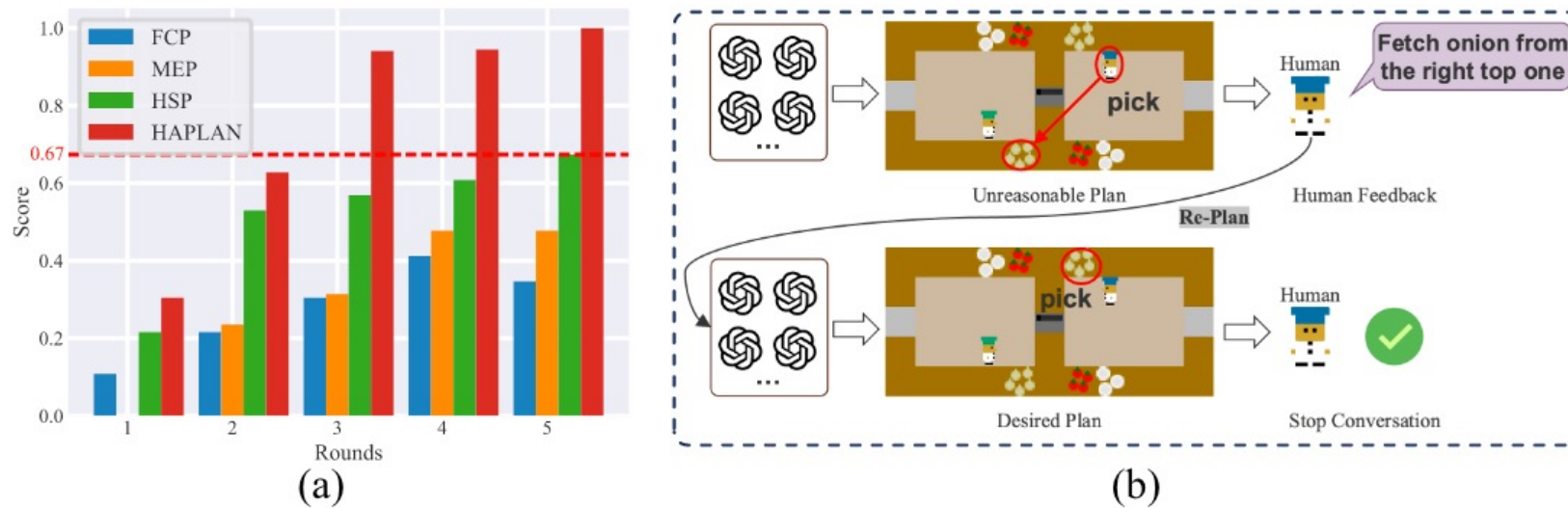


Figure 5: (a) Normalized scores on *Many Orders*. (b) An example of human-AI conversation.

Analysis of LLMs in Human-AI Coordination

- Human-AI value alignment

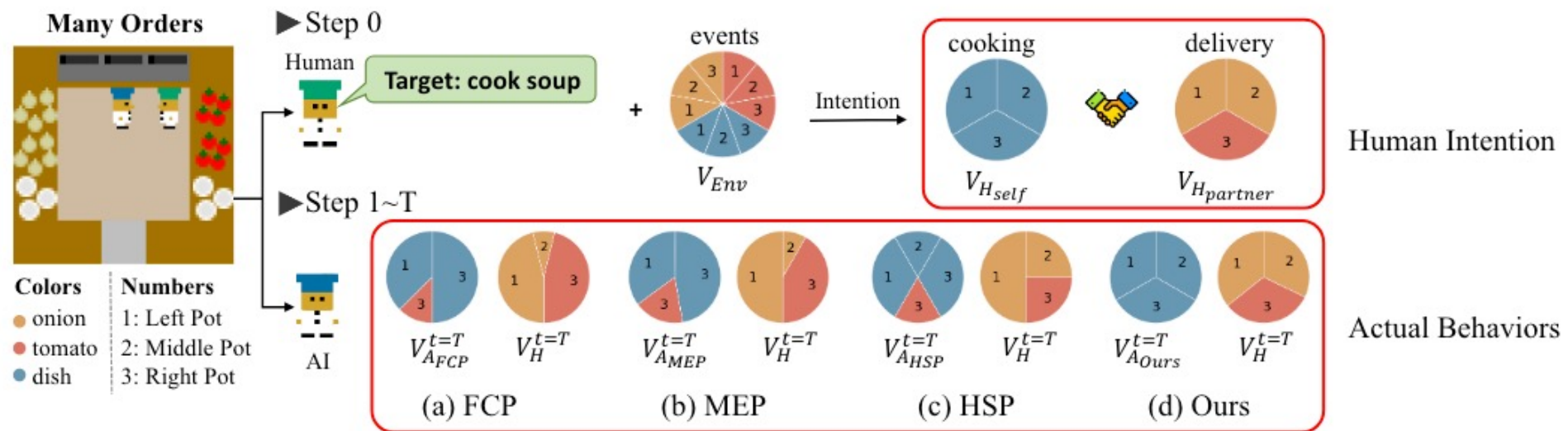


Figure 6: Overview of the human-AI value alignment. Colors denote task types and numbers indicate pot usage, e.g., the red sector of label 1 means placing onions to pot 1, the blue sector of label 2 means delivering the soup in pot 2. $V_{H_{self}}$ and $V_{H_{partner}}$ denote the human's initial intention regarding what they do respectively. Subsequent pie charts show actual event proportions post-trajectory.

Evaluating Multi-Agent Coordination Abilities in Large Language Models

Saaket Agashe, Yue Fan, Xin Eric Wang

Under review (ICLR 2024)

Yudi Zhang

[1] Agashe, S., Fan, Y., & Wang, X. E. (2023). Evaluating Multi-Agent Coordination Abilities in Large Language Models. *arXiv preprint arXiv:2310.03903*.

LLM-Coordination (LLM-Co) Framework

- evaluation with three game environments and organize the evaluation into five aspects:
 - Theory of Mind, Situated Reasoning
 - Ability to infer the partner's intention and reason actions accordingly
 - Sustained Coordination, Robustness to Partners
 - the ability of LLMs to coordinate with an unknown partner in complex long-horizon tasks, outperforming Reinforcement Learning baselines.
 - Explicit Assistance:
 - the ability of an agent to offer help proactively: prioritize helping their partners, sacrificing time that could have been spent on their tasks.
 - two novel layouts into the Overcooked-AI benchmark

LLM-Co Coordination Games

- Collab Capture: two agents chase an adversary through a maze of rooms
- Collab Escape: two agents need to coordinate to escape from an adversary
- Overcooked: two players cook and deliver onion soup
- LLM-Co Framework: provides agents with contextual state information and feasible actions & interprets agents' output for execution in real-time.

Coordination Games – Collab Capture

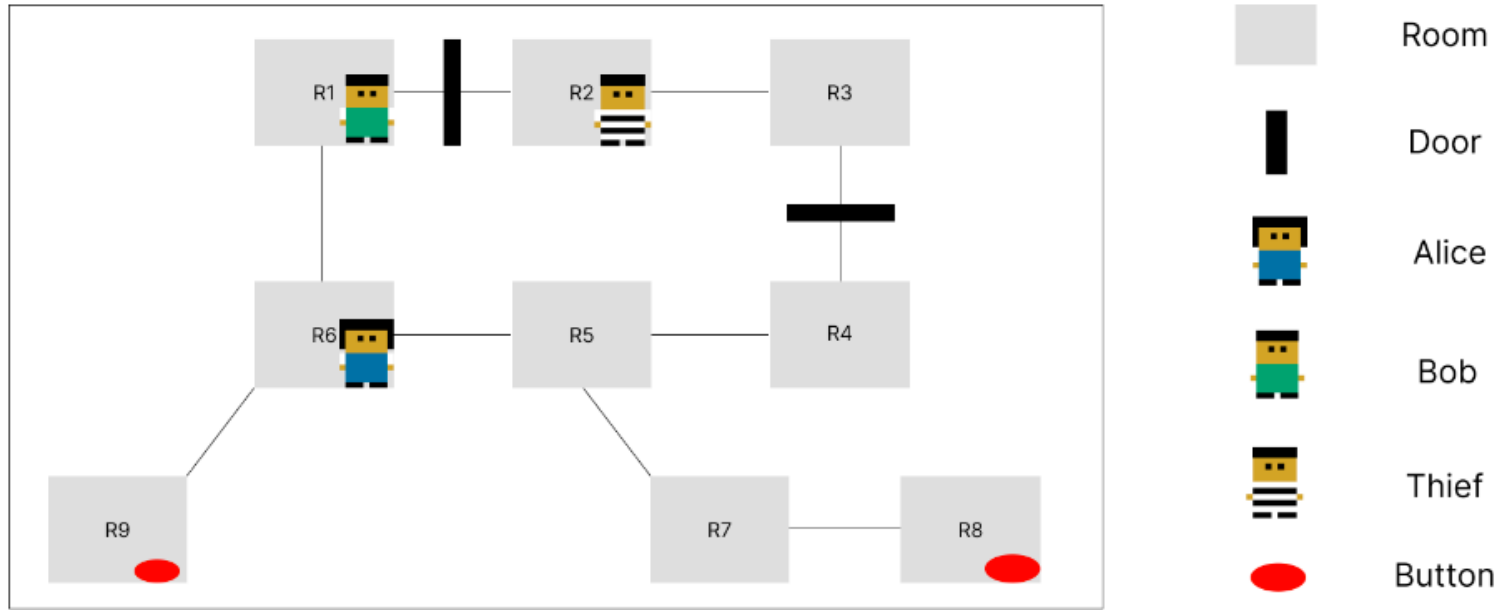


Figure 1: The CollabCapture game involves two agents, Alice (Blue) and Bob (Green), chasing a thief across multiple rooms. Some rooms are connected by doors, which can be controlled by buttons in different rooms.

The agent's task is to capture the adversary in the least amount of time using effective strategies including cornering the adversary, disabling the adversary, or enabling their partners.

Coordination Games – Collab Escape

- Based on the popular Video Game "Dead-by-Daylight", Collaborative Escape involves two agents trying to escape an adversary in a maze of interconnected rooms. They need to **fix two generators located in randomly selected rooms** to open an exit portal. The adversary tries to catch the agents, and the win condition is **any one agent escaping**.
- This game requires strategies like **luring the adversary away from the partner, sacrificing for the partner's safety, and manipulating the movement of the adversary**.

Coordination Games – Overcooked

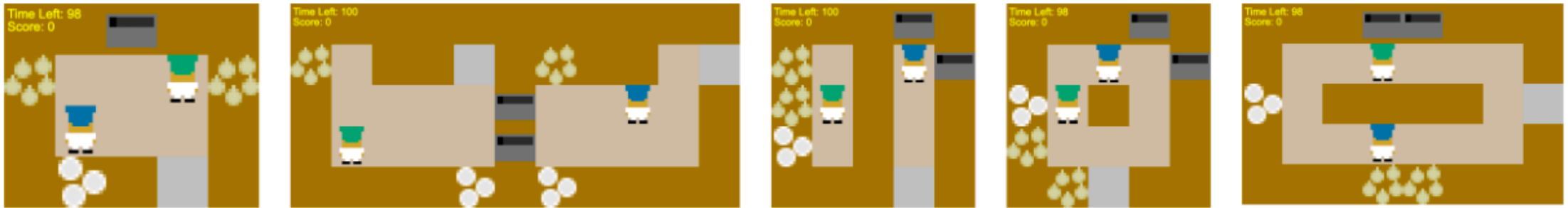


Figure 2: All layouts from the overcooked environment we use for our tests. The two agents Alice (Blue) and Bob (Green) need to collaborate to cook, plate, and deliver onion soups. From Left to Right: Cramped Room, Asymmetric Advantages, Forced Coordination, Coordination Ring, and Counter Circuit.

Coordination Games – Overcooked-Assist



Figure 3: Additional Layouts that require agents to explicitly help their partner complete a delivery. These new layouts utilize **walls** and **gates** to create situations requiring explicit assistance.

Gates: can be opened by an agent provided they are not holding anything in their hand, and only can remain open for a short time.

Wall: prevent the agent from placing their soups temporarily on counters to open the gates.

LLM-Co Framework

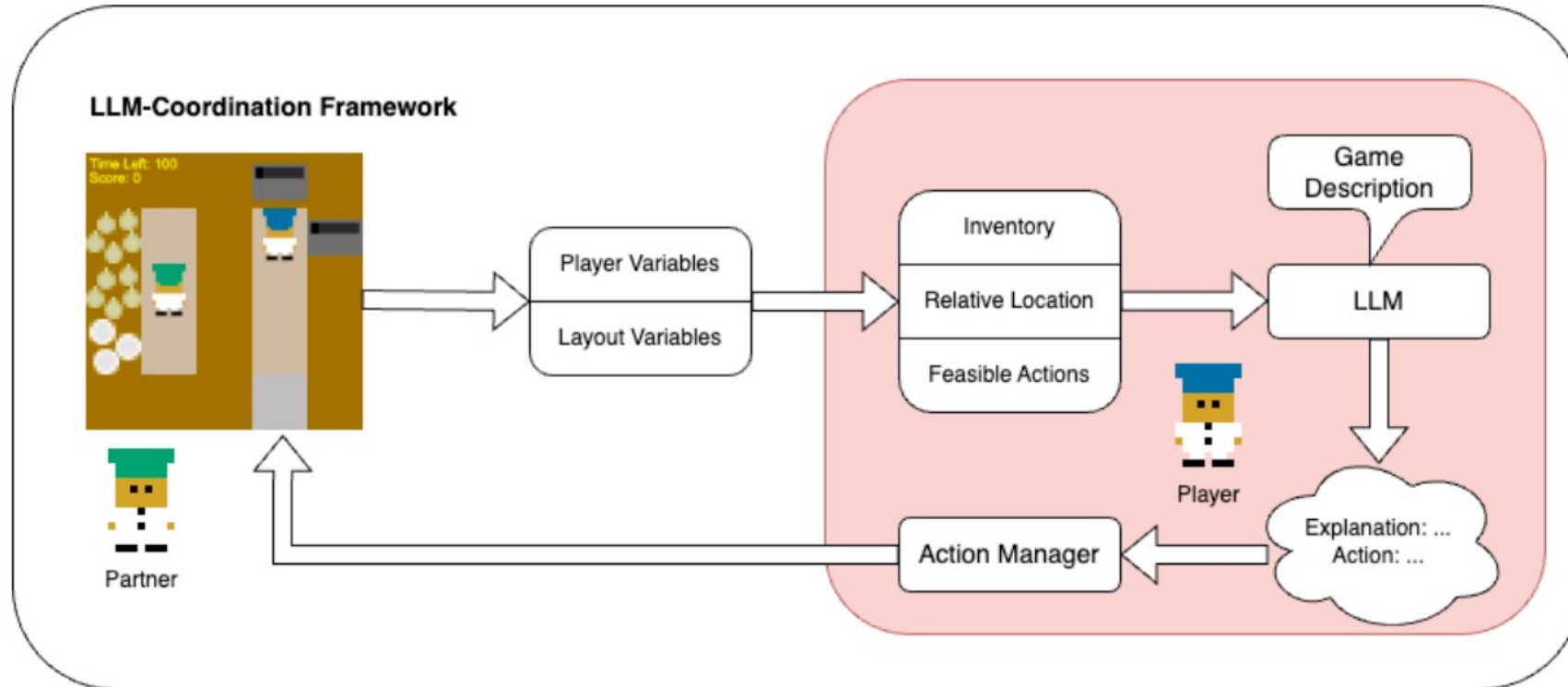


Figure 4: Visual summary of the LLM-Co framework. Our framework serves as the backbone for an individual agent, focusing on bringing out its coordination ability. The framework translates abstract game details into an LLM-compatible format and then utilizes the generated LLM output to take actions in the game world.

Prompt Design

- Game Description (G): details of the game along with the rules and the layout of the map
- Directives (D_i)
- At each turn,
 - State description ($D(S)$): programmatically obtained from the environment and the player state S .
 - Relative distances from the agent to each location of interest in $D(S)$
 - Partner's inventory and relative position
- medium-level action space: verb-based actions
 - Pick, place, move
 - feasible actions M_f : according to inventory and accessibility of locations

LLM-ToM-Reasoning Test Set

- A **hand-picked** suite of 18 scenarios posed with questions among all three games
- the agent under-tests to first take their partner's possible next actions into active consideration, reason about the current state, and adjust their actions that “indirectly” lead to the best possible outcome.
- Formed by State Description and Feasible Action Generator
- Labeled by partner's potential next action (ToM) & the optimal next action from the perspective of a player (Situating Reasoning)

ToM and Situated Reasoning

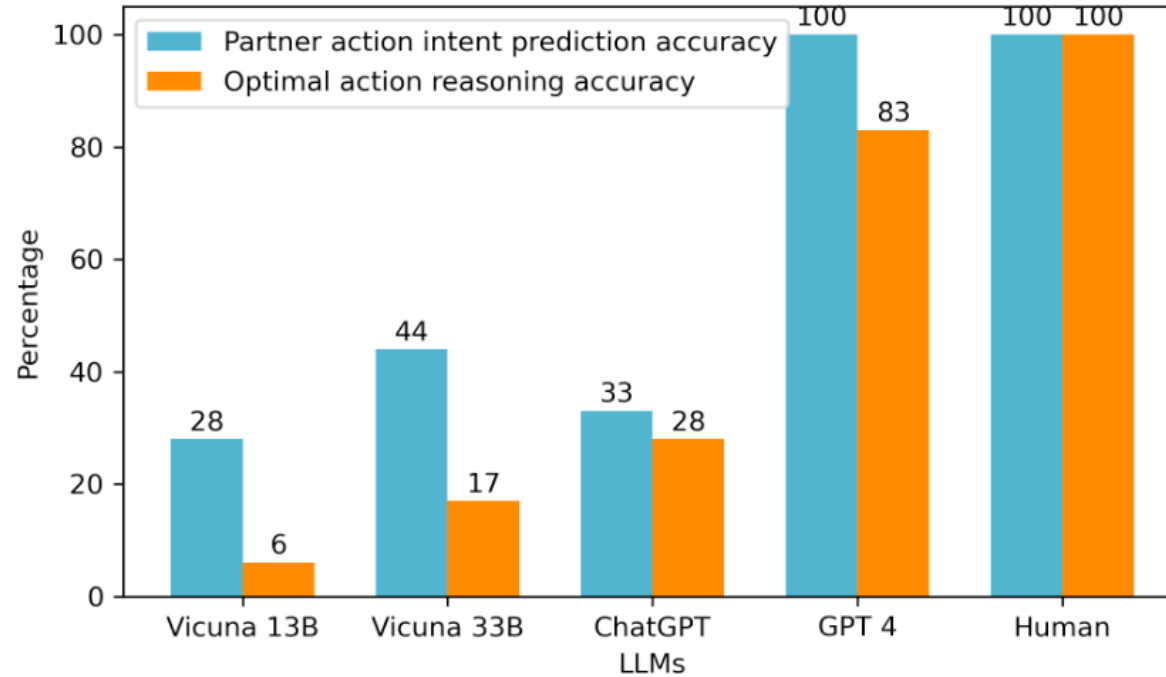


Figure 5: LLMs performance on the LLM-ToM-Reasoning test set. Partner action intent prediction accuracy shows the Theory Of Mind ability of LLMs under test and the optimal action reasoning accuracy infers the Situated Reasoning effect of LLMs under test. GPT-4 achieves the best performance among tested LLMs.

Sustained Coordination and Robustness to Partners

- Sustained coordination: the ability of agents to continuously collaborate and adapt their actions over extended periods.
- 400 timestep, each delivery wins the agents 20 points

Agent Type	Layouts				
	Cramped Rm.	Asymm. Adv.	Coord. Ring	Forced Coord.	Counter Circ.
PPO _{SP}	198.8 ± 4.06	167.2 ± 3.63	190.8 ± 4.25	151.9 ± 3.28	122.3 ± 3.80
PBT	216.9 ± 1.31	190.1 ± 8.64	173.8 ± 18.27	169.5 ± 10.09	140.1 ± 13.86
LLM-Co	220 ± 0	280 ± 0	180 ± 0	200 ± 0	160 ± 0

Table 1: Comparison of game play between self-play baselines (PPO, and PBT) and LLM-Co Agents. LLM-Co agents outperform RL methods on 4 out of 5 layouts, demonstrating highly effective reasoning under sustained coordination.

LLM Agents are capable of achieving sustained coordination, adjusting to their partners, and correcting their own actions consistently.

Sustained Coordination and Robustness to Partners

- self-play agents, when paired with humans, tend to struggle because their behavior diverges from what they consider to be the optimal strategy, while LLM-Co agent not

Agents	Layouts				
	Cramped Rm.	Asymm. Adv.	Coord. Ring	Forced Coord.	Counter Circ.
BC	103.5 \pm 3.38	136.5 \pm 7.00	59.0 \pm 5.38	20.5 \pm 4.33	38.0 \pm 3.99
PPO _{BC}	156.4 \pm 1.48	72.6 \pm 19.44	126.4 \pm 3.24	58.9 \pm 2.98	69.5 \pm 2.18
LLM-Co	160 \pm 0	180 \pm 0	160 \pm 0	120 \pm 0	140 \pm 0
Playing from swapped positions:					
BC	110.0 \pm 3.39	137.5 \pm 8.40	70.0 \pm 4.00	31.0 \pm 5.00	44.0 \pm 3.02
PPO _{BC}	163.9 \pm 1.61	178.8 \pm 2.65	129.8 \pm 3.59	76.9 \pm 2.29	57.6 \pm 2.50
LLM-Co	180 \pm 0	140 \pm 0	160 \pm 0	80 \pm 0	120 \pm 0

Table 2: Comparison of AI-Human Proxy Game play. We compare Behavior Cloning Agents, PPO_{BC} Agents with LLM-Co agents utilizing the GPT-4 LLM. The LLM-Co agents are able to outperform or match the performance of Reinforcement Learning models, indicating that LLM agents are robust to the choice of partner agents.

Explicit Assistance

Conditions	Layouts	
	Locked	Gated Delivery
Without Helper Directive	160	0
With Helper Directive	240	180

Table 3: Comparison of Gameplay in the Overcooked-Assistance Layouts with and without Helper Directive. The results indicate that the Large Language Model needs to be prompted to be aware of situations where their partner might need assistance in order to be effective in the Overcooked-Assistance layouts.

- A simple directive: help their partners when the situation demands
 - The agent tends to help partner agents during the time they are waiting for their own soup to be cooked by choosing the open gates for the waiting agent.
 - Not most efficient strategy but always help others during coordination

Explicit Assistance

Agents	Layouts	
	Locked	Gated Delivery
PPO _{SP}	132.83 ± 7.31	134.88 ± 5.99
PBT	175.8 ± 1.69	178.6 ± 9.76
LLM-Co	220 ± 0	180 ± 0

Table 4: Comparison of Gameplay on Overcooked-Assistance Layouts between RL baselines and LLM Agents. The RL baselines being able to effectively solve the deliveries indicates that the environments are solvable through self-play training. The high scores achieved by LLM agents demonstrate that LLM agents are capable of reasoning for providing explicit assistance to their partners.

- The LLM-Co agent outperforms MARL methods at Overcooked-Co-op layouts

Conclusion

- Theory of Mind & Situated Reasoning abilities: only GPT4 can provide acceptable ToM and Situated Reasoning skills, via **LLM-ToM-Reasoning Test Set**.
- LLM-Co Agent (with GPT4) performs better than or equal to the RL baseline in both AI-AI and AI-human proxy gameplay without any fine-tuning, enjoying the interpretability.
- In the newly designed Overcooked env, LLM-Co agent can proactively help out their partners, requiring a 'helper directive'.