Closed-Loop Self-Organizing Agent System

Anonymous submission

Abstract

Building on the triumph of the planning-action-perception closed-loop in large language models (LLMs) that has propelled embodied agents to exhibit human-like behaviors, there is an escalating interest in multi-agent collaboration to further tackle large-scale tasks and navigate more intricate challenges. In this study, we unveil the concept *"Embodied Organization"*, a framework designed to self-organize a group of LLM-powered agents towards the successful completion of diverse tasks. However, a mere interlinking of agents disrupts the clear information loop within an individual's planning-action-perception cycle, leading to unpredictable behaviors when information traverses multiple agents. In our approach, we devise Closed-Loop Self-Organizing Agent System, which is characterized by 1) an acyclic structure for multi-agent interconnections, ensuring unidirectional intra-organizational information flow; 2) agents with general functionalities, facilitating independent intra-organizational and cross-environment interactions; 3) a feedback processor to assimilate both intra-organizational and cross-environment interactions, supporting an organizational closed-loop cycle. We've constructed an array of systematic collective tasks within the Minecraft environment - encompassing collective collection, and crafting, construction - to assess organizational behaviors. The exploration of embodied organizations pushes the frontiers of Embodied AI, nudging it toward a more human-like organizational structure.

Introduction

Drawing upon large language models (LLMs) (Brown et al. 2020), encapsulated with human knowledge, communicative competence, decision-making capabilities, and more, the embodied agents exhibit human-like intelligence in executing functions such as playing games (Park et al. 2023; Wang et al. 2023a; Shinn, Labash, and Gopinath 2023), coding (Qian et al. 2023; Hong et al. 2023), and completing robotic tasks (Mandi, Jain, and Song 2023; Zhang et al. 2023). At the heart of these multifaceted implementations lies the essence of individual intelligence. It's anchored in a planning-action-perception cycle, which underpins a seamless closed-loop control, characterized by the interplay of input and feedback information between agents and their environment, known as the ***cross-environment*** interaction. Currently, inspired by the remarkable strength of human collective unity over individual prowess, interests are raising

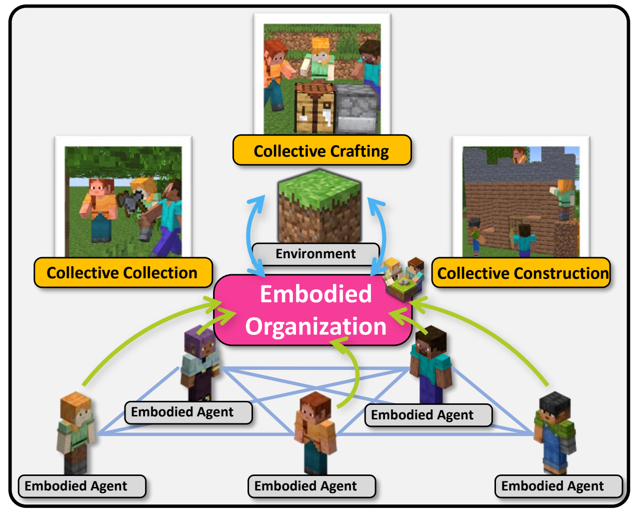

Figure 1: In this study, we extrapolate the concept of the Embodied Agent to an organizational dimension. This collective of agents autonomously self-organizes to undertake collaborative endeavors, ranging from the collective resources collection, collective tools crafting, and collective shelter construction.

to concentrate on the organic integration of these embodied agents. Empowered by the additional ***intra-organization*** interaction, the integration facilitates the execution of large-scale tasks and more complex, systematic tasks that necessitate parallel processing, labor division, and collaboration.

Therefore, we propose the concept ***Embodied Organization***. Drawing from the term "Embodied" in Embodied AI, which signifies the capacity to perceive, comprehend, and navigate an environment, the Embodied Organization similarly integrates multiple agents. This collective entity, as a result, is endowed with the ability for self-organization, enabling it to perceive, comprehend, and navigate diverse environments.

However, the naive approach that merely stitches together communication pathways among multiple agents governed by the planning-action-perception cycle culminates in chaos. This inadequacy arises from the convoluted intra-organization and cross-environment information path, which spiral endlessly within the intricate maze woven by the organization.

As a solution, a direct input and feedback pathway should be clear out, the intra-organization interaction and cross-environment interaction should be cooperated organically. Thus, we propose **C**losed-Loop **S**elf-Organizing **A**gent **S**ystem (CSAS). In this aproach, we devise 1) an acyclic structure inside the organization, ensuring unidirectional intra-organizational flow; 2) agents with general functionalities, facilitating independent intra-organizational and cross-environment interaction; 3) a feedback processor to assimilate both intra-organizational and cross-environment interactions, finally supporting an organizational closed-loop cycle.

To assess the performance of the Embodied Organizations, we devise a carefully crafted set of collective benchmarks within the Minecraft milieu. These encompass collective resource gathering, architectural house construction, and tool fabrication.

In summation, our work has led to several pivotal contributions: 1) The introduction of the ”Embodied Organization” paradigm, a leap from individual embodied intelligence to an organizational level; 2) The development of Closed-Loop Self-Organizing Agent System, crafted to actualize the organizational closed-loop planning-action-perception cycle; and 3) The formulation of a comprehensive suite of benchmarks within Minecraft to evaluate the prowess of Embodied Organizations.

Related work

Embodied agents in Minecraft Minecraft is an open-ended, three-dimensional world that is a free experimental environment for building numerous benchmarks and agent methods. ***1) Benchmarks***: MineDojo (Fan et al. 2022), Malmo (Johnson et al. 2016) and MineRL (Guss et al. 2019, 2021; Kanervisto et al. 2022) are established benchmarks for evaluating single-agent algorithms. While IGLU (Mohanty et al. 2022; Kiseleva et al. 2022a,b; Mohanty et al. 2023) and CraftAssist (Gray et al. 2019) focus on designing specific structures based on human instructions, their scope is limited to human-machine interaction and cannot perform more complex tasks. On the other hand, Malmo (Perez-Liebana et al. 2019) offers a artificially designed game environment for multi-agent cooperation but lacks the necessary openness and diversity in tasks and environments. ***2) Agent design***: Building upon these benchmarks, several advanced works explore different approaches to realizing embodied agent. Many prior works utilize reinforcement learning to learn human game behavior (Kanitscheider et al. 2021; Lin et al. 2021; Mao et al. 2022; Skrynnik et al. 2021; Hafner et al. 2023). MineDojo (Fan et al. 2022) and VPT (Baker et al. 2022) perform large-scale pre-training on game-playing videos. (Wang et al. 2023b) proposes a closed-loop feedback framework for single agent, allowing the agent to achieve its goals vis interact with environment. (Zhu et al. 2023) implements long-term planning by combining the knowledge base with LLMs. (Wang et al. 2023a) makes use of LLMs to automatically generate the next task based on the environment, continuously enriching the skill library and exploring the world.

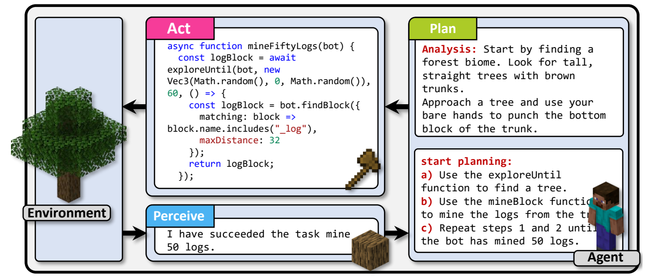

Figure 2: **An illustration of individual cross-environment interaction.** The agent begins by interpreting the main objective, converting it into a long-term plan. This is then broken down into detailed short-term tasks. The action module generates JavaScript code for environment interaction, while the perception module collects feedback, feeding it back to the planning module for adaptive action.

Previous research has mostly focused on the decision-making capabilities (Wang et al. 2023b; Zhu et al. 2023; Wang et al. 2023a) of single agents or on human-machine cooperation and interaction (Mohanty et al. 2023; Perez-Liebana et al. 2019). However, there has been scarce previous exploration into the possibility of constructing a structured, scalable autonomous embodied organization in an open world, which shows enormous potential value.

LLM powered multi-agent collaboration Large Language Models (LLMs) have demonstrated phenomenal capabilities in a wide range of domains. ***1) AI sociological simulation***: Other studies (Park et al. 2023; Akata et al. 2023; Xiang et al. 2023) drive multiple agents with LLM and simulate human-like conversations as well as some social behaviors; ***2) Multi-agent programming***: Recent attempts (Li et al. 2023; Dong et al. 2023; Qian et al. 2023) found that multi-agent embodied collaboration could develop software following a fixed process, but lacked the ability to coordinate autonomously; ***3) Embodied collaboration***: Multiple robotic arms (Mandi, Jain, and Song 2023), and multiple agents (Zhang et al. 2023) working in collaboration all bring higher efficiency, but cannot be scaled up; However, the autonomous collaborative behavior of multi-agent embodied organizations in the open world remains an unclear topic in the current research.

While previous methodologies have demonstrated leading-edge performance on certain tasks, their organizational structures and collaboration mechanisms are preordained, deviating from the true essence of an embodied organization. An embodied organization ought to autonomously structure its group, inherently determining its collaboration framework and being versatile enough to tackle a multitude of tasks.

Methodology

In this section, we present the Closed-Loop Self-Organizing Agent System—a novel architecture conceived to actualize the embodied organization. Initially, we differentiate between cross-environment and intra-organizational interactions. Subsequently, we delineate the concept of the embod-

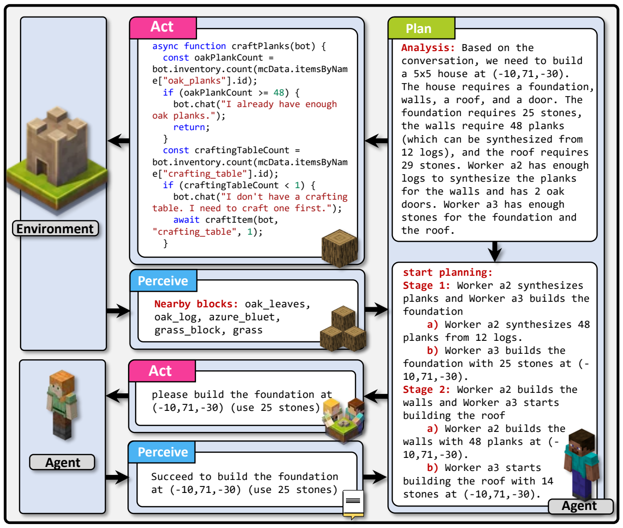

Figure 3: The diagram elucidates both the individual's cross-environmental and intra-organizational interactions. Analogous to the environmental counterpart, the intra-organizational communications strictly adhere to a planning-action-perception model. Within this framework, the depicted action signifies directives issued to another agent, assigning specific tasks. The subsequent perception phase involves monitoring the agent's feedback and ascertaining the successful execution of the prescribed order.

ied graph, a conceptual representation that encapsulates the organization's structure. We then elaborate on the agents' prowess to simultaneously navigate cross-environment interactions and engage in intra-organization interactions. Finally, agents are equipped with a feedback processor, enabling them to strategically re-plan based on inputs derived from both cross-environment and intra-organization exchanges.

Cross-environment and intra-organization and interaction As illustrated in Figure 2, each individual agent operates within a singular planning-action-perception cycle, engaging with its environment. The agent starts by planning its long-term plan and short-term plan, subsequently executing actions within the environment. Upon perceiving the resultant alterations in the environment, the agent recalibrates and iteratively planning the succeeding plan. This cycle epitomizes the closed-loop information exchange between the agents and their environment.

To foster intra-organizational interactions, we augment the foundational agents with communication functionalities. Parallel to the environmental interaction, intra-organization interaction (communications) similarly adheres to the planning-action-perception paradigm. The agent begins by crafting its strategy, conceptualized as an inquiry list in this context, and then communicates with fellow agents. Upon receiving feedback from its peers, the agent recalibrates, iteratively devising the subsequent inquiry list similarly.

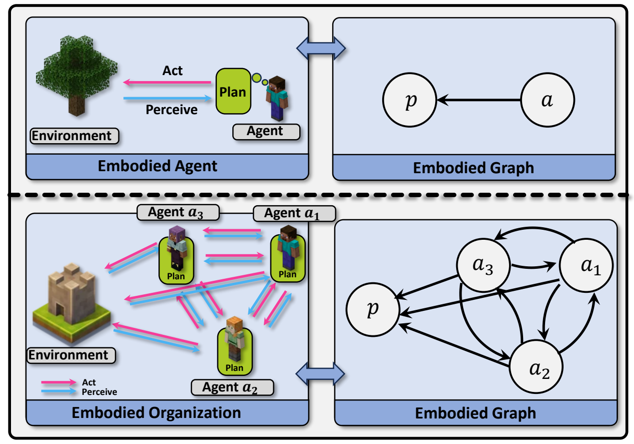

Figure 4: **An illustration of embodied graph construction.** The directed edge $e_{ij} = (v_i, v_j)$ for $v_i, v_j \in \mathcal{V}$ is defined if v_i actively acts upon and passively perceives from v_j. The magenta arrows delineate the direction of active actions, while the cyan arrows represent the pathway of passive perceptions. Within the embodied organization illustration, the trajectory of directed edges aligns with the direction indicated by the magenta arrows.

Organization Structure Grounded in the intra-organization and cross-environment interaction, we can construct a graph, called ***Embodied Graph*** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ based on the sets of agents, denoted as $\mathcal{A}$, and the environment, represented as p. The graph's vertex set is formulated as $\mathcal{V} = \mathcal{A} \cup \{p\}$. We subsequently define a directed edge $e_{ij} = (v_i, v_j)$ for $v_i, v_j \in \mathcal{V}$ if v_i ***actively acts upon*** and ***passively perceives*** from v_j. For instance, when considering a single agent a, the graph can be succinctly represented as $\mathcal{V} = \{a, p\}$, with the edges defined as $\mathcal{E} = \{(a, p)\}$.

From the described graph structure, several inherent properties can be discerned: **1)** The environment vertex p has an out-degree of 0, indicating that the environment doesn't actively act on any agents, but merely responds to their actions. Conversely, its in-degree is equal to $|\mathcal{A}|$, signifying that every agent in set $\mathcal{A}$ can act upon or perceive changes from the environment. **2)** A closed-loop interaction between vertices v_i and v_j is established if a path exists from v_i to v_j. This implies that the agent associated with v_i can act upon and then perceive changes or feedback from the entity associated with v_j. **3)** Chaos, or unpredictable and conflicting behaviors, can arise when the graph contains cycles. This cyclical structure indicates the possibility of agents receiving contradictory information, which can disrupt the intended flow of actions and perceptions within the organization.

Directed Acyclic Graph Considering the inherent structure of the environment p — which doesn't have an out-degree as outlined in property 1 — cyclical loops originating from the environment are naturally prevented. To ensure stability and avoid chaotic behaviors as identified in property 3, it is imperative that the sub-graph $\mathcal{G}_A$, comprising merely of agents, forms a directed acyclic graph (DAG). Moreover, upon closer examination, we discern that agents receiving simultaneous directives from multiple counterparts often lead

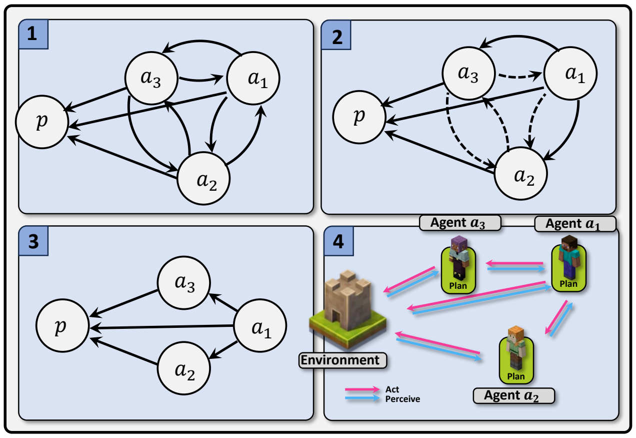

Figure 5: **An illustration of acyclic process.** Commencing with a comprehensive agent sub-graph constituted by agents a_1, a_2, and a_3, we designate a_1 as the root agent, with a_2 and a_3 functioning as leaf agents. In stage 2, we eliminate redundant edges, leading to the formation of a directed tree in stage 3. Stage 4 exemplifies the embodied organizational relations aligned with the tree structure.

to conflicting tasks. Such contradictions invariably induce disorder and inefficiencies within the system. Consequently, we mandate that the in-degree of each agent be restricted to less than 1. This configuration naturally evolves into what is recognized as a ***directed tree***.

In determining the tree's structure, we balance between its depth and breadth. A tree with greater depth results in an extended control chain, introducing increased complexity in control. Conversely, a tree with wider breadth imposes a higher burden on individual agents, as they grapple with managing multiple control loops simultaneously. Leveraging the capabilities of Large Language Models, individual agents have the prowess to handle vast amounts of information, rendering parallization concerns less pressing for LLM-enhanced agents. Thus, the tree's structural design should prioritize ***increased breadth***, allowing for a more expansive horizontal spread, while accommodating a ***shallower depth***. In practical application, we adopt a 1-depth tree structure. Within this setup, one agent assumes the role of the root (***root agent***), while the remaining agents function as the leaves (***leaf agents***).

Agents in the Embodied Graph Acting as the root node of the sub-graph, the root agent predominantly shoulders the responsibility of guiding and instructing the other agents. This role mirrors what we traditionally understand as "leadership" in human contexts. Equipped with the capabilities of the LLM, the root agent is adept at evaluating collective objectives, allocating tasks, and processing feedback.

Conversely, as peripheral nodes in the sub-graph, the leaf agents operate under the directives of the root agent. This role mirrors what we traditionally understand as "member" in human contexts. Their main functions involve interacting with the environment, executing assigned tasks, and subsequently providing feedback to the root agent regarding task outcomes.

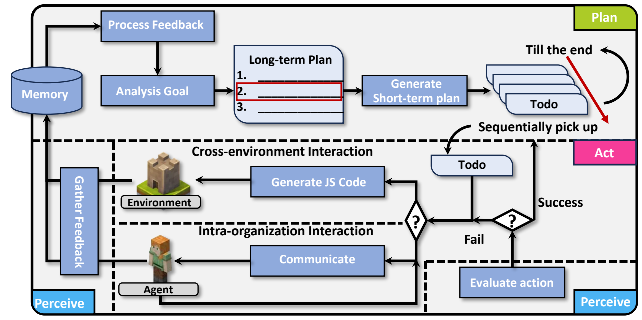

Figure 6: **An illustration of agent general functionalities.** The diagram segregates the agent functionalities into planning, action, and perception, divided by dashed lines. Blue squares, indicative of modules powered by LLM, interface with white diamonds that function as switches.

Agents with general functionalities The general functionality of agents also composes planning, action and perception, as depicted in Figure 6.

1) Planning Module: The planning phase commences with an LLM analyzing the perceived goals and subsequently producing a long-term plan. This is followed by another LLM which selects the current long-term stage and crafts a comprehensive short-term plan.

2) Action Module: Tasks enumerated on the short-term list are executed in sequence. When a task pertains to cross-environmental actions, an LLM is employed to generate the necessary JavaScript code for its realization in the environment. For tasks internal to the organization, the information is directly relayed to the conversation generation LLM. Importantly, Given scenarios where agents will communicate with the agents for rounds in one short-term plan, it becomes necessary for the agent to ***independently*** handle intra-organizational dynamics. In practical terms, we launch two auxiliary threads dedicated to executing actions in the environment and among agents, respectively. Conversely, the processes of planning and perception are integrated into a singular thread.

3) Perception Module: The perception module is bifurcated into short-term and long-term perceptual processes. Short-term perception is invoked immediately following the execution of a short-term task by the action module. This rapid response perception is primed to validate the success of the recent action. An LLM model evaluates the ensuing perception against the task's predefined criteria. If the action meets the stipulated conditions, a success notification is dispatched. However, if there's a misalignment, the code-generation LLM is alerted with specifics of the deviation, prompting it to produce a modified code iteration. This corrective loop persists for a set number of cycles. Should the task remain unfulfilled after these iterations, a failure alert will be recorded to the long-term perception. The long-term perception remains perpetually active, meticulously accumulating feedback from both the environment and fellow agents. This reservoir of insights is archived in the memory, equipping the planning module with a rich dataset to inform subsequent feedback-driven processes.

Figure 7: **An illustration of the task of "mine 100 stones".** The root agent (a1) allocates tasks to facilitate parallel processing, while the leaf agent (a2 & a3) autonomously refines these tasks into actionable steps.

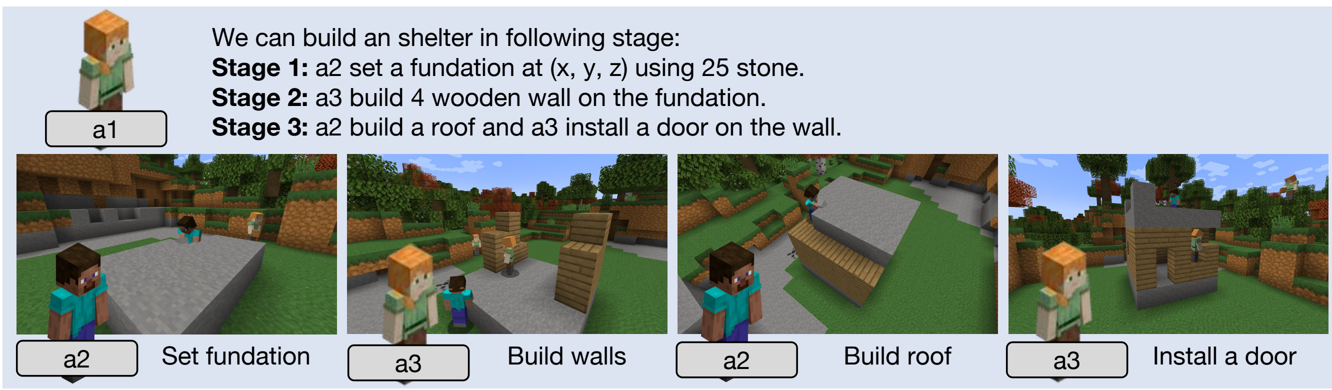

Figure 8: **An illustration of collective shelter construction in stages.** The root agent (a1) systematically arranges the tasks and schedules the leaf agents (a2 & a3) for phased execution.

Feedback Processor The feedback processor assimilates information from the mentioned memory, including both intra-organizational and cross-environmental sources and is responsible for strategizing actions directed towards the organization and the environment. The feedback processor is powered by a LLM, which consolidates the source information into a good prompt for the following goal analyzer. Subsequently, the goal analyzer recalibrates the long-term plan, taking cues from the feedback prompt.

However, it's crucial to note that the re-planning diverges based on the nature of tasks: cross-environment versus intra-organization. For cross-environmental tasks, re-planning is typically triggered by failures in the short-term tasks. This necessitates a revision in the long-term planning strategy to either circumvent or adapt to these problematic short-term tasks.

Conversely, for intra-organizational tasks, the re-planning becomes a tad more intricate. It requires a reassessment of fellow agents' competencies, a revaluation of shared objectives, a redistribution of tasks, and necessary tweaks to ensure optimal synergy and performance. As an instance, if a task assigned to a particular agent falters, the re-planning process might entail reassigning it to a more suited agent.

This feedback-driven approach ensures the planning module possesses a comprehensive grasp of the system's holistic state, thereby fostering more insightful and integrative decision-making.

Experiments

Experimental setup

Simulator & Benchmark Suite In earlier studies, open-ended single-agent research in Minecraft focused on the exploration of the environment and skill acquisition, a domain that appeared nebulous and unstructured (Wang et al. 2023a; Zhu et al. 2023; Fan et al. 2022). Contrarily, large-scale structured endeavors are commonplace in human societies. Addressing such tasks typically demands considerable manpower for specialized roles, such as logging and mining, coupled with superior coordination abilities. Such coordination is anticipated when constructing sophisticated multi-agent systems characterized by enhanced organizational structures and explicit team objectives. This benchmark seeks to validate two primary concerns: 1) the potential of embodied organizations to spontaneously parallelize

Tasks	Single agent TE ↓	CGI ↓	**CSAS-3** (Ours) TE ↓	CGI ↓	**CSAS-5** (Ours) TE ↓	CGI ↓
Mine 50 logs	8.3	2	5.2	2	6.2	7
Mine 50 stones	15.0	10	13.3	15	12.5	32
Mine 50 irons	56.5	8	24.0	27	25.2	37
Mine 100 logs	14.0	2	10.2	3	8.0	3
Mine 100 stones	23.2	9	11.6	10	14.7	40
Mine 100 irons	74.3	12	36.5	31	28.0	40

Table 1: **Performance on collective resource collection tasks**. `TE` denotes time efficiency (minutes), `CGI` denotes code generation iterations (times). The notation CSAS-3 signifies an organization comprising 3 agents, while CSAS-5 represents an organization encompassing 5 agents.

manual tasks and realize efficiency improvements, and 2) the capacity of these organizations to effectuate spontaneous and judicious labor allocation based on workforce attributes.

To rigorously investigate these challenges, we devised three distinct experimental tasks within the Minecraft framework: collective resources collection, collective tools crafting, and collective shelter construction. In these trials, a primary agent assumes the role of coordinator, orchestrating the activities of either three or five subordinate agents. The specific configurations of the three tasks are as follows:

1) Collective Collection: All subordinate agents commence with empty hands, tasked with amassing an ample quantity of rudimentary materials such as logs, stones, and raw iron. This assesses the capacity of the embodied organization to efficiently segment and concurrently allocate tasks. (Presented in Figure 8)

2) Collective Crafting: Starting with empty hands, all subordinate agents are mandated to produce a predetermined number of tools (encompassing axes, hoes, pickaxes, shovels, and swords) set at 5. The agents must first accumulate the requisite resources, then employ workbenches and furnaces to finalize the tool fabrication. This evaluates the embodied organization's proficiency in task division and parallel execution.

3) Collective Construction: Each subordinate agent possesses varied resources, for instance, logs and stones. The primary agent is responsible for designating appropriate tasks based on the resources each agent holds, culminating in the construction of a shelter. (Displayed in Figure 8)

Evaluation Metrics To thoroughly evaluate the performance of our proposed approach, we utilize the subsequent metrics:

Time Efficiency (TE): Measures the time resources needed to execute a designated task. Reduced time expenditure suggests greater system efficiency.

Code Generation Iterations (CGI): Represents the cumulative count of code iterations essential for task completion.

Tasks	Single agent TE ↓	CGI ↓	**CSAS-3** (Ours) TE ↓	CGI ↓
Craft 5 wooden tools	22.1	9	16.0	11
Craft 5 stone tools	24.0	15	17.2	19
Craft 5 iron tools	32.5	23	28.8	17
Build shelter	11.3	9	10.2	7

Table 2: **Performance on collective crafting, and collective construction**. `TE` denotes time efficiency (minutes), `CGI` denotes code generation iterations (times). The notation CSAS-3 signifies an organization comprising 3 agents, while CSAS-5 represents an organization encompassing 5 agents.

Method DAG	CLF	Collection TE ↓	CGI ↓	Crafting TE ↓	CGI ↓	Construction TE ↓	CGI ↓
		N/A	N/A	N/A	N/A	N/A	N/A
✓		N/A	N/A	N/A	N/A	N/A	N/A
	✓	25.2	29	19.7	20	N/A	N/A
✓	✓	24.0	27	17.2	19	10.0	7

Table 3: **Ablation study on close-loop feedback processor and Directed acyclic graph**. `DAG` stands for Directed Acyclic Graph. `CLF` stands for closed-loop feedback processor. `N/A` denotes the experiment failed.

Quantitative results

In our comprehensive experimental setup, our Closed-Loop Self-Organizing Agent System (CSAS) demonstrates remarkable coordination and task delegation capabilities. The root agent, functioning as an analog to human managers, effectively orchestrated task allocation for the agents, playing a pivotal role in scheduling and making corrective interventions throughout the process.

As illustrated in Table 1 and Table 2, the performance of our proposed Embodied Organization notably surpasses that of single agent in time efficiency metric. This superiority is even more pronounced in tasks with substantial workload; for instance, in scenarios such as "mining 100 stones", our system achieved a twofold increase in speed. This boost in performance can be attributed to the parallelized task handling abilities of multiple agents working in tandem.

Ablation studies

As depicted in Table 3, we undertook exhaustive ablation studies involving Directed Acyclic Graph (DAG) and closed-loop feedback processor. We employed a three-agent system as our test subject, conducting experiments in collective collection (mining 50 iron ores), collective crafting (producing 5 stone tools), and collective construction.

Directed Acyclic Graph ensure success execution. In the absence of DAG, agents fail to perform commands adeptly. They frequently delegate tasks bestowed by the root agent to other operatives or even revert them to the initiator. This engenders a chaotic operational sequence. Subsequently, agents manifest a hesitancy in addressing specific duties, culminating in heightened temporal expenses and an escalation in code generation occurrences. In the collective construction task, each leaf agent possesses designated resources, and the construction stages are interrelated. Any abdication of duties among the agents would consequently precipitate the experiment's failure.

The closed-loop feedback processor is the basis for self-organization. Abrogating the Closed-loop feedback processor at the organizational echelon effectively disrupts the feedback conduit from operatives to the root agent. Consequently, the root agent is rendered ineffective, incapacitated in promulgating directives grounded on the contemporaneous conditions of its constituents, thus obliterating its supervisory capacity and catalyzing the experiment's collapse.

Qualitative Results

Self-organization: division of labor As shown in Figure 7 and Example 1, the root agent identifies the workload and distributes it equitably among various leaf agents, enabling them to process the tasks in parallel. Moreover, as shown in Example 2, for distinct tasks that can be executed simultaneously, the root agent strategically allocates them to different leaf agents to maximize efficiency.

Self-organization: Error correction through feedback The root agent adjusts task allocations based on feedback from members. Specifically, there are two scenarios:

1) Cross-environment Feedback: As shown in Example 3, if a leaf agent fails in a particular task, the root agent aids in identifying the underlying causes and formulates a more pragmatic work plan to navigate challenges, mirroring the role of mentors in human societies.

Example 1: Equivalent splitting work
```
[Task] Mine 100 logs.
[Division by root agent] a2 mine 50
logs, a3 mine 50 logs.
```

Example 2: Assign different tasks separately
```
[Task] Craft 1 iron axe and 1 iron
hoe.
[Division by root agent] a2 craft 1
iron axe, while a3 craft 1 iron hoe.
```

Example 3: Task refinement
```
[Previous plan]
Stage 1: a2 gather resources
 a2 mine 10 woods.
 a2 mine 10 stones.
... ...
[Feedback] a2 failed to mine 10
stones, because he lacked of a
pickaxe.
```

Continuation of exmaple 3
```
[refined plan]
Stage 1: a2 gather resources
 a2 mine 10 woods.
 a2 craft a wooden pickaxe.
 a2 mine 10 stones with a wooden
pickaxe.
... ...
```

Example 4: Task redistribution
```
[Previous plan]
Stage 1: Gather resources
 a2 mine 10 irons.
 a3 mine 10 irons.
... ...
[Feedback] a2 failed to mine 10
irons, because he lacked of a
pickaxe. a3 succeeded to mine 10
irons.
[refined plan]
Stage 1: Gather resources
 a2 mine another 10 irons.
 a3 mine 10 oak logs.
... ...
```

2) Intra-organization Feedback: As shown in Example 4, root agent observes that leaf agent a2 excels in a specific task while another leaf agent a3 consistently underperforms, the task will be reassigned to leaf agent a2, and a more straightforward task will be allocated to leaf agent a3. This strategy enhances the robustness of the multi-agent team, allowing for redundancy and ensuring the seamless progression of tasks.

Conclusion

In this research, we delved into the potential of integrating the planning-action-perception closed-loop, which has previously empowered embodied agents with human-analogous functionalities, into a broader organization level, which we called Embodied Organization. To this end, we unveiled the Closed-Loop Self-Organizing Agent System (CSAS), a pioneering framework that endorses an acyclic architecture for agent interconnections, upholds agents with general functionality of independent cross-environment and intra-organization interaction, and a feedback mechanism. Our evaluations within the dynamic realm of the Minecraft universe substantiate our system's proficiency in agent self-organization, its enhanced efficiency over standalone agents, and its dexterity in navigating diverse challenges. Ultimately, this exploration augments the scope of Embodied AI, directing it ever closer to replicating the sophistication and intricacy of human organizational systems.

References

Akata, E.; Schulz, L.; Coda-Forno, J.; Oh, S. J.; Bethge, M.; and Schulz, E. 2023. Playing repeated games with Large Language Models. *arXiv preprint*.

Baker, B.; Akkaya, I.; Zhokov, P.; Huizinga, J.; Tang, J.; Ecoffet, A.; Houghton, B.; Sampedro, R.; and Clune, J. 2022. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*.

Dong, Y.; Jiang, X.; Jin, Z.; and Li, G. 2023. Self-collaboration Code Generation via ChatGPT. *arXiv preprint*.

Fan, L.; Wang, G.; Jiang, Y.; Mandlekar, A.; Yang, Y.; Zhu, H.; Tang, A.; Huang, D.-A.; Zhu, Y.; and Anandkumar, A. 2022. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*.

Gray, J.; Srinet, K.; Jernite, Y.; Yu, H.; Chen, Z.; Guo, D.; Goyal, S.; Zitnick, C. L.; and Szlam, A. 2019. Craftassist: A framework for dialogue-enabled interactive agents. *arXiv preprint*.

Guss, W. H.; Castro, M. Y.; Devlin, S.; Houghton, B.; Kuno, N. S.; Loomis, C.; Milani, S.; Mohanty, S.; Nakata, K.; Salakhutdinov, R.; et al. 2021. The minerl 2020 competition on sample efficient reinforcement learning using human priors. *arXiv preprint*.

Guss, W. H.; Houghton, B.; Topin, N.; Wang, P.; Codel, C.; Veloso, M.; and Salakhutdinov, R. 2019. Minerl: A large-scale dataset of minecraft demonstrations. *arXiv preprint*.

Hafner, D.; Pasukonis, J.; Ba, J.; and Lillicrap, T. 2023. Mastering diverse domains through world models. *arXiv preprint*.

Hong, S.; Zheng, X.; Chen, J.; Cheng, Y.; Zhang, C.; Wang, Z.; Yau, S. K. S.; Lin, Z.; Zhou, L.; Ran, C.; et al. 2023. MetaGPT: Meta Programming for Multi-Agent Collaborative Framework. *arXiv preprint*.

Johnson, M.; Hofmann, K.; Hutton, T.; and Bignell, D. 2016. The Malmo Platform for Artificial Intelligence Experimentation. In *IJCAI*.

Kanervisto, A.; Milani, S.; Ramanauskas, K.; Topin, N.; Lin, Z.; Li, J.; Shi, J.; Ye, D.; Fu, Q.; Yang, W.; et al. 2022. Minerl diamond 2021 competition: Overview, results, and lessons learned. *NeurIPS 2021 Competitions and Demonstrations Track*.

Kanitscheider, I.; Huizinga, J.; Farhi, D.; Guss, W. H.; Houghton, B.; Sampedro, R.; Zhokhov, P.; Baker, B.; Ecoffet, A.; Tang, J.; et al. 2021. Multi-task curriculum learning in a complex, visual, hard-exploration domain: Minecraft. *arXiv preprint*.

Kiseleva, J.; Li, Z.; Aliannejadi, M.; Mohanty, S.; ter Hoeve, M.; Burtsev, M.; Skrynnik, A.; Zholus, A.; Panov, A.; Srinet, K.; et al. 2022a. Interactive grounded language understanding in a collaborative environment: Iglu 2021. In *NeurIPS 2021 Competitions and Demonstrations Track*.

Kiseleva, J.; Skrynnik, A.; Zholus, A.; Mohanty, S.; Arabzadeh, N.; Côté, M.-A.; Aliannejadi, M.; Teruel, M.; Li, Z.; Burtsev, M.; ter Hoeve, M.; Volovikova, Z.; Panov, A.; Sun, Y.; Srinet, K.; Szlam, A.; and Awadallah, A. 2022b. IGLU 2022: Interactive Grounded Language Understanding in a Collaborative Environment at NeurIPS 2022.

Li, G.; Hammoud, H. A. A. K.; Itani, H.; Khizbullin, D.; and Ghanem, B. 2023. Camel: Communicative agents for" mind" exploration of large scale language model society. *arXiv preprint*.

Lin, Z.; Li, J.; Shi, J.; Ye, D.; Fu, Q.; and Yang, W. 2021. Juewu-mc: Playing minecraft with sample-efficient hierarchical reinforcement learning. *arXiv preprint*.

Mandi, Z.; Jain, S.; and Song, S. 2023. RoCo: Dialectic Multi-Robot Collaboration with Large Language Models. *arXiv preprint*.

Mao, H.; Wang, C.; Hao, X.; Mao, Y.; Lu, Y.; Wu, C.; Hao, J.; Li, D.; and Tang, P. 2022. Seihai: A sample-efficient hierarchical ai for the minerl competition. In *DAI*.

Mohanty, S.; Arabzadeh, N.; Kiseleva, J.; Zholus, A.; Teruel, M.; Awadallah, A.; Sun, Y.; Srinet, K.; and Szlam, A. 2023. Transforming Human-Centered AI Collaboration: Redefining Embodied Agents Capabilities through Interactive Grounded Language Instructions. *arXiv preprint*.

Mohanty, S.; Arabzadeh, N.; Teruel, M.; Sun, Y.; Zholus, A.; Skrynnik, A.; Burtsev, M.; Srinet, K.; Panov, A.; Szlam, A.; et al. 2022. Collecting Interactive Multi-modal Datasets for Grounded Language Understanding. *arXiv preprint*.

Park, J. S.; O'Brien, J. C.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative Agents: Interactive Simulacra of Human Behavior.

Perez-Liebana, D.; Hofmann, K.; Mohanty, S. P.; Kuno, N.; Kramer, A.; Devlin, S.; Gaina, R. D.; and Ionita, D. 2019. The Multi-Agent Reinforcement Learning in Malm\" O (MARL\" O) Competition. *arXiv preprint*.

Qian, C.; Cong, X.; Yang, C.; Chen, W.; Su, Y.; Xu, J.; Liu, Z.; and Sun, M. 2023. Communicative Agents for Software Development. *arXiv preprint*.

Shinn, N.; Labash, B.; and Gopinath, A. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint*.

Skrynnik, A.; Staroverov, A.; Aitygulov, E.; Aksenov, K.; Davydov, V.; and Panov, A. I. 2021. Hierarchical deep q-network from imperfect demonstrations in minecraft. *Cognitive Systems Research*.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023a. Voyager: An open-ended embodied agent with large language models. *arXiv preprint*.

Wang, Z.; Cai, S.; Liu, A.; Ma, X.; and Liang, Y. 2023b. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint*.

Xiang, J.; Tao, T.; Gu, Y.; Shu, T.; Wang, Z.; Yang, Z.; and Hu, Z. 2023. Language Models Meet World Models: Embodied Experiences Enhance Language Models. *arXiv preprint*.

Zhang, H.; Du, W.; Shan, J.; Zhou, Q.; Du, Y.; Tenenbaum, J. B.; Shu, T.; and Gan, C. 2023. Building Cooperative

Embodied Agents Modularly with Large Language Models. *arXiv preprint*.

Zhu, X.; Chen, Y.; Tian, H.; Tao, C.; Su, W.; Yang, C.; Huang, G.; Li, B.; Lu, L.; Wang, X.; et al. 2023. Ghost in the Minecraft: Generally Capable Agents for Open-World Enviroments via Large Language Models with Text-based Knowledge and Memory. *arXiv preprint*.