Hierarchical Auto-Organizing System for Open-Ended Multi-Agent Navigation

Zhonghan Zhao[*,1]**, Kewei Chen**[*,2]**, Dongxu Guo**[*,2]**, Wenhao Chai**[†,3]**, Tian Ye**[*,4]**,**
Yanting Zhang[2]**, and Gaoang Wang**[1,✉]

[1] Zhejiang University   [2] Donghua University   [3] University of Washington
[4] Hong Kong University of Science and Technology (GZ)
{zhonghan.22, gaoangwang}@intl.zju.edu.cn, wchai@uw.edu

Figure 1: Illustration of the functionality of **HAS**, in the Minecraft environment. Given a navigation goal in the form of images, objects, or audio, collectives of agents autonomously self-organize to undertake collaborative endeavors of navigation like searching and exploring.

Abstract

Due to the dynamic and unpredictable open-world setting, navigating complex environments in Minecraft poses significant challenges for multi-agent systems. Agents must interact with the environment and coordinate their actions with other agents to achieve common objectives. However, traditional approaches often struggle to efficiently manage inter-agent communication and task distribution, crucial for effective multi-agent navigation. Furthermore, processing and integrating multi-modal information (such as visual, textual, and auditory data) is essential for agents to comprehend their goals and navigate the environment successfully and fully. To address this issue, we design the HAS framework to auto-organize groups of LLM-based agents to complete navigation tasks. In our approach, we devise a hierarchical auto-organizing navigation system, which is characterized by 1) a hierarchical system for multi-agent organization, ensuring centralized planning and decentralized execution; 2) an auto-organizing and intra-communication mechanism, enabling dynamic group adjustment under subtasks; 3) a multi-modal

---

[*]Equal contribution, [†]Equal engineering contribution, ✉ Corresponding author.

information platform, facilitating multi-modal perception to perform the three navigation tasks with one system. To assess organizational behavior, we design a series of navigation tasks in the Minecraft environment, which includes searching and exploring. We aim to develop embodied organizations that push the boundaries of embodied AI, moving it towards a more human-like organizational structure.

1 INTRODUCTION

Drawing on large language models (LLMs) with human knowledge, communicative competence Mao et al. (2020a;a), and decision-making capabilities, embodied agents exhibit human-like intelligence in playing games (Park et al., 2023; Wang et al., 2023a; Zhao et al., 2023), programming (Qian et al., 2023; Hong et al., 2023), and robotic tasks (Zhang et al., 2023b; Mandi et al., 2023). The development of individual intelligence has led to creating a new, collaborative framework where multiple agents Chen et al. (2023a;b) work together.

This shift towards a multi-agent system Hong et al. (2023); Chen et al. (2023a) uses advanced language understanding and decision-making abilities to enhance intelligence through interactions and data sharing. Agents in this system specialize in various tasks, sharing insights to enhance overall efficiency. This collaborative approach not only optimizes execution but also fosters a learning environment where agents continually improve through shared intelligence. Their interaction leads to refined skills, equipping them to handle complex, large-scale tasks through parallel processing and coordinated collaboration, a notable advancement in autonomous systems.

Multi-modal navigation (Wijmans et al., 2019; Yu et al., 2021; Du et al., 2020; Kwon et al., 2023; Chen et al., 2021; Moudgil et al., 2021) stands at the forefront of contemporary AI research, representing a rapidly evolving field that aims to integrate various sensory inputs into a cohesive navigational strategy. This emerging domain extends the capabilities of multi-agent systems by requiring them to interpret and act upon a confluence of sensory data. Unlike traditional methods focusing on rendered images or static virtual environments, works on dynamic environments Deng et al. (2023), such as multi-modal navigation in open-ended environments like Minecraft, pose a more complex challenge. Agents in Minecraft Wang et al. (2023a); Zhao et al. (2023) navigate a world full of freedom and variability, making it an ideal testbed for embodied agent systems. Minecraft's unpredictable and open-ended environment is an ideal testing ground for advanced AI systems. These systems emulate human adaptability and intelligence, processing and sharing multi-modal data to strategically navigate and complete tasks. This approach goes beyond simple improvements, marking a significant leap in developing autonomous systems that handle complex tasks with remarkable autonomy and skill. Integrating Multi-modal Language Models (MLMs) into embodied agents enhances navigational efficiency. Equipped with LLMs, these agents engage intelligently with their environment. The zero-shot performance capabilities of LLMs enable them to interpret and act upon data without explicit programming. This makes them ideal for open-ended environments.

Our vision is centered on employing MLM-based embodied agents to revolutionize navigation within the intricate landscapes of Minecraft. Environments in this game present unique challenges, such as the need to respond to image, audio, and object cues in real-time, as highlighted in Figure 1. By harnessing the zero-shot learning capabilities and the nuanced decision-making abilities of LLMs, our agents can navigate these spaces with efficiency and adaptability akin to human intuition. Such versatility is achieved without requiring exhaustive retraining or complex reconfigurations, marking a substantial leap towards autonomous systems that engage with their surroundings in more sophisticated and human-like ways.

We summarize our contributions as follows:

• We introduce **HAS**, a hierarchical structure for multi-agent navigation based on LLMs in the Minecraft environment. It utilizes centralized planning with decentralized execution, enabling efficient multi-modal navigation in open-ended environments.

• We design an auto-organizing and intra-communication mechanism to dynamically adjust the key role and action group based on the task allocation and maintain inter-group communication to ensure efficient collaboration.

• We achieve state-of-the-art performance on the asynchronous multi-modal navigation task on image, audio, and object goals in Minecraft's open-ended environment.

2 RELATED WORKS

2.1 INTELLIGENT AGENT IN MINECRAFT

As an open-ended sandbox game, Minecraft has always been an ideal setting for testing the performance of intelligent agents (Johnson et al., 2016; Hofmann, 2019). The agents must autonomously perform various tasks in Minecraft, such as chopping trees, crafting tools, and mining diamonds. At the beginning, much of the works focus on exploring reinforcement learning (Lin et al., 2021; Mao et al., 2022; Skrynnik et al., 2021; Lifshitz et al., 2023) or imitation learning (Amiranashvili et al., 2020; Baker et al., 2022), without satisfactory performance. VPT (Baker et al., 2022) and MineDojo (Fan et al., 2022) collect internet-scale datasets for their model pre-training. VPT enables direct learning to act during video pre-training and using these behaviors as exploration priors for reinforcement learning. Yet, recent works found that the pre-trained LLMs could serve as a strong "mind" that provides planning ability to the agents. Voyager (Wang et al., 2023a) is a single-robot multiple agent system that leverages multiple groups of GPT-4 (OpenAI, 2023) as a high-level planner, low-level action code generator, critic generator, and curriculum manager. Plan4MC (Yuan et al., 2023) proposes a skill graph pre-generated by the LLMs. DEPS (Wang et al., 2023e), an interactive planning method based on LLMs, addresses multi-step reasoning issues in open-world planning. GITM (Zhu et al., 2023b) develops a set of structured actions and leverages LLMs to generate action plans for the agents to execute, achieving impressive results in various tasks.

2.2 EMBODIED MULTIMODAL MODEL

Embodied agents integrate sensory perceptions, physical actions, and computational intelligence to accomplish tasks and goals within their environment. Key areas are wide-ranging, including Navigation (Wijmans et al., 2019; Yu et al., 2021; Du et al., 2020; Kwon et al., 2023; Chen et al., 2021; Moudgil et al., 2021), Embodied Question Answering (Das et al., 2018; Yu et al., 2019; Datta et al., 2022), Active Visual Tracking (Luo et al., 2018; Zhong et al., 2021; Luo et al., 2019; Zhong et al., 2019), and Visual Exploration (Liu & Okatani, 2022; Dean et al., 2020; Chen et al., 2018). The field is evolving rapidly with the development of Large Language Models (LLMs) (Song et al., 2022) and Multimodal LLMs (MLLMs) (Alayrac et al., 2022; Zhu et al., 2023a; Li et al., 2023a; 2022; 2023b; Gong et al., 2023; Lyu et al., 2023; Ye et al., 2023; Dai et al., 2023; Wang et al., 2023c; Liu et al., 2023a; Maaz et al., 2023; Su et al., 2023; Gao et al., 2023), integrating multiple modalities for more effective processing. A prime example of this innovation is PaLM-E (Driess et al., 2023), a sophisticated multimodal model with 562B parameters, adept at a broad spectrum of embodied tasks and demonstrating exceptional capabilities in visual reasoning.

2.3 LLM-BASED MULTI-AGENT FRAMEWORKS

Large Language Models (LLMs) are skilled at completing new tasks when given prompt-based instructions. Autonomous agents based on Large Language Model-based (LLM-based) models have gained significant interest in industry and academia (Wang et al., 2023b). Several works (Wang et al., 2023d; Du et al., 2023; Zhuge et al., 2023; Hao et al., 2023; Akata et al., 2023; Zhang et al., 2023a) have augmented the problem-solving abilities of LLMs by incorporating discussions among multiple agents. Stable-Alignment (Liu et al., 2023b) generates instruction datasets by reaching a consensus on value judgments through interactions among LLM agents in a sandbox. Some works in the field of artificial intelligence focus on studying sociological phenomena. For instance, Generative Agents (Park et al., 2023) creates a virtual "town" comprising 25 agents to investigate language interaction, social understanding, and collective memory. The Natural Language-Based Society of Mind (Zhuge et al., 2023) involves agents with different functions interacting to solve complex tasks through multiple rounds of Mindstorms. In addition, others (Cai et al., 2023) propose a model for cost reduction by combining large models as tool makers and small models as tool users.

Some works emphasize cooperation and competition related to planning and strategy (Bakhtin et al., 2022), some propose LLM-based economies (Zhuge et al., 2023), and others propose LLM-based

programming (Hong et al., 2023). Developing an LLM-based multiple embodied agent system with control capability is challenging. We face difficulties in multi-agent cooperation, such as low efficiency of global perception and generalizing dynamic groups. To overcome these challenges, we aim to apply a hierarchical auto-organizing system in multi-agent navigation.

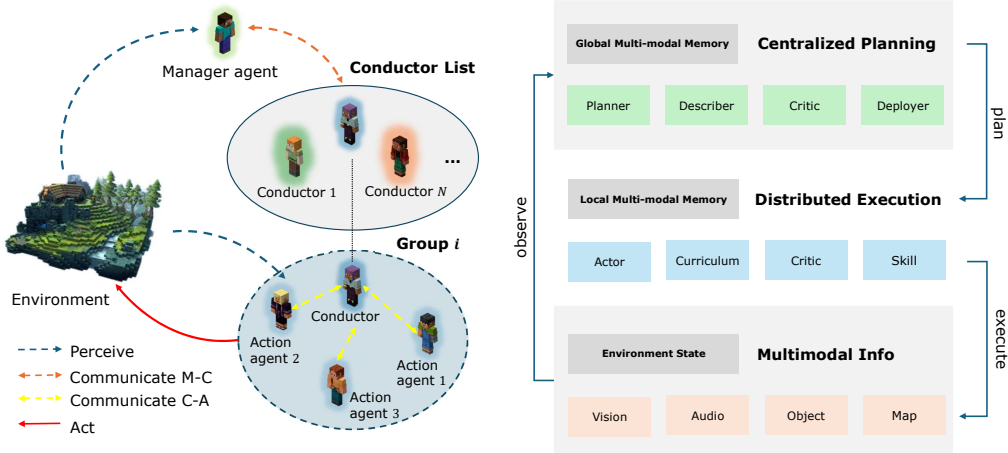# 3 HAS: A MULTI-AGENT NAVIGATION FRAMEWORK

Figure 2: **HAS framework.** M-MLM and A-MLM correspond to the memory-augmented multi-modal language models of the manager and action agent, respectively. It also utilizes a multi-modal memory to store and obtain experiences as references for planning. HAS can improve its planning skills by exploring its own proposed tasks with self-instruction and using its growing memory to plan tasks it has visited before.

As shown in Figure 2, the **HAS** is a hierarchical LLM-based multi-modal multi-agent navigation framework denoted as $\mathcal{F}$, which can manage and execute complex multi-agent navigation tasks on image (Im), object (Oj), and audio (Au) goals with perception on the state list of vision (v), audio (a), and other properties p within open-ended environments by leveraging cognitive and collaborative capabilities of the multi-modal language model (**MLM**):

$$\mathbf{a}_i^c = \mathcal{F}(s, g, \mathcal{M}), \ s = (v_i^c, a_i^c, p_i^c)_{i \leq N}, \ g \in G = \{Im, Oj, Au\} \tag{1}$$

where (v_i^c, a_i^c, p_i^c) represents the state list of vision, audio, and other properties of one conductor agent, g is the original goal, and $\mathcal{M}$ is the dynamic map. Then, we get $\mathbf{a}_i^c$ as the action for the conductor agent i.

The Hierarchical architecture consists of two primary operational domains: higher-order centralized planning, which is managed by the manager multi-modal language model ($\mathbf{MLM}^M$), and ground-level decentralized execution, which is conducted by the conductor model ($\mathbf{MLM}^C$), then the action a_i^c of the conductor can be obtained as follows,

$$\mathcal{F}(s, g, \mathcal{M}) = \mathbf{MLM}^C(\mathbf{MLM}^M(s, g, \mathcal{M}), s_i), \quad s_i = (v_i^c, a_i^c, p_i^c) \tag{2}$$

where $\mathbf{MLM}^C$ and $\mathbf{MLM}^M$ represent the multi-modal language models of the conductor and manager agents.

3.1 CENTRALIZED PLANNING WITH DECENTRALIZED EXECUTION

Evoked from the Centralized Training with Decentralized Execution (CTDE) framework (Hu et al., 2023; Sunehag et al., 2017; Lowe et al., 2017; Mao et al., 2018) for cooperative Multi-Agent Reinforcement Learning (MARL), we propose Centralized Planning with Decentralized Execution (CPDE) for the cooperative LLM-based Multi-Agent system. This architecture leverages global state information to inform the planning process, while the execution of tasks is carried out by local agents independently. CPDE is tailored to maximize the synergy of global oversight and local autonomy, ensuring efficient navigation and task completion in complex environments.

Centralized planning for manager agent. The centralized planning process for a manager agent M simulates a high-level understanding like that of a human planner. The process includes understanding the environment's dynamic global states, recognizing the conductor agents' capabilities and limitations, and devising a strategy. It contains $\mathbf{MLM}^M$ consisting of 4 MLM modules with different functions for the manager as mentioned in Section 3.2 and a Global Multi-modal Memory for storing multi-modal information as mentioned in Section 3.3.

Decentralized execution for action agent. The decentralized execution process is designed to capitalize on the autonomy and flexibility of conductor agents C. These agents navigate the environment, perform tasks, and learn from their interactions guided by the strategic direction from the centralized planning of the manager agent. Note that these conductors can deploy several action agents for similar goals of one sub-goal. It contains $\mathbf{MLM}^C$ comprising 4 MLM modules for conductors with different functions as mentioned in Section 3.2 and a Local Multi-modal Memory for storing multi-modal information as mentioned in Section 3.3.

Auto-organizing mechanism. The auto-organizing mechanism is a spontaneous grouping mechanism to promote multi-agents' efficiency. In the centralized planning phase, the manager agent auto-organizes several conductor agents based on the global environment and tasks, which takes advantage of the planning capabilities of MLM and is inspired by AutoAgents (Chen et al., 2023a) to deploy agents with different roles automatically. During the decentralized execution stage, we are inspired by the Self-Organized Group (SOG) (Shao et al., 2022) to solve the issue of zero-shot generalization ability with dynamic team composition and varying partial observability. We improve and use a novel auto-organizing mechanism. Each group has conductor-action agent consensus, where the action agents can only communicate with their conductor. Additionally, we utilized the summarization ability (Ma et al., 2023) from LLM to summarize and distribute the messages received to all affiliate group members to hold a unified schedule.

3.2 Multi-modal Language Model

The Multi-modal Language Model $\mathbf{MLM} = \{\mathbf{MLM}^M, \mathbf{MLM}^C\}$, consisting of two different types of MLM of the manager and conductor agents. $\mathbf{MLM}^M = \{\mathcal{P}_l, \mathcal{D}_s, \mathcal{C}_r^m, \mathcal{D}_p\}$, which consists of Planner, Describer, Critic, and Deployer for the manager. They formulate aligned task plans, condense and translate multi-modal data, refine strategies through feedback, and assign and direct agent subtasks respectively. $\mathbf{MLM}^C = \{\mathcal{A}_c, \mathcal{C}_u, \mathcal{C}_r^c, \mathcal{S}_k\}$, which consists of Actor, Curriculum, Critic, and Skill module for conductors. They translate strategic plans into executable actions, orchestrate dynamic group formations, and distribute tasks across agents, ensuring alignment with centralized directives and facilitating continuous learning and adaptation through a curriculum of complex tasks.

Adaptive planning with MLM. Our approach integrates the environment's observations and task directives to plan actions based on the current scenario. We begin by translating multi-modal observations into textual descriptions, utilizing a method that avoids direct scene captioning by the MLM. Instead, we extract keywords for items from the STEVE-21K dataset (Zhao et al., 2023) and employ GPT-4 to craft sentences that articulate these observations. The MLM identifies relevant condition sentences from textual observations during the planning phase. It also incorporates additional context, such as biome types and inventory levels, into text formats via predefined templates. We generate action plans by re-engaging the MLM's linguistic component with the task instructions and these descriptive texts. This methodology leverages the MLM's capabilities in a layered manner, yielding more accurate situational descriptions and plans that significantly reduce the likelihood of generating unrealistic elements compared to fully integrated models.

Autonomous error correction and proactive planning. HAS enhances its planning through a closed-loop feedback mechanism, automatically correcting failures by analyzing feedback and identifying errors using its self-explanation capabilities. Unlike other agents, it generates improved plans without human input or extra information. Additionally, HAS simulates and evaluates each plan step to identify potential flaws early, reducing the likelihood of encountering difficult situations due to plan failures. This proactive approach enables it to foresee issues like insufficient resources, which could hinder task completion.

3.3 MULTI-MODAL MEMORY

Research (Hong et al., 2023) has shown that memory mechanisms play a crucial role in the functioning of generalist agents. Equipping HAS with multi-modal memory enables it to plan using pre-existing knowledge and real-world experiences, improving planning accuracy and consistency. The MLM in HAS allows leveraging these experiences in context without requiring additional model updates.

We have illustrated the design of our multi-modal memory system. At a high level, this system is a key-value memory model with multi-modal keys comprising both the task and the observation of the state when this memory entry is created. The values stored in this memory system are the plans that were successfully executed. Since the plans are in an open-ended environment, Minecraft, there could be multiple entries with the same task but different observations and plans. As a result, HAS needs to generate multi-modal queries based on the current task and situations to retrieve the relevant memory entries.

Retrival-augmented storage. Retrieval-augmented storage (RAS) enables long-term planning capability by Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Mao et al., 2020b). RAG improves the quality of responses generated by language models with external sources of knowledge to complement the model's internal representation. Instead of external knowledge libraries, we use the collected multi-modal memory as the knowledge library and retrieve interactive experiences as demonstration prompts to improve the planning results. The formulation is as follows:

$$p(y \mid x) \approx \sum_{z \in \text{top-k}(p(\cdot \mid x))} p_\eta(z \mid x) \cdot p_\theta(y \mid x, z), \tag{3}$$

where x, y, and z represent instructions, plans, and retrieved memory entries. p_η and p_θ denote retrieval and planning models. This retrieval-augmented planning method helps HAS to ground its internal knowledge into open-ended environments efficiently. It also leverages historical interaction feedback to solve hallucinations within LLMs and produce more accurate plans.

Multi-modal retrieval. Multi-modal retrieval (MMR) enables efficient access to a rich repository of multi-modal memories. This process is initiated with a query containing textual and visual elements. To align this query with the trajectories stored within the multi-modal memory, we utilize the manager's Describer $\mathcal{D}_s$. The Describer converts visual information into a textual format. This textual description serves as an image tag, amalgamating with other textual data or as a textual representation for audio information.

When a retrieval request is made from the multi-modal memory, especially when the information is an amalgamation of image and text, the Describer module $\mathcal{D}_s$ is employed to transcribe the image into text. Subsequently, this description is used to compute the similarity across the multi-modal memory entries. The top-k most similar entries are retrieved for further processing. The formalization of this retrieval process is as follows:

$$\mathcal{R}(q_t, q_v) = \sigma(\mathcal{D}_s(q_v), q_t), \tag{4}$$

where q_t denotes the textual query, q_v represents the visual query, and $\mathcal{R}$ signifies the retrieval function. The function $\mathcal{S}$ computes the similarity between the multi-modal memory entries and the query, using the textual description provided by $\mathcal{D}_s$.

Dynamic map. The dynamic map visually represents the exploration domain, showcasing only the areas the agents have explored. It is the main way for the manager agent to see the global environment. When using the map, it will be used in the form of an image. It is updated in real-time with information from all action agents, providing a strategic overview of the environment. This map is instrumental in planning and executing navigation tasks as it reflects the current knowledge and discoveries made by the agent collective.

The following equations can formalize the dynamic map's function:

$$\mathcal{M}_t = \mathcal{M}_{t-1} \cup \bigcup_{i=1}^{n} S_{i,t}, \qquad S_{i,t} = F(\mathbf{txt}_{i,t}) \tag{5}$$

Where M_t represents the dynamic map at time t, $S_{i,t}$ is the state information from agent i at time t, including text data $\mathbf{txt}_{i,t}$ like the place name, special materials. F is the function that integrates the text data into the map, and n is the number of active agents. This real-time updating mechanism ensures that the dynamic map remains an accurate and current representation of the exploration field.

4 EXPERIMENT

Our experiments aim to achieve three goals using HAS in challenging Minecraft navigation tasks. Firstly, we want to evaluate the performance of HAS against baselines that do not fully address the issues faced by open-world agents. Secondly, we aim to understand the factors that contribute to these results. Lastly, we want to explore the potential of HAS for life-long learning and its benefits in long-horizon tasks. We will first introduce the evaluation settings, present the main comparative results and ablation studies, and conclude with an exploratory trial on long-horizon tasks.

4.1 EXPERIMENTAL SETUPS

We select gpt-4-1106-vision-preview (Yang et al., 2023) as the base model. Our simulation environment is based on MineDojo (Fan et al., 2022) and uses Mineflayer (PrismarineJS, 2013) APIs for motor controls. The maximum number of robots that can be allocated based on this environment is 8, which is also our experimental robots' upper limit.

4.2 BASELINES

Currently, no MLM-driven multi-agents (robots) work out of the box for Minecraft, so we carefully selected several representative algorithms as baselines for our experiment. They rely on extracting information from a system's backend, presenting a significant divergence from real-world scenarios.

Voyager (Wang et al., 2023a) is a blind, single-robot system, relying only on textual grounding for perception. It has a long-term procedural memory that stores a hierarchical library of code-based grounding procedures. Complex skills can use simpler skills as sub-procedures. Voyager is known for its ability to explore areas and master the tech tree. However, its main focus is to prompt GPT-4 (OpenAI, 2023) on background text messages in embodied agents. We use multiple hosts to deploy multiple models to work directly on the server. We convert the input of the image goal into a text task through GPT-4V, using the same Describer module as ours.

STEVE (Zhao et al., 2023) is a multi-modal single-robot system that combines the vision unit with the STEVE-13B with the code database. It focuses on introducing a visual module to endow the model with visual perception capabilities in processing visual perception information and handling task reasoning for skill-related execution. Similarly, using multiple hosts, we deploy multiple models to work directly on the server. This single-agent method optimizes performance. We also convert the input of the image goal into a text task by using the same Describer module as ours.

4.3 TASK SETTING

We set the world on peaceful mode and agents always start in survival mode with an empty inventory so that navigation tasks can be performed without interruption. Meanwhile, we choose over 15 environment seeds of 6 different terrains from the STEVE-21K dataset (Zhao et al., 2023) for evaluation. The tasks we chose mainly test the efficiency of long-distance directivity navigation, short-range non-directional navigation, and free-world exploration.

A task is considered successful when the target object is close to the target or when a certain number of targets are reached. Due to the open-world nature of Minecraft, the world and initial position that the agent is spawned at could vary a lot. Therefore, we conduct at least 30 tests for each task and report the average efficiency and success rate to ensure a thorough assessment.

4.4 EVALUATION RESULTS

Method	# agents	Image Goal		Object Goal		Audio Goal	
		# iters (↓)	success rate (↑)	# iters (↓)	success rate (↑)	# iters (↓)	success rate (↑)
Voyager	1	95	0.21	64	0.41	21	0.67
	3 / 2 / 5	45	0.47	36	0.59	6	0.85
STEVE	1	85	0.25	71	0.31	13	0.71
	5 / 5 / 4	32	0.52	29	0.57	6	0.82
HAS (Ours)	1	27	0.76	15	0.83	4	0.87
	8 / 7 / 3	**6**	**0.84**	**4**	**0.95**	**2**	**0.99**

Table 1: **Comparison on goal search task.** # iters represent the average number of iterations required to finish each task (find goals) with a maximum of 100 prompting iterations (only within the statistical range). The success rate is for task fulfillment. We list the one-agent and best performance with the number of agents (from left to right represents different goals) on the bottom line for each method according to # agents. Note that due to dynamic robots, this number for our method refers to the peak number.

Method	# agents	# iters (↓)	# blocks (↑)
Voyager[(Wang et al., 2023a)]	1	34	28
	3	14	81
STEVE[(Zhao et al., 2023)]	1	15	65
	4	6	211
HAS	1	14	68
	8	**2**	**367**

Table 2: **Comparison on continues block search task.** # iters represent the average number of iterations required to locate the first ten diamond blocks with a maximum of 100 prompting iterations; the lower this number, the higher the task completion efficiency. # blocks denotes the average number of diamond blocks found over 100 iterations, with higher values indicating better performance. We list the one-agent and best performance with the number of agents on the bottom line for each method according to # agents.

Multi-modal goal search. Multi-modal goal search includes goals of Image, Object, and Audio. Object labels identify in-game items such as villages, pyramids, and animals. Image labels help locate objects using images. Audio labels are used to detect sounds outside of the player's range. Due to an insufficient audio library, we set the perceptible range around the target and passed it to the agent as text. It is similar to finding objects at close range, expressing distance through feedback intensity values.

As shown in Table 1, **HAS** achieves the best performance as a multi-agent system. Due to the auto-organizing mechanism, multiple agents can be better planned than directly adding multiple free agents. Even with a single-agent system, our performance is still the best. This is because our system decomposes tasks layer by layer, allowing both managers and action agents to maintain focus on their specific tasks. With the help of dynamic maps, the agent can be dynamically updated to understand the environment better globally. Our method has sufficient potential for performance growth. With the improvement of robots, it does not require too many robots to find a reasonable number of robots in the audio goal at close range.

Continuous block search. Continuous block search is a close exploration mission to assess the agent's exploratory capabilities and proficiency in locating multiple blocks in a row. Diamond blocks are placed at every 16-block interval across the mainland map.

As shown in Table 2, we experiment with the block-searching task (Zhao et al., 2023) to assess the agent's exploratory capabilities and proficiency in locating specified blocks. Dynamic map is to identify as many blocks as possible within the fewest iterations, which indicates the method's efficiency. The dynamic map and the self-organization of the hierarchical structure allow more blocks to be found and deployed.

Map exploration. Map exploration aims to let the agent update the map as much as possible. We set up the same status awareness: when in an unreached area, status information prompts the agent in text. We set each step's maximum movement distance not to exceed 50 blocks.

Method	# agents	# iters (↓)	area (↑)
Voyager[(Wang et al., 2023a)]	1	6	175
	5	3	755
STEVE[(Zhao et al., 2023)]	1	6	161
	6	3	696
HAS	1	1	201
	8	**1**	**1368**

Table 3: **Comparison on map exploration task.** # iters represent the average number of iterations required to locate the area of 100 blocks; the lower this number, the higher the task completion efficiency. # area denotes the average squares of blocks over 5 iterations, with higher values indicating better performance. We list the one-agent and the best performance with the number of agents on the bottom line for each method according to # agents.

Setting	# agents	Goal Search		Block Search		Map Exploration	
		# iters (↓)	success rate (↑)	# iters (↓)	# blocks (↑)	# iters (↓)	area (↑)
w/o DM	1	53	0.46	14	67	6	160
	6 / 4 / 5	22	0.64	5	237	3	624
w/o AO	1	41	0.55	35	29	6	172
	5 / 5 / 5	15	0.78	11	106	3	706
HAS (Ours)	1	15	0.82	14	68	1	201
	6 / 8 / 8	**4**	**0.93**	**2**	**367**	**1**	**1368**

Table 4: **Ablation studies** for multi-modal goal search, continuous block search, and map exploration. The setting is the same as the above 3 experiments. Note that w/o DM is without the dynamic map, and w/o AO is without the auto-organizing mechanism.

As shown in Table 3, **HAS** has achieved the best performance, especially on multi-agent, which can better reflect the hierarchical structure. It is of considerable value for macro-control of multi-agent exploration and avoids agents repeatedly wandering in the explored area.

4.5 ABLATION STUDY

To understand the impact of different components on the performance of our system, we conducted ablation studies. The results, as shown in Table 4, provide insights into the effectiveness of the dynamic map and auto-organizing mechanism. Note that in w/o AO, we directly block the planner and deployer modules of the Manager to remove the auto-organizing mechanism and only retain the describer's perception of the environment, including dynamic maps into status information, thereby directly acting on the same target task on a fixed number of agents.

Dynamic map is critic. Dynamic map is a multi-modal information body including images and status information. It can integrate data in the simplest way to provide global understanding, reduce additional overhead and provide the possibility of macro layout for multi-person cooperation.

Comparison with GPT-4. Auto-organizing mechanism can dynamically adjust dynamic groups and automatically assign tasks based on the existing environment, allowing efficient multi-person cooperation.

5 CONCLUSION

In conclusion, **HAS** brings a significant advancement to multi-agent systems for complex environment navigation. Our contributions include a hierarchical auto-organizing system that ensures effective centralized planning and decentralized execution, an innovative auto-organizing and intra-communication mechanism for dynamic task adaptation, and a multi-modal information platform that integrates diverse sensory inputs. These advancements collectively enhance the autonomy, efficiency, and adaptability of AI agents, marking a pivotal step forward in the field of Embodied AI.

REFERENCES

Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models. *arXiv preprint*, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Artemij Amiranashvili, Nicolai Dorka, Wolfram Burgard, Vladlen Koltun, and Thomas Brox. Scaling imitation learning in minecraft. *arXiv preprint arXiv:2007.02701*, 2020.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. Large language models as tool makers. *arXiv preprint*, 2023.

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F Karlsson, Jie Fu, and Yemin Shi. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*, 2023a.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847, 2021.

Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *International Conference on Learning Representations*, 2018.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2023b.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–10, 2018.

Samyak Datta, Sameer Dharur, Vincent Cartillier, Ruta Desai, Mukul Khanna, Dhruv Batra, and Devi Parikh. Episodic memory question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19119–19128, 2022.

Victoria Dean, Shubham Tulsiani, and Abhinav Gupta. See, hear, explore: Curiosity via audio-visual association. *Advances in neural information processing systems*, 33:14961–14972, 2020.

Jie Deng, Wenhao Chai, Jianshu Guo, Qixuan Huang, Wenhao Hu, Jenq-Neng Hwang, and Gaoang Wang. Citygen: Infinite and controllable 3d city layout generation. *arXiv preprint arXiv:2312.01508*, 2023.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Heming Du, Xin Yu, and Liang Zheng. Vtnet: Visual transformer network for object goal navigation. In *International Conference on Learning Representations*, 2020.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35: 18343–18362, 2022.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. Multimodal-gpt: A vision and language model for dialogue with humans. *arXiv preprint arXiv:2305.04790*, 2023.

Rui Hao, Linmei Hu, Weijian Qi, Qingliu Wu, Yirui Zhang, and Liqiang Nie. Chatllm network: More brains, more intelligence. *arXiv preprint*, 2023.

Katja Hofmann. Minecraft as ai playground and laboratory. In *Proceedings of the annual symposium on computer-human interaction in play*, pp. 1–1, 2019.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.

Zican Hu, Zongzhang Zhang, Huaxiong Li, Chunlin Chen, Hongyu Ding, and Zhi Wang. Attention-guided contrastive role representations for multi-agent reinforcement learning. *arXiv preprint arXiv:2312.04819*, 2023.

Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. The malmo platform for artificial intelligence experimentation. In *Ijcai*, pp. 4246–4247, 2016.

Obin Kwon, Jeongho Park, and Songhwai Oh. Renderable neural radiance map for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9099–9108, 2023.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474, 2020.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023a.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023b.

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. Steve-1: A generative model for text-to-behavior in minecraft (abridged version). In *NeurIPS 2023 Workshop on Goal-Conditioned Reinforcement Learning*, 2023.

Zichuan Lin, Junyou Li, Jianing Shi, Deheng Ye, Qiang Fu, and Wei Yang. Juewu-mc: Playing minecraft with sample-efficient hierarchical reinforcement learning. *arXiv preprint arXiv:2112.04907*, 2021.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023a.

Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Denny Zhou, Andrew M Dai, Diyi Yang, and Soroush Vosoughi. Training socially aligned language models in simulated human society. *arXiv preprint arXiv:2305.16960*, 2023b.

Shuang Liu and Takayuki Okatani. Symmetry-aware neural architecture for embodied visual exploration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17221–17230. IEEE, 2022.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking via reinforcement learning. In *International conference on machine learning*, pp. 3286–3295. PMLR, 2018.

Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking and its real-world deployment via reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1317–1332, 2019.

Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv preprint arXiv:2306.09093*, 2023.

Weiyu Ma, Qirui Mi, Xue Yan, Yuqiao Wu, Runji Lin, Haifeng Zhang, and Jun Wang. Large language models play starcraft ii: Benchmarks and a chain of summarization approach. *arXiv preprint arXiv:2312.11865*, 2023.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. *arXiv preprint arXiv:2307.04738*, 2023.

Hangyu Mao, Zhengchao Zhang, Zhen Xiao, and Zhibo Gong. Modelling the dynamic joint policy of teammates with attention multi-agent ddpg. *arXiv preprint arXiv:1811.07029*, 2018.

Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. Learning multi-agent communication with double attentional deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 34:1–34, 2020a.

Hangyu Mao, Chao Wang, Xiaotian Hao, Yihuan Mao, Yiming Lu, Chengjie Wu, Jianye Hao, Dong Li, and Pingzhong Tang. Seihai: A sample-efficient hierarchical ai for the minerl competition. In *Distributed Artificial Intelligence: Third International Conference, DAI 2021, Shanghai, China, December 17–18, 2021, Proceedings 3*, pp. 38–51. Springer, 2022.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*, 2020b.

Abhinav Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, and Dhruv Batra. Soat: A scene-and object-aware transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34:7357–7367, 2021.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv: Arxiv-2303.08774*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.

PrismarineJS. Prismarinejs/mineflayer: Create minecraft bots with a powerful, stable, and high level javascript api., 2013. URL `https://github.com/PrismarineJS/mineflayer/tree/master`.

Chen Qian, Yufan Dang, Jiahao Li, Wei Liu, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Experiential co-learning of software-developing agents. *arXiv preprint arXiv:2312.17025*, 2023.

Jianzhun Shao, Zhiqiang Lou, Hongchang Zhang, Yuhang Jiang, Shuncheng He, and Xiangyang Ji. Self-organized group for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:5711–5723, 2022.

Alexey Skrynnik, Aleksey Staroverov, Ermek Aitygulov, Kirill Aksenov, Vasilii Davydov, and Aleksandr I Panov. Hierarchical deep q-network from imperfect demonstrations in minecraft. *Cognitive Systems Research*, 65:74–78, 2021.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. *arXiv preprint*, 2022.

Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*, 2023.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*, 2023b.

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023c.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint*, 2023d.

Zihao Wang, Shaofei Cai, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*, 2023e.

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations*, 2019.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L Berg, and Dhruv Batra. Multi-target embodied question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6309–6318, 2019.

Yinfeng Yu, Wenbing Huang, Fuchun Sun, Changan Chen, Yikai Wang, and Xiaohong Liu. Sound adversarial audio-visual navigation. In *International Conference on Learning Representations*, 2021.

Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. Plan4mc: Skill reinforcement learning and planning for open-world minecraft tasks. *arXiv preprint arXiv:2303.16563*, 2023.

Bin Zhang, Hangyu Mao, Jingqing Ruan, Ying Wen, Yang Li, Shao Zhang, Zhiwei Xu, Dapeng Li, Ziyue Li, Rui Zhao, et al. Controlling large language model-based agents for large-scale decision-making: An actor-critic approach. *arXiv preprint arXiv:2311.13884*, 2023a.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023b.

Zhonghan Zhao, Wenhao Chai, Xuan Wang, Li Boyi, Shengyu Hao, Shidong Cao, Tian Ye, Jenq-Neng Hwang, and Gaoang Wang. See and think: Embodied agent in virtual environment. *arXiv preprint arXiv:2311.15209*, 2023.

Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1467–1482, 2019.

Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Towards distraction-robust active visual tracking. In *International Conference on Machine Learning*, pp. 12782–12792. PMLR, 2021.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023a.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*, 2023b.

Mingchen Zhuge, Haozhe Liu, Francesco Faccio, Dylan R Ashley, Róbert Csordás, Anand Gopalakrishnan, Abdullah Hamdi, Hasan Abed Al Kader Hammoud, Vincent Herrmann, Kazuki Irie, et al. Mindstorms in natural language-based societies of mind. *arXiv preprint arXiv:2305.17066*, 2023.