UNIVERSITY OF CALIFORNIA

Los Angeles

Embodied Multi-Agent Systems in 3D Environments

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

Zhi Li

2024

ABSTRACT OF THE THESIS

Embodied Multi-Agent Systems in 3D Environments

by

Zhi Li
Master of Science in Computer Science
University of California, Los Angeles, 2024
Professor Demetri Terzopoulos, Chair

This thesis advances multi-agent (MA) systems designed to emulate human-like perception. It focuses on the use of first-person partial RGB visual perception alongside limited state observations such as tools, inventory, and body status, to support MA cooperation in a 3D, open-ended environment, specifically that in the game of Minecraft. Central to this exploration are two novel contributions: The first introduces MA-Minecraft, a comprehensive benchmark in the Minecraft environment that challenges agents with RGB image and language inputs in dynamic interaction scenarios. The second presents a novel control strategy, MA offline imitation learning via social Gradient Fields (MAGF), which leverages the same pixel-based partial vision inputs for precise, low-level action execution during MA interactions. Experimental results validate the efficacy of our approach by demonstrating the ability of trained agents to effectively collaborate in complex environments. Together, these contributions aim to enhance strategic decision-making and operational precision in partial observation settings, pushing the boundaries of how embodied agents perceive and interact in complex 3D environments.

The thesis of Zhi Li is approved.

Bolei Zhou

Kai-Wei Chang

Demetri Terzopoulos, Committee Chair

University of California, Los Angeles

2024

To my beloved parents, family and friends.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I extend my heartfelt gratitude to Professor Demetri Terzopoulos, whose exemplary guidance and mentorship have profoundly influenced my academic journey. His leadership in the field and incisive feedback have been instrumental in shaping both the approach and execution of my research.

I am deeply indebted to my colleague, Qian Long, whose invaluable guidance and steadfast support have been crucial from the outset of my exploration into multi-agent systems. His academic leadership and collaboration have been pivotal in the development and success of the research included in this thesis.

My sincere appreciation also goes to my colleagues at the Center for Vision, Cognition, Learning, and Autonomy (VCLA), especially Ran Gong and Xiaofeng Gao. Their expert advice was instrumental in the formulation of the benchmark paper, enriching the research with their perspectives.

I am thankful to Professor Ying Nian Wu for his supportive guidance and insightful contributions, which have enriched my study significantly.

Finally, I express my profound gratitude to my family and friends, whose unwavering support and encouragement have been my constant source of strength and inspiration throughout this journey.

VITA

2018–2022	B.A. Degree in Computer Science Boston University Boston, MA, USA
2019–2021	Grader, Department of Computer Science Boston University Boston, MA, USA
2021	Teaching Assistant, Department of Computer Science Boston University Boston, MA, USA
2022–2024	M.S. Degree in Computer Science University of California, Los Angeles Los Angeles, CA, USA
2022–2023	Reader, Computer Science Department University of California, Los Angeles Los Angeles, CA, USA
2023–2024	Teaching Assistant, Computer Science Department University of California, Los Angeles Los Angeles, CA, USA

CHAPTER 1

Introduction

Recent advancements in artificial intelligence have paved the way for the development of embodied agents capable of performing complex tasks in dynamic 3D environments. Crucial to further advancement is the challenge of creating agents that can perceive and interact with their surroundings in a human-like manner. This thesis focuses on exploring human-like perception within Multi-Agent (MA) systems, particularly through bounded, egocentric fields of view and limited state observations such as tools, inventory, and body status.

While research in vision-language task-planning for autonomous agents has primarily centered on navigation and object-based interactions, the collaborative potential of multiple agents in an open-ended world to enhance task efficiency remains largely untapped. Efforts to develop agents that can tackle a broad spectrum of tasks in intricate, embodied MA environments are still limited.

One avenue of prior research aims to establish MA methodologies within 2D environments solely using abstract vector inputs (Leibo et al., 2021a; Suarez et al., 2021; Long et al., 2020, 2024); however, such inputs lack realism and rich information. Another line of work focuses on developing agents across a diverse range of tasks in domains encompassing both gaming and robotics (Wang et al., 2023b,a; Ahn et al., 2022; Huang et al., 2022b,a); however, these studies address only single-agent scenarios. Considering the intricate interplay and uncertainties that arise within interactions among multiple agents, formulating multi-task agents within an MA setting is considerably more challenging and complex.

A primary contribution of this thesis is the development of MA-Minecraft, a new

benchmark built upon the Minecraft game, which provides a controlled yet challenging setting for evaluating the capabilities of MA systems. MA-Minecraft not only introduces tasks that require sophisticated interaction strategies among agents, but also includes diverse object and background imagery, demanding a high level of generalization from the agents.

Leveraging the MA-Minecraft dataset from the benchmark, which includes demonstrations encompassing first-perspective RGB image and state observation for agents, orchestrated by hand-designed planners, the thesis further investigates how State-Of-The-Art (SOTA) Vision-Language Models (VLMs) and abstract high-level action frameworks can process human-like percepts of complex 3D scenes and effectively direct agent behavior.

The exploration extends into a second study, in which we propose a novel Multi-Agent control via Social Gradient Field (MAGF) approach, based on the same human-like perception within MA systems, to develop autonomous agents capable of generalized control and broad applicability, mimicking the flexible and dynamic learning abilities of humans. Our MAGF method leverages social gradient fields to translate RGB partial observations directly into discrete low-level 3D navigation commands for individual agents, showcasing an advanced form of visual imitation learning and demonstrating the potential for agents to achieve generalized control and broad applicability, much like humans.

1.1 Contributions

More specifically, this thesis makes the following contributions:

1. To facilitate research on multi-agent (MA) systems, we introduce MA-Minecraft, a novel and challenging MA benchmark dataset based on the Minecraft game. Instead of the abstract vector inputs commonly provided to agents in MA systems research, MA-Minecraft agents receive RGB image and language inputs. Such multi-modal inputs pose a higher level of difficulty, since agents must generalize across diverse object and background imagery, different numbers of agents, a wide

range of tasks, etc. Our planner-generated dataset includes various tasks, such as building construction, ground clearing, and farming, with a total of 100,000 procedurally-generated demonstrations that feature 2 to 3 agents interacting with over 30 objects across 10 different backgrounds. Given only the three-view graph of the environment or target state, along with task descriptions and initial inventory settings, the agents must efficiently collaborate to complete the assigned tasks. We test the generalization abilities of several baseline Vision-Language Model (VLM) multi-agent control strategies in centralized and decentralized settings.

2. Drawing inspiration from the efficacy of representing MAS environments through social gradient fields, we propose a novel approach: Multi-Agent control via the social Gradient Field representation (MAGF). Our method begins by gathering low-level control demonstrations from planners in Minecraft, which constitutes our offline dataset. Subsequently, we employ an encoder to translate pixels into latent space, embedding the resultant output as a gradient field. Then, utilizing a refined version of imitation learning, we determine the ultimate actions of the agents. Experimental results validate the efficacy of our approach by demonstrating the ability of trained agents to effectively collaborate in complex environments. Comparative analyses against various baselines confirm the superiority of our method across all the evaluated tasks.

1.2 Thesis Overview

The remainder of this thesis is organized as follows:

Chapter 2 surveys the relevant literature related to multi-agent reinforcement learning, embodied agents in multi-agents systems, open-ended environments, and imitation learning.

Chapter 3 introduces our MA-Minecraft benchmark to evaluate agent abilities to process visual RGB and language inputs for multi-agents cooperative tasks.

Chapter 4 develops our MAGF method for solving low-level multi-agent navigation control with partial RGB image observations.

Chapter 5 presents our conclusions and suggests avenues for future work.

CHAPTER 2

Related Work

2.1 Environments for Multi-Agent Reinforcement Learning

The recent success of Multi-Agent Reinforcement Learning (MARL) methods (Lowe et al., 2020; Yu et al., 2021; Long et al., 2020) has garnered attention, as these methods explore cooperation and competence behaviors among agents. These methodologies have been developed and tested on prominent platforms. However, many of these platforms involve 2D environments (Leibo et al., 2021b; Suarez et al., 2021; Mordatch and Abbeel, 2017; Vinyals et al., 2019) and rely solely on vector observations. This limited scope poses challenges in terms of extending applicability to real-world scenarios.

2.2 Embodied Agents in Multi-Agent Systems

A cluster of works has taken shape within the multi-agent embodied setting, predominantly within the AI2-THOR environment (Kolve et al., 2022). Jain et al. (2019) delved into the communication dynamics that enhance collaboration between two agents. Tan et al. (2020a) and Liu et al. (2022a) propounded the efficient exploration of environments as a central task for agents. Meanwhile, Liu et al. (2022b) introduced a model that dynamically decomposes tasks among different agents, enabling dynamic task allocation. It is noteworthy, however, that the task propositions thus far have primarily revolved around navigation, inherently tethered to the environment's constraints. However, Minecraft is a multidimensional, visually immersive realm characterized by procedurally generated landscapes and extraordinarily versatile game mechanics. This supports an extensive

spectrum of activities, fostering an environment ripe for intricate collaborations and the emergence of competence.

2.3 Open-Ended Environments in Minecraft

In contrast to vision-based simulators primarily designed for single-agent scenarios that lack the complexity required for advanced AI research (e.g., (James et al., 2020), (Jiang et al., 2023)), or platforms like SMAC (Samvelyan et al., 2019) and various board game environments (Claus and Boutilier, 1998) typically offer only state representations and come with predefined task structures, Minecraft, facilitated by the Malmo platform (Johnson et al., 2016), provides a visually immersive and dynamic environment conducive to exploring intricate interactions among multiple agents, with a Gym-style API. This endeavor paved the way for subsequent developments, such as Minerl (Guss et al., 2019), which augmented the dataset and introduced a suite of benchmarking tasks. Despite this potential, few researchers have fully exploited Minecraft's capabilities for multi-agent scenarios and diverse visual stimuli. While MineDojo (Fan et al., 2022) and VPT (Baker et al., 2022) concentrate on single-agent tasks, leveraging YouTube videos for extensive pre-training within Minecraft enhances its applicability to developing agents adept at comprehending and autonomously navigating complex visual landscapes. Additionally, Gong et al. (2023) employ a Large Language Model (LLM) as a planner to generate high-level plans for multi-agent scenarios, albeit utilizing vector inputs in a centralized manner. To our knowledge, our study represents the first attempt at offline reinforcement learning in a multi-agent setting with visual inputs.

2.4 Imitation Learning

Imitation Learning offers an alternative to manually designing rewards for cultivating desired behaviors. Various approaches exist for utilizing demonstrations, ranging from Behavioral Cloning (Pomerleau, 1988), which focuses on directly replicating expert actions

within the agent's training policy, to Inverse Reinforcement Learning (Ziebart et al., 2008), which derives a reward function from demonstrations and subsequently trains a policy to optimize it. Nonetheless, these methods lack an explicit model of higher representation, leading to suboptimal performance in complex environments.

CHAPTER 3

MA-Minecraft: A Benchmark for Embodied Multi-Agent Systems

MA-Minecraft is tailored to MA embodied systems. It utilizes the acclaimed Minecraft game as our experimental platform. Our focus is directed toward unraveling the intricate dynamics of interactions among agents. Our framework encompasses the design of four multi-modal tasks: building construction, ground clearing, farming, and object acquisition. Within the cooperative tasks, each assignment necessitates consideration of fellow agents, spanning factors such as their spatial positioning, inventory holdings, skill differentials, and initial vitality. This nuanced assessment yields divergent role allocation and task strategies within the planning phase. The collaborative actions encompass resource sharing and joint pursuit, unfolding within the execution phase.

3.1 Features of MA-Minecraft

Our MA-Minecraft dataset encompasses fundamental skills and tasks, meticulously orchestrated by hand-designed planners. This planning approach concurrently serves as a benchmark ceiling across all tasks. Moreover, we introduce two alternative baseline models, both utilizing the provided dataset for training, thereby validating the efficacy of the generated data. The first model, MA-GPT4-o, employs a Large Language Model (LLM) as the planner, generating subgoals that guide an individual agent. The second model, MA-LLAVA, comprehensively encodes input facets and subsequently fuses embeddings through an attention mechanism, culminating in the prediction of ultimate actions. Our experimental findings demonstrate that both baseline models achieve competence across

Benchmark	RGB	Language	3D	Allocation	Multi-Agents	(De)centralized	Tool Use	Interaction	Generalization
Alfred (Shridhar et al., 2020)	✓	✓	Obs	✗	✗	✗	✓	✓	100,000+
DialFRED (Gao et al., 2022)	✓	✓	Obs	✓	✗	✗	✓	✓	53,000+
MultiagentEQ (Tan et al., 2020b)	✓	✓	Obs	✗	✓	✗	✓	✓	✗
EmbodiedMA (Liu et al., 2022b)	✓	✓	Obs	✓	✓	✓	✗	✗	✗
Cordial Sync (Jain et al., 2020)	✓	✓	Obs w/ Action	✓	✓	✓	✗	✓	✗
MineLand (Yu et al., 2024)	✗	✓	✓	✓	✓	✗	✓	✓	6,000+
MindAgent (Gong et al., 2023)	✗	✓	Obs	✓	✓	✗	✓	✓	100,000+
Creative Agents (Zhang et al., 2023)	✓	✓	✓	N/A	✗	N/A	✓	✓	✗
MineDojo (Fan et al., 2022)	✓	✓	✓	N/A	✗	N/A	✓	✓	1,000+
Overcooked-AI (Carroll et al., 2020)	✗	✗	Obs	✓	✓	N/A	✗	✓	✗
Watch&Help (Puig et al., 2021)	✗	✗	Obs	✓	✓	✗	✗	✓	✗
Too many cooks (Wang et al., 2020)	✗	✗	✗	✓	✓	✓	✗	✓	✗
SQA3D (Ma et al., 2023)	✓	✓	✓	✗	✗	N/A	✗	✗	40,000+
OpenEQA (Majumdar et al., 2024)	✓	✓	Obs	✗	✗	N/A	✗	✗	2,000+
AlexaArena (Gao et al., 2023)	✓	✓	Obs	✗	✗	N/A	✓	✓	✗
MA-Minecraft	✓	✓	✓	✓	✓	✓	✓	✓	100,000+

Table 3.1: Comparison of MA-Minecraft with other benchmarks. MA-Minecraft features RGB image and language inputs for multi-agents control with a large number of widely-varied demonstrations in Minecraft. *RGB:* Real-time first-person perspective RGB images are provided to agents and serve as observations. *Language:* Task goals are specified by human language instruction. *3D:* Task requires agents to have perception and be able to interact with the 3D world (i.e., movement in 3D, objects interacted with have 3D relations). Note: "Obs" denotes only support of 3D observation, no movement or action in 3D. *Allocation:* Multiple tasks must be dynamically allocated to multiple agents to obtain maximum benefit. Agents must use visual perception to understand other agents' states and make decisions to increase efficiency. *Multi-Agents:* This denotes that multiple agents can be present in a single experiment. *(De)centralized:* This denotes that agents can be operated separately in both centralized and decentralized settings. *Tool Use:* Completing tasks necessitates the use of specific tools by the agents, or using tools results in different task efficiencies. *Interaction:* Agents must manipulate or engage with different items or environmental elements or objects to achieve certain goals with irreversible actions. *Generalization*: Standardized generalization across a diversity of goals, objects, backgrounds, and inventories.

a subset of tasks.

Table 3.1 compares MA-Minecraft with other benchmarks. Existing benchmarks in MA systems research are founded on state or voxel-based observation in a controlled, closed environment. Actions and task types are also limited by the environments. MA-Minecraft advances the state-of-the-art in multi-agent benchmarks by exploiting the dynamic and open-ended Minecraft environment, offering:

1. first-person perspective high-quality RGB observations on top of the traditional voxel-based and state-based observations,

2. the ability to benchmark on existing MA cooperation tasks and define custom tasks with a variety of interactions,

3. the ability to control multiple agents in a open world to perform open-ended 3D tasks, in both centralized or decentralized settings,

4. the capacity to execute hundreds of actions individually for multiple agents that expand all possible task spaces with high-level, abstract language input, and

5. the ability to provide expansive visual diversity in tools, blocks, entities, and richly-detailed backgrounds.

3.1.1 Simulation Environment

MA-Minecraft utilizes Minecraft as its foundational environment, providing a complex, open-world setting for multi-agent interaction. Each agent is individually controlled via the Mineflayer interface, which provides high-level API functionalities for interaction with the environment. This setup is integrated into the MA-Minecraft and enables Gym-like interactions while supporting multiple agents. This allows for intricate command executions via both pre-written code and natural language instructions, facilitated by integration with prominent LLMs such as OpenAI GPT and LLaVA.

Figure 3.1 illustrates the MA-Minecraft architecture.

To cater to a variety of human users, our system supports interactive modalities ranging from keyboard and mouse inputs to VR headset controls, enhancing the immersive experience and dynamic interaction capabilities within the simulation environment.

3.1.2 Observation and Actions

MA-Minecraft captures a wide array of observational data to ensure agents have a comprehensive understanding of their environment:

RGB images: It provides real-time, high-resolution (up to 8K) image streams from first-person views for each agent and front, side, and top views of each object of interest.

Agent states: It provides detailed reporting on each agent's location, orientation, health status, inventory, and equipment.

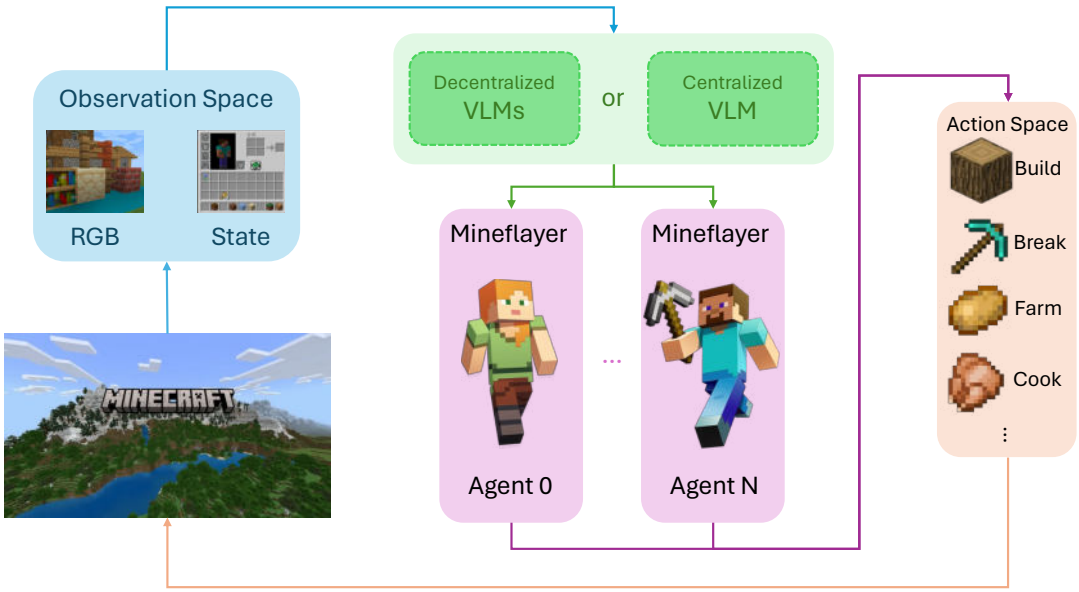

Figure 3.1: MA-Minecraft architecture

The action space in MA-Minecraft allows primitive motor skills such as "place" and "break". We provide over 12 actions that explore the full feature set of Minecraft, enabling complex and diverse agent interactions.

3.1.3 Centralized and Decentralized Agents

We have implemented two different settings for the agents: centralized agents and decentralized agents.

Centralized agents: These agents are given complete observational access to the environment, including the first person view, action history, and inventory information of all the agents. Based on this comprehensive data, the model generates the actions for all agents simultaneously. This approach leverages the full scope of information available in the environment to coordinate and optimize the actions of all agents collectively.

Decentralized agents: These agents do not receive information about other agents except for the initial environment settings, which may include some inventory details of other agents. The model generates actions solely for the individual agent based on its limited view. This setting simulates a more realistic scenario where agents operate

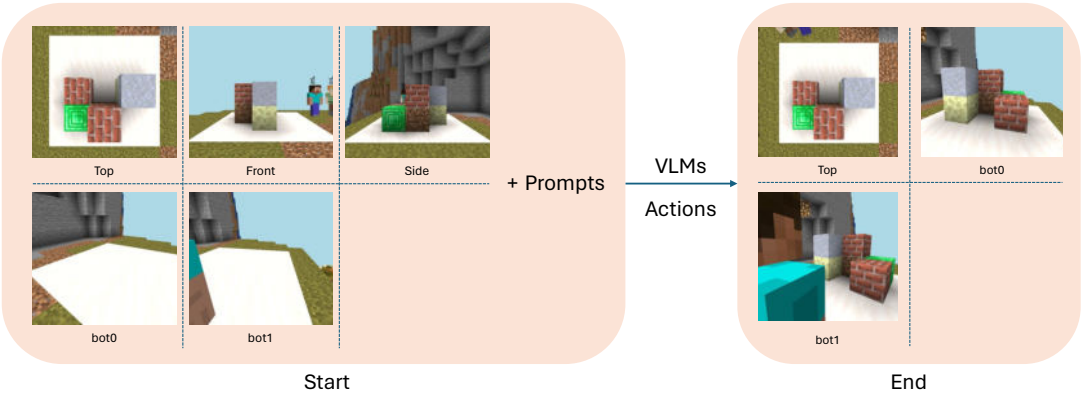

Figure 3.2: Building construction

independently with restricted information, focusing on their own observations and actions without centralized coordination.

3.1.4 Task Design

MA-Minecraft introduces a variety of complex and interactive tasks that challenge the agents' capabilities in planning, coordination, and execution within a collaborative and dynamic environment. Each task is designed to test different facets of MA interaction, including communication strategies, role distribution, real-time decision-making, and adaptability to changing environments. Tasks require capabilities in visual observation understanding, agent status intercepting, action capability understanding, language prompt understanding, continuous state understanding, and task action sequence planning. Here, we detail the specific tasks included in the MA-Minecraft benchmark:

Building construction: This task (Figure 3.2) requires agents to collaboratively erect a structure based on a provided three-view blueprint (front, side, and top). Each agent possesses a unique inventory of building blocks necessary for the construction. The task requires agents not only to understand their individual capabilities and inventories, but also to plan their movements and actions in coordination with other agents to efficiently construct the building on a designated 5×5 foundation.

Ground clearing: This task (Figure 3.3) challenges agents to remove all blocks

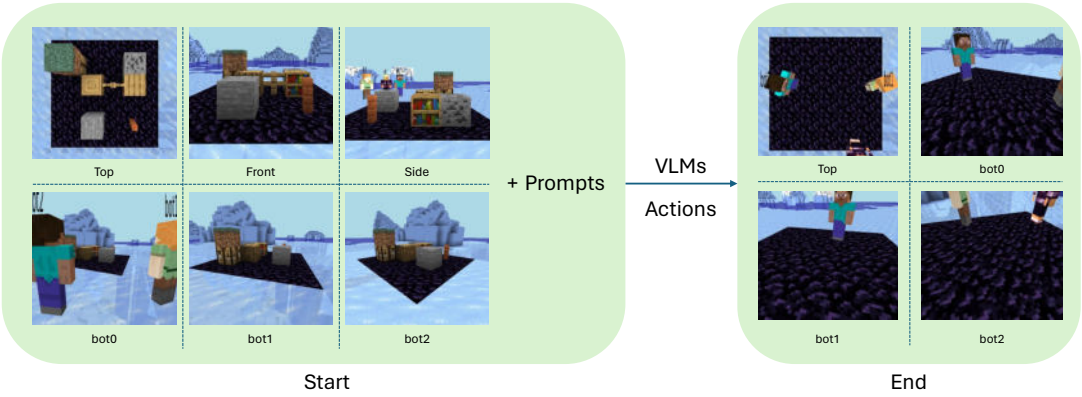

Figure 3.3: Ground clearing

from a specified 6×6 area. Agents must employ appropriate tools to break the blocks, which vary in durability, thereby requiring multiple interactions for complete removal. The use of correct tools can dramatically reduce the time required for block removal (up to $3\times$ speedup). Agents must manage their tool assignments to optimize block-breaking efficiency, so that the time steps needed for one task can be minimized. Strategic coordination is essential in this task as agents need to dynamically decide which blocks to target based on their current tools and help each other to minimize the overall time taken to clear the area.

Farming This task (Figure 3.4) is designed to simulate agricultural activities, where agents must sow and harvest crops. Agents are required to plant seeds on designated farmland plots and observe plantings until the crops reach maturity. Each crop has several growth stages from Level 0 (newly planted) to Level 7 (fully grown), and agents must identify when crops are ready to be harvested. The challenge lies in dynamically allocating tasks among agents based on their positions, available seeds, and the maturity of different crops. Effective task distribution and coordinated actions ensure maximum yield and efficiency. For example, some agents can sow while others are planning, and they should stop when their total crop yield is satisfactory.

Cooking This task requires the cooking of a meal. Agents are initially placed on separate islands, each containing different ingredients needed to prepare a meal. A central stove, accessible to all agents, acts as the communal cooking and item exchange point.

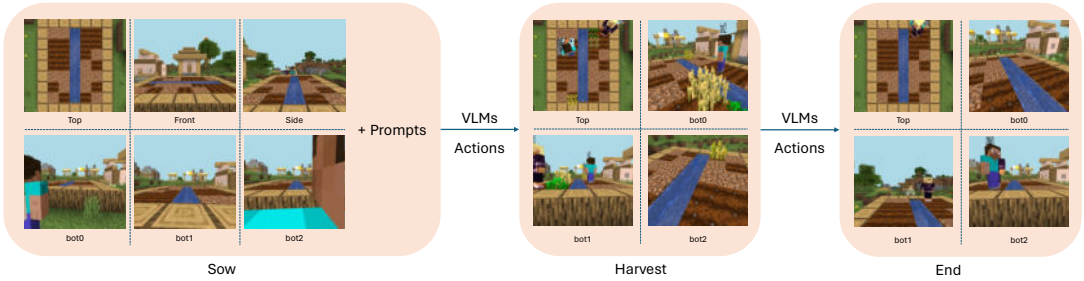

Figure 3.4: Farming

Figure 3.5: Diversity

The agents must collect and prepare ingredients using appropriate tools, which might be distributed among them initially or need to be exchanged midway through the task, depending on the resources within their reach. Agents must effectively communicate and strategically pass tools and ingredients to one another to assemble the required meal. This task tests the agents' ability to plan based on limited resources, cooperate subject to spatial constraints, and execute complex sequences of actions involving resource gathering, item management, and cooking processes. An example of this task could involve a series of actions from handing over a sword to an agent that has access to a chicken, handing over an axe to an agent that has access to a tree, and finally find and navigate to a stove to cook the chicken meal.

3.1.5 Diversity

The design of these tasks incorporates several layers of complexity to test and develop robust multi-agent systems capable of operating in diverse and unpredictable environments (Figure 3.5).

Object diversity: More than 30 objects are used for each task. Objects, such as a fence, an anvil, or a stone block, have different shapes and different textures, such as pink wool and dirty blocks. Farm crops will have different shapes in all 7 growth stages.

Inventory diversity: Each agent's inventory might include essential items mixed with non-essential ones, realistically simulating scenarios where agents must choose the right tools or materials for specific tasks while managing inventory constraints.

Background diversity: More than 10 backgrounds are included for all tasks, including village, mountain, forest, swamp, desert, etc.

Ground diversity: Tasks take place on grounds with diverse textures such as glass, concrete, and quartz. Certain tasks may involve additional complexity such as farmland intermixed with non-plantable blocks.

Goal diversity: Goals vary between tasks. For a place and construction task, we introduce different block placement shapes; e.g., a $2 \times 4 \times 2$ tower with top right intentionally not occupied. For a farming task, the total target corp type and counts are randomized. For the cooking task, the target cooked meal is randomized.

Agent count diversity: Agent counts vary from 2 agents to 4 agents, challenging the flexibility and adaptability of the coordination and task execution algorithms.

3.2 MA-Minecraft Dataset

To create a rich learning environment and effective training dataset for the MA-Minecraft tasks, systematic scenario design and data collection methods are employed:

Planner-based scenario design: Each task scenario is carefully crafted using ad-

vanced planning algorithms that consider all possible interactions within the environment. This includes optimal paths, resource distribution, and agent role assignments based on capabilities and task requirements.

Trajectory generation: Using Mineflayer interfaces controlled by heuristic methods such as the Hungarian Algorithm and dynamic programming, the planner orchestrates the agents to execute the task, ensuring that actions are taken optimally. Each step's effectiveness is assessed to guarantee efficient task completion.

Real-time interaction and feedback: Agents receive immediate feedback on their actions, which includes success, failure, and updates on environmental states. This real-time data is crucial for adjusting strategies and learning from interactions.

We leverage our planner to generate a large dataset of expert trajectories for learning. Our dataset includes 25K trajectories per task, resulting in 100K variants, and 90K successful trajectories in total. We hold aside a subset of variants, and the split is further shuffled and divided into Train / Validate / Test sets proportioned at 70% / 20% / 10%.

3.3 Experiments and Results

3.3.1 Baselines

The baseline methods consist of two main components: the GPT4-o method and the LLaVA method.

GPT4-o: For the GPT4-o method, we employ a one-shot learning approach. The prompt provided to the model includes a single successful demonstration of the task from the training set. Based on this example, we then ask the GPT4-o model to generate the actions for agents in response to new observations. This approach leverages the model's ability to generalize from a minimal amount of information.

LLaVA: We use the pretrained LLaVA-v1.6-Vicuna-7B model for our experiments. Under a centralized setting, we fine-tune this model using two different methods for feeding images. In the first method, we input four separate images into the model. We

refer to this model as LLaVA-M. In the second method, we concatenate these images into a single composite image before feeding it into the model and we refer to this model as LLaVA-S. This allows us to compare the effectiveness of different image input strategies on the model's performance.

3.3.2 Evaluation Metrics

We evaluate the performance of the methods based on two key metrics: task success rate and competence percentage.

Task success rate: The task success rate is determined by the ratio of the number of completed tasks to the total number of tested tasks. This metric indicates the proportion of test cases that the model can successfully complete from start to finish. A higher success rate reflects the model's ability to consistently achieve the desired outcomes in various scenarios.

Competence percentage: The competence percentage is calculated by dividing the accumulated rewards by the total number of tested tasks. This metric measures the overall effectiveness of the agents in performing the tasks, considering partial successes and the extent to which the tasks are completed. It provides a more granular view of the model's performance, highlighting how well the agents can handle different aspects of the tasks even if they don't fully complete them.

3.3.3 Evaluation Results

Due to limited computation resources, we trained the model only with the first 1/10 of the training dataset. The task success rates (Table 3.2) and competence percentages (Table 3.3) gleaned from our experiments provided significant insights into the performance capabilities of the two baseline models, MA-GPT4-o and MA-LLaVA, across different operational contexts.

The centralized settings generally yielded higher success rates and competence levels, indicating the advantage of having comprehensive environmental data available for decision-

Tasks	Condition	Centralized			Decentralized	
		LLaVA-M	LLaVA-S	GPT4-o	LLaVA	GPT4-o
Building	Seen	0.325	0.270	0.250	0.125	0.180
	Goal	0.240	0.330	0.210	0.205	0.280
	Object	0.155	0.180	0.165	0.140	0.080
	Agents	0.135	0.120	0.075	0.090	0.065
	Background	0.310	0.195	0.230	0.165	0.150
Clearing	Seen	0.350	0.260	0.245	0.180	0.150
	Goal	0.195	0.310	0.305	0.170	0.100
	Object	0.160	0.220	0.135	0.095	0.120
	Agents	0.105	0.190	0.080	0.115	0.055
	Background	0.225	0.200	0.175	0.150	0.125
Farming	Seen	0.195	0.285	0.260	0.150	0.170
	Goal	0.340	0.325	0.290	0.195	0.180
	Object	0.160	0.140	0.120	0.095	0.130
	Agents	0.120	0.070	0.085	0.065	0.090
	Background	0.210	0.300	0.270	0.220	0.155
Cooking	Seen	0.280	0.250	0.230	0.135	0.120
	Goal	0.265	0.345	0.315	0.200	0.190
	Object	0.140	0.165	0.155	0.110	0.100
	Agents	0.130	0.085	0.095	0.070	0.055
	Background	0.225	0.210	0.195	0.180	0.165

Table 3.2: Task success rates

making processes. The MA-LLaVA model, particularly the LLaVA-M variant, consistently outperformed others in most tasks under centralized scenarios. This model's ability to separately process multi-modal inputs likely contributed to its enhanced performance by allowing for finer granularity in perception and action planning.

Conversely, decentralized settings yielded a noticeable dip in performance metrics. Both LLaVA and GPT4-o decentralized configurations faced challenges, particularly in complex scenarios such as those requiring intricate coordination among agents or detailed environmental interactions. The reduced information flow inherent to decentralized settings evidently limited the models' ability to formulate and execute effectively cohesive strategies.

An interesting observation was the variance in model performance across different

Tasks	Condition	Centralized			Decentralized	
		LLaVA-M	LLaVA-S	GPT4-o	LLaVA	GPT4-o
Building	Seen	0.393	0.218	0.575	0.175	0.361
	Goal	0.546	0.234	0.433	0.190	0.320
	Object	0.339	0.219	0.246	0.160	0.192
	Agents	0.251	0.158	0.111	0.145	0.190
	Background	0.516	0.162	0.352	0.125	0.280
Clearing	Seen	0.642	0.412	0.590	0.305	0.470
	Goal	0.535	0.325	0.445	0.290	0.335
	Object	0.379	0.240	0.295	0.225	0.215
	Agents	0.288	0.175	0.140	0.160	0.120
	Background	0.467	0.285	0.360	0.245	0.275
Farming	Seen	0.520	0.308	0.485	0.290	0.365
	Goal	0.501	0.312	0.421	0.275	0.320
	Object	0.358	0.225	0.280	0.210	0.200
	Agents	0.270	0.160	0.191	0.150	0.200
	Background	0.449	0.275	0.350	0.230	0.260
Cooking	Seen	0.500	0.300	0.470	0.280	0.355
	Goal	0.489	0.305	0.410	0.265	0.310
	Object	0.345	0.220	0.265	0.205	0.195
	Agents	0.260	0.155	0.125	0.145	0.295
	Background	0.435	0.270	0.340	0.225	0.255

Table 3.3: Competence percentages

conditions within tasks. For instance, tasks labeled under "Seen" generally showed higher competence percentages, suggesting that the models were more adept at handling familiar scenarios where the environmental variables were within the expected parameters. However, under the "Agents" or "Background" conditions, where unpredictable elements influenced task dynamics, model performance declined. This indicates a potential area for improvement in enhancing model robustness and adaptability to unforeseen changes or less-controlled environments.

The competence percentages, in particular, highlighted not just the ability of models to complete tasks but also how effectively they could handle different task aspects. High competence in the "Goal" condition across several tasks emphasized the models' capabilities in strategic thinking and goal-oriented processing. Meanwhile, lower competence in

"Object" or "Agents" conditions pointed towards potential deficiencies in object recognition or agent-specific interaction strategies.

CHAPTER 4

MAGF: Multi-Agent Control via Social Gradient Fields

Our Multi-Agent offline visual reinforcement learning via social Gradient Field (MAGF) method yields low-level agent controllers in a MA setting, particularly within complex environments involving visual input. Selecting the Minecraft game as our testing environment, in this chapter we develop our method based on its rich game mechanics. Initially, we build our own dataset for the cooperative navigation task using a hand-crafted planner containing 45,000 demonstrations, which serves as our offline training dataset. We employ the social gradient field for visual embedding, which has been shown to provide a higher-level representation of the multi-agent environment. Comparing our methods with other baseline approaches, we find that our approach outperforms them on the tested tasks.

4.1 Technical Background

4.1.1 Learning Gradient Fields via Score-Matching

The score-based generative model aims to learn the *gradient field* of a log-data-density; *i.e.*, the *score function*. Given samples $\{\mathbf{x}_i\}_{i=1}^N$ from an unknown data distribution $\{\mathbf{x}_i \sim p_{\text{data}}(\mathbf{x})\}$, the goal is to learn a *score function* to approximate $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ via a *score network* $\mathbf{s}_\theta(\mathbf{x}) : \mathbb{R}^{|\mathcal{X}|} \to \mathbb{R}^{|\mathcal{X}|}$.

$$\mathcal{L}(\theta) = \frac{1}{2}\mathbb{E}_{p_{\text{data}}}\left[\|\mathbf{s}_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})\|_2^2\right]. \tag{4.1}$$

During the test phase, a new sample is generated by Markov Chain Monte Carlo (MCMC) sampling; *e.g.*, Langevin Dynamics (LD), which is beyond our interest since we focus on gradient field estimation.

However, the objective of score-matching in (Equation 4.1) is intractable, since $p_{\text{data}}(\mathbf{x})$ is unknown. The Denoising Score-Matching (DSM) (Vincent, 2011) proposes a tractable objective by pre-specifying a noise distribution $q_\sigma(\widetilde{\mathbf{x}}|\mathbf{x})$, and training a score network to denoise the perturbed data samples, where $q_\sigma(\widetilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\widetilde{\mathbf{x}}; \mathbf{x}, \sigma^2 I)$ is a Gaussian kernel with tractable gradient, in our case:

$$\begin{aligned}\mathcal{L}(\theta) &= \mathbb{E}_{\substack{\widetilde{\mathbf{x}}\sim q_\sigma(\widetilde{\mathbf{x}}|\mathbf{x}),\\ \mathbf{x}\sim p_{\text{data}}(\mathbf{x})}} \left[\|\mathbf{s}_\theta(\widetilde{\mathbf{x}}) - \nabla_{\widetilde{\mathbf{x}}} \log q_\sigma(\widetilde{\mathbf{x}}|\mathbf{x})\|_2^2\right] \\ &= \mathbb{E}_{\substack{\widetilde{\mathbf{x}}\sim q_\sigma(\widetilde{\mathbf{x}}|\mathbf{x}),\\ \mathbf{x}\sim p_{\text{data}}(\mathbf{x})}} \left[\left\|\mathbf{s}_\theta(\widetilde{\mathbf{x}}) - \frac{1}{\sigma^2}(\mathbf{x} - \widetilde{\mathbf{x}})\right\|_2^2\right].\end{aligned} \tag{4.2}$$

DSM guarantees that the optimal score network satisfies $\mathbf{s}^*_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ for almost all $\mathbf{x}$.

In practice, we adopt an extension of DSM (Song et al., 2020) that estimates a *time-dependent score network* $\mathbf{s}_\theta(\mathbf{x}, t) : \mathbb{R}^{|\mathcal{X}|} \times \mathbb{R}^1 \to \mathbb{R}^{|\mathcal{X}|}$ to denoise the perturbed data from different noise levels simultaneously:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(\epsilon,T)} \left\{ \mathbb{E}_{\substack{\widetilde{\mathbf{x}}\sim q_{\sigma(t)}(\widetilde{\mathbf{x}}|\mathbf{x}),\\ \mathbf{x}\sim p_{\text{data}}(\mathbf{x})}} \lambda(t) \left[\left\|\mathbf{s}_\theta(\widetilde{\mathbf{x}}, t) - \frac{1}{\sigma^2(t)}(\mathbf{x} - \widetilde{\mathbf{x}})\right\|_2^2\right] \right\}, \tag{4.3}$$

where T, ϵ, $\lambda(t) = \sigma^2(t)$, $\sigma(t) = \sigma_0^t$, and σ_0 are hyper-parameters. The optimal time-dependent score network satisfies $\mathbf{s}^*_\theta(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log q_{\sigma(t)}(\mathbf{x})$, where

$$q_{\sigma(t)}(\widetilde{\mathbf{x}}) = \int q_{\sigma(t)}(\widetilde{\mathbf{x}}|\mathbf{x}) p_{\text{data}}(\mathbf{x}) d\mathbf{x} \tag{4.4}$$

is the perturbed data distribution.

4.1.2 Image Tokenization With the CLIP Model

Learning direct control actions from raw RGB images via Reinforcement Learning (RL) exceeds the traditional design capabilities of RL systems. This is primarily because images contain vast amounts of abstract information unrelated to actionable controls. Image-text CLIP models, such as those described by Radford et al. (2021), offer a solution by tokenizing images into formats understandable by machines.

We utilize MineCLIP (Fan et al., 2022), a contrastive video-language model, which correlates video snippets with natural language descriptions. MineCLIP is inherently multi-task, trained on open-vocabulary and diverse English transcripts, thus aligning closely with the multi-faceted nature of real-world tasks. The model architecture mirrors that of CLIP4Clip (Luo et al., 2021), employing a similar dual-encoder setup: a text encoder ϕ_G for embedding language goals and a video encoder ϕ_V for processing sequences of 16 frames at 160×256 resolution. Specifically, ϕ_G reutilizes the pretrained text encoder from OpenAI's CLIP, while ϕ_V is divided into a frame-wise image encoder ϕ_I and a temporal aggregator ϕ_a. The aggregator condenses the image feature sequence into a single video embedding. Unlike CLIP4Clip, MineCLIP enhances its feature representation by adding two layers of residual CLIP Adapter (Gao et al., 2021) subsequent to ϕ_a and fine-tuning only the last layers of ϕ_I and ϕ_G.

Our adaptation of MineCLIP omits its final layer, which traditionally computes the CLIP reward based on the similarity between frame observations and skill descriptions. We repurpose the backbone of MineCLIP to tokenize images within the Minecraft domain, effectively bridging these visual inputs with the social gradient field for enhanced comprehension by the RL algorithm. This method not only streamlines the tokenization of complex visual information but also tailors the learning process to the specific semantics and dynamics of the environment, thus promising a more intuitive integration of visual data into RL decision-making processes.

4.2 Problem Definition

From a multi-agent dataset $D(I, V, A)$ containing images, vectors of coordinates, and actions of agents, we aim to develop a method to generate a decentralized policy $\pi(I)$ that performs tasks using only visual inputs.

The primary challenge arises from the pixel-based input, which, unlike vector inputs, contains a vast array of information with pronounced dynamism. Therefore, acquiring a large, high-quality dataset becomes imperative. It is crucial to sift through these data to distill the most salient information from the visual inputs.

Another significant challenge emerges in the multi-agent and partial observation setting. Agents must navigate their environment and interact with other agents, introducing a plethora of environmental diversities. Given that this task operates under a first-person view observation, agents can only perceive views within their limited, forward-looking visual fields. However, in the realm of multi-agent reinforcement learning, the strategy of centralized training and decentralized execution has proven critical for enhancing performance. It would also be beneficial to consider this approach in behavior cloning for multi-agent tasks. It remains a challenge to effectively utilize the fully observable data from the dataset to boost policy performance in the decentralized setting.

4.3 Methods

In this chapter, we elucidate the structural framework of our approach. Our method is detailed as Algorithm 1 and the entire pipeline is illustrated in Figure 4.1. The methodology unfolds in three distinct phases. First, we generate a dataset D_{gf} for the gradient field function from the original dataset D. This D_{gf} will later be used to train the *gf* function, which is then utilized to construct the gradient field ground truth from visual observations. Second, we train a visual input embedding network f_{gf} that processes the raw pixels and outputs the *gf* function. Finally, we apply f_{gf} in imitation learning. We will next delve into the specifics of each phase of our algorithm.

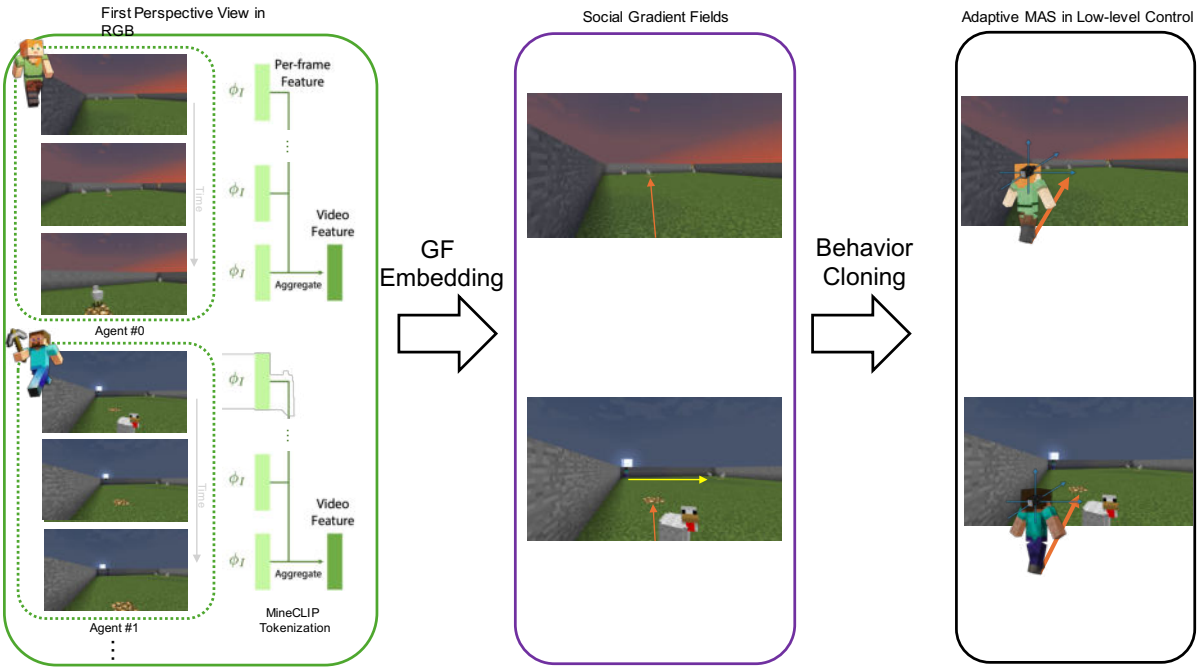

Figure 4.1: Pipeline for multi-agent control via *GF* embeddings. We first embed the visual inputs as Gradient Fields *GF*, which we then use as inputs for further behavior cloning.

4.3.1 Training the *gf* Functions

Inspired by the work of Long et al. (2024), we employ the GF as a superior representation of visual input within a multi-agent context. The GF offers several advantages: It drastically reduces the dimensionality of inputs and is goal-related, guiding agents at a higher level. This addresses the challenge of high implicit input in vision tasks. Additionally, the GF embedding fully utilizes global information, specifically relative coordinates, from the dataset.

Algorithm 1: Imitation learning using SocialGF

Generate gf from datasets by applying (4.1);
for gf_i *in* GF **do**
    Training gf_i representation from pixels for each type i via supervised learning;
`// Keep the vision embedding network fixed, we do the imitation learning with the gf embedding`
for $t = 1, \ldots,$ *max-iterations* **do**
    $O_{GF} \leftarrow \{gf_0, gf_1, \ldots, gf_n\}$; `// GFs Embedding`
    Do the imitation learning;

We must first construct the example set for the training of the *gf* function from the vector ground truth. We propose two kinds of *gf*: Environment gf_E and Agents gf_A. The gf_E will be used to capture the knowledge of the environmental constraints, and the gf_A will model the entities' dynamics and relations.

4.3.1.1 Environment *gf*

This will be considered as the single agent *gf* since it is related only to the environment. Vectors of other entities' information will not be considered. It can be denoted as $gf_E = \phi(v_{i,i})$, where $v_{i,i}$ represents the self-coordinates of agent i. Every data point from the dataset D can be treated as examples of legal vectors. We employ (4.2) in the training of gf_E.

4.3.1.2 Agent *gf*

This models the relations of different entities within the environment. gf_A will lead the agents toward task competence. To construct the example set, we extract entity coordinates from instances where success flags are raised during demonstrations, thereby creating our sets of successful examples. Employing score matching functions, per (4.4), we build the *GF* functions ϕ based on the relative coordinates: $gf_A = \phi(v_{i,j})$, where $v_{i,j}$ represents the relative coordinates of Agent i. This serves as the foundation of our training data for the *GF* embedding network.

After training gf_E and gf_A, the *gf* generated will be applied to the vector V of the original dataset $D(I, V, A)$ to obtain the ground truth $GF = (gf_E(V), gf_A(V))$. We form the (I, GF) pairs as the training dataset for the subsequent training of the *GF* embedding network.

4.3.2 Training the Pixel Embedding Network

Subsequently, we train the functions f_{gf}^E and f_{gf}^A using the (I, GF) pairs from the previous phase. This pixel embedding function takes in raw pixels and outputs the *GF* of the

environment and agents, aiming to minimize the losses $L_E = L_2(f_{gf}^E(x), gf_E(V))$ and $L_A = L_2(f_{gf}^A(x), gf_A(V))$, where x denotes the raw pixels. This network aims to extract the gradient information from the partially observable pixels. It leverages the concept of centralized training and decentralized operation. The *gf* extracted from the global information guides the training of the pixel embedding network, specifically leading to the target states.

For these two pixel embedding networks, we concatenate their outputs to form the extracted information from the images. This high-level *GF* representation is then fed into a final fully-connected neural network for action execution.

4.3.3 Training the MAGF

Finally, we integrate everything for imitation learning on the dataset D. The policy network is defined as $\pi(x) = h_i(f_{gf}^A(x), f_{gf}^E(x))$. Here, h_i is a two-layer fully connected network that inputs the concatenation of the *GF* embeddings and outputs the final action. We minimize the loss $L = L_2(\pi(x), A)$ using imitation learning methods. This is an end-to-end training process. The f_{gf}^A and f_{gf}^E functions trained in the previous phases are used as model initialization. Their final parameters will be fixed as training progresses.

The MAGF training pipeline is illustrated in Figure 4.2.

4.4 Experiments and Results

Our experimental setup utilizes the MineRL environment, which interfaces with Minecraft in a manner consistent with the OpenAI Gym framework. Each timestep within this environment provides a first-person RGB image observation for each agent, which serves as input to our policy. This setup then advances to the next timestep using the policy's output—low-level control actions that mimic real-world human interactions, including gaze yaw and pitch changes, movements, and other actions. MineRL also enables the customization of reward functions, calculated based on the environmental state to assess

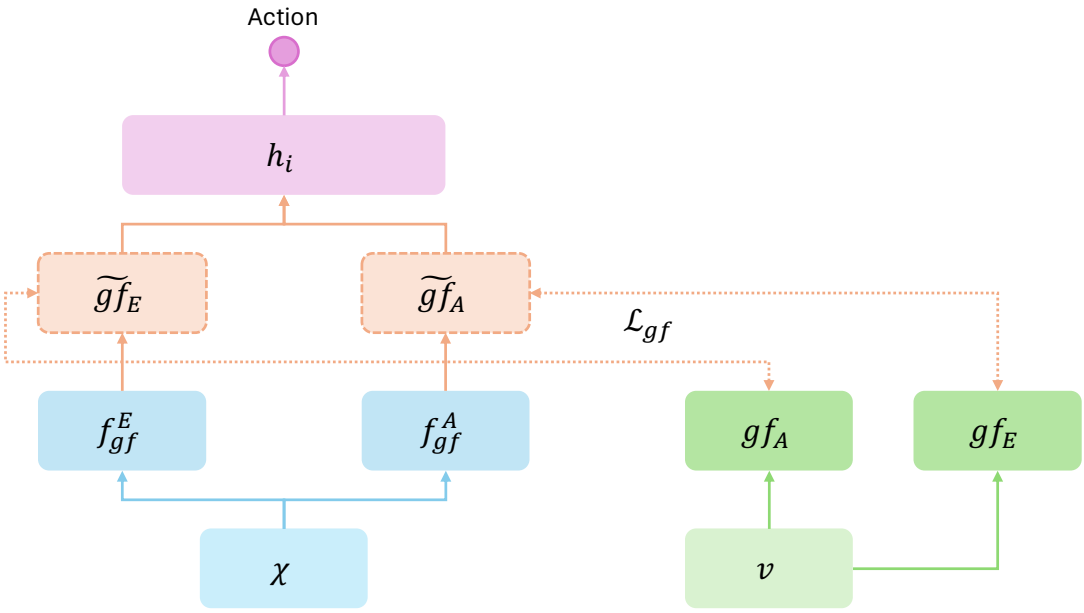

Figure 4.2: Pipeline for MAGF training

agent behavior. Notably, in our offline RL approach, the reward does not influence the policy's action output directly; rather, it serves solely as a performance evaluation metric.

4.4.1 Task Design

Our experimental setup encompasses three distinct cooperative multi-agent navigation tasks within Minecraft environments: two-dimensional navigation, three-dimensional navigation, and two-dimensional navigation with obstacles. Each task is designed to challenge the agents in different aspects of spatial awareness, strategic planning, and cooperative dynamics. We show samples of all three cooperative navigation tasks from the datasets in Figures Figure 4.3, Figure 4.4, and Figure 4.5.

4.4.1.1 2D Navigation

We implement a 2D navigation-based task within a confined environment surrounded by walls in Minecraft. Two to four agents spawn at the corners of the environment. The objective centers around two to four landmarks randomly placed within the walled area, which act as goals for the agents. Each agent's task is twofold: 1) occupy a unique

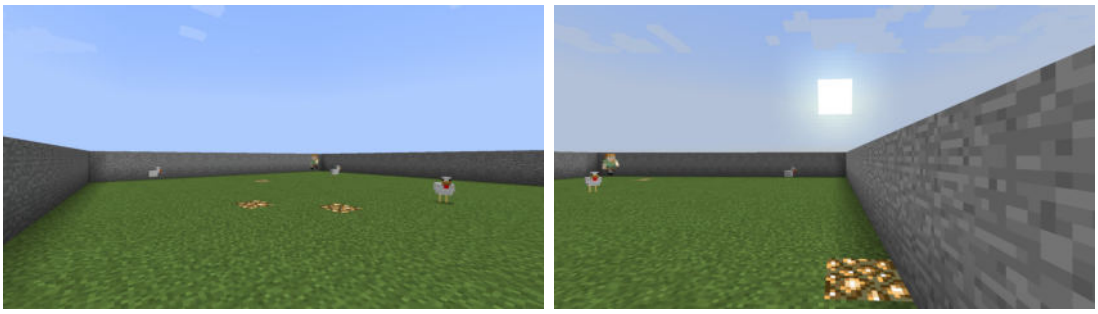

Figure 4.3: 2D Navigation task sample from the dataset

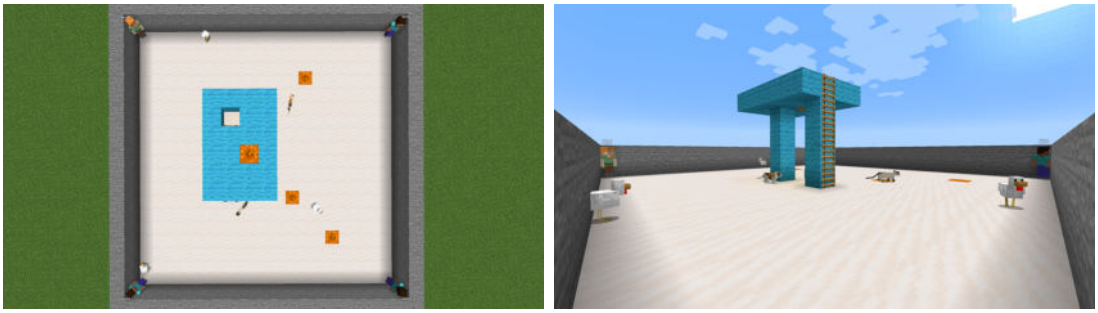

Figure 4.4: 3D Navigation task sample from the dataset

landmarks without sharing it with another agent, and 2) achieve this positioning with the fewest possible steps. Therefore, all agents must cooperative according to their position and their fellow agent's position to find a optimal path and target. The reward for this task is computed based on the number of landmarks occupied exclusively by only one agent, with a maximum score of 4 in a scenario involving four agents.

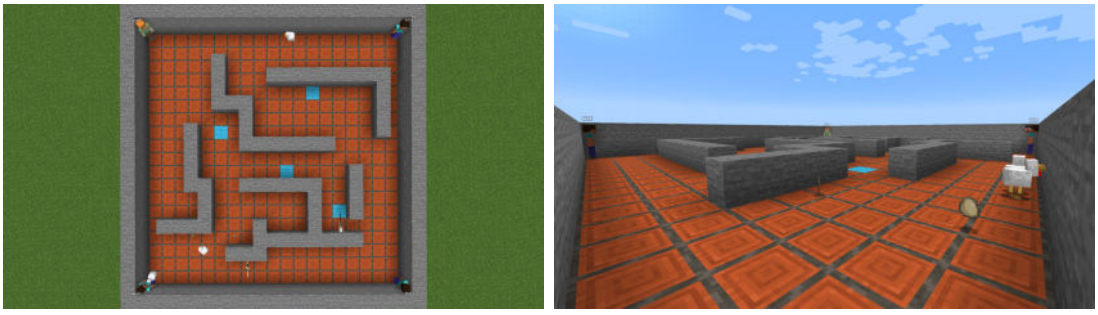

Figure 4.5: 2D Navigation with obstacles task sample from the dataset

4.4.1.2 3D Navigation

Building on the 2D navigation setup, the 3D navigation task introduces an additional vertical dimension by placing landmarks on elevated platforms. These platforms are accessible via multiple staircases or ladders, requiring agents to navigate both horizontally and vertically. While landmarks are visible from ground level, reaching them necessitates strategic vertical movement, thereby testing the agents' abilities to negotiate three-dimensional spaces and solve complex spatial puzzles. The task completion criterion demands that an agent not only reach a platform but also precisely position itself on the landmark.

4.4.1.3 2D Navigation with obstacles

This variation complicates the 2D navigation framework by integrating visual obstacles into the environment. The layout mimics a maze with short walls that obstruct direct lines of sight to the landmarks, effectively hiding them from the agents' initial positions. However, agents are able to see other agents over these obstructions, allowing for indirect informational cues to influence their strategies. Agents are required to explore the environment to locate landmarks or deduce their positions by observing the actions of their peers, such as stopping movements near targets. This task emphasizes not only physical navigation but also the importance of inferential reasoning and strategic information gathering.

4.4.1.4 Generalization

To enhance the complexity and assess the robustness of the agents' learning and adaptability, we introduce generalizations that modify the task environment:

- *Agent Counts* are randomly chosen from 2 to 4 to participate.
- *Initial Conditions* including starting position and initial orientation are randomized within a small area around the corners for each agent.

- *Landmark Diversity* involving different types of targets or landmarks (e.g., glowstone, gold block, stone block) varying in visibility and contrast against the background. The exact positions of landmarks within the environment are also randomized in each trial.

- *Ground and Wall Material* varies from grass to sand to stone for the ground and from wood to stone to brick for wall materials.

- *Dynamic Obstacles* introduces random non-player entities such as chickens and sheep, adding additional obstacles or interactions. The number of these entities is also varied.

- *Environmental Conditions* are set during different times of day—daylight or night. Nighttime settings decrease visibility, altering the agents' ability to detect and navigate toward the landmarks.

4.4.2 Data Construction

Data are captured by recording each agent's interactions in the Minecraft on MineRL, with each agent's frame rate set at approximately 24 frames per second. All RGB images are captured at a resolution of 1,920 $\times$ 1,080. To generate a diverse and challenging dataset, we employ a planner that is responsible for the following key tasks:

1. *Scene Variation Generation:* The planner initiates each episode by setting up the environment with variations as discussed in the task design section, including randomizing agent start positions and orientations, changing the types of landmarks and ground textures, adjusting environmental conditions (day or night), and varying the number and type of dynamic obstacles. These variations are integrated at the world/environment initialization phase, which are decided by the planner at the beginning of each session.

2. *Optimized Path Planning/Target Decision:* The planner utilizes a linear sum assignment algorithm to distribute targets among agents optimally, aiming to minimize

the total number of steps required for all agents to reach their respective targets.

3. *Optimized Action Control:* The planner employs voxel-based observations to determine each agent's current and target positions within the 3D space. It incorporates special awareness of the environment to navigate effectively towards the target. Actions are designed to facilitate movement toward the target by combining forward, backward, left, and right movements, along with quantized camera yaw adjustments. The camera yaw is discretized into 10-degree increments to ensure smooth and realistic turns without sudden changes.

4. *Reward Calculation:* Rewards are calculated by the planner based on general environmental feedback, including the positions of entities, if there is a correct occupation of each landmark by an agent within a specified range, calculated using the Euclidean distance between the agent and the landmark.

The Dataset includes total of 45,000 offline demonstration variants for all three tasks. Each demonstration contains 1) a JSON file with backgrounds/environment configuration, and action, reward, voxel observations for each agent in every time step, and 2) a set of RGB images from each agent's first-person perspective in every corresponding time step.

4.4.3 Baselines

In our experiments, we compared the following approaches:

1. *Vanilla Imitation Learning:* This method uses an adaptive horizon prediction module to learn goal-conditioned policies (Cai et al., 2023) on image-action pairs. Each agent's first-person view image is processed through the MineClip model to tokenize it into image features, which are then utilized to produce corresponding low-level controls for each agent.

2. *Vector Representation Imitation learning:* This method introduces a middle representation layer to encode the relative coordinates from image features. Similar to

vanilla imitation learning, where RGB pixels and image features encapsulate all the information necessary for action decisions, this approach first transforms the image features into a single vector. This vector represents the relative position from the agent to all the surrounding entities, and derives corresponding low-level actions.

3. *MAGF:* This is our proposed method which uses the social gradient field as an intermediate representation layer to better capture the multi-agent settings.

4.4.4 Evaluation Method

We evaluate our agent-learning approach on two combined dimensions: 1) task specified rewards and 2) the total time (or total environment steps) to accomplish the goal, as follows:

1. *Rewards within Time Limit:* This metric quantifies the success rate of task completion within a predefined time limit and task completion rate. It serves as an indicator of an agent's ability to efficiently execute tasks under temporal constraints.

2. *Time Score:* The time score is calculated by the completion time. It reflects the temporal duration taken by the agent to accomplish a given task. This metric offers insights into the efficiency of an agent's decision-making and execution processes.

4.4.5 Results

We tested on all three cooperative tasks using 2 to 4 agents, and the number of landmarks is the same as the agent. Each task has over 2,000 variants and during the training contains at least one unseen element, such as block texture or dynamic obstacle. We scaled the rewards within the time limit to [0,1], and time score from [0,100]. For each environment, we trained three groups of policies using each method. The training lasted for a total of 150 epochs. We then tested on 100 tracks per environment.

The results are shown in Table 4.1. From the table, we can see that our MAGF dominates most of the scales. For adaptive ability, the MAGF reward drops as the agent

Table 4.1: MAGF Performance

Group	Tasks	**Ours** (MAGF)	Vector Representation	Imitation Learning
2 Agents	2D	$\mathbf{0.8854 \pm 0.0587}$	0.5552 ± 0.1573	0.4897 ± 0.1166
	3D	0.6714 ± 0.1121	0.1334 ± 0.0754	$\mathbf{0.6821 \pm 0.1598}$
	2D w/ obstacles	$\mathbf{0.7183 \pm 0.0918}$	0.6008 ± 0.0249	0.6654 ± 0.0584
3 Agents	2D	$\mathbf{0.7581 \pm 0.0446}$	0.4417 ± 0.0547	0.3875 ± 0.0468
	3D	$\mathbf{0.5987 \pm 0.0395}$	0.1319 ± 0.0954	0.5527 ± 0.1546
	2D w/ obstacles	0.6865 ± 0.0636	$\mathbf{0.7102 \pm 0.1118}$	0.4623 ± 0.0875
4 Agents	2D	$\mathbf{0.7953 \pm 0.0408}$	0.3745 ± 0.115	0.2955 ± 0.0451
	3D	$\mathbf{0.6518 \pm 0.0456}$	0.1521 ± 0.1085	0.4589 ± 0.1537
	2D w/ obstacles	$\mathbf{0.6916 \pm 0.059}$	0.5259 ± 0.1333	0.4319 ± 0.0879

count scales, only with small variants. By comparison, the vector representation and imitation learning rewards dropped sharply as they tended to consider agents as three individuals, but failed to understand the cooperative strategy. In the 2-agent setting, imitation learning slightly outperformed MAGF for the 3D task, but it has a higher variance.

Utilizing the time score, we also computed cross match rewards within the time limit. This evaluates how efficiently the task is done; higher scores indicate that the task can be completed within a smaller number of time steps, which implies higher efficiency. The results are shown in Figure 4.6. The blue bars indicate the MAGF time score. It once again outperformed in most of the tasks. These results suggest that agents trained with MAGF have adapted successfully in the vision-based 3D environment and that MAGF is an effective representation of multi-agent collaboration information with only partial visual observations.

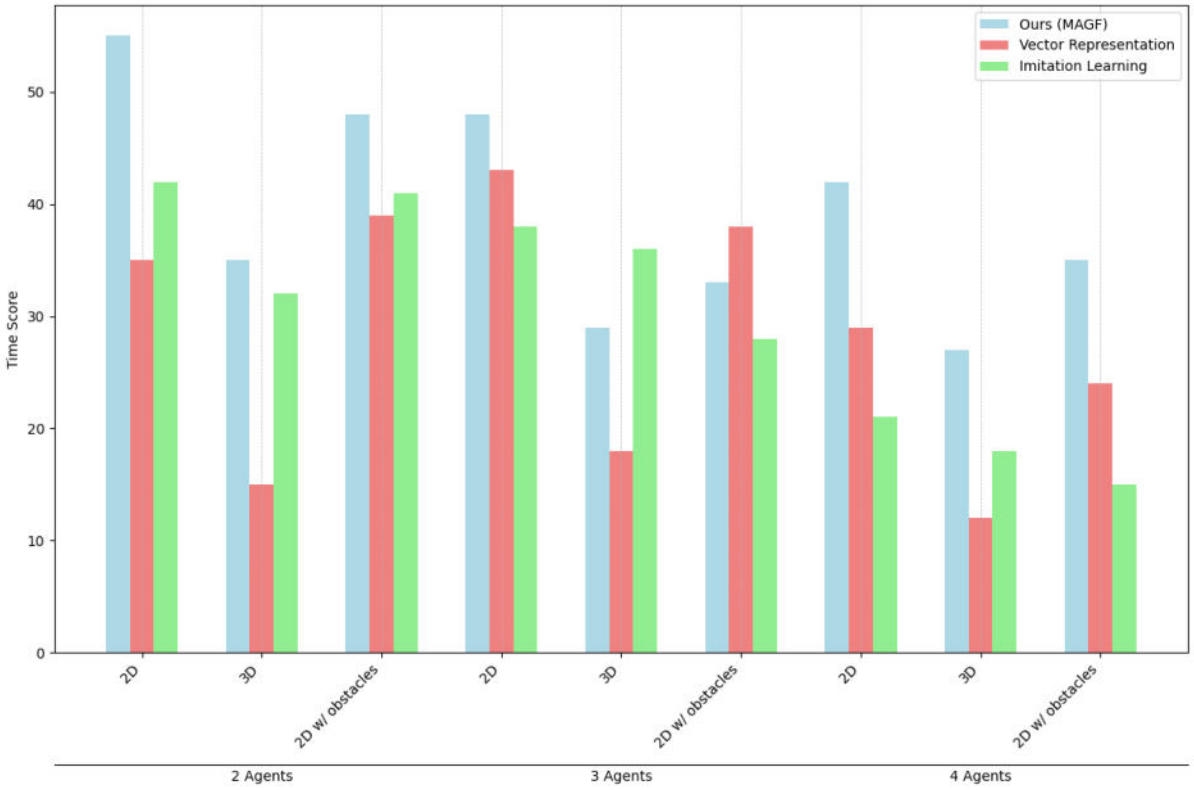

Figure 4.6: MAGF Time score results

CHAPTER 5

Conclusions

5.1 Summary

The findings of this thesis have significantly advanced our understanding of multi-agent dynamics in environments that simulate human-like perception. By incorporating a diverse array of tasks, ranging from construction to farming, to cooking, coupled with dynamic interactions among agents and objects, MA-Minecraft challenges the conventional paradigms of multi-agent research and paves the way for new explorations in embodied intelligence. The implementation of RGB image and language inputs as opposed to traditional abstract vector inputs has enabled a more realistic simulation of human-like perception and interaction. This setup has effectively demonstrated the necessity and impact of high-level strategic planning and real-time decision-making in a controlled yet challenging environment.

Our experimental results highlight the strengths and limitations of current Vision-Language Models (VLMs) in managing complex and dynamic task environments. While the centralized models exhibited robust performance across most tasks, reflecting their ability to leverage comprehensive environmental data for decision-making, the decentralized models underscored the challenges faced when agents operate with limited information. This dichotomy not only enriches our understanding of agent interaction dynamics but also underscores the critical role of information accessibility in strategic multi-agent environments.

Furthermore, the introduction of the MAGF method marks a substantial progression in the field of visual imitation learning, demonstrating that direct mapping of RGB

visual inputs to agent actions can result in highly effective navigation and task execution in multi-agent scenarios. This approach not only enhances the capability of agents to perform in visually complex and unpredictable environments, but also solidifies the role of social dynamics in the strategic planning and execution of tasks.

Overall, this thesis has contributed to the broader goal of developing intelligent agents that can perform a wide array of tasks with efficiency and precision comparable to human capabilities. The methodologies and insights yielded by this research provide a valuable framework for future explorations into embodied intelligence and open new avenues for the practical deployment of autonomous systems in real-world applications.

5.2 Future Work

The findings from this study underscore the crucial impact of information accessibility on the effectiveness of multi-agent systems in complex task environments. Centralized models benefit significantly from comprehensive data, facilitating better strategic planning and execution. By contrast, the challenges observed in decentralized setups mimic real-world scenarios where agents must often operate with incomplete knowledge, highlighting the need for developing more sophisticated information synthesis and decision-making mechanisms.

Future research should focus on enhancing the adaptability of decentralized models by incorporating methods for dynamic information sharing that maintain the decentralized nature of operations. Additionally, refining the models' ability to process diverse and unpredictable environmental inputs is crucial for improving task performance under varied conditions. The impact of the scale of the dataset on model performance should also be investigated to understand how data volume and variety affect learning outcomes. Exploration of advanced models such as GPT-4V, and LLAMA2, LLAMA3, or fine-tuned GPT-3.5 powered MA-LLAVA, could further our understanding of complex multi-agent interactions. Extending the use of images from single snapshots to time-series data could provide richer context and improve decision-making processes for each agent. Moreover,

enhancing prompts with detailed information such as tool efficiency, specific task settings, or advanced deduction logic like Chain-of-Thought(Wei et al., 2023) could refine the models' operational capabilities.

Future studies could also consider expanding the action space within the MAGF framework to include a broader range of activities beyond navigation, such as fighting, placing items, or using tools. This expansion would allow the exploration of more complex task dynamics and interactions. Introducing more elaborate cooperative, competitive, or mixed tasks could further challenge the current methodologies and encourage the development of sophisticated agent strategies.

Looking forward, the MA-Minecraft benchmark and MAGF can serve as a foundational platform for future research that aims to enhance the perceptual and cognitive faculties of autonomous agents. Such a platform would provide an excellent testbed for evaluating agent models. Potential research directions include refining the integration of sensory modalities, improving the robustness and adaptability of agents in decentralized settings, and exploring the implications of advanced agent cooperation strategies under varying environmental complexities. Widespread adoption of the platform and further development to broaden its application across various domains and environments promises to catalyze significant advancements in the field of multi-agent systems that improve the abilities of AI agents to navigate and manipulate complex, unpredictable environments. This could lead to innovative applications of multi-agent systems in real-world scenarios, enhancing the robustness and effectiveness of autonomous agents, ultimately enabling embodied AI agents to perceive, think, and act in a manner akin to humans. Furthermore, such research could be extended to facilitate human-robot collaboration, enabling seamless interactions via natural language.

REFERENCES

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M., and Zeng, A. (2022). Do as i can, not as i say: Grounding language in robotic affordances. 1

Baker, B., Akkaya, I., Zhokhov, 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. *arXiv preprint arXiv: Arxiv-2206.11795*. 6

Cai, S., Wang, Z., Ma, X., Liu, A., and Liang, Y. (2023). Open-world multi-task control through goal-aware representation learning and adaptive horizon prediction. *arXiv preprint arXiv: Arxiv-2301.10034*. 32

Carroll, M., Shah, R., Ho, M. K., Griffiths, T. L., Seshia, S. A., Abbeel, P., and Dragan, A. (2020). On the utility of learning about humans for human-AI coordination. 9

Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI Conference on Artificial Intelligence*. 6

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. *arXiv preprint arXiv: Arxiv-2206.08853*. 6, 9, 23

Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., and Qiao, Y. (2021). CLIP-Adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv: Arxiv-2110.04544*. 23

Gao, Q., Thattai, G., Gao, X., Shakiah, S., Pansare, S., Sharma, V., Sukhatme, G., Shi, H., Yang, B., Zheng, D., et al. (2023). Alexa arena: A user-centric interactive platform for embodied AI. *arXiv preprint arXiv:2303.01586*. 9

Gao, X., Gao, Q., Gong, R., Lin, K., Thattai, G., and Sukhatme, G. S. (2022). DialFRED: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics and Automation Letters*, 7(4):10049–10056. 9

Gong, R., Huang, Q., Ma, X., Vo, H., Durante, Z., Noda, Y., Zheng, Z., Zhu, S.-C., Terzopoulos, D., Fei-Fei, L., and Gao, J. (2023). MindAgent: Emergent gaming interaction. 6, 9

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. In Kraus, S., editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial*

Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, pages 2442–2448. ijcai.org. 6

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. (2022a). Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S., editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 9118–9147. PMLR. 1

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Brown, N., Jackson, T., Luu, L., Levine, S., Hausman, K., and Ichter, B. (2022b). Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv: Arxiv-2207.05608*. 1

Jain, U., Weihs, L., Kolve, E., Farhadi, A., Lazebnik, S., Kembhavi, A., and Schwing, A. (2020). A cordial sync: Going beyond marginal policies for multi-agent embodied tasks. 9

Jain, U., Weihs, L., Kolve, E., Rastegari, M., Lazebnik, S., Farhadi, A., Schwing, A., and Kembhavi, A. (2019). Two body problem: Collaborative visual task completion. 5

James, S., Ma, Z., Arrojo, D. R., and Davison, A. J. (2020). RLBench: The robot learning benchmark and learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026. 6

Jiang, Y., Gupta, A., Zhang, Z., Wang, G., Dou, Y., Chen, Y., Fei-Fei, L., Anandkumar, A., Zhu, Y., and Fan, L. (2023). VIMA: General robot manipulation with multimodal prompts. 6

Johnson, M., Hofmann, K., Hutton, T., and Bignell, D. (2016). The Malmo platform for artificial intelligence experimentation. In Kambhampati, S., editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4246–4247. IJCAI/AAAI Press. 6

Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., Kembhavi, A., Gupta, A., and Farhadi, A. (2022). AI2-THOR: An interactive 3D environment for visual AI. 5

Leibo, J. Z., Duéñez-Guzmán, E., Vezhnevets, A. S., Agapiou, J. P., Sunehag, P., Koster, R., Matyas, J., Beattie, C., Mordatch, I., and Graepel, T. (2021a). Scalable evaluation of multi-agent reinforcement learning with melting pot. 1

Leibo, J. Z., Duéñez-Guzmán, E., Vezhnevets, A. S., Agapiou, J. P., Sunehag, P., Koster, R., Matyas, J., Beattie, C., Mordatch, I., and Graepel, T. (2021b). Scalable evaluation of multi-agent reinforcement learning with melting pot. 5

Liu, X., Guo, D., Liu, H., and Sun, F. (2022a). Multi-agent embodied visual semantic navigation with scene prior knowledge. *IEEE Robotics and Automation Letters*, 7(2):3154–3161. 5

Liu, X., Li, X., Guo, D., Tan, S., Liu, H., and Sun, F. (2022b). Embodied multi-agent task planning from ambiguous instruction. *Proceedings of Robotics: Science and Systems, New York City, NY, USA*, pages 1–14. 5, 9

Long, Q., Zhong, F., Wu, M., Wang, Y., and Zhu, S.-C. (2024). SocialGFs: Learning social gradient fields for multi-agent reinforcement learning. 1, 25

Long, Q., Zhou, Z., Gupta, A., Fang, F., Wu, Y., and Wang, X. (2020). Evolutionary population curriculum for scaling multi-agent reinforcement learning. 1, 5

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2020). Multi-agent actor-critic for mixed cooperative-competitive environments. 5

Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., and Li, T. (2021). CLIP4Clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv: Arxiv-2104.08860*. 23

Ma, X., Yong, S., Zheng, Z., Li, Q., Liang, Y., Zhu, S.-C., and Huang, S. (2023). SQA3D: Situated question answering in 3d scenes. 9

Majumdar, A., Ajay, A., Zhang, X., Putta, P., Yenamandra, S., Henaff, M., Silwal, S., Mcvay, P., Maksymets, O., Arnaud, S., et al. (2024). OpenEQA: Embodied question answering in the era of foundation models. In *2nd Workshop on Mobile Manipulation and Embodied Intelligence at ICRA 2024*. 9

Mordatch, I. and Abbeel, P. (2017). Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*. 5

Pomerleau, D. A. (1988). Alvinn: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems*, 1. 6

Puig, X., Shu, T., Li, S., Wang, Z., Liao, Y.-H., Tenenbaum, J. B., Fidler, S., and Torralba, A. (2021). Watch-And-Help: A challenge for social perception and human-ai collaboration. 9

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR. 23

Samvelyan, M., Rashid, T., Schroeder de Witt, C., Farquhar, G., Nardelli, N., Rudner, T. G., Hung, C.-M., Torr, P. H., Foerster, J., and Whiteson, S. (2019). The StarCraft multi-agent challenge. *International Conference on Autonomous Agents and Multi-Agent Systems*. 6

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. (2020). ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10737–10746. IEEE. 9

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*. 22

Suarez, J., Du, Y., Zhu, C., Mordatch, I., and Isola, P. (2021). The neural mmo platform for massively multiagent research. In Vanschoren, J. and Yeung, S., editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. 1, 5

Tan, S., Xiang, W., Liu, H., Guo, D., and Sun, F. (2020a). Multi-agent embodied question answering in interactive environments. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 663–678. Springer. 5

Tan, S., Xiang, W., Liu, H., Guo, D., and Sun, F. (2020b). Multi-agent embodied question answering in interactive environments. In *European Conference on Computer Vision*. 9

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674. 22

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354. 5

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. 1

Wang, R. E., Wu, S. A., Evans, J. A., Tenenbaum, J. B., Parkes, D. C., and Kleiman-Weiner, M. (2020). Too many cooks: Bayesian inference for coordinating multi-agent collaboration. 9

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. 1

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models. 38

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., and Wu, Y. (2021). The surprising effectiveness of MAPPO in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*. 5

Yu, X., Fu, J., Deng, R., and Han, W. (2024). MineLand: Simulating large-scale multi-agent interactions with limited multimodal senses and physical needs. 9

Zhang, C., Cai, P., Fu, Y., Yuan, H., and Lu, Z. (2023). Creative agents: Empowering agents with imagination for creative tasks. 9

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. (2008). Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA. 7