StarDojo: Benchmarking Open-Ended Behaviors of Agentic Multimodal LLMs in Production–Living Simulations with Stardew Valley

Weihao Tan[1⋆], Changjiu Jiang[2⋆], Yu Duan[1⋆], Mingcong Lei[2], Jiageng Li[1], Yitian Hong[3], Xinrun Wang[4⋆⋆], and Bo An[1]

[1] Nanyang Technological University, Singapore
[2] Harbin Institute of Technology, Shenzhen, China
[3] The Chinese University of Hong Kong, Shenzhen, China
[4] East China University of Science and Technology, China
[5] Singapore Management University, Singapore
weihao001@ntu.edu.sg
https://weihaotan.github.io/StarDojo

Abstract. Autonomous agents navigating human society must master both production activities and social interactions, yet existing benchmarks rarely evaluate these skills simultaneously. To bridge this gap, we introduce StarDojo, a novel benchmark based on Stardew Valley, designed to assess AI agents in open-ended production–living simulations. In StarDojo, agents are tasked to perform essential livelihood activities such as farming and crafting, while simultaneously engaging in social interactions to establish relationships within a vibrant community. StarDojo features 1,000 meticulously curated tasks across five key domains: farming, crafting, exploration, combat, and social interactions. Additionally, we provide a compact subset of 100 representative tasks for efficient model evaluation. The benchmark offers a unified, user-friendly interface that eliminates the need for keyboard and mouse control, supports all major operating systems, and enables the parallel execution of multiple environment instances, making it particularly well-suited for evaluating the most capable foundation agents, powered by multimodal large language models (MLLMs). Extensive evaluations of state-of-the-art MLLMs agents demonstrate substantial limitations, with the best-performing model, GPT-4.1, achieving only a 12.7% success rate, primarily due to challenges in visual understanding, multimodal reasoning and low-level manipulation. As a user-friendly environment and benchmark, StarDojo aims to facilitate further research towards robust, open-ended agents in complex production-living environments.

Keywords: Interactive Benchmark · Multimodal LLMs · Open-Ended Agents · Production–Living Simulation

⋆ Equal contribution
⋆⋆ Corresponding author

1 Introduction

The complexity of human society is embodied in "**production-living systems**" [9], where individuals simultaneously engage in productive activities (e.g., farming, crafting, resource generation) and living activities (e.g., social interaction, relationship building, cultural participation). These activities are not independent processes but mutually reinforcing and constraining components of daily life. Productive outputs provide the material foundation that enables social participation, while social relationships, community engagement, and economic exchanges directly influence access to resources, knowledge, and opportunities for production. Time, energy, and environmental dynamics further bind these processes together, forcing individuals to continuously balance resource generation, social commitments, and long-term planning under shared constraints.

While recent models have demonstrated strong performance on well-defined tasks such as question answering [1], code synthesis [18], and mathematical problem solving [37], developing human-level agentic multimodal LLMs (MLLMs) capable of operating within interactive production-living systems poses a fundamentally different challenge. Traditional interactive benchmarks [5, 8, 13, 38] largely emphasize either skill execution and strategic gameplay, where textual information plays only a minimal role, and socially situated interactions are either absent or highly simplified. Other social deduction benchmarks [2, 19, 25] focus on multi-agent reasoning communication, without structured production activities or resource-generation mechanisms. As a result, social interactions in these environments are largely decoupled from material constraints and long-term developmental trade-offs. As a result, the field still lacks a systematic assessment of an agent's ability to operate within the coupled, constraint-bound structures that characterize human life.

To bridge the gap, we introduce StarDojo, a novel environment and benchmark consisting of 1,000 comprehensive tasks based on the production-living dynamics of the popular simulation game, Stardew Valley. As illustrated in Figure 1, agents undertake tasks such as clearing farmland, tilling soil, planting and watering crops, harvesting produce, raising animals, mining and foraging for resources, and crafting essential tools and equipment.

Additionally, agents must explore diverse maps, collect various items, combat monsters, trade with merchants to generate income, upgrade and expand their farms, and complete numerous in-game quests. Social progress involves participating in festivals and community events, building interpersonal relationships, and may eventually lead to marriage and raising children. Beyond these, agents must explore diverse maps, collect various items, combat monsters, trade with merchants to generate income, upgrade and expand their farms, and complete numerous in-game quests.

Crucially, production and social progress form a bidirectional feedback loop. On one hand, social progression can unlock new recipes, discounts, services, and collaboration opportunities that directly boost production efficiency (e.g., gaining access to advanced crafting recipes or other production, enhancing resources through relationships and community participation). On the other hand,

Fig. 1: StarDojo leverages the open-world richness of Stardew Valley to provide a diverse array of scenarios and activities. Agents are required not only to engage in a wide range of production activities, such as farming, crafting, mining, logging, and animal husbandry, but also to participate in various social events, including trading, conversing with NPCs, and even starting families. In addition, agents need to adapt to dynamic changes in time and weather, thereby reflecting the multifaceted nature of real-world living and social interaction.

production enables social advancement by generating gifts and valuable items, providing income and materials for participating in festivals and community events, and supporting longer-term milestones such as marriage and raising children. Social progress involves engaging in festivals and community events, building interpersonal relationships, and may eventually lead to marriage and family life, each of which can further reshape production possibilities through newly available items, rewards, and opportunities.

The environment realistically simulates time, stamina, weather patterns, and seasonal cycles, all of which significantly impact both production tasks and social interactions, presenting further challenges for agents. To facilitate development and evaluation for researchers, we also present StarDojo-Lite, a curated subset comprising 100 core tasks that focus on the essential skills typically encountered during the game's early stages.

To facilitate the development and evaluation of agent behaviors, StarDojo offers four key features: 1) **Unified User-friendly Interface**: Provides an intuitive Python interface to interact seamlessly with Stardew Valley's game

engine implemented in C#, simplifying interaction and internal state capture, eliminating the need for manual screenshots and keyboard/mouse inputs to obtain observation and action. 2) **Automated Evaluation**. Includes comprehensive evaluation scripts for all tasks, ensuring reliable and reproducible agent performance assessments. 3) **System Compatibility**. Supports major operating systems (Ubuntu, macOS, and Windows) to ensure wide accessibility. 4) **Parallelized Environments**. Allows multiple headless environment instances to run concurrently, significantly enhancing efficiency for evaluation and data collection.

Through extensive evaluations, we demonstrate that tasks within StarDojo present significant challenges even for agents with state-of-the-art MLLMs. Our assessments cover cutting-edge models, including GPT-4.1 (& mini) [23], Claude-3.7 Sonnet [3], Gemini 2.5 Pro [10], and the open-source Llama 4 Maverick [21], Qwen2.5-VL-72B [4] and Gemma 3 27B [35]. These agents achieve performance ranging from 4% to 12.7% on the StarDojo-Lite task suite. While agents successfully complete some easy-level tasks requiring fewer than 30 steps, they exhibit near-zero success rates on medium and hard tasks demanding extended action sequences. We conducted a detailed error analysis and attributed these failures to four primary categories: visual understanding, multimodal reasoning, long-term planning, and low-level control. We release StarDojo as an open-source environment and benchmark, providing setup instructions, robust evaluation scripts, comprehensive documentation, and baseline implementations. We hope it can facilitate research into robust, open-ended decision-making agents capable of lifelong learning and long-term planning in production-living systems.

Table 1: Comparison of StarDojo with representative existing environments. The columns indicate whether the environment supports open-ended interaction and continuous learning, evaluates the ability to perform long-term planning, reflects scenarios relevant to real-world daily life, includes production activities (e.g., farming, hunting, and crafting), simulates human society with social interactions, implements a realistic economy supporting production and social activities, and provides language-based APIs for interaction with LLMs.

Environment	Open-Endness	Long-Term Planning	Routine Planning	Production Activities	Social Interaction	Economy System	Language APIs
Atari [5]	×	×	×	×	×	×	×
VirtualHome [27]	×	×	√	×	×	×	×
SMAC [30]	×	×	×	×	×	×	×
Crafter [13]	√	×	×	√	×	×	×
MineDojo [8]	√	√	×	√	×	×	√
OSWorld [44]	×	√	√	×	×	×	√
TextWorld [7]	×	×	√	×	×	×	√
AvalonBench [19]	×	√	×	×	√	×	√
Smallville [25]	√	√	√	×	√	×	√
CivRealm [29]	√	√	×	√	√	√	√
StarDojo	√	√	√	√	√	√	√

2 Related Work

Interactive Benchmarks. The development of robust benchmarks for evaluating decision-making agents across various scenarios has been a critical focus in AI research. Traditional reinforcement learning (RL) benchmarks [5, 11, 30, 36] predominantly emphasize low-level control tasks and non-realistic environments. More realistic simulators [6, 16, 17, 27, 28], usually focus on embodied tasks in household scenarios, lacking significant environmental dynamics and diverse social activities. Recent advancements in generative agents have enabled large language models (LLMs) to simulate human social behaviors in interactive environments [2, 7, 19, 25]. Their limited action spaces, primarily restricted to dialogue, hinder engagement in broader, real-world-inspired tasks that involve production, consumption, and resource management. On the other side, while GUI benchmarks [15, 32, 44, 49] also provide interactive environments, they focus on short-term software manipulations. The most relevant work is MineDojo [8], which offers a diverse range of tasks within the open-ended environment of Minecraft. However, its gameplay primarily focuses on interactions with nature, with limited opportunities for human-like social interactions. Additionally, its complex 3D navigation controls pose significant challenges, particularly for LLM-based agents to complete even simple tasks, further limiting its suitability for more complex activities. Another relevant benchmark, CivRealm [29], evaluates agents' strategic decision-making at a country level within a turn-based, Civilization-like game. While CivRealm presents a variety of tasks, including managing population, production, and economy, its scope remains at a macro-strategic level, distinct from StarDojo's focus on granular, individual-level decision-making. Additionally, recent works [24, 48] tend to evaluate MLLMs on traditionally RL environments in simple game scenarios, lacking semantic richness and social interaction.

As shown in Table 1, StarDojo addresses the limitations of prior benchmarks by providing a comprehensive, open-ended evaluation platform for decision-making agents in a dynamic environment. Although StarDojo, like most of the aforementioned benchmarks, adopts an abstract visual style, it effectively reconstructs and abstracts real-world events and social dynamics in a structured and operational manner. Its unique combination of complexity, realism, and diversity makes it a valuable testbed for advancing agents in real-world-like environments.

MLLMs Agents. Traditional reinforcement learning (RL) agents [12, 20, 22, 31] primarily focus on low-level control and fail to leverage natural language understanding, making them unsuitable for tasks requiring complex reasoning, long-term planning, and social interactions. Recent advancements in LLM-based agents have significantly expanded AI capabilities by integrating reasoning mechanisms such as chain-of-thought (CoT) prompting [45] and reflection [33]. Modular frameworks and multi-agent architectures have enabled LLM-based agents to achieve remarkable performance in tasks like code generation [14, 41, 42] and GUI manipulation [40, 43, 46, 47]. Additionally, Voyager [39] has demonstrated strong in-context lifelong learning abilities, showcasing exceptional proficiency in the open-ended world of Minecraft. However, Voyager's strong reliance on

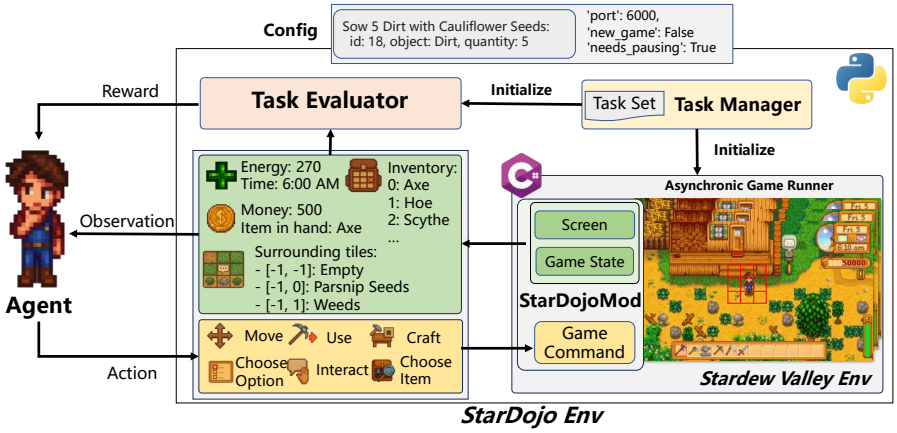

Fig. 2: StarDojo environment is initiated by configurable task files. It communicates with parallel game engines through StarDojoMod to obtain internal game states and execute commands, which will be encapsulated as observations and actions by the Python Wrapper.

built-in APIs makes it challenging to adapt to other games. Cradle [34] successfully completes meaningful tasks across multiple commercial video games and software applications with a unified interface without the need to access APIs. Its preliminary experiments in Stardew Valley highlight the limitations of current state-of-the-art agents, particularly in handling multi-modal understanding and long-term planning, which reveals the importance of extending Stardew Valley as a well-developed benchmark for decision-making agents.

3 StarDojo

3.1 Introduction to Stardew Valley

Stardew Valley is a globally popular open-ended simulation RPG game where players inherit a run-down farm. Players must thoughtfully manage their farming strategies, explore the surrounding village, build meaningful relationships with villagers, and gather diverse resources to revitalize the farm.

Realistic Dynamics. The game incorporates a realistic cycle, featuring days that run from 6 AM to 2 AM, a daily energy system, and four 28-day seasons with changing weather, all of which influence both production and social activities. Success depends on effective time and energy management, as well as the ability to quickly adapt to dynamics.

Rich Production Activities. As the core gameplay of Stardew Valley, players engage in a wide range of productive activities, including clearing land, cultivating crops, raising animals, mining, and foraging. The fruits of their labor, such as harvested crops, animal products, and crafted goods, can be sold for

income, reinvested to expand and improve the farm, or given to villagers to strengthen friendships and complete quests or community objectives. Through these interconnected systems, players gradually enhance both the sustainability of their farm and their overall quality of life within the game world.

Diverse Social Interaction. The game also features social interaction with 45 unique non-player characters (NPCs), each with their own personality and routines. Players can build friendships and may even date, marry, and raise children with villagers. Improving friendships not only enriches the narrative experience but also facilitates production, as villagers may send useful gifts, share recipes, or offer assistance in various activities. Rich town festivals and quests provide further opportunities for community engagement and resource gathering.

Comprehensive Economic System. Agents must engage in strategic resource management, investment, and efficient planning to generate income from both production and social activities while adapting to seasonal demands and market conditions to ensure long-term financial success.

Overall, Stardew Valley serves as an ideal environment for decision-making agents in the production–living simulation. Its well-integrated systems of time management, resource allocation, economic planning, and social interaction provide a dynamic and complex environment that requires strategic thinking and adaptability. The game's structured yet open-ended nature makes it an excellent testbed for evaluating decision-making capabilities in simulated real-world conditions.

3.2 Architecture

As a typical commercial video game, Stardew Valley only supports human-like interaction, e.g., observing gameplay through screenshots and using keyboard and mouse to control. Additionally, the game window must remain active and in the foreground, significantly restricting automated gameplay and preventing simultaneous execution of multiple instances. As shown in Figure 2, to overcome these limitations, we introduce the carefully designed StarDojo environment, enabling efficient interaction and comprehensive evaluation of agents.

Unified User-friendly Interface. We present StarDojoMod, a novel extension built upon the Stardew Modding API (SMAPI) [26], which is a widely adopted, open-source modding framework designed specifically for Stardew Valley. SMAPI offers developers extensive APIs that expose key game events and internal states, facilitating the creation of interactive and sophisticated mods. Based on SMAPI, StarDojoMod provides structured and efficient interactions between agents and the game environment. It communicates in real-time with the Stardew Valley game engine through a socket server, granting agents direct access to rendered gameplay images, saving the time-consuming screen captures, internal game states (such as character positions, statuses, and environmental information), and enabling diverse callable functions as action skills beyond traditional keyboard and mouse inputs. Moreover, we implemented a configurable pause-and-resume mechanism by directly modifying the inner states of the game, allowing the game to pause during model inference and agent planning, and resume before

action execution. Inherited from SMAPI, StarDojoMod is implemented in C# to be consistent with the game engine. To enhance ease-of-use and accessibility of the environment, we provide a user-friendly Python Wrapper based on the StarDojoMod for observation retrieval, action execution, and task customization, empowering users to engage with the StarDojo environment effortlessly.

System Compatibility. Stardew Valley is one of the few games that can be played on all mainstream operating systems (Linux, macOS and Windows). We also ensured the compatibility of StarDojoMod and the Python Wrapper, enabling the entire environment to run seamlessly across different systems.

Parallel Execution. Our architecture is designed for scalability and parallel execution. Each instance of Stardew Valley is independently managed through unique ports, enabling simultaneous control of multiple game instances without interference. Communication efficiency is further enhanced through the use of shared memory, reducing observation retrieval time to as little as 30 ms. Furthermore, StarDojoMod supports headless operation through the X Virtual Framebuffer (Xvfb), enabling compatibility with Linux systems without graphical interfaces, thus broadening accessibility across diverse hardware and system configurations.

3.3 Observation and Action Spaces

Observation Space. StarDojo offers a comprehensive observation space that integrates both visual and textual modalities to accommodate a wide range of agent architectures. Each observation includes a gameplay screenshot alongside detailed textual information describing the game state. This textual state contains character status (such as health and energy), local tile information ($n \times n$ tiles surrounding the agent), and global information (such as time, weather, and the positions of NPCs and buildings). By combining visual and textual observations, StarDojo ensures that agents can leverage both high-level context and fine-grained environmental cues for more robust and informed decision-making.

Action Space. The action space in StarDojo is designed to encompass the full range of activities that can be performed in the original Stardew Valley using a keyboard and mouse. Actions are carefully abstracted to eliminate redundant operations while retaining the core decision-making challenges inherent to the game. We define ten fundamental actions, which together are sufficient to cover most gameplay scenarios: *move(x, y)*, *use(direction)*, *interact(direction)*, *choose_item(slot_index)*, *attach_item(slot_index)*, *detach_item()*, *craft(item)*, *choose_option(option_index, quantity, direction)* and *menu(option, menu_name)*.

3.4 Tasks

As shown in Figure 3, we carefully curate 1000 tasks to benchmark agents' various behaviors in StarDojo. These tasks are divided into five distinct categories: Farming, Crafting, Exploration, Combat and Social, which cover most of the production-living activities in the early and middle stages of the game. Each task

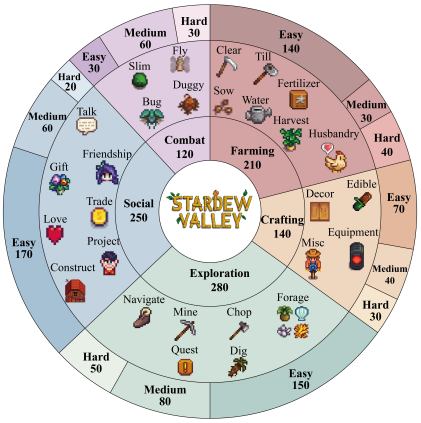

Fig. 3: Distribution of 1000 tasks across five categories: Farming, Crafting, Exploration, Combat and Social in StarDojo, each with Easy, Medium, and Hard difficulties.

Table 2: Task statistics of StarDojo-Lite. The task suite is made up of the most representative early-stage tasks from each category.

Category	Easy	Medium	Hard	Total
Farming	14	3	4	21
Crafting	7	4	3	14
Exploration	15	8	5	28
Combat	3	6	3	12
Social	17	6	2	25
Total	56	23	21	100

is classified into three difficulties: easy, medium, and hard. For easy-level tasks, agents are provided with all necessary items or tools from the start (e.g., mature crops ready for harvest, ingredients for crafting). Agents are initialized near the target location (e.g., inside a shop with enough budget for trading). These tasks primarily test the agent's basic ability to complete atomic activities. For medium-level tasks, agents need to fulfill the prerequisites of the tasks on their own, such as planting and harvesting crops, gathering crafting ingredients, traveling from the farm to the target area, and earning sufficient funds to purchase specific items. Many of these resources can be acquired through multiple sources and methods. This flexibility gives agents considerable freedom in choosing their strategies. Hard-level tasks typically require several in-game days to complete and are often composed of multiple medium-level tasks. Agents have greater freedom to allocate their time and energy each day, choosing how to prioritize activities and sequence actions. Success often demands strategic long-term planning, adaptive decision-making, and efficient resource management, as there are multiple viable paths to achieving the goal.

As shown in Table 2, to facilitate efficient agent evaluation, we curate a representative smaller task suite, called StarDojo-Lite, comprising 100 core tasks from the full task collection, balancing coverage and practicality. This lite task set covers most of the representative activities in the early stage of the game.

Playthrough. In addition to modular tasks, StarDojo also provides an extended Playthrough task designed to comprehensively assess an agent's strategic decision-making over prolonged gameplay. Starting from scratch, the agent's objective is to accumulate 1 million in-game currency, which is a significant milestone typically achieved over hundreds of in-game days. Success requires efficient seasonal planning, unlocking new maps through quests, careful resource

and energy management, skill progression for advanced crafting, and strategic relationship-building with NPCs.

Initial Config and Setup. Some tasks require the agent to possess specific equipment (e.g., a sword), have certain items (e.g., sufficient crop seeds in inventory), or have completed particular game progress (e.g., unlocking the mines). To establish the initial state for each task, we provide multiple saved game files reflecting various stages of progression, along with specialized task-specific functions utilizing StarDojoMod commands. At the start of each task, StarDojo automatically loads the corresponding saved game and executes these task-specific commands, ensuring all necessary prerequisites, such as items, equipment, skill levels, date and farm status, are appropriately configured.

Automatic Evaluation. Given the extensive number of existing tasks and the vast potential for future additions, it is essential to establish a reusable, scalable, adaptive, and efficient evaluation mechanism. StarDojo addresses this need by implementing a general evaluation system that continuously monitors task progression and provides immediate reward feedback to agents. Specifically, after agents finish executing actions at each step, the evaluator is invoked to determine whether tasks have been successfully completed or have reached the predefined maximum number of steps. To achieve this, the evaluator maintains the previous game state information, compares it with the current state obtained via StarDojoMod, identifies incremental changes indicating progress, and accurately assesses task completion criteria based on these observed differences. By leveraging a comprehensive observation space for incremental progress tracking, our evaluation approach effectively mitigates discrepancies caused by varying initial conditions, ensuring robust adaptability across diverse tasks.

4 Empirical Studies

In this section, we benchmark multiple advanced MLLMs on StarDojo and provide a comprehensive study and analysis of their behaviors and limitations.

MLLM Baselines. We evaluated seven MLLMs on StarDojo-Lite task set: GPT-4.1 series [23](gpt-4.1-2025-04-14 & gpt-4.1-mini-2025-04-14), Claude 3.7 Sonnet [3] (claude-3-7-sonnet-20250219), Gemini 2.5 Pro [10] (gemini-2.5-pro-preview-03-25), from the closed-source community and Llama 4 Maverick [21] (Llama-4-Maverick-17B-128E-Instruct), Qwen2.5 VL [4] (qwen2.5-vl-72b-instruct) and Gemma 3 [35] (gemma-3-27b-it) from the open-source community.

Settings. If not mentioned explicitly, all experiments are conducted under the following settings: Agents have access to both visual and textual observations for their decision-making process. Visual observations are provided at a resolution of 720P (1280×720). Textual observations include 7×7 agent-centered surrounding information and other global information like time, date and budget. In addition to receiving observations from the current timestep, agents are also provided with history information from the previous timestep, enabling agents to reflect on past states and facilitating consistency in decision-making. Agents can output at most two skills as an action to be executed sequentially. After executing all

Table 3: Success rates(%) and standard deviation of agents with different base models on StarDojo-Lite task set, ranging over five categories (Farming, Crafting, Exploration, Combat and Social) and three levels of difficulty. Each task is evaluated over three runs.

Model	Task	Farming	Crafting	Exploration	Combat	Social	Total
GPT-4.1	**Easy**	**31.0**±4.1	**52.4**±8.3	**15.6**±3.9	**11.1**±19.3	7.8±6.8	**21.4**±1.8
	Medium	0.0±0.0	0.0±0.0	8.3±7.2	0.0±0.0	0.0±0.0	2.7±2.3
	Hard	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	Total	**20.6**±2.8	**26.2**±4.1	**10.7**±0.0	**2.8**±4.8	5.3±4.6	**12.7**±0.6
Gemini 2.5 Pro	**Easy**	26.2±4.1	47.6±8.3	6.7±6.7	0.0±0.0	**11.8**±5.9	17.9±1.8
	Medium	0.0±0.0	0.0±0.0	8.3±7.2	0.0±0.0	0.0±0.0	2.7±2.3
	Hard	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	Total	17.5±2.8	23.8±4.1	6.0±2.1	0.0±0.0	**8.0**±4.0	10.7±0.6
Claude 3.7 Sonnet	**Easy**	**31.0**±4.1	28.6±0.0	8.9±10.2	0.0±0.0	7.8±3.4	16.1±4.7
	Medium	0.0±0.0	0.0±0.0	**16.7**±7.2	0.0±0.0	0.0±0.0	**5.3**±2.3
	Hard	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	Total	**20.6**±2.8	14.3±0.0	9.5±5.5	0.0±0.0	5.3±2.3	10.3±2.5
GPT-4.1 mini	**Easy**	26.2±4.1	4.8±8.3	4.4±7.7	**11.1**±19.3	7.8±3.4	11.3±2.7
	Medium	0.0±0.0	0.0±0.0	8.3±7.2	0.0±0.0	0.0±0.0	2.7±2.3
	Hard	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	Total	17.5±2.8	2.4±4.1	4.8±5.5	**2.8**±4.8	5.3±2.3	7.0±1.7
Llama 4 Maverick	**Easy**	28.6±0.0	28.6±0.0	0.0±0.0	0.0±0.0	7.8±3.4	13.1±1.0
	Medium	0.0±0.0	0.0±0.0	4.2±7.2	0.0±0.0	0.0±0.0	1.3±2.3
	Hard	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	Total	19.1±0.0	14.3±0.0	1.2±2.1	0.0±0.0	5.3±2.3	7.7±0.6
Qwen2.5 VL 72B	**Easy**	16.7±8.3	9.5±8.3	13.3±13.3	0.0±0.0	9.8±3.4	11.9±5.5
	Medium	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	Hard	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	Total	11.1±5.5	4.8±4.1	7.1±7.1	0.0±0.0	6.7±2.3	6.7±3.1
Gemma 3 27B	**Easy**	9.5±4.1	0.0±0.0	2.2±3.9	0.0±0.0	**11.8**±0.0	6.6±2.1
	Medium	0.0±0.0	0.0±0.0	4.2±7.2	0.0±0.0	0.0±0.0	1.3±2.3
	Hard	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
	Total	6.4±2.8	0.0±0.0	2.4±2.1	0.0±0.0	**8.0**±0.0	4.0±1.0

the actions, the environment is paused until the agent outputs the next action. Each task is evaluated over three runs with a heuristic maximum of 30, 50 and 150 steps for easy, medium and hard level tasks.

4.1 Qualitative Analysis

Overall Results. As shown in Table 3, a clear gap between flagship commercial models and open-source models can be observed. GPT-4.1, Gemini 2.5 Pro, and Claude 3.7 Sonnet all exceed a 10% overall success rate, with GPT-4.1 achieving the highest at 12.7%. In contrast, all open-source models remain below 8%, with Llama 4 Maverick performing best, largely due to its larger model size. Most successful completions are limited to easy tasks, whereas all models struggle significantly with medium and hard tasks, achieving near-zero success rates due

to increased task complexity and longer sequences of required actions. Models show some proficiency in farming and crafting tasks but exhibit considerable difficulty in exploration, combat, and social interactions.

Low-Level Control as Tool-Use. All models demonstrate reasonable ability to control agents with non-trivial movements and interactions with actions clearly specified in the prompts. Larger models such as GPT-4.1, Gemini 2.5 Pro, Claude 3.7 Sonnet and Llama 4 Maverick perform significantly better than others on Crafting tasks, highlighting their superior tool-use capabilities and in-context understanding when handling more complex function calls. This contrast explains the weaker performance of smaller models like GPT-4.1 Mini and Gemma 3 27B, which often struggle to provide valid and accurate parameters when calling actions like *choose_ option* and *craft*. However, all models continue to struggle in fast-paced tasks such as Combat, which demand dynamic control.

Navigation is the Key Bottleneck. We found that all models struggled most severely with navigation, which is the core skill required in the game. Due to limited image understanding, they consistently failed to accurately identify and locate target objects, building entrances or exits, and map transitions, which usually only appear in the provided image rather than the textual information. These shortcomings heavily impaired tasks that involve moving across scenarios or exploring areas beyond the immediate field of view. As a result, success rates were particularly low in Exploration and Social tasks, both of which often require cross-map navigation to locate targets. In contrast, agents are less influenced and perform relatively better in easy-level Farming tasks, since these are usually confined to a small farmland where target products remain within visual proximity and are directly accessible.

Long-Horizon Tasks Remain Far From Solved. Medium and hard-level tasks can typically be decomposed into multiple simpler subtasks involving different category combinations. However, given the low success rates even on easy-level tasks, which serve as the atomic activities in the benchmark, all models remain far from being able to complete complex long-horizon tasks in StarDojo. This gap reveals substantial potential for more capable MLLM models.

4.2 Error Analysis & Case Studies

To better understand model behavior, we conduct an error analysis of the GPT-4.1-based agent on the StarDojo-Lite task set to identify and categorize its major failure modes in Figure 4. Figure 5 shows four representative error cases, highlighting common failure patterns of current agents during environment interaction. For clarity, we include only the key parts of the textual observations and reasoning traces. For tasks with multiple errors, we report primary errors that block further progress.

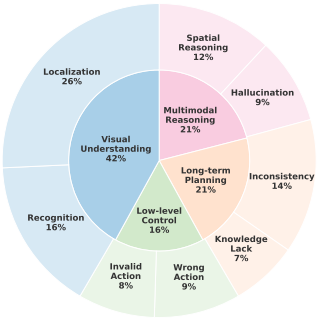

Fig. 4: Error Analysis.

The error analysis reveals several key failure modes, with the most significant being limited

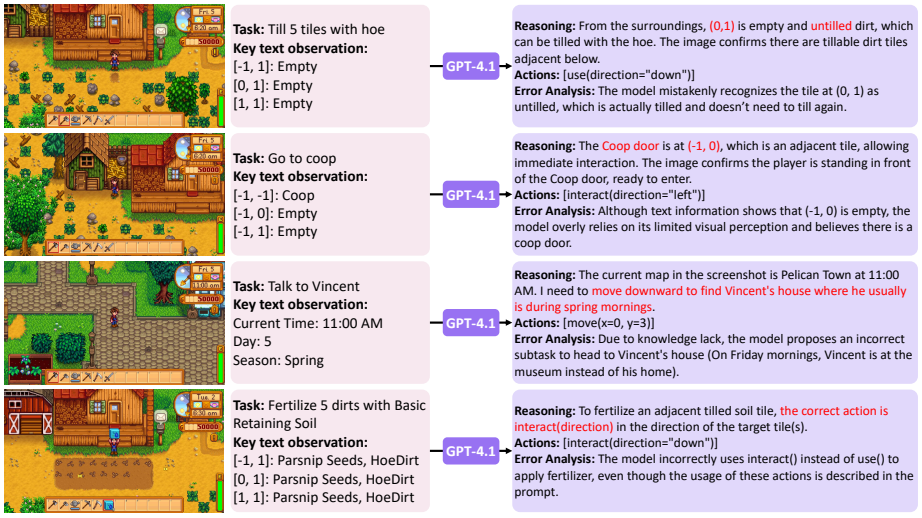

Fig. 5: Cases illustrating four typical errors identified in the error analysis section: visual understanding, multimodal reasoning, long-term planning, and low-level control. For simplicity, we present only the key portions of the textual observations. Incorrect reasoning is highlighted in red, and we provide corresponding analyses to explain the causes of these errors.

Visual Understanding, which accounts for 42% of all errors. The model often struggles to reliably recognize target objects, many of which are only around 10x10 pixels shown in the image. And even when detected, it frequently fails to accurately determine their position relative to the character, hindering effective navigation and interaction. Additionally, models often struggle to interpret the status of tiles, even those immediately adjacent to the character, such as whether a tile is tilled, seeded, watered, or obstructed. The second major source of failure, at 21%, is **Multimodal Reasoning**, where the model overly relies on its limited visual perception instead of integrating both visual and textual inputs, leading to confusion even when clear positional or status information is available in the text. Additionally, the model sometimes hallucinates task progress by incorrectly assuming the success of previous actions. **Long-Term Planning** issues also contribute to 21% of failures, as the model, while capable of proposing reasonable subtasks, often abandons plans prematurely and switches strategies inconsistently, which undermines progress on longer-horizon tasks; this is sometimes compounded by insufficient domain knowledge, such as not knowing a crafting recipe or which NPC to interact with. Lastly, **Low-Level Control** accounts for a smaller but persistent 16% of errors, where the model, despite generally choosing appropriate actions, occasionally selects incorrect or suboptimal ones, demonstrating instability in fine-grained execution, with occasional formatting or parameter errors.

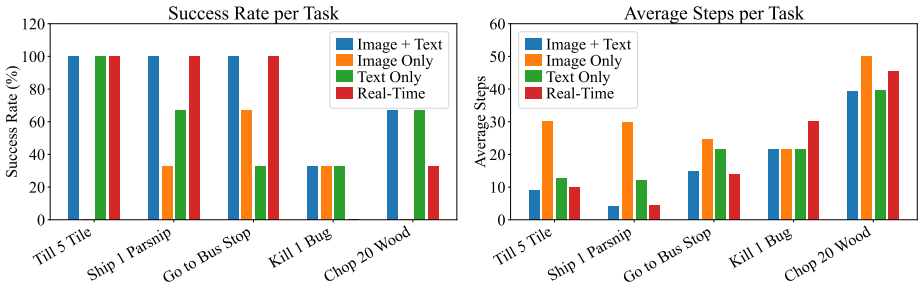

Fig. 6: Ablation results of GPT-4.1-based agents on five representative tasks under four different settings: *Image + Text* (default setting), *Image Only*, *Text Only*, and *Real-Time* (without game pausing). Except for *Chop 20 Wood*, which is a medium-level task with a maximum of 50 steps, the remaining four are easy-level tasks capped at 30 steps. Each task is evaluated over three runs.

4.3 Ablation Studies

To demonstrate the flexible customization capabilities of StarDojo and enable more comprehensive evaluations of agent performance, we benchmark GPT-4.1 under the following three additional settings using five representative basic tasks. 1) **Image Only**: Agents receive only visual observations without textual state information, evaluating their capability to rely solely on visual cues. 2) **Text Only**: Agents receive textual state observations without visual input, restricting the agent's perception to local 7×7 grid-based information. This setting is particularly relevant for LLM agents. 3) **Real-time**: The environment progresses continuously during action generation, simulating real gameplay conditions where agents must plan and respond without game pausing.

As illustrated in Figure 6, removing textual input (*Image Only*) significantly affects agents' performance across all tasks, reflecting the defect of base models' poor visual-based control, emphasizing textual information's importance in grounding detailed action decisions for the current stage of agents. On the other side, eliminating visual input (*Text Only*) substantially reduces success in tasks that require navigation like *Ship 1 Parsnip* and *Go to Bus Stop*, demonstrating the essential contribution of visual cues to spatial reasoning and movement. Disabling the feature to pause the environment (*Real-time*) remarkably affects performance across tasks demanding timely reactions or prolonged action sequences. For instance, in combat-related tasks such as *Kill 1 Bug*, the bug keeps moving during the model's inference. It usually takes more than 10 seconds to get the GPT4.1's results through the API. The bug has already moved away after the model's inference and executed the actions based on the position 10 seconds ago. Similarly, in long-horizon tasks like *Chop 20 Wood*, the in-game time keeps elapsed during models' inference, if passing, complete the task will usually take 2h in-game hours, if not pausing, it will take more than 12 in-game hours, which can extend into nighttime or span multiple in-game days that greatly delay and affect the completion of the task. Seldom previous benchmarks that emphasised the

importance of evaluating the effect of real-timeness. These observations highlight the practical importance of real-time evaluation, a factor often overlooked in prior benchmarks. In many cases, agents may appear effective in offline evaluations but require substantially longer wall-clock time in real deployment settings. This issue is particularly pronounced for advanced agents built upon cutting-edge closed-source models accessed through APIs, where inference latency can become a bottleneck. In contrast, smaller open-source models typically offer faster inference but often suffer from substantially weaker task performance, leading to a fundamental trade-off between inference efficiency and capability.

5 Limitations and Conclusion

Limitations. This work has several potential limitations. 1) Although StarDojo is an open-source environment and benchmark, users need an official copy of Stardew Valley to run it. 2) Fishing, an optional gameplay, is currently not included in StarDojo, due to its nature as a complex, real-time mini-game. The operations are independent of other game controls, which is not applicable to evaluate MLLMs. 3) The benchmark primarily focuses on early- and mid-game content within the main valley map. Other advanced areas, such as the Desert and Ginger Island, are not yet supported. 4) Since "social" is a very broad concept, as the first benchmark that combines production and living, we do not intend to claim that StarDojo evaluates open-ended social intelligence across all aspects. Instead, it focuses on socially situated decision-making within an embodied environment. 5) Our evaluations were conducted mainly on StarDojo-Lite with seven models. A more comprehensive assessment across a broader range of tasks, models, and agentic approaches remains an important direction for future work.

Conclusion. We introduce StarDojo, a novel environment and benchmark designed to evaluate the open-ended behaviors of MLLM agents in Stardew Valley. StarDojo bridges the gap in existing environments by enabling comprehensive assessment of agents across both production and daily living activities within a simulated nature and society. Featuring a set of diverse tasks, StarDojo exposes significant challenges in current agents' visual understanding, multimodal reasoning, long-term planning, and real-time inference, highlighting key areas for future research and development.

Acknowledgements

This research is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG18/24). This research is also supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant (Proposal ID: 23-SIS-SMU-037).

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Agarwal, M., Rana, S., Sundoro, T., Berhe, H., Kim, S., Sharma, V., O'Brien, S., Zhu, K.: Wolf: Werewolf-based observations for llm deception and falsehoods. arXiv preprint arXiv:2512.09187 (2025)
3. Anthropic: Claude 3.7 sonnet and claude code (2025), accessed: 2025-05-10
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
5. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research **47**, 253–279 (2013)
6. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
7. Côté, M.A., Kádár, A., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Hausknecht, M., El Asri, L., Adada, M., et al.: Textworld: A learning environment for text-based games. In: Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7. pp. 41–75. Springer (2019)
8. Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.A., Zhu, Y., Anandkumar, A.: Minedojo: Building open-ended embodied agents with internet-scale knowledge. Advances in Neural Information Processing Systems **35**, 18343–18362 (2022)
9. Fu, J., Gao, Q., Jiang, D., Li, X., Lin, G.: Spatial–temporal distribution of global production–living–ecological space during the period 2000–2020. Scientific Data **10**(1), 589 (2023)
10. Google: Gemini 2.5: Our most intelligent ai model (2025), accessed: 2025-05-10
11. Guss, W.H., Houghton, B., Topin, N., Wang, P., Codel, C., Veloso, M., Salakhutdinov, R.: Minerl: A large-scale dataset of minecraft demonstrations. arXiv preprint arXiv:1907.13440 (2019)
12. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. pp. 1861–1870. PMLR (2018)
13. Hafner, D.: Benchmarking the spectrum of agent capabilities. arXiv preprint arXiv:2109.06780 (2021)
14. Hong, S., Zheng, X., Chen, J., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S.K.S., Lin, Z., Zhou, L., et al.: Metagpt: Meta programming for multi-agent collaborative framework. arXiv preprint arXiv:2308.00352 (2023)
15. Koh, J.Y., Lo, R., Jang, L., Duvvur, V., Lim, M.C., Huang, P.Y., Neubig, G., Zhou, S., Salakhutdinov, R., Fried, D.: Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. arXiv preprint arXiv:2401.13649 (2024)
16. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., et al.: Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474 (2017)

17. Li, C., Xia, F., Martín-Martín, R., Lingelbach, M., Srivastava, S., Shen, B., Vainio, K., Gokmen, C., Dharan, G., Jain, T., et al.: igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. arXiv preprint arXiv:2108.03272 (2021)
18. Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Dal Lago, A., et al.: Competition-level code generation with alphacode. Science **378**(6624), 1092–1097 (2022)
19. Light, J., Cai, M., Shen, S., Hu, Z.: Avalonbench: Evaluating llms playing the game of avalon. arXiv preprint arXiv:2310.05036 (2023)
20. Lillicrap, T.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)
21. Meta: The llama 4 herd: The beginning of a new era of natively multimodal ai innovation (2025), accessed: 2025-05-10
22. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)
23. OpenAI: Introducing gpt-4.1 in the api (2025), accessed: 2025-05-10
24. Paglieri, D., Cupiał, B., Coward, S., Piterbarg, U., Wolczyk, M., Khan, A., Pignatelli, E., Kuciński, Ł., Pinto, L., Fergus, R., et al.: Balrog: Benchmarking agentic llm and vlm reasoning on games. arXiv preprint arXiv:2411.13543 (2024)
25. Park, J.S., O'Brien, J., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S.: 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)
26. Plamondon-Willard, J.: SMAPI: Stardew Modding API, accessed: 2025-07-09
27. Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., Torralba, A.: Virtualhome: Simulating household activities via programs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8494–8502 (2018)
28. Puig, X., Undersander, E., Szot, A., Cote, M.D., Yang, T.Y., Partsey, R., Desai, R., Clegg, A.W., Hlavac, M., Min, S.Y., et al.: Habitat 3.0: A co-habitat for humans, avatars and robots. arXiv preprint arXiv:2310.13724 (2023)
29. Qi, S., Chen, S., Li, Y., Kong, X., Wang, J., Yang, B., Wong, P., Zhong, Y., Zhang, X., Zhang, Z., et al.: Civrealm: A learning and reasoning odyssey in civilization for decision-making agents. arXiv preprint arXiv:2401.10568 (2024)
30. Samvelyan, M., Rashid, T., De Witt, C.S., Farquhar, G., Nardelli, N., Rudner, T.G., Hung, C.M., Torr, P.H., Foerster, J., Whiteson, S.: The starcraft multi-agent challenge. arXiv preprint arXiv:1902.04043 (2019)
31. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
32. Shi, T., Karpathy, A., Fan, L., Hernandez, J., Liang, P.: World of bits: An open-domain platform for web-based agents. In: International Conference on Machine Learning. pp. 3135–3144. PMLR (2017)
33. Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K.R., Yao, S.: Reflexion: language agents with verbal reinforcement learning. In: Thirty-seventh Conference on Neural Information Processing Systems (2023), https://openreview.net/forum?id=vAElhFcKW6
34. Tan, W., Zhang, W., Xu, X., Xia, H., Ding, G., Li, B., Zhou, B., Yue, J., Jiang, J., Li, Y., et al.: Cradle: Empowering foundation agents towards general computer control. In: NeurIPS 2024 Workshop on Open-World Agents (2024)
35. Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al.: Gemma 3 technical report. arXiv preprint arXiv:2503.19786 (2025)

36. Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. pp. 5026–5033. IEEE (2012)
37. Trinh, T.H., Wu, Y., Le, Q.V., He, H., Luong, T.: Solving olympiad geometry without human demonstrations. Nature **625**(7995), 476–482 (2024)
38. Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al.: Grandmaster level in StarCraft II using multi-agent reinforcement learning. nature **575**(7782), 350–354 (2019)
39. Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A.: Voyager: An open-ended embodied agent with large language models. arXiv preprint arXiv:2305.16291 (2023)
40. Wang, J., Xu, H., Jia, H., Zhang, X., Yan, M., Shen, W., Zhang, J., Huang, F., Sang, J.: Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. arXiv preprint arXiv:2406.01014 (2024)
41. Wang, X., Li, B., Song, Y., Xu, F.F., Tang, X., Zhuge, M., Pan, J., Song, Y., Li, B., Singh, J., et al.: Openhands: An open platform for ai software developers as generalist agents. arXiv preprint arXiv:2407.16741 (2024)
42. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., Li, B., Jiang, L., Zhang, X., Wang, C.: Autogen: Enabling next-gen llm applications via multi-agent conversation framework. arXiv preprint arXiv:2308.08155 (2023)
43. Wu, Z., Han, C., Ding, Z., Weng, Z., Liu, Z., Yao, S., Yu, T., Kong, L.: OS-copilot: Towards generalist computer agents with self-improvement. arXiv preprint arXiv:2402.07456 (2024)
44. Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Toh, J.H., Cheng, Z., Shin, D., Lei, F., et al.: Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. Advances in Neural Information Processing Systems **37**, 52040–52094 (2025)
45. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: ReAct: Synergizing reasoning and acting in language models. In: International Conference on Learning Representations (ICLR) (2023)
46. Zhang, C., Li, L., He, S., Zhang, X., Qiao, B., Qin, S., Ma, M., Kang, Y., Lin, Q., Rajmohan, S., et al.: UFO: A UI-focused agent for Windows OS interaction. arXiv preprint arXiv:2402.07939 (2024)
47. Zheng, L., Wang, R., Wang, X., An, B.: Synapse: Trajectory-as-exemplar prompting with memory for computer control. In: ICLR (2024)
48. Zheng, X., Li, L., Yang, Z., Yu, P., Wang, A.J., Yan, R., Yao, Y., Wang, L.: V-mage: A game evaluation framework for assessing visual-centric capabilities in multimodal large language models. arXiv preprint arXiv:2504.06148 (2025)
49. Zhou, S., Xu, F.F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., et al.: Webarena: A realistic web environment for building autonomous agents. arXiv preprint arXiv:2307.13854 (2023)