MineAnyBuild: Benchmarking Spatial Planning for Open-world AI Agents

Ziming Wei[1]*, **Bingqian Lin**[2]*, **Zijian Jiao**[1]*, **Yunshuang Nie**[1], **Liang Ma**[3]
Yuecheng Liu[4], **Yuzheng Zhuang**[4], **Xiaodan Liang**[1]†
[1]Shenzhen Campus of Sun Yat-sen University [2]Shanghai Jiao Tong University
[3]Mohamed bin Zayed University of Artificial Intelligence [4]Huawei Noah's Ark Lab
Project Website: `https://mineanybuild.github.io/`

Abstract

Spatial Planning is a crucial part in the field of spatial intelligence, which requires the understanding and planning about object arrangements in space perspective. AI agents with the spatial planning ability can better adapt to various real-world applications, including robotic manipulation, automatic assembly, urban planning *etc*. Recent works have attempted to construct benchmarks for evaluating the spatial intelligence of Multimodal Large Language Models (MLLMs). Nevertheless, these benchmarks primarily focus on spatial reasoning based on typical Visual Question-Answering (VQA) forms, which suffers from the gap between abstract spatial understanding and concrete task execution. In this work, we take a step further to build a comprehensive benchmark called **MineAnyBuild**, aiming to evaluate the spatial planning ability of open-world AI agents in the *Minecraft* game. Specifically, MineAnyBuild requires an agent to generate *executable architecture building plans* based on the given multi-modal human instructions. It involves 4,000 curated tasks and provides a paradigm for infinitely expandable data collection by utilizing rich player-generated content. MineAnyBuild evaluates spatial planning through four core supporting dimensions: spatial understanding, spatial reasoning, creativity, and spatial commonsense. Based on MineAnyBuild, we perform a comprehensive evaluation for existing MLLM-based agents, revealing the severe limitations but enormous potential in their spatial planning abilities. We believe our MineAnyBuild will open new avenues for the evaluation of spatial intelligence and help promote further development for open-world AI agents capable of spatial planning.

1 Introduction

Spatial intelligence, an emerging research field gradually attracting the attention of AI researchers, requires AI agents to understand, reason and memorize the visual-spatial relationships between objects and spaces [1, 2, 3, 4]. Spatial planning is a pivotal capability regarding spatial intelligence, which requires agents to not only perform spatial perception and cognition, but also generate executable planning in 3D space. Spatial planning is widely needed in various human-centric real-world applications, including automatic assembly, architectural design, environmental urban planning, *etc*.

AI Agents integrated with Multi-modal Large Language Models (MLLMs) have demonstrated astonishing capabilities in various tasks in text (1D) and image (2D) domains [5, 6, 7]. To investigate how existing MLLM-based agents can handle space dimension tasks, several benchmarks designed for evaluating the spatial intelligence have been proposed recently [8, 9, 10, 11]. These benchmarks reveal that although AI agents perform well in tasks of text and image domains, they still present

*Equal Contribution. †Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

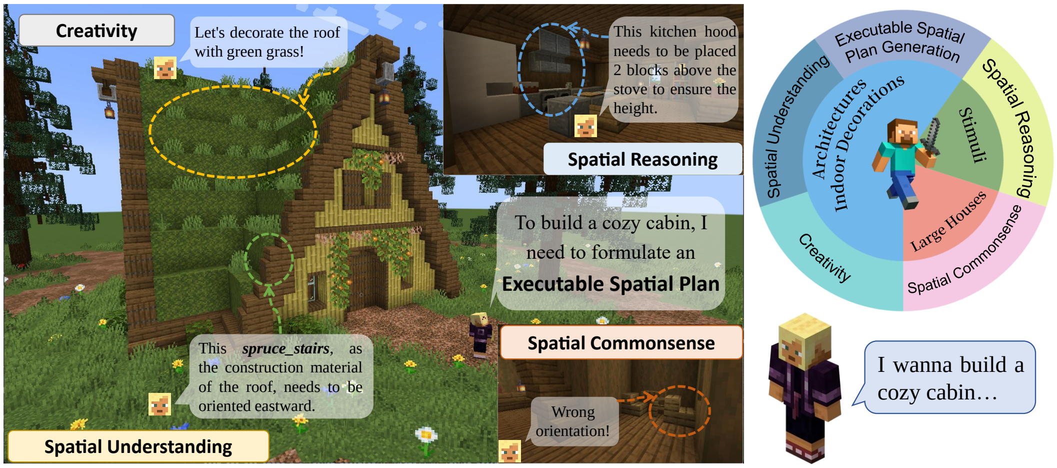

Creativity
Let's decorate the roof with green grass!
This kitchen hood needs to be placed 2 blocks above the stove to ensure the height.
Spatial Reasoning
To build a cozy cabin, I need to formulate an Executable Spatial Plan
Spatial Commonsense
This spruce_stairs, as the construction material of the roof, needs to be oriented eastward.
Wrong orientation!
Spatial Understanding
Executable Spatial Plan Generation
Spatial Reasoning
Spatial Commonsense
Creativity
Spatial Understanding
Architectures
Indoor Decorations
Stimuli
Large Houses
I wanna build a cozy cabin…

Figure 1: Overview of **MineAnyBuild**. Our MineAnyBuild is a novel benchmark built on the Minecraft game, which aims to evaluate the spatial planning capabilities of open-world AI agents. In MineAnyBuild, the agent needs to generate *executable spatial plans* to construct a building or indoor decoration following given multi-modal human instructions. We introduce four core dimensions, including spatial understanding, creativity, spatial reasoning, and spatial commonsense to fulfill a comprehensive assessment for spatial planning.

relatively poor performance in spatial dimension tasks. However, current benchmarks have critical constraints in evaluating spatial intelligence. They mainly focus on metric-level spatial understanding tasks and predominantly employ Visual Question-Answering (VQA) pairs, requiring AI agents to answer geometric attributes (e.g., distance, positional coordinates, or spatial relations of objects in 3D space), while neglecting the gap between abstract spatial understanding and concrete task execution.

In this work, we propose **MineAnyBuild**, which is an innovative benchmark designed to evaluate an important yet unexplored aspect of spatial intelligence, i.e., spatial planning, for open-world AI agents. In our MineAnyBuild, the agents need to generate *executable spatial plans* following human instructions for constructing a building or indoor decoration, which requires both spatial reasoning and task execution. We build our benchmark on the popular *Minecraft* game, where a player journeys through a 3D world with diverse biomes to explore, tools to craft, and architectures to build. Compared to benchmarks focusing on skills learning or tech-tree tasks [12, 13], architecture building has always been a vital attraction to millions of players to present the openness and freedom of Minecraft. Unlike most other games, Minecraft defines go-as-you-please goals, making it well suited for developing open-ended tasks for AI agent research.

Our MineAnyBuild benchmark consists of 4,000 curated tasks where four core evaluation dimensions are introduced. As shown in Figure 1, given a multimodal human instruction, agents are requested to perform **spatial understanding** to abstract a pivotal basic structure according to specific or brief demands, where agents emulate architects in our real world, and plan the composition of each basic units. Agents also need to reason and think about whether the units of the architecture from different perspectives conforms to spatial rules by **spatial reasoning**. For the overall appearance of the architecture, agents exert their **creativity** and imagination to make it more aesthetically unique, or to simulate some well-designed or delicate structural designs in the real world through the combination of fixed-shaped blocks, e.g., using a variety of stairs and slabs to design unique Chinese-style or castle-style roofs. For some architectures like modern houses, agents implement **spatial commonsense** to judge the rationality of each designs inside the buildings.

We design and construct different tasks based on multiple aspects that a human player would consider, to evaluate several capabilities of AI agents. Agents are supposed to response with concrete layout sequences of expected architectures to present their spatial planning. For some tasks that are not easy to evaluate directly like spatial reasoning, we customize tasks inspired by classical mental rotation experiments [14, 15] to test agents. For creativity, we score and vote on the overall aesthetics and structural strategy by human evaluation or critique-based MLLMs. We also propose an infinitely expandable paradigm to utilize Minecraft data on the Internet, where millions of active players provide their creation and shares, to build our tasks. Through our data curation pipeline, we can collect endless tasks evaluating spatial intelligence for open-world agents, making further contributions to promoting AI agents research.

We test the tasks on several state-of-the-art MLLM-based AI agents and observe that even the most powerful MLLMs like GPT-4o and Claude-3.7-Sonnet demonstrate significant limitations in most tasks, where GPT-4o obtains an overall score of 41.02, far lower than the maximum score of 100. Open-source models generally have poor capabilities to generate executable spatial plan, reflecting a serious deficiency in their understanding of spatial data. These results reveal the foresight of our MineAnyBuild for AI evaluation.

To summarize, the main contributions of this work are as follows:

- We propose MineAnyBuild, which benchmarks the spatial planning evaluation for open-world AI agents in the Minecraft game. MineAnyBuild covers diverse evaluation dimensions, including spatial reasoning, creativity, spatial commonsense, *etc*. Through requiring the agent to generate executable architecture building plans, our MineAnyBuild significantly mitigate the gap between abstract spatial understanding and concrete task execution.
- We test various existing MLLM-based AI agents for spatial planning in multiple perspectives and difficulties, which exposes the insufficiency of the existing AI agents' capabilities in spatial planning. We provide the visualization results on executable planning outputs and failure cases, revealing that current AI agents are still facing tough issues such as spatial misunderstanding and implementation gap to be handled.
- We propose an infinitely expandable data curation pipeline to scale our benchmark and datasets, where we can collect endless player-generated content on the Internet and automatically convert it into processable data. Our pipeline well utilize the abundant creations made by millions of players to benefit the training and evaluation of open-world AI agents.

2 Benchmark and Task Suite

In this section, we describe our MineAnyBuild benchmark in detail. Specifically, we first present the overview of our benchmark in Section 2.1. Then, we define various spatial planning tasks in MineAnyBuild in Section 2.2. Finally, we introduce our data curation pipeline in Section 2.3.

2.1 Benchmark Overview

MineAnyBuild is designed to evaluate an AI agent's capabilities in spatial planning to conduct infinite architecture creations in Minecraft game. Spatial planning is a critical capability, aiming to examine agents' understanding and disassembly of combinations in 3D space, and to construct each sub-units and judge the rationality of them by reasoning or commonsense. Our benchmark examines various MLLM-based agents to conduct planning on architectures, which requires them to generate executable architectural construction plans according to different forms of instructions or visual inputs. Evaluating the creativity of agents becomes essential in our particular architectural construction tasks, as it reflects human-centric assessments of aesthetic value across spatial planning and conception designing domains. For some evaluating dimensions that are not easily presented directly in spatial planning, such as spatial reasoning and spatial commonsense, we design series of visual question-answering pairs to indirectly reflect the manifestation of these two capabilities of agents. The next sections detail the specific tasks and the process of our data curation.

2.2 Tasks

Our MineAnyBuild involves approximately 4,000 spatial planning tasks with 500+ buildings/indoor decoration assets. These tasks, including Executable Spatial Plan Generation, Spatial Understanding, Creativity, Spatial Reasoning, and Spatial Commonsense, correspond to diverse evaluation dimensions, thereby conducting a comprehensive assessment of AI agents' spatial planning capabilities. In Executable Spatial Plan Generation, Spatial Understanding, and Creativity tasks, the agent needs to generate executable spatial plans for building an architecture according to the given instruction. While in Spatial Reasoning and Spatial Commonsense tasks, we introduce $\sim$2,000 VQA pairs, where we ask the agent to answer the given questions accompanied by the related images. In the following, we define each task in detail, and present the corresponding task examples in Figure 2.

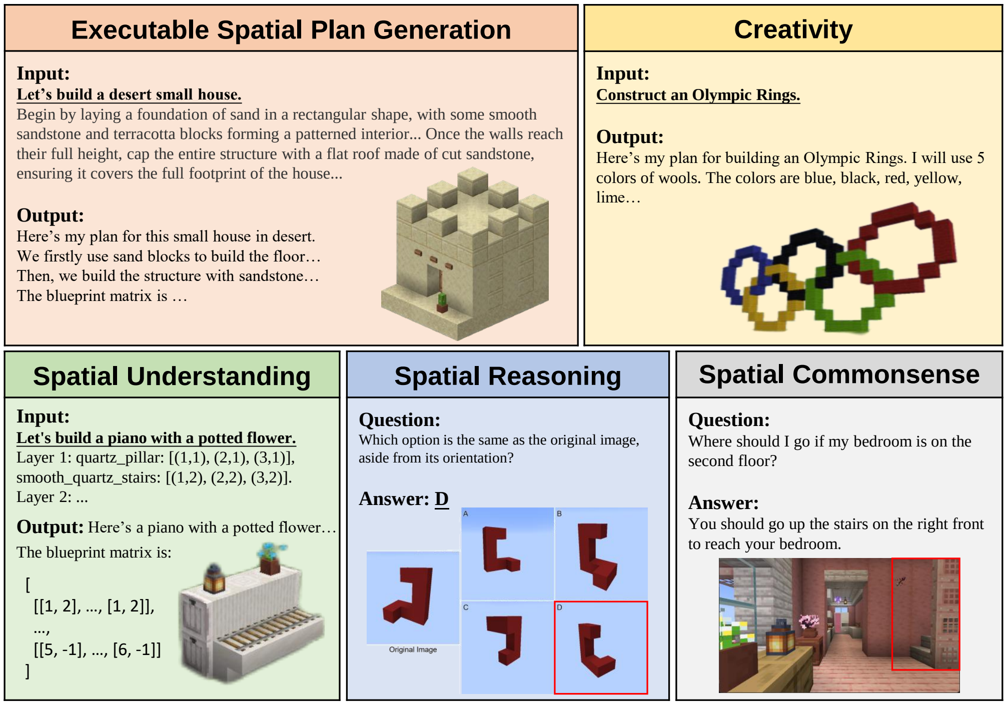

Executable Spatial Plan Generation
Input:
Let's build a desert small house.
Begin by laying a foundation of sand in a rectangular shape, with some smooth sandstone and terracotta blocks forming a patterned interior... Once the walls reach their full height, cap the entire structure with a flat roof made of cut sandstone, ensuring it covers the full footprint of the house...
Output:
Here's my plan for this small house in desert.
We firstly use sand blocks to build the floor…
Then, we build the structure with sandstone…
The blueprint matrix is …
Creativity
Input:
Construct an Olympic Rings.
Output:
Here's my plan for building an Olympic Rings. I will use 5 colors of wools. The colors are blue, black, red, yellow, lime…
Spatial Understanding
Input:
Let's build a piano with a potted flower.
Layer 1: quartz_pillar: [(1,1), (2,1), (3,1)], smooth_quartz_stairs: [(1,2), (2,2), (3,2)].
Layer 2: ...
Output: Here's a piano with a potted flower…
The blueprint matrix is:
[
[[1, 2], ..., [1, 2]],
...,
[[5, -1], ..., [6, -1]]
]
Spatial Reasoning
Question:
Which option is the same as the original image, aside from its orientation?
Answer: D
A
B
C
D
Original Image
Spatial Commonsense
Question:
Where should I go if my bedroom is on the second floor?
Answer:
You should go up the stairs on the right front to reach your bedroom.

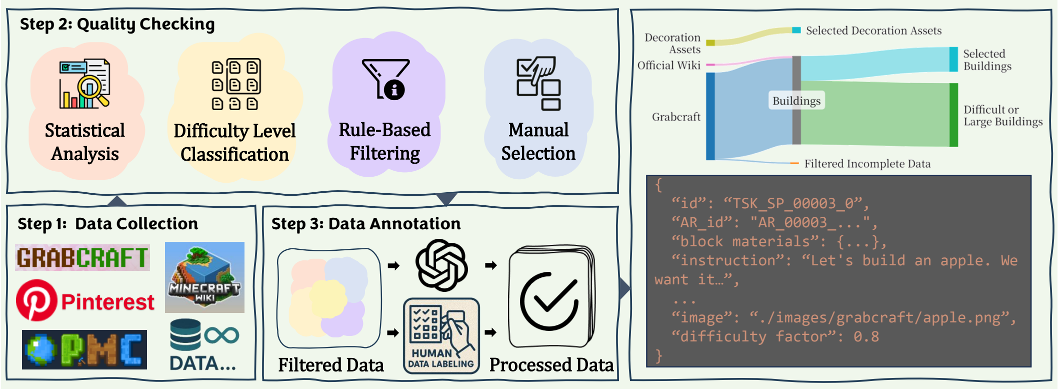
Figure 2: Task examples of **MineAnyBuild**. We present five task examples with specific inputs (questions) and outputs (answers). Some of them are simplified to illustrate the core presentations.

Executable Spatial Plan Generation. To construct an architecture, an agent first needs to design the layout of the architecture and accordingly generate the executable spatial plans based on its spatial perception, spatial understanding, and abundant knowledge. Based on this motivation, we propose the Executable Spatial Plan Generation task, which evaluates agents' abilities to perform Spatial Planning. The task input is an abstract architecture building instruction accompanied by precise explanations. Under the given task input, the agents are required to think on the decomposition of architecture substructures and corresponding connections to generate executable spatial plans for architecture building, just like completing a jigsaw puzzle.

For example, in this task instruction "*Build an apple...The apple also needs to have a stem, which we can use black_terracotta to make it.*" for architecture building, we lead the agent to think on the *stem* substructure in *apple* architecture and how to connect it with other substructures. If the agent could understand and plan in spatial perspective, the result should be better than planning in an abstract perspective. For the instruction regarding the indoor decoration, the agents are challenged to make more delicate and exquisite design and planning. More details of the Executable Spatial Plan Generation task are provided in the Supplementary Material.

Spatial Understanding. Inspired by the popular instruction following tasks [16, 17, 18] which are widely developed for evaluating MLLM-based agents, we introduce a Spatial Understanding task, where the agent needs to build the architecture according to the step-by-step instruction containing the positions of each building block through abstract spatial understanding. Specifically, we label the parts of our data with ground-truth annotations and generate the instructions with a mapping table of relative coordinate corresponding to the pivot position, for instance, *Layer 2: "red_wool": [(0,0),(1,0)]...*, where the block types and relative positions are provided. The agents are required to translate it into a blueprint matrix, which reflects the cognitive transition and integration of relative and holistic spatial understanding, simulating human-like cognitive mapping mechanisms that dynamically balance egocentric (body-centered) and allocentric (world-centered) perspectives.

Creativity. Architecture constructing and indoor decoration designing are attracting evaluation tasks than the previous tasks [12, 19, 20, 21, 22], like textual reasoning and coding. In our MineAnyBuild, we introduce a novel Creativity task for evaluating architecture building, where agents receive an instruction and are required to brainstorm block combinations for different parts of the architecture and outline a rough structure layout, to find ways to maximize creativity and the dynamic range of possible builds.



```
{
  “id”: “TSK_SP_00003_0”,
  “AR_id”: "AR_00003_...",
  “block materials”: {...},
  “instruction”: “Let's build an apple. We
want it…”,
  ...
  “image”: “./images/grabcraft/apple.png”,
  “difficulty factor”: 0.8
}
```

Figure 3: Data curation pipeline of **MineAnyBuild**. We conduct three core steps to curate our datasets: data collection, quality checking, and data annotation. On the right side, a Sankey diagram showing our data processing flow is presented, along with an example of simplified format of processed data.

It is worth elaborating that although this evaluation dimension is different from other traditional ones, creativity is an aesthetic and humanistic criterion that is more in line with human intuition and consensus, akin to the aesthetic assessment of image generation tasks. Creativity is a crucial component of scientific thinking that reflects the process of new things replacing old ones, which is precisely the spirit that scientific researchers have always pursued. Therefore, creativity not only reflects the cognitive depth of future AI systems, but also emerges as a novel and important criterion for agents towards Artificial General Intelligence (AGI). We test creativity through state-of-the-art MLLM-based critic models and human evaluations. Despite the challenges in standardizing evaluation criteria and achieving inter-rater reliability, the majority of recognized favorable comments or votes indicate that agents exhibit measurable creativity.

Spatial Reasoning. Spatial reasoning is the ability to imagine, visualize and differentiate objects in 3D space [5, 23, 24]. Inspired by the classic experiments in psychology named *mental rotation* [25, 15, 26], we construct 48 geometric objects made of blocks, denoted as *stimuli*, and generated 1,900 tasks for evaluation of spatial reasoning. As shown in Figure 2, for example, the agent needs to analyze the geometric structure of the given stimuli and determine whether others are the same or not as the given one. We conduct these tasks with Visual Question-Answer pairs, which is easier and more accurate for evaluation.

Spatial Commonsense. Spatial commonsense is the critical intuitive comprehension in our daily life that humans possess about the spatial attributes of objects in the physical world, including location, orientation, distance, shape, *etc.* [27, 28, 29]. Spatial commonsense is generally reflected in several aspects: 1) navigation and sense of direction: human can roughly orient the bedroom without a map. 2) rationality of object placement: a refrigerator cannot be placed in a bathroom. We evaluate these tasks on agents and we place the complete commonsense tasks tested in the Supplementary Material.

2.3 Data Curation

MineAnyBuild is a comprehensive benchmark with diverse architectures and indoor decorations, aligned with various instructions and visual reference images. We build our benchmark based on the following steps: 1) data collection, 2) quality checking, and 3) data annotation. Figure 3 presents our data curation pipeline for constructing MineAnyBuild.

Data Collection. Benefiting from the abundant and creative player-generated content on the Internet, we first collect ∼7000 architectures from several websites, e.g., GrabCraft [30] and Minecraft Official Wiki [31, 32], and collect ∼500 indoor decoration assets from sharing platforms by Minecraft creators. For some player-uploaded data containing potential issues, we filter out these problematic data with some quality standards. For spatial reasoning tasks, we collect 48 stimuli referencing the classic mental rotation experiments [14] and generate three groups of chiral stimuli symmetrical about the X/Y/Z coordinate axes. We design three major types of questions to construct VQA data for the spatial reasoning task based on these generated stimuli. These questions involve having agents select the only one among the four options that differs from or is the same as the stimulus in reference image, or to determine whether the stimuli in the two images are consistent. We utilize the data of large-scale buildings incorporating interior decorations to generate VQA pairs that complies with spatial commonsense, and further request agents to plan how to construct these decoration assets.

Quality Checking and Data Annotation. We implement some codes to filter the problematic data, and then we conduct a human review process to maintain high quality for data annotation. We annotate the instructions of tasks by human or state-of-the-art MLLMs. Specifically, we first carefully design some instructions that guide the agents to think about the decomposition and construction of the architectures, thereby more closely aligning with the motivation of spatial planning. For spatial commonsense tasks, we manually design the VQA pairs that well-fit with questions in the real world.

Infinitely Expandable Paradigm. As shown in Figure 3, we provide an infinitely expandable paradigm for data curation, facilitating the subsequent development of training and evaluation resources to advance AI agents research for spatial planning. Through our infinitely expandable paradigm, we can collect the majority of the existing player-generated content on the Internet and import it into the Minecraft game. Specifically, we manually mark the starting block (the minimum values on the X/Y/Z coordinates) and the ending block (the maximum values on the X/Y/Z coordinates) of the 3D coordinates as the three-dimensional coordinate box of the entire building, and obtain all the block information corresponding to each position through *mineflayer* simulator [33]. After filtering the *"air"* blocks, corresponding *three_d_info*, *blueprint* and *block_materials* can be acquired, with which we can generate this building by calling high-level commands in a blank Minecraft environment and obtain the corresponding visual images through manual screenshot for MLLM or manual annotation. The data that finally generate follows the requirements of our datasheet in terms of format, ultimately ensuring that all data has its unified format.

3 Experiments

In this section, we describe the agents evaluated on MineAnyBuild and corresponding evaluation metrics, followed by the results and analysis of performance of agents on MineAnyBuild.

3.1 Agents

We mainly conduct our evaluation on MLLM-based agents that suitable to address the spatial planning task in our benchmark. To adapt MLLM-based agents to our spatial planning task, we ask the agents to directly output the executable blueprint matrices. Then, the matrices are subsequently utilized by *mineflayer* simulator [33] to automatically generate corresponding architectures in Minecraft environment. We evaluate 13 MLLMs for our MineAnyBuild. For proprietary models, we evaluate Claude-3.5-Sonnet, Claude-3.7-Sonnet [34], Gemini-1.5-Flash, Gemini-1.5-Pro, Gemini-2.0-Flash [7], GPT-4o, GPT-4o-mini [35]. For open-source models, we evaluate InternVL2.5-[2B/4B/8B] [36], Qwen2.5VL-[3B/7B] [37], LLava-Onevision-7B [38].

All evaluations are conducted in a zero-shot manner for a fair comparison. We also provide RL-based agents for future research and adaptation. We place the detailed information of them and compute resources for agents in the Supplementary Material.

3.2 Evaluation Metrics

We introduce diverse metrics for evaluating different tasks in our benchmark. For the Executable Spatial Plan Generation, Creativity, and Spatial Commonsense tasks, as their results do not have a definitely correct or perfect answer, we use the state-of-the-art MLLM (GPT-4.1 [39]) as the critic model to score the planning. Specifically, we query GPT-4.1 to score separately based on different evaluation sub-dimensions and calculate a weighted "Evaluation Score". For different tasks, we obtain a comprehensive score based on the score, denoted as **"Score" (out of 10)** shown in Table 1, through corresponding weighting to indicate the performance of agents in each task. For some cases where the plans generated by agents are not executable, we directly set the scores of these cases to 0, showing that agents have failed in these cases. For the Spatial Reasoning task, we directly calculate the **Accuracy(%)** of the agent's responses as our results. More details about the evaluation metrics and the weighted formulas of scores are given in the Supplementary Material.

Table 1: Evaluation results of AI agents on **MineAnyBuild**. Gray indicates the best performance of each evaluation dimension among all agents and Light Gray indicates the second best results. We also highlight the top three agents based on their overall performance with Dark Orange, Orange, Light Orange, respectively.

Models	Executable Spatial Plan Generation	Spatial Understanding	Spatial Reasoning	Creativity	Spatial Commonsense	Overall
	Score ↑	Score ↑	Accuracy ↑	Score ↑	Score ↑	
Proprietary						
Claude-3.5-Sonnet	3.21	4.63	19.8	3.24	6.90	39.92
Claude-3.7-Sonnet	3.48	5.07	17.6	3.10	6.94	40.70
Gemini-1.5-Flash	2.87	2.49	25.8	2.71	7.12	35.54
Gemini-1.5-Pro	3.53	4.80	16.9	2.73	7.52	40.54
Gemini-2.0-Flash	2.63	4.19	16.0	2.44	6.82	35.36
GPT-4o	3.27	4.75	24.4	2.73	7.32	41.02
GPT-4o-mini	2.08	2.52	26.7	2.38	7.14	33.58
Open-source						
InternVL2.5-2B	0.24	0.34	19.8	0.28	4.94	15.56
InternVL2.5-4B	0.32	0.42	20.0	0.63	5.66	18.06
InternVL2.5-8B	0.68	0.62	20.4	0.66	5.62	19.24
Qwen2.5VL-3B	0.63	0.61	17.0	0.54	5.46	17.88
Qwen2.5VL-7B	1.29	1.12	16.0	1.34	6.30	23.30
LLava-Onevision-7B	0.73	0.92	19.6	0.98	5.54	20.26

3.3 Results Analyses

We include evaluation results of tested agents and Output Success Rate (OSR) in Table 1 and Figure 4, respectively. We also provide some output results and failure cases in Figure 5 for specific analyses.

Task Performance Results. We evaluate 13 MLLM-based agents on our MineAnyBuild benchmark, including 7 proprietary models and 6 open-source models. From Table 1, we can see that for most proprietary models, the performances are much better than those of open-source models. However, these proprietary models still perform relatively poorly in terms of the average absolute scores, e.g., even the GPT-4o with the highest overall score of 41.02 achieves less than half of the full score of 100. We analyze the task-specific findings as follows:

(1) **Executable Spatial Plan Generation**: For some low-parameter MLLM-based agents (e.g. InternVL2.5-2B/4B), they tend to understand the basic elements in the given instruction or image, and offer a simple or detailed plan for constructing. However, they often encounter difficulties when generating the executable spatial plan and cannot convert their planning into an executable 3D matrix, thus leading to scores under 0.4 as shown in Table 1. For some large-parameter models, they can usually understand the block materials partly compared to low-parameter ones, and can actively select some diverse blocks to build structures. Nevertheless, they often fail to understand the correlations between various combinations of block materials, resulting in a faulty completion of the final building and thus poor scores (from 0.63 to 1.29 in Table 1) by critic model. Proprietary models often achieve a relatively good balance in this aspect, but their planning is generally limited to a boxy or conservative design, and therefore their creations are not highly appreciated by the critic model with the average score of 3.01 ultimately.

(2) **Creativity**: Most proprietary MLLM-based agents can leverage their imagination to construct relatively novel designs, but their 3D architectural capabilities are weak, leading to the difficulty in outputting the executable plans corresponding to their planning and design. Conversely, open-source MLLMs receive lower scores frequently due to their invalid output results rather than the creative plans they generate, yielding a maximum score of 1.34 as quantified in Table 1.

(3) **Spatial Understanding**: In Table 1, the majority of proprietary models achieve solid results, while Gemini-1.5-Flash frequently generates matrices with more than three dimensions resulting execution errors, which suggests a limited grasp of structural understanding. For open-source models, they struggle with the building structures and mainly respond with repeated or increasing matrix results shown in Figure 5, which points to unclear interpretations of task goals.

(4) **Spatial Reasoning**: Spatial reasoning tasks, i.e., mental rotation experiments, require agents to simulate how humans' brain recognizes and moves the stimuli by rotating 3D objects in the mental representation based on the reference stimulus. The distractors are generally mirror-reversed geometries of the stimuli with extra rotations to increase task difficulty. From Table 1, we can observe that most MLLM-based agents perform poorly on this task, where even the top-performing model,

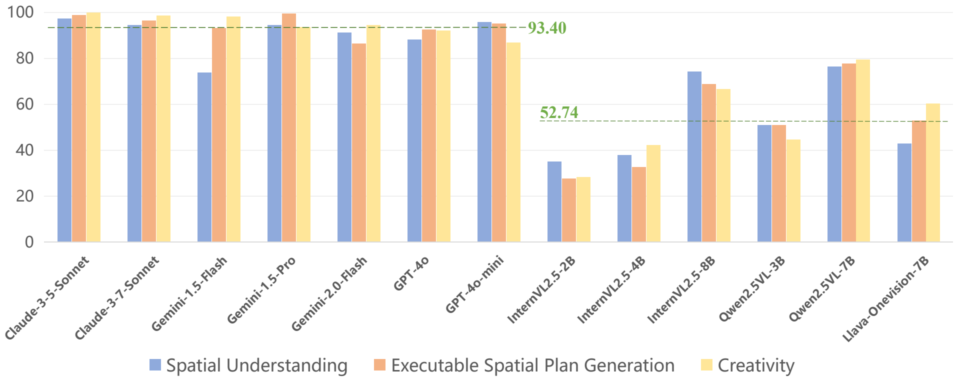


Figure 4: Bar chart of Output Success Rate (OSR) for MLLMs. Two green dotted lines indicate the average OSR of proprietary models and open-source models, respectively.

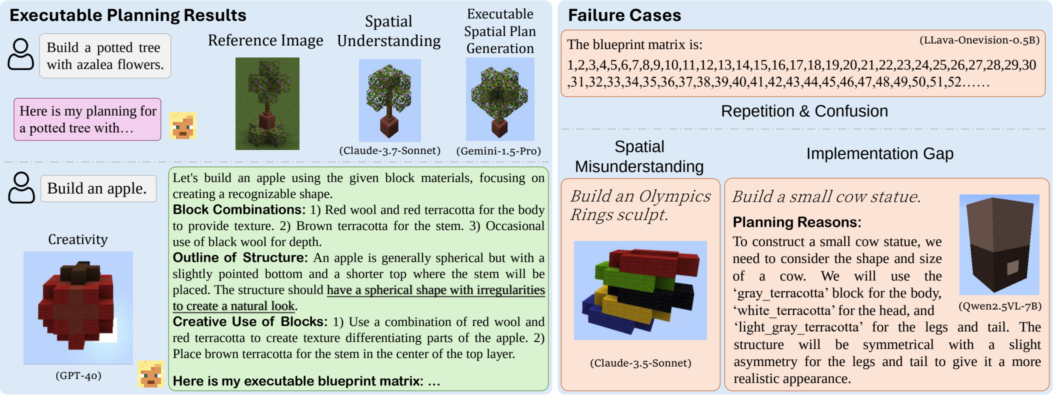
GPT-4o-mini, obtains merely 26.7% accuracy. Some models that are more capable in general AI tasks, e.g. GPT-4o, perform worse than GPT-4o-mini, which indicates that our spatial reasoning task still remains challenging for most MLLM-based agents.

(5) **Spatial Commonsense**: We evaluated these agents on spatial commonsense tasks relevant to humans' daily life, including the rationality of location, *etc*. The results in Table 1 reveal that most proprietary models have abundant spatial commonsense, achieving comparable responses to human-annotated answers. Open-source MLLM-based agents also show competent performances though with marginally lower consistency scores.

Output Success Rate. We statistically calculate the proportion of these MLLM-based agents that successfully respond with executable plans. In the Figure 4, we can observe that for the majority of proprietary models, their instruction-following capability and comprehension of 3D data are relatively strong, thus they can generate the corresponding executable blueprints according to their spatial planning. Gemini-1.5-Flash scores 73.81 on OSR which is below the mean line of 93.40 due to its incorrect understanding of the output dimension of the executable plan. The average line of open-source MLLM-based agents is quite lower than that of proprietary models, revealing that these agents are only effective for basic visual or textual understanding, while further training is still required for these 3D spatial tasks, such as spatial planning. Full metrics of Figure 4 are provided in the Supplementary Material.

Planning Output Visualization. We visualize some planning results in Figure 5. We can find that for some relatively easier tasks, most agents with strong capabilities can achieve great performance similar to the structure in the reference image. For example, as shown in Figure 5, agents are required to *build a potted tree with azalea flowers*, and Claude-3.7-Sonnet and Gemini-1.5-Pro show effective results under the tasks of spatial understanding and executable spatial plan generation, respectively. For the creativity task, the agent accessing GPT-4o can analyze and plan what blocks should be utilized and in what form to combine sub-structures into an integral whole. Moreover, the agent tend to consider how to increase the diversity and creativity of the overall structure and appearance, revealing its capabilities of spatial intelligence. More visualization results of all tasks are provided in the Supplementary Material.

Failure Cases Analysis. We provide some failure cases in Figure 5. We can observe that there are several causes of failure, which leads to agents being unable to generate the executable results based on their planning. For some low-parameter open-source MLLM-based agents, they have difficulty handling the 3D executable structures well, generating repetitive or confusing blueprints, leading to compilation failure. Some powerful proprietary models can understand some requirements and build the substructures, but there is still a spatial misunderstanding in their planning of combining them. For example, Claude-3.5-Sonnet wrongly overlaps the five rings of the Olympics Rings instead of laying them flat on the same plane, which is not in line with commonsense. For most MLLM-based agents, there is usually a severe implementation gap, i.e., they can not convert their textural planning into a spatial structure, which is precisely their huge defect in spatial planning. More visualization results of failure cases are provided in the Supplementary Material. Moreover, we provide deep analyses of the failure reasons for the cases with a summarization as follows:



Build a potted tree with azalea flowers.

Here is my planning for a potted tree with…

Build an apple.

Let's build an apple using the given block materials, focusing on creating a recognizable shape.
Block Combinations: 1) Red wool and red terracotta for the body to provide texture. 2) Brown terracotta for the stem. 3) Occasional use of black wool for depth.
Outline of Structure: An apple is generally spherical but with a slightly pointed bottom and a shorter top where the stem will be placed. The structure should have a spherical shape with irregularities to create a natural look.
Creative Use of Blocks: 1) Use a combination of red wool and red terracotta to create texture differentiating parts of the apple. 2) Place brown terracotta for the stem in the center of the top layer.

Here is my executable blueprint matrix: …

The blueprint matrix is:
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30
,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52……

Build an Olympics Rings sculpt.

Build a small cow statue.

Planning Reasons:
To construct a small cow statue, we need to consider the shape and size of a cow. We will use the 'gray_terracotta' block for the body, 'white_terracotta' for the head, and 'light_gray_terracotta' for the legs and tail. The structure will be symmetrical with a slight asymmetry for the legs and tail to give it a more realistic appearance.

Figure 5: Visualization of executable planning results (left) and failure cases (right).

(1) Spatial Misunderstanding: Agents frequently misinterpret 3D positional relationships or fail to maintain the correct spatial arrangements, which highlights a persistent weakness in spatial grounding and planning.

(2) Implementation Gap: The agents have a central issue that they can not transform high-level textual plans into precise and executable blueprint matrices. The integration of substructures often fails due to incorrect block indexing, orientation errors or inconsistent spatial logic, leading to blueprint parsing or execution failures. This is essentially because the model's understanding of data structures such as codes or DSL and 3D matrices from a numerical perspective is still limited. If these models strengthen the training of spatial data, it may enhance the capabilities of these agents.

(3) Structural Degeneration under Complexity: When the tasks demand non-cubic, asymmetric or creative designs, the agents tend to collapse into simple and box-like outputs or disorganized results. This indicates that their limited ability to scale from basic patterns to more abstract and complex architectural concepts.

These failure modes reflect deeper limitations in MLLM's capabilities to perform hierarchical spatial planning, maintain geometric consistency and ground language into manipulable 3D structures. They also provide more research directions for MLLMs, e.g., to improve multi-modal spatial understanding, align linguistic abstraction with executable plans or enhance agent's ability for structural composition in open-ended 3D environments.

4 Related Works

Spatial Intelligence. Spatial intelligence involves thinking about the shapes and arrangements of objects in space and about spatial processes, such as the deformation of objects, and the movement of objects and other entities through space. Current works mainly focus on spatial understanding and spatial reasoning [8, 9, 10, 11, 40, 41, 42, 43, 44, 45]. VSI-Bench [8] first introduces the definition of visual-spatial intelligence and proposes a benchmark for it. SpatialVLM [9] presents an automatic framework generating millions of VQA samples of spatial reasoning for VLMs' evaluation. Lego-Puzzles [11] introduces a scalable benchmark with several VQA samples including tasks in multi-step spatial reasoning. However, these benchmarks suffer from the gap between abstract spatial understanding and concrete task execution. In this paper, we introduce an innovative benchmark concentrating on spatial planning, where the open-world AI agents need to generate executable spatial plans based on its spatial perception and cognition for architecture and indoor decorations. We also introduce diverse evaluation dimensions such as creativity and spatial commonsense to realize a comprehensive assessment for spatial planning capabilities.

Minecraft for AI Research. Minecraft is a 3D world sandbox video game with diverse game mechanics supporting various tasks and activities. Benefiting from its open-ended property, the training and evaluation of autonomous agents built on Minecraft are quite inspiring for the research in the field of artificial intelligence and embodied AI. There are several related works [12, 46, 47, 13, 48, 49, 50, 51, 52] contributing to the development in recent years. VPT [47] utilizes Youtube videos for agents' large-scale pretraining. MineDojo [12] features a massive database collected automatically from the Internet and learns a MineCLIP model by watching thousands of Youtube videos. Voyager [13] imitate behavior by pseudo-labeling actions by plugging GPT-4 while Optimus-2 [49] learned a VLA-based model with MLLMs for high-level planning. These works are merely confined to traditional embodied planning tasks like skill learning or tech-tree goals. Compared to these works, we propose a new benchmark MineAnyBuild to evaluate AI agents in spatial intelligence, which is an emerging research field regarding the ability of AI agents to reason about 3D space.

5 Conclusion

We introduce MineAnyBuild, an innovative benchmark designed to evaluate spatial planning for open-world AI agents. Our MineAnyBuild consists of 4,000 curated tasks with 500+ buildings and decoration assets for evaluating spatial planning, and approximately 2,000 VQA pairs for spatial reasoning and commonsense evaluation. Extensive experiments on 13 advanced MLLM-based agents reflects that there is still a great growth space for spatial intelligence of agents. We believe that our MineAnyBuild will pioneer a novel paradigm to evaluate spatial intelligence, while advancing the development of open-world AI agents with spatial planning capabilities.

6 Acknowledgments

This work is supported by National Key Research and Development Program of China (2024YFE0203100), National Natural Science Foundation of China (NSFC) under Grants No.62476293, National Postdoctoral Program for Innovative Talents under Grant Number BX20250379, China Postdoctoral Science Foundation under Grant Number 2025M771521, and General Embodied AI Center of Sun Yat-sen University.

References

[1] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. *arXiv preprint arXiv:2411.16537*, 2024.

[2] Arijit Ray, Jiafei Duan, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, Kuo-Hao Zeng, et al. Sat: Spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2024.

[3] Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. *arXiv preprint arXiv:2412.04383*, 2024.

[4] Yue Zhang, Zhiyang Xu, Ying Shen, Parisa Kordjamshidi, and Lifu Huang. Spartun3d: Situated spatial understanding of 3d world in large language models. *arXiv preprint arXiv:2410.03878*, 2024.

[5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[6] OpenAI. Gpt-4v(ision) system card, 2023.

[7] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[8] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. *arXiv preprint arXiv:2412.14171*, 2024.

[9] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

[10] Xianda Guo, Ruijun Zhang, Yiqun Duan, Yuhang He, Chenming Zhang, Shuai Liu, and Long Chen. Drivemllm: A benchmark for spatial understanding with multimodal large language models in autonomous driving. *arXiv preprint arXiv:2411.13112*, 2024.

[11] Kexian Tang, Junyao Gao, Yanhong Zeng, Haodong Duan, Yanan Sun, Zhening Xing, Wenran Liu, Kaifeng Lyu, and Kai Chen. Lego-puzzles: How good are mllms at multi-step spatial reasoning? *arXiv preprint arXiv:2503.19990*, 2025.

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

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

[14] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

[15] Shenna Shepard and Douglas Metzler. Mental rotation: effects of dimensionality of objects and type of task. *Journal of experimental psychology: Human perception and performance*, 14(1):3, 1988.

[16] Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. *arXiv preprint arXiv:2310.07641*, 2023.

[17] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

[18] Renze Lou, Kai Zhang, and Wenpeng Yin. Large language model instruction following: A survey of progresses and challenges. *Computational Linguistics*, 50(3):1053–1095, 2024.

[19] Shiying Hu, Zengrong Huang, Chengpeng Hu, and Jialin Liu. 3d building generation in minecraft via large language models. In *2024 IEEE Conference on Games (CoG)*, pages 1–4. IEEE, 2024.

[20] Matthew Barthet, Antonios Liapis, and Georgios N Yannakakis. Open-ended evolution for minecraft building generation. *IEEE Transactions on Games*, 15(4):603–612, 2022.

[21] Jun Yu Chen and Tao Gao. Apt: Architectural planning and text-to-blueprint construction using large language models for open-world agents. *arXiv preprint arXiv:2411.17255*, 2024.

[22] Sam Earle, Filippos Kokkinos, Yuhe Nie, Julian Togelius, and Roberta Raileanu. Dreamcraft: Text-guided generation of functional 3d environments in minecraft. In *Proceedings of the 19th International Conference on the Foundations of Digital Games*, pages 1–15, 2024.

[23] Ruth MJ Byrne and Philip N Johnson-Laird. Spatial reasoning. *Journal of memory and language*, 28(5):564–575, 1989.

[24] Douglas H Clements and Michael T Battista. Geometry and spatial reasoning. *Handbook of research on mathematics teaching and learning: A project of the National Council of Teachers of Mathematics*, pages 420–464, 1992.

[25] Mary Hegarty. Components of spatial intelligence. In *Psychology of learning and motivation*, volume 52, pages 265–297. Elsevier, 2010.

[26] Steven G Vandenberg and Allan R Kuse. Mental rotations, a group test of three-dimensional spatial visualization. *Perceptual and motor skills*, 47(2):599–604, 1978.

[27] Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015.

[28] Xiao Liu, Da Yin, Yansong Feng, and Dongyan Zhao. Things not written in text: Exploring spatial commonsense from visual signals. *arXiv preprint arXiv:2203.08075*, 2022.

[29] Guillem Collell, Luc Van Gool, and Marie-Francine Moens. Acquiring common sense spatial knowledge through implicit spatial templates. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[30] GrabCraft LLC. Grabcraft - the biggest library of minecraft objects, models, floor plans, ideas, and blueprints. `https://www.grabcraft.com/`.

[31] Citricsquid. Minecraft wiki. `https://minecraft.wiki/`.

[32] Fandom. Fandom wiki for minecraft. `https://minecraft.fandom.com/wiki/Minecraft_Wiki`.

[33] PrismarineJS. Mineflayer. `https://github.com/PrismarineJS/mineflayer`.

[34] Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024.

[35] OpenAI:Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, FlorenciaLeoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, HyungWon Chung, Dave Cummings, and Jeremiah Currier. Gpt-4 technical report. Dec 2023.

[36] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[37] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[38] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[39] OpenAI. Introducing gpt-4.1 in the api. `https://openai.com/index/gpt-4-1/`, 2025.

[40] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision language models. *arXiv preprint arXiv:2406.01584*, 2024.

[41] Yun Li, Yiming Zhang, Tao Lin, XiangRui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. Sti-bench: Are mllms ready for precise spatial-temporal world understanding? *arXiv preprint arXiv:2503.23765*, 2025.

[42] Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. *arXiv preprint arXiv:2406.05756*, 2024.

[43] Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. *arXiv preprint arXiv:2412.18194*, 2024.

[44] Yuecheng Liu, Dafeng Chi, Shiguang Wu, Zhanguang Zhang, Yaochen Hu, Lingfeng Zhang, Yingxue Zhang, Shuang Wu, Tongtong Cao, Guowei Huang, et al. Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning. *arXiv preprint arXiv:2501.10074*, 2025.

[45] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[46] Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. Mineworld: a real-time and open-source interactive world model on minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

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

[48] Zaijing Li, Yuquan Xie, Rui Shao, Gongwei Chen, Dongmei Jiang, and Liqiang Nie. Optimus-1: Hybrid multimodal memory empowered agents excel in long-horizon tasks. *arXiv preprint arXiv:2408.03615*, 2024.

[49] Zaijing Li, Yuquan Xie, Rui Shao, Gongwei Chen, Dongmei Jiang, and Liqiang Nie. Optimus-2: Multimodal minecraft agent with goal-observation-action conditioned policy. *arXiv preprint arXiv:2502.19902*, 2025.

[50] Shaofei Cai, Zhancun Mu, Anji Liu, and Yitao Liang. Rocket-2: Steering visuomotor policy via cross-view goal alignment. *arXiv preprint arXiv:2503.02505*, 2025.

[51] Qian Long, Zhi Li, Ran Gong, Ying Nian Wu, Demetri Terzopoulos, and Xiaofeng Gao. Teamcraft: A benchmark for multi-modal multi-agent systems in minecraft. *arXiv preprint arXiv:2412.05255*, 2024.

[52] Shaofei Cai, Zihao Wang, Kewei Lian, Zhancun Mu, Xiaojian Ma, Anji Liu, and Yitao Liang. Rocket-1: Mastering open-world interaction with visual-temporal context prompting. *arXiv preprint arXiv:2410.17856*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state our contributions and problem scope in Section 1 of our paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss our limitations in Section. A of the Supplementary Material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper does not include theoretical results such as theory assumptions and proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the information, code and datasets in Section D of the Supplementary Material for reproduction. We will also make our datasets and codes public in the future.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our paper in this submission track provides Dataset URL and Code URL in order to facilitate the review process. We will make our datasets and codes public with sufficient instructions in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide all the experimental details in Section 3 of our paper and additional results in Section F of the Supplementary Material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide some analyses on experimental results, such as scoring by critic models, in Section F of the Supplementary Material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the information of our experimental compute resources in Section F of the Supplementary Material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conducts in every respect with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts of our work in Section A of the Supplementary Material.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We conduct manual reviews on our data obtained from the Internet from containing unsafe images and curate our data for safety as safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and models are properly cited in our main text and our Supplementary Material.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the information and document of our datasets in Section D of the Supplementary Material.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We provide the declaration of our LLM usage in Section H in our Supplementary Material. However, the usage of LLMs in our work does not impact the scientific rigorousness and originality of the research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.