Spatial Awareness in LLMs

Siddarth Madala *smadala2@illinois.edu*
smadala2

Risham Sidhu *rsidhu3@illinois.edu*
rsidhu3

Tianjiao Yu *ty41@illinois.edu*
ty41

Reviewed on OpenReview: ..

Abstract

We seek to survey the spatial awareness capabilities of Large Language Models (LLMs), as well as related tasks/benchmarks that measure their performance in embodied reasoning/planning.

1 Introduction

The emergence of Large Language Models (LLMs) has revolutionized artificial intelligence, demonstrating remarkable proficiency across diverse tasks, from natural language understanding to code generation and scientific reasoning. Initially confined to processing textual data, these models have rapidly evolved to integrate multimodal capabilities, enabling them to handle images, audio, and even video. As they continue to expand their reach, a critical challenge arises: Can LLMs effectively interpret, understand, and navigate three-dimensional (3D) environments?

Spatial reasoning is fundamental to many real-world applications, including robotics, autonomous navigation, augmented reality, and geospatial analysis. Unlike text, which follows a linear structure, spatial information is inherently multidimensional, requiring models to process relationships between objects, depth, orientation, and movement. While LLMs can describe spatial concepts linguistically, their training primarily relies on textual data, limiting their ability to develop an intrinsic understanding of physical space, geometry, and embodied experience. This survey aims to examine the current capabilities, limitations, and potential solutions for enabling LLMs to reason about 3D space, investigating the spatial awareness of LLMs and their mechanisms for spatial reasoning.

1.1 Motivation

Understanding how Large Language Models (LLMs) conceptualize space is crucial for optimizing their application across various domains. By identifying their weaknesses in spatial reasoning — such as difficulties with 3D navigation, object permanence, and relative positioning — we can enhance their performance through additional training, multimodal learning, or specialized reasoning modules. At the same time, recognizing their strengths, such as their ability to infer spatial relationships from linguistic cues or generate structured navigation plans, allows us to frame problems in ways that align with their capabilities.

Beyond spatial reasoning, this principle extends to broader emergent abilities in LLMs, including logical inference, commonsense reasoning, and adaptability to novel tasks. For example, while LLMs exhibit strong performance in abstract reasoning tasks, they struggle with tasks requiring grounded perception or real-world interaction.

Additionally, a comprehensive understanding of LLMs' cognitive architecture — not just in space but in temporal reasoning, social awareness in multi-agent environment, and embodied cognition — will be key to

unlocking their full potential in domains ranging from robotics to augmented reality, scientific discovery, and autonomous systems.

2 Spatial Understanding

First, we will cover existing works that explore spatial understanding in LLMs.

2.1 Text-Only Understanding

Text-only understanding has become less popular with the advent of multimodal LLMs that are able to incorporate other modalities like images or videos. However, it provides a solid foundation for understanding what spatial relations can be acquired by LLMs during their extensive text-only pretraining processes.

At first, it may not be clear how only training on text can produce a model that understands modalities outside of it. However, existing work has shown that models are able to learn colors, cardinal directions, and state tracking to varying degrees of success Patel & Pavlick (2022); Abdou et al. (2021). Liu et al. (2022) even show that some degree of commonsense understanding of relative sizes and relations is available to language-only models. Yamada et al. (2024) explore this idea for navigating a 2D grid through word problem style QA. In their setup, they measure the performance of GPT and Llama on various types of grids. They find that models perform best on square grids, which they hypothesize is because of the prevalence of similar structures in tables, charts, graphs, and other human artefacts. For paths arranged around a ring or on a triangular grid, Llama 2 performance worse than random chance at correctly identifying vertices, and for paths arranged on a hexagonal grid, both GPT4 and Llama 2 perform worse than random chance. The performance of both models trails behind human performance by a significant margin.

Furthermore, they explore whether spatial information is better utilized by these models when presented in one cohesive global description or piece by piece as if the agent is acquiring the new information as it travels the grid. They find that the models perform better when the information is provided locally instead of globally. They perform several other experiments including varying the ordering of coordinates in the textual representation of the grid to see if it affects understanding (a standard row-by-row approach is most effective), an error analysis of incorrect answers (which shows that mistakes are often physically near the correct answer on the grid or temporally near the correct answer in the textual representation, suggesting limitations of the textual representation), and a code based approach to explore code as another possible representation (but it underperforms a purely textual representation). From this, we can establish that LLMs have some base level of spatial understanding that allows them to "visualize" simple 2D grids.

Similar to Yamada et al. (2024) initial work on "navigating" tree-like structures, Momennejad et al. (2023) evaluate cognitive maps, which are abstract representations of a task. For example, when navigating a maze or a set of rooms, a cognitive map would represent each possible location or room as a node and the adjacent locations or next rooms as joined by edge. They consider a variety of underlying task structures and evaulate the performance of 8 different LLMs on instructions that include finding a shortcut through the given map, choosing the next location to maximize the reward, or drawing the map from the given textual description. They found that all LLMs struggled with the presented tasks, though they sometimes succeeded on sparse graphs or smaller graphs. Generally, the models hallucinated edges between desirable nodes, took suboptimal paths through the graphs, or got stuck in loops. This suggests that rather than generalizing and understanding the general graph structure, they are succeeding through memorization, simple setups, or possible data contamination in their training.

Mirzaee et al. (2021) suggest a task less restricted to a grid or graph. The task verbally describes simple images of arranged shapes and asks the model to answer questions about their relative locations and sizes. However, they do not evaluate on LLMs. Preliminary exploration by the authors of this survey suggest that the most recent LLMs may be more than capable of spatially reasoning over tasks like Mirzaee et al. (2021); Yamada et al. (2024).

2.2 Multimodal LLM Understanding

While these text-only models are capable of some degree of spatial awareness, it seems clear that achieving more complex and realistic spatial understanding necessitates integrating another modality. In line with earlier exploration, VQA is often the preferred means of evaluating this ability . A variety of such benchmarks have been proposed Malinowski & Fritz (2015); Andreas et al. (2017); Johnson et al. (2016).

Datasets Azuma et al. (2022) focus on answering questions about scenes with full color photorealistic 3D scans. The answer to the natural language question is both a textual response and a bounding box in the scan information. On the other hand, Majumdar et al. (2024) present OpenEQA specifically for foundation models, integrating full color images with depth information and camera pose, as well as open scenes and open vocabulary evaluated by an LLM judge. To make the task more realistic and practical than simply answering questions about where an object is, they include aspects such as episodic memory that require reasoning over multiple images in a history. They find that even the highest scoring GPT-4V based models fall far short of human performance on these tasks.

Cheng et al. (2024) explore a similar setup for SpatialRGPT but focus on constructing scene graphs of 2D images to answer questions on, which allows for questions that cover both 2D and 3D images and explore depth perception. They include templatic questions that expect the model to understand the size of specific objects or the relative size and positions of multiple objects, as well as more complex questions that test the same concepts over more varied language. These Llama generated questions also ask the model to estimate the size and depth information in different units (such as the time required to traverse a distance rather than the distance itself or the number of people that can sit in space rather than the square footage), requiring some amount of reasoning about space. SpatialVLM

Chen et al. (2024) create a similar dataset, but at a larger scale that does not require human annotation. They labels their questions types as qualitative and quantitative, but the differences aligns fairly well with SpatialRGPT's broad categories of relative questions vs object specific ones. They do, however, create more complex "chain-of-thought" questions such as asking if 3 objects roughly form an equilateral triangle with their positions.

Q-Spatial Bench is another spatial reasoning/understanding dataset created by Liao et al. (2024) that takes the opposite approach. Rather than generating answers or evaluating with an LLM, they opt for a completely manual dataset focusing specific on object dimensions and relative distances. Li et al. (2024a) frame the problem of spatial understanding a task better suited to a top-down view and still find that VLMs perform far behind humans.

Training on Datasets The proposed model in ScanQA (Azuma et al. (2022)) is a standard highly engineered modular setup that prevailed before the advent of LLMs: it involves separate language and vision embeddings, a transformer layer, and 3 separate classifiers (two for the answer components and one for the question). Unlike the modular networks of Andreas et al. (2017) or the approach of ScanQA, approaches for LLMs tend to focus on supplementing the main LLM training with additional fine-tuning examples and trusting the LLM base to handle most of the processing after some initial visual step. SpatialRGPT Cheng et al. (2024) propose a frozen visual model that produces visual encodings and a trainable features extractor that operates over image regions to create embeddings for both the image and depth modalities, both of which simply plug into a trainable LLM. A model trained on this data outperforms other models, but no comparison to human answers is provided. Still, the model is able to reason over the complex question in the selected qualitative examples. Chen et al. (2024) also take this approach, using a ViT to process their images, before passing in embeddings and natural language to an LLM to process.

Hong et al. (2023) use the 3D pointcloud information instead of images as they offer more comprehensive views of the environment and the objects within it. Using 2D VLMs to label information in the environment and 3D feature extractors, they train various LLMs capable of reasoning over pointcloud information. By doing so, they improve performance on datasets like ScanQA. From qualitative results it seems that the model is capable of understanding the scene composition and some degree of relative descriptions.

2.3 Using LLM Spatial Understanding

We will also briefly explore methods that are used to improve LLMs grasps of space beyond simply fine-tuning on a spatial dataset. Ma et al. (2024) present SpatialPIN, which utilizes prompting to create text descriptions of images that help the model understand various aspects of an image. For example, and LLM can be prompted to describe a 2D image by listing the objects and locations and these additional text descriptions can be used to help the model create better segementation masks for each object. This can be repeated with other ways of breaking up an image (e.g. depth) to produce a cohesive 3D representation of the image. Zhao et al. (2023) use scene graphs while Gu et al. (2023) utilize concept graphs created by foundation model outputs to determine the relations between objects and potential uses/interactions with specific objects.

2.4 What do Models Understand?

Early VQA datasets focused on relatively simple questions like object recognition or relative placements of objects in 2D images. From the discussed datasets, we can see that the trend is now towards more complex setups. Rather than just 2D images, VQA is evaluated over depth information, scene graphs, camera poses, and even sets of images that can act separately or as still frames of a memory. Now that LLMs can be used to write less templatic questions, we see complex automated questions being created as well. With LLMs as judges, we also observe the evaluation of answers being given to LLMs to evaluate, reducing the burden of formatting information in a certain way. Finally, we also see that the question types have expanded. More and more datasets are exploring information grounded in the real-life qualities of the images, e.g. object dimensions and distances. The expectation now, after models seems to be able to locate and identify objects and some relative traits, is that models should be able to extract real world measurements from images accurately.

Liao et al. (2024) explore a zero-shot prompting setup for evaluating model understanding on their Q-Spatial Bench dataset. They found that some GPT-based models could accurately estimate the sizes of objects by using reference objects (nearby objects with dimensions that are predictable through commonsense/world knowledge) and then comparing the size of this reference object with the target. They found that doing so correlated with high accuracy and, thus, created a prompt that encouraged models to use existing objects as references, founding that even in a zero-shot setting that it dramatically improved model performance across various models. The question remains, however, if this truly indicates spatial understanding. Measurement based questions are easier to evaluate and thus automate evaluation for, but when we consider the use of reference objects and the reasoning done by the model, it's possible that what the model really understands is how large objects are supposed to be.

Liu et al. (2023) do a more thorough analysis of relative spatial relations and model behavior when predicting them from images. They consider if models are able to understand proximity, direction and orientation, and other potential relations. Their analysis provides insight into which relations are vague by nature and lead to human annotator disagreement (e.g. closeness is often subjective for human annotators and therefore may not be clear for LLMs) and also show results on VLMs. These results show that models struggle with many possible relations, e.g. orientations of objects are particularly difficult for models to grasp.

Fu et al. (2024) explore similar concepts of multimodal model understanding , posing tasks that are simple for humans but difficult for current leading VLMs because they do not use language as an intermediary step. They found that replacing images with dense text captions for existing VQA tasks still allowed for high model performance, suggesting that much of the prowess was coming from the language reasoning side of the model rather than understanding the visual modality. They propose a different styles of questions that require reasoning over multiple viewpoints, lighting conditions, or times, differentiating between real and generated images, evaluating potential bounding boxes, filling in missing image regions, counting objects, finding similar objects across different images, or choosing the most similar images in a set. They found that for these tasks most LLMs did barely better than random chance and even the strongest performing GPT based models only answer correctly around half of the time, trailing far behind humans who are almost always correct on such tasks. They also note that for some models, performance improves with a dense

caption as opposed to an image, possibly suggesting the the methods of integrating images into LLMs do not properly convey all the necessary information.

Overall, there is certainly more focus on understanding the spatial awareness of LLMs with visual components, which follows from their inherently more spatially grounded nature. However, the types of tasks that are explored for both settings are fairly similar in that they require understanding spatial relations between existing objects or moving around an environment. There seems to be a larger focus for text models to see what spatial information the model has acquired during pre-training whereas for many VLMs the expectation is to further fine-tune them to achieve better spatial reasoning. We feel that complementing both of these avenues of research by considering how text-only models might better understand spatial information and what pre-trained spatial understanding models with visual components have would lead to interesting insights. One potential future direction might be to experiment with new probing methods/measuring activations across neurons with LLMs.

Still, from the results of these papers and the applications of various models, it seems clear that LLMs, both with and without textual information, have some amount of comprehension of space, though it seems to be somewhat brittle and not as well internally represented as human perceptions of space. Furthermore, it may be possible that the advances we see in the area are due more to correlations in the training datasets than true reasoning over a spatial modality.

3 Games

Outside of well-defined Visual QA tasks, it can be difficult to create settings to effectively probe the spatial abilities of LLMs. One potential avenue that is both readily available and easy to label is games. Many games have a spatial element and thus approaches to solve them may lend insight into how to formulate certain spatial relations effectively for LLM understanding.

3.1 Chess

Chess is an area where we have seen fairly strong performance from LLMs, such as ChessGPT Feng et al. (2023) which seems to receive ELO scores on par with an average strong chess player. Notably, models like ChessGPT have high accuracies of converting between chess notation systems, which might indicate some understanding of the physical chessboard. Furthermore, ChessGPT can often predict moves that lead to checkmate, which might suggest mastery of both movement restrictions and some understanding of the physical positions of the pieces. More recent works, such as Zhang et al. (2025)'s ChessLLM, show even stronger scores across more varied tasks (e.g. predicting the best move for any game stage instead of just checkmate in one), by training on full games. Srivastava et al. (2023) explore chess data in their large multi-task dataset. They find that before models accurately predict checkmate in one move, they learn to produce valid/legal moves. While the latter is not necessarily a prerequisite for the former, a model that performs well in the former likely understands the rules and may understand the physical space of the board in some abstract sense. Further, analysis of the moves predicted by models show that a majority of moves are valid and occupy expected spaces, but some involve moving incorrectly for that particular piece or ignore pieces that are in the path of travel, essentially blocking the move. So, whatever understanding the models do have of space, it is far from perfect or equivalent to an image in a human mind.

Chess, thus provides us with two interesting potentials for LLM spatial awareness. Firstly, existing work with LLMs and VLMs may be extended into this domain as a means of understanding how the model is conceptualizing the game. We may explore VQA style queries over text-image pairs or text-only descriptions to provide insight into how the model is representing the game internally and if that includes a spatial component. Or, models created to solve chess games might give us some insight on how to set up grid-like structures to effectively convey space or movement restrictions to models. While much of their strong performances is likely due to these models learning the rules of chess within a narrow notational space, it might be interesting to refer to smaller grids with chess notations that combine numbers and letters to represent the pieces and locations rather than a standard 2D Cartesian plane. Using setups similar to this

that foundation models are likely already familiar with or as a potential source of augmenting an existing dataset may help familiarize the model with physical space.

3.2 ARC-AGI

The ARC-AGI (Chollet (2019))tasks presents another interesting way to set up physical spaces for an LLM. The tasks in this competition require deducing patterns in pairs of grids and then applying that pattern to a final unpaired grid to produce a the final answer. These patterns may range from moving shapes across the grid to augmenting or completing shapes or changing their colors. The grids are generally represented with pure text meant to resemble the actual grid. That is, a 3x3 grid with a long red rectangular block at the top and a small blue square in the bottom right corner may be represented as:

```
R R R
* * *
* * B
```

This might be a valid alternate to purely text based descriptions of grids, such as were seen in Yamada et al. (2024) or code based representations that are becoming more common. It would be interesting to compare such a representation to these existing ones, especially as humans may struggle to follow such a structure when it represents a 3D space, but structures that humans find to be adequate representations of 3D space are not always the easiest for models to parse.

4 Minecraft: A Practical Application

While large language models (LLMs) have shown impressive capabilities in abstract reasoning and symbolic manipulation, their ability to understand and operate within physical space—particularly in open-ended, partially observable environments—remains limited and poorly understood. To investigate this gap, we turn to Minecraft, a richly structured 3D environment where spatial understanding is not optional but essential. Tasks such as locating resources, navigating terrain, or constructing multi-block structures require agents to reason over egocentric observations, maintain spatial memory, and translate abstract instructions into grounded actions.

What makes Minecraft especially valuable for studying spatial reasoning in LLM-based agents is its combination of embodiment, language, and compositionality. Unlike synthetic spatial benchmarks with static layouts or tightly scoped tasks, Minecraft provides a dynamic, first-person world with a high degree of spatial and temporal complexity. Its native action space mirrors real-world interactions—turning, placing, crafting, traversing—while its use of natural language as a control modality enables the study of how spatial concepts are represented, grounded, and acted upon by language-driven agents. Prior work has explored a wide spectrum of agent architectures in this setting, ranging from low-level behavioral cloning models trained on human demonstrations (e.g., MineRLGuss et al. (2019), Steve-1Lifshitz et al. (2024)) to hierarchical agents that integrate LLM-based planning or goal generation (e.g., VoyagerWang et al. (2023), DECKARDNottingham et al. (2023), OdysseyLiu et al. (2024)). These systems offer a natural scaffold for probing how well LLMs can perceive spatial layouts, track relative positions, and reason over object configurations and environmental constraints.

In this section, we use Minecraft as a lens to examine the spatial capacities of LLM-based agents. Rather than presenting an exhaustive survey of methods, we focus on how spatial understanding emerges or fails to emerge across different stages of the agent pipeline: from interpreting spatial instructions to planning over spatially dependent goals, and maintaining consistent spatial memory during execution.

4.1 How Spatial Reasoning Manifests in Minecraft Agents

Spatial Goal Understanding A core strategy across recent work is to embed spatial goal understanding within a latent goal space, a modality-agnostic intermediate representation that links natural language instructions, visual observations, and action plans. For instance, Steve-1 Lifshitz et al. (2024) extends

the Video Pre-Training model with a latent variable z_T^{goal}, derived from a MineCLIP-encoded trajectory segment Fan et al. (2022), to condition low-level action policies on high-level textual intent. This latent goal embedding is injected into the ResNet-based visual encoder and subsequently passed to a Transformer-XL policy head, enabling goal-aware action generation. Here, goal understanding is implicitly captured through contrastive alignment between text and behavior over pre-collected gameplay videos, grounding spatial terms like "build," "place," or "navigate" in actual Minecraft trajectories.

However, this approach inherits the limitations of fixed-length embeddings and limited expressivity of the CLIP-based space. MineDreamer Zhou et al. (2024) pushes this further by leveraging a goal imagination module to predict future visual states (subgoal images) from current observations and textual instructions. It introduces learnable [GOAL] tokens into the LLM vocabulary and uses a Goal Q-Former transformer to produce a latent representation of the imagined future state. This latent is then fused with the agent's visual encoder output and passed through a latent diffusion model to synthesize visual representations of the goal. The result is a more grounded form of goal understanding—rather than encoding "build a house" as a vector, the agent generates a visual blueprint of what the house should look like, enhancing spatial consistency during planning and control.

Another compelling direction is the use of reference video as goal specification. The GROOT Cai et al. (2023) framework explores aligning natural language or video instructions with a learned goal space derived from gameplay demonstrations. While the original model used video clips as input to the encoder, a later variant replaces this with a fine-tuned text encoder (e.g., BERT), optimized through behavior cloning to match the pre-trained goal space. This creates a shared semantic space between video-based demonstrations and text instructions, allowing agents to execute unseen textual commands by mapping them onto known goal trajectories. Spatial goals such as "mine wood" or "build snow golem" are thus represented as latent codes that inform policy decoding. Building on this paradigm, GROOT-2Cai et al. (2024) further enhances spatial goal representation through a variational autoencoder (VAE) that encodes latent intentions from reference videos. This allows the agent to generalize from specific demonstrations to broader spatial goals, such as digging toward hidden resources or navigating toward unobserved landmarks. While latent spaces provide flexibility, they are also subject to ambiguities in alignment, particularly when the spatial referents are underspecified. GROOT-2 addresses this by leveraging a behavior tokenizer that discretizes trajectory segments into behavior tokens compatible with LLM reasoning, enabling fine-grained spatial control through language-conditioned latent variables

An important aspect that emerges across these works is the gap between abstract language and spatially grounded execution. Language instructions rarely specify absolute coordinates or detailed spatial plans; instead, they rely on qualitative terms ("near the tree", "on top of the house") or relational concepts ("next to", "behind", "to the left"). Bridging this gap requires agents to reason over partial observations, leverage spatial priors, and maintain a representation of unseen or occluded regions. OmniJarvis Wang et al. (2024) takes steps in this direction by tokenizing behaviors into discrete embeddings aligned with vision and language, enabling chain-of-thought-style planning over spatial actions. ROCKET-2Cai et al. (2025a), introduces a human-friendly goal specification method where users specify targets via segmentation masks from their own (third-person) perspective, which the agent must then align with its own egocentric view.

Spatial Planning and Navigation Beyond recognizing spatial goals, agents operating in Minecraft must reason over sequences of actions that unfold across space and time. Spatial planning and navigation in this context involve breaking down long-horizon tasks into subgoals, identifying spatial dependencies among them, and executing them under egocentric constraints. As Minecraft environments are vast, partially observable, and continuously changing, successful agents must not only interpret the structure of the task but dynamically adjust to failures and new spatial observations during execution.

A representative example of structured spatial planning is Plan4MCYuan et al. (2023). This framework constructs a skill graph using an LLM, which encodes dependencies between atomic "skills" like find tree, craft planks, or mine cobblestone. Each skill node in the graph encodes its requirements, outputs, and spatial dependencies (e.g., needing to be near a crafting table). For a given target goal (like crafting a stone pickaxe), Plan4MC performs a backward search over this graph to generate a valid execution plan. The result is a directed acyclic graph (DAG) of interdependent skills—each mapping to spatial actions the

agent must perform (e.g., finding a tree before reaching a furnace). This architecture allows the agent to adaptively generate skill sequences based on available resources and locations, translating high-level task structure into concrete navigation and manipulation plans. Plan4MC's approach is complemented by its hierarchical control system. A high-level planner decomposes tasks into subgoals via skill graphs, while a low-level controller executes each skill. Notably, navigation skills (e.g., "find sheep") are learned via PPO or DQN with intrinsic rewards for exploring new spatial locations.

A more memory-integrated planning framework is seen in Optimus-1Li et al. (2024b), which augments LLM-based planning with an episodic reflection loop. The planner, called the Knowledge-Guided Planner, uses an LLM conditioned on visual observations to generate a subgoal sequence. Unlike prior work which plans in a vacuum, Optimus-1 explicitly incorporates current environmental observations and history to generate context-sensitive plans. Importantly, an Experience-Driven Reflector module monitors plan execution and triggers re-planning if a subgoal fails—e.g., if a fish cannot be found in a nearby cave, the agent may re-plan to "leave the cave and find a river." Here, spatial planning is not a one-shot operation but a feedback loop sensitive to dynamic context, embodying a more realistic model of navigation.

A related architecture is seen in MrSteve Park et al. (2025), which builds on VPT (Video Pre-Training) by adding a goal-conditioned navigation head. Rather than training end-to-end, MrSteve separates goal encoding (via coordinates of current and target location) from image encoding and injects the goal signal late into the policy network. This architectural choice is driven by empirical findings that early fusion of spatial goals with visual embeddings degrades performance in complex terrains (e.g., mountains, rivers). With LoRA-based fine-tuning and a custom reward function based on distance reduction, MrSteve's navigation policy exhibits robust behavior across difficult spatial layouts. The agent is explicitly rewarded for minimizing Euclidean distance to the goal and is considered successful if it stops within 3 blocks of the target. In this setup, spatial planning is treated as continuous online adaptation rather than discrete symbolic planning.

Other systems rely on LLMs not only for subgoal sequencing but also for code generation. Voyager Wang et al. (2023), for example, takes a radically different approach by treating the entire planning process as language-level code synthesis. The agent uses GPT-4 to write, modify, and execute code to achieve spatially grounded goals such as exploring, building, or mining. The agent maintains a skill library and generates new skills as Python code based on its current state and past experience. It leverages a reward function based on discovery (e.g., new items or biomes found) and continues to expand its plan based on state changes. Spatial planning here is entirely symbolic: the LLM reasons over spatial state described in text (e.g., "no trees nearby") and produces spatially grounded programs such as "walk north until you find trees." Although this removes the need for learned navigation policies, it introduces brittleness due to the reliance on textual descriptions and the assumption of a stable code interface.

Spatial Memory and Temporal Continuity Spatial reasoning in embodied agents is rarely a one-shot computation. Unlike static benchmarks, Minecraft requires agents to persist in the environment over long periods, encountering spatially distributed cues, deferred goals, and incomplete observations. To succeed in such settings, agents must retain a memory of what they have seen, where it occurred, and when it happened. This introduces a crucial but underexplored dimension of spatial reasoning: temporal continuity and episodic spatial memory.

The need for such capabilities becomes especially clear in sequential, multi-stage tasks, where the location of one object might become relevant only minutes after it was first seen. Traditional low-level controllers like Steve-1 rely on a short-term Transformer-based memory (e.g., Transformer-XL with 128-step recurrence), which covers only a few seconds of interaction and lacks the capacity for long-term spatial recall. Consequently, Steve-1 often resorts to repeated exploration when it fails to retain information about previously seen resources or landmarks. This is a fundamental limitation for spatial reasoning: the agent cannot reason about previously observed spatial relations or plan using prior knowledge of the world's structure. To address this, MrSteve Park et al. (2025) introduces a hierarchical episodic memory module called Place Event Memory (PEM). PEM is structured around the "what-where-when" paradigm: it encodes visual events (e.g., seeing a cow), their spatial location, and the time of occurrence. During execution, the agent queries PEM to retrieve spatial memories relevant to the current task—e.g., "Where did I last see a cow?" or "Was there water nearby earlier?" If a relevant memory is found, the agent switches to an "execution mode," using a

goal-conditioned navigator to reach the memorized location. If not, it defaults to an "exploration mode" to gather new data. This strategic modulation between memory recall and exploration reflects a cognitive model of spatial intelligence: not just perceiving space, but remembering and reusing it for future decisions.

Another approach to spatial memory appears in AdamYu & Lu (2024), which constructs and updates a causal graph of world interactions through a memory-enabled controller module. While not an episodic memory system in the classic sense, Adam's memory pool stores intermediate task trajectories, inventory states, and interaction outcomes, enabling the agent to infer persistent structural relations between actions and spatial outcomes. For example, it can learn that a crafting table is causally upstream of acquiring a wooden pickaxe, or that a furnace must be placed near coal for optimal smelting. Although Adam's causal memory is more symbolic than spatial, it supports the construction of a technology tree—a topological representation of Minecraft's action-space—which is itself a form of spatial abstraction grounded in temporal logic. This allows Adam to generalize and recompose previously learned subgraphs in novel contexts, demonstrating a unique blend of spatial persistence and causal reasoning

Temporal grounding also plays a critical role in visual perception. In ROCKET-1 Cai et al. (2025b), a novel visual-temporal context prompting strategy is employed to bind past visual frames to present action decisions. ROCKET-1 leverages SAM-based object segmentation to highlight objects of interest across time and trains a transformer policy to link these object trajectories to current goals.

4.2 Reflections: What Minecraft Teaches Us About LLMs and Space

Minecraft offers a uniquely revealing lens into the spatial reasoning capabilities of large language models (LLMs), precisely because it demands continual spatial awareness across long-horizon, open-ended tasks. Across recent agent designs, we see consistent patterns of failure and success that echo broader questions in LLM spatial understanding.

1) Most notably, LLM-based agents struggle with reference frame consistency. Spatial language like "to the left of the tree" or "build behind the house" is easy to produce but difficult to ground when agents operate from an egocentric perspective without persistent spatial anchoring. 2) Another challenge involves grounding abstract spatial language into concrete geometry. Instructions such as "build a wall" or "place the furnace inside" encode high-level spatial intent, but require agents to translate that into precise object configurations. 3) LLMs lack spatial memory over long horizons. In dynamic environments like Minecraft, relevant spatial information—such as the location of a previously seen landmark—may occur minutes before it becomes actionable.

However, several trends highlight how these limitations can be partially addressed. Visual grounding, through subgoal images or VLMs, improves geometric fidelity and object localization. Models like MineDreamer Zhou et al. (2024) and OmniJarvisWang et al. (2024) demonstrate that aligning language with visual targets leads to more coherent spatial behavior. By addressing challenges in Minecraft, it paves the way to applications of embodied AI in the real world.

5 Discussion and Future Directions

Overall, we have seen that even text-only foundation models have some concepts of spatial relations and can simulate small interactions with their internal representations, though more complex or unique environments pose immediate and currently insurmountable challenges. We find that once a visual modality is introduced, LLMs are able to reason much better over simple spatial tasks like relative object positions, but they struggle to incorporate more complex aspects of space such as multiple viewpoints, strange perspectives, or individual object orientations. While there are many datasets aiming to evaluate these qualities, the general approach to improving VLM understanding of space is simply fine-tuning on those datasets, as opposed to architectural additions.

We have also considered some text-based games as a potentially related domain where the success of models might have useful lessons on spatial representations for LLMs or present a new avenue to explore how models might internally represent space. Perhaps, models do not conform to the human expectation of these games

as inherently spatial and reason about them in some other manner, or perhaps, they simply learn to optimize the game without considering the game environment.

Finally, we explore Minecraft as a strong test ground for many of these approaches to simulate the real world. From this we can learn the weaknesses of current models and trial stronger methods.

We would like to expand our coverage of this problem by incorporating more papers evaluating LLM performance on tasks that include spatial awareness, exploring methods that claim to improve performance on such tasks, and including analyses of LLM behavior on spatial awareness (e.g. probing).

References

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. Can language models encode perceptual structure without grounding? a case study in color, 2021. URL https://arxiv.org/abs/2109.06129.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks, 2017. URL https://arxiv.org/abs/1511.02799.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding, 2022. URL https://arxiv.org/abs/2112.10482.

Shaofei Cai, Bowei Zhang, Zihao Wang, Xiaojian Ma, Anji Liu, and Yitao Liang. Groot: Learning to follow instructions by watching gameplay videos, 2023. URL https://arxiv.org/abs/2310.08235.

Shaofei Cai, Bowei Zhang, Zihao Wang, Haowei Lin, Xiaojian Ma, Anji Liu, and Yitao Liang. Groot-2: Weakly supervised multi-modal instruction following agents, 2024. URL https://arxiv.org/abs/2412.10410.

Shaofei Cai, Zhancun Mu, Anji Liu, and Yitao Liang. Rocket-2: Steering visuomotor policy via cross-view goal alignment, 2025a. URL https://arxiv.org/abs/2503.02505.

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, 2025b. URL https://arxiv.org/abs/2410.17856.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities, 2024. URL https://arxiv.org/abs/2401.12168.

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, 2024. URL https://arxiv.org/abs/2406.01584.

François Chollet. On the measure of intelligence, 2019. URL https://arxiv.org/abs/1911.01547.

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. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=rc8o_j8I8PX.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. Chessgpt: Bridging policy learning and language modeling, 2023. URL https://arxiv.org/abs/2306.09200.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive, 2024. URL https://arxiv.org/abs/2404.12390.

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning, 2023. URL https://arxiv.org/abs/2309.16650.

William H. Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations, 2019. URL https://arxiv.org/abs/1907.13440.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models, 2023. URL https://arxiv.org/abs/2307.12981.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning, 2016. URL https://arxiv.org/abs/1612.06890.

Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. Topviewrs: Vision-language models as top-view spatial reasoners, 2024a. URL https://arxiv.org/abs/2406.02537.

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, 2024b. URL https://arxiv.org/abs/2408.03615.

Yuan-Hong Liao, Rafid Mahmood, Sanja Fidler, and David Acuna. Reasoning paths with reference objects elicit quantitative spatial reasoning in large vision-language models, 2024. URL https://arxiv.org/abs/2409.09788.

Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, and Sheila McIlraith. Steve-1: A generative model for text-to-behavior in minecraft, 2024. URL https://arxiv.org/abs/2306.00937.

Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning, 2023. URL https://arxiv.org/abs/2205.00363.

Shunyu Liu, Yaoru Li, Kongcheng Zhang, Zhenyu Cui, Wenkai Fang, Yuxuan Zheng, Tongya Zheng, and Mingli Song. Odyssey: Empowering minecraft agents with open-world skills, 2024. URL https://arxiv.org/abs/2407.15325.

Xiao Liu, Da Yin, Yansong Feng, and Dongyan Zhao. Things not written in text: Exploring spatial commonsense from visual signals. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2365–2376, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.168. URL https://aclanthology.org/2022.acl-long.168/.

Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors, 2024. URL https://arxiv.org/abs/2403.13438.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Alexander Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16488–16498, 2024. doi: 10.1109/CVPR52733.2024.01560.

Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input, 2015. URL https://arxiv.org/abs/1410.0210.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. SPARTQA: A textual question answering benchmark for spatial reasoning. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4582–4598, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.364. URL https://aclanthology.org/2021.naacl-main.364/.

Ida Momennejad, Hosein Hasanbeig, Felipe Vieira, Hiteshi Sharma, Robert Osazuwa Ness, Nebojsa Jojic, Hamid Palangi, and Jonathan Larson. Evaluating cognitive maps and planning in large language models with cogeval, 2023. URL https://arxiv.org/abs/2309.15129.

Kolby Nottingham, Prithviraj Ammanabrolu, Alane Suhr, Yejin Choi, Hannaneh Hajishirzi, Sameer Singh, and Roy Fox. Do embodied agents dream of pixelated sheep: Embodied decision making using language guided world modelling, 2023. URL https://arxiv.org/abs/2301.12050.

Junyeong Park, Junmo Cho, and Sungjin Ahn. Mrsteve: Instruction-following agents in minecraft with what-where-when memory, 2025. URL https://arxiv.org/abs/2411.06736.

Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*, 2022. URL https://api.semanticscholar.org/CorpusID:251647156.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones,

Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2023. URL https://arxiv.org/abs/2206.04615.

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, 2023. URL https://arxiv.org/abs/2305.16291.

Zihao Wang, Shaofei Cai, Zhancun Mu, Haowei Lin, Ceyao Zhang, Xuejie Liu, Qing Li, Anji Liu, Xiaojian Ma, and Yitao Liang. Omnijarvis: Unified vision-language-action tokenization enables open-world instruction following agents, 2024. URL https://arxiv.org/abs/2407.00114.

Yutaro Yamada, Yihan Bao, Andrew K. Lampinen, Jungo Kasai, and Ilker Yildirim. Evaluating spatial understanding of large language models, 2024. URL https://arxiv.org/abs/2310.14540.

Shu Yu and Chaochao Lu. Adam: An embodied causal agent in open-world environments, 2024. URL https://arxiv.org/abs/2410.22194.

Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. Skill reinforcement learning and planning for open-world long-horizon tasks, 2023. URL https://arxiv.org/abs/2303.16563.

Yinqi Zhang, Xintian Han, Haolong Li, Kedi Chen, and Shaohui Lin. Complete chess games enable llm become a chess master, 2025. URL https://arxiv.org/abs/2501.17186.

Yongqiang Zhao, Zhenyu Li, Zhi Jin, Feng Zhang, Haiyan Zhao, Chengfeng Dou, Zhengwei Tao, Xinhai Xu, and Donghong Liu. Enhancing the spatial awareness capability of multi-modal large language model, 2023. URL https://arxiv.org/abs/2310.20357.

Enshen Zhou, Yiran Qin, Zhenfei Yin, Yuzhou Huang, Ruimao Zhang, Lu Sheng, Yu Qiao, and Jing Shao. Minedreamer: Learning to follow instructions via chain-of-imagination for simulated-world control, 2024. URL https://arxiv.org/abs/2403.12037.