iVISPAR — An Interactive Visual-Spatial Reasoning Benchmark for VLMs

Julius Mayer[*] **Mohamad Ballout**[†] **Serwan Jassim**[†]
Farbod Nosrat Nezami[†] **Elia Bruni**
Institute of Cognitive Science, Osnabrück University, Osnabrück, Germany
research@jmayer.ai

Abstract

Vision-Language Models (VLMs) are known to struggle with spatial reasoning and visual alignment. To help overcome these limitations, we introduce iVISPAR, an interactive multimodal benchmark designed to evaluate the spatial reasoning capabilities of VLMs acting as agents. iVISPAR is based on a variant of the sliding tile puzzle—a classic problem that demands logical planning, spatial awareness, and multi-step reasoning. The benchmark supports visual 3D, 2D, and text-based input modalities, enabling comprehensive assessments of VLMs' planning and reasoning skills. We evaluate a broad suite of state-of-the-art open-source and closed-source VLMs, comparing their performance while also providing optimal path solutions and a human baseline to assess the task's complexity and feasibility for humans. Results indicate that while VLMs perform better on 2D tasks compared to 3D or text-based settings, they struggle with complex spatial configurations and consistently fall short of human performance, illustrating the persistent challenge of visual alignment. This underscores critical gaps in current VLM capabilities, highlighting their limitations in achieving human-level cognition. Project website: `https://microcosm.ai/ivispar`.

1 Introduction

The rapid advancement of Vision-Language Models (VLMs) has spurred significant debate regarding their capacity to achieve human-level cognition. These models are increasingly deployed as general reasoning systems capable of addressing complex problems across diverse domains, with applications extending into dynamic, real-world scenarios such as physical agent-based tasks and planning (Wang et al., 2024a; Xi et al., 2023; Zeng et al., 2023). However, critical gaps persist in their spatial reasoning and visual alignment capabilities, areas essential for understanding, interpreting, and manipulating objects and their spatial relationships (Kamath et al., 2023a; Bordes et al., 2024; Campbell et al., 2024).

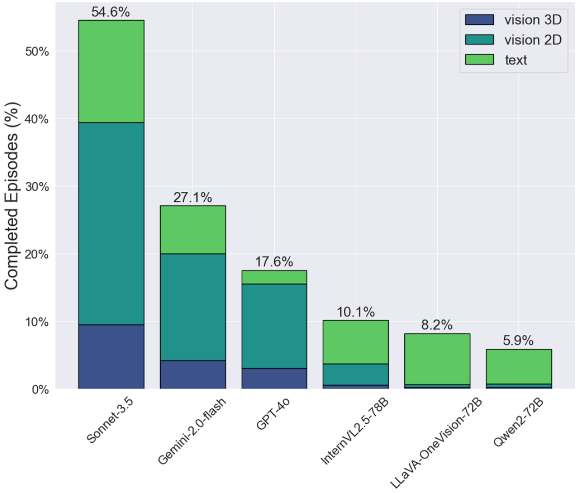


Figure 1: VLMs' success rates of completed games over 900 episodes across vision 3D, vision 2D, and text.

Spatial reasoning, a foundational aspect of problem-solving, navigation, and interaction with the physical world, requires models to bridge vision and cognition by interpreting visual information to understand spatial arrangements. Tasks such as mentally rotating shapes, predicting object movement, and recognizing patterns exemplify the importance of visual-spatial reasoning. Despite these critical requirements, progress in VLMs has been hampered by evaluation benchmarks that fail to capture the dynamic and multi-step complexity of real-world spatial reasoning. Existing benchmarks predominantly rely on static, text- or image-based setups that often oversimplify spatial contexts, focusing on 2D environments without interactivity or dynamic problem-solving capabilities.

---

[*] Corresponding author.
[†] Equal contribution.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 26757–26781
November 4-9, 2025 ©2025 Association for Computational Linguistics

This limitation perpetuates a lack of meaningful progress in visual-spatial reasoning within more realistic 3D environments.

Contributions. To bridge this gap, we introduce iVISPAR (Interactive Visual-Spatial Reasoning), a novel benchmark designed to systematically evaluate VLMs as agents in dynamic 3D environments. iVISPAR is built around the sliding tile puzzle, a well-established problem in developmental psychology that demands logical planning, spatial awareness, and multi-step problem-solving. As part of our contributions, we introduce the Sliding Geom Puzzle, a variant that replaces traditional numbered tiles with geometric objects distinguished by their color and shape, adding an additional layer of visual reasoning.

Notably, iVISPAR is grounded in a well-studied, formalized problem with access to optimal solutions, ensuring a robust framework for evaluation.[1] The benchmark supports scalable task complexity by adjusting factors such as board size, the number of tiles, and solution paths, ranging from simple configurations to NP-complete challenges that surpass baseline human performance.

Leveraging a prompt-based API, iVISPAR enables VLMs to interact with a simulated environment through an iterative action-perception loop. Experimentation results demonstrate that while state-of-the-art VLMs can handle basic spatial reasoning tasks, they face significant difficulties with more complex scenarios, especially in 3D environments. Evaluating models in such 3D settings is essential, as they more closely mirror the spatial complexity of real-world environments. By contrasting their performance against optimal solutions and human baselines, we highlight the persistent gap between current VLM capabilities and human-level spatial reasoning.

Our contributions are threefold: (i) a novel interactive benchmark that systematically evaluates visual-spatial reasoning in VLMs; (ii) a scalable task design rooted in a formalized problem with optimal solutions; and (iii) empirical insights into the strengths and limitations of VLMs across varying task complexities and modalities. iVISPAR lays the foundation for advancing VLM research toward overcoming critical gaps in reasoning and alignment capabilities.

2 Related work

2.1 Spatial Reasoning Benchmarks

Physical understanding in interactive agents has long been studied through simulation-based benchmarks (Li et al., 2024b; Mecattaf et al., 2024; Jassim et al., 2024; Wang et al., 2025; Hu et al., 2023; Zhao et al., 2025; Guruprasad et al., 2024; Su et al., 2024; Feng et al., 2025), although many of these frameworks are not directly suited for VLM evaluation due to limited language interfaces, low task fidelity, or demanding simulation requirements. Several datasets targeting visual reasoning have been applied to deep learning models (Johnson et al., 2016; Li et al., 2023), but they do not support interactive planning or action execution by language agents. Other works have explored similar setups using geometric object games, primarily in the context of language game learning with deep learning agents (Wang et al., 2016; Kuhnle and Copestake, 2017); related efforts such as Sliding Puzzles Gym and PUZZLES (Oliveira et al., 2024; Estermann et al., 2024) have been proposed as RL benchmarks, but lack the language interface and fine-grained 3D problem generation introduced in our setting.

2.2 Spatial Reasoning in LLMs

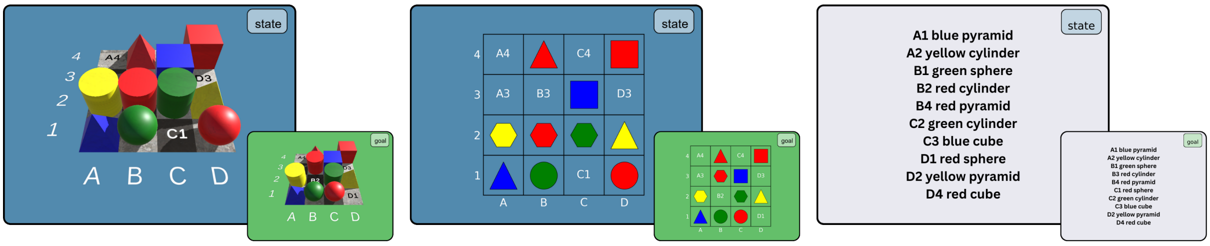
Even though Large Language Models (LLMs) are primarily trained via next-token prediction on textual corpora, their capacity for spatial reasoning have attracted recent attention (Abdou et al., 2021; Patel and Pavlick, 2021). LLMs have also been explored as agents for spatial planning (Bohnet et al., 2024), path planning (Aghzal et al., 2024), and spatial path generation (Rizvi et al., 2024) in purely textual or symbolic environments. Several recent studies have examined whether LLMs implicitly encode spatial structures and geometric reasoning, ranging from digital twin generation via symbolic rules (Wang et al., 2024c), to textual spatial question answering in diverse settings (Mirzaee et al., 2021), and evaluations across grid, ring, and tree topologies (Yamada et al., 2024).

2.3 Spatial Reasoning in VLMs

Visual reasoning has emerged as a key focus in evaluating VLMs, with growing interest in their capacity to interpret spatial relationships and object configurations (Zhang et al., 2024; Rajabi and Kosecka, 2024b; Roberts and Roberts, 2024; Campbell et al., 2025); concurrently, several studies have examined the degree to which these mod-

---

[1] The formalization is achieved through the adaptation of the sequential generalized sliding-tile puzzle, as described in the Appendix A.3. Optimal solutions are computed using the A* algorithm, detailed in Section 4.2.



A1 blue pyramid
A2 yellow cylinder
B1 green sphere
B2 red cylinder
B4 red pyramid
C2 green cylinder
C3 blue cube
D1 red sphere
D2 yellow pyramid
D4 red cube

A1 blue pyramid
A2 yellow cylinder
B1 green sphere
B3 red cylinder
B4 red pyramid
C1 red sphere
C2 green cylinder
C3 blue cube
D2 yellow pyramid
D4 red cube

Figure 2: Example of VLMs' observations for a state (blue) and the goal (green) at each step during an episode of the Sliding Geom Puzzle environment, on a 4×4 board with 10 geoms and an optimal path length of 2. Left to right, each tested modality: vision 3D, vision 2D, and text-based representation. For more examples, see Appendix A.1.2

els align visual inputs with linguistic representations (Merullo et al., 2023; Ilharco et al., 2021). Recent advancements in VLMs have prompted a surge in evaluations, yet most studies primarily rely on visual question-answering tests (Liu et al., 2023; Rajabi and Kosecka, 2024a; Wang et al., 2024b; Cheng et al., 2024; Tang et al., 2024; Duan et al., 2025; Wang et al., 2023; Kamath et al., 2023b). Beyond static evaluations, a growing body of work explores the use of VLMs and foundation models as interactive agents within simulated environments, where they are tasked with manipulating objects, navigating spaces, or executing spatial instructions in grounded contexts (Wu et al., 2024; Li et al., 2024b; Mecattaf et al., 2024; Jassim et al., 2024; Wang et al., 2025; Su et al., 2024). This includes applications in embodied AI and robotics, where VLMs are increasingly integrated into control loops to support visuomotor reasoning and spatial decision-making (Hu et al., 2023; Zhao et al., 2025; Guruprasad et al., 2024; Feng et al., 2025).

In this context, we present iVISPAR, an interactive multimodal benchmark designed to evaluate the spatial reasoning capabilities of VLMs acting as agents.

3 The iVISPAR Benchmark

iVISPAR[2] is an interactive, multimodal puzzle simulator that presents agents with a board state in one of three input modalities: a 3D rendered image, a 2D top-down view, or a text-based representation (see Figure 2). By rendering scenes in 3D space, iVISPAR offers a more realistic depiction of spatial environments compared to traditional 2D grid visualizations and enables systematic comparisons across modalities. Agents interact with the board by issuing natural language commands through a text-based API to apply actions to the board (see Figure 3). iVISPAR supports procedural generation of puzzle instances with finely controlled parameters, allowing for a scalable dataset of tasks with adjustable complexity across many spatial properties, and benchmarking performance with multiple baseline models.

3.1 Sliding Geom Puzzle

A central environment in iVISPAR is the Sliding Geom Puzzle (SGP), a reimagining of the classic sliding tile puzzle (see Appendix A.3). Instead of numbered tiles, SGP uses geometric objects (geoms) uniquely defined by combinations of color and shape, increasing visual-spatial complexity and enhancing task scalability. This design shift requires models to interpret object features rather than follow numerical sequences, mirroring real-world spatial reasoning where items are distinguished by appearance, size, or structure. The task draws inspiration from physical scenarios such as organizing items, assembling structures, or packing, promoting a more authentic evaluation of real-world spatial capabilities.

3.2 Game dynamics

The objective is to rearrange the pieces on the board by moving them over free spaces to match a given goal configuration. In each episode, agents receive observations of the start and goal states (see Figure 2), accompanied by task instructions (see Appendix A.1.1). Agents apply move actions to geoms by referencing their unique color and shape combination and specifying the direction of intended movement. Geoms can be moved in cardinal directions (*LEFT, RIGHT, UP, DOWN*), with actions formatted as "move <color> <shape> <direction>”:

```
"move blue sphere right"
```

---

[2] All source code is available under MIT license at: https://github.com/SharkyBamboozle/iVISPAR

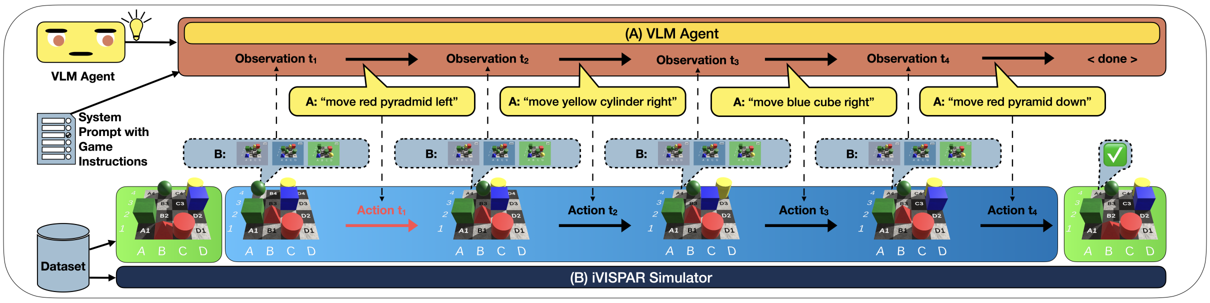


Figure 3: Depiction of the interaction flow between VLM agents and the iVISPAR simulator with a progression through an episode with the shortest path solution of 4 steps being solved by prompted actions from a VLM agent. For a full example of an episode progression, see Appendix A.1.4.

Actions are validated and applied if legal, with agents receiving updated board states regardless of the action's success after each move command. Effective and ineffective actions both result in valid new board states but, respectively, decrease or increase the path length to the goal state. Invalid moves, such as occupied destination and out-of-bounds actions, fail to alter the board state, as do illegal commands, which violate the instructed action format. This action-perception loop repeats until the goal state is achieved or a step limit is reached. Due to limited context windows, VLM agents receive task instructions at each time step. A sample agent-environment interaction is provided in Appendix A.1.3.

3.3 Observation Spaces

Agents observe a combination of the current board state and the goal state. Additionally, they can receive a sequence of past state-action pairs, determined by the size of the configured context window. Images for 3D observations are presented from an angled top-down perspective and may include partially occluded objects, whereas 2D observations follow a graph-like layout with fully visible elements. Both may optionally include embedded, text-based chess-style coordinate labels as spatial cues along the outer edge of the grid board as well as on free tiles. In 2D observations, shapes are mapped consistently from their 3D counterparts to preserve object identity across modalities. Images can also be marked with an embedded text label and a colored background to differentiate between past (grey), current (blue), and goal state (green). Figure 2 shows 3D vision (left) and 2D vision (middle) for the active state (top) and the goal state (bottom). The text-based representation encodes past, active, and goal states directly in the prompt string supplied to the agent. Agents receive the list of geoms in the order of board coordinates. A visualization of the text-based active (top) and goal states (bottom) is shown in Figure 2 (right). This modality does not rely on images.

3.4 Complexity Scalability

The GSTP is a well-known NP-hard problem due to the need for multi-step planning across a constrained grid (Gozon and Yu, 2024). SGP inherits this complexity but introduces greater flexibility in scaling difficulty without altering the game's core mechanics. This flexibility provides more degrees of freedom, making the task more tractable for VLM agents. Key scaling factors include board size, number of objects, object variability, length of the shortest path solution, and the geom interference factor (see Appendix A.1.2). The shortest path solution for all episode configurations is calculated using the A* algorithm (Hart et al., 1968), as detailed in Appendix A.7.1. The interference factor denotes the extent to which objects obstruct one another's optimal paths, increasing the global solution length beyond the cumulative Manhattan distances of individual paths. This interference can create configurations with short optimal paths but increased planning requirements, significantly raising the problem's difficulty. Available geometric shapes include [“cube,” “pyramid,” “sphere,” “cylinder,” “cone,” “prism”], with colors freely selectable by referencing RGB values. Agents must navigate combinatorial complexity by matching shapes and colors, promoting spatial strategies over the sequential patterns seen in numerical tile puzzles. Episode configurations are generated procedurally, requiring models to generalize across puzzle instances. Human and algorithmic benchmarks for these experiments are detailed in Section

4.2.

4 Experiments

Performance of VLMs is tested for the SGP to assess their capabilities in scene understanding, problem-solving, and multi-step planning within constrained environments.

4.1 Dataset Generation

Experiments were conducted on a dataset of SGPs on a fixed board size to 4×4: smaller grids (e.g., 3×3) collapse many spatial-relation cases, while larger ones ($\geq 5 \times 5$) dilute object visibility without yielding further complexity benefits. Performance is assessed by varying complexity across two parameters: the number of objects (2-11) and the shortest path length (2-11). Configurations maintain a geom interference factor of 0, ensuring the shortest path equals the cumulative Manhattan distance. Initial experiments indicated that VLM agents faced significant challenges at higher task complexities. Three episodes are sampled for each complexity level, producing a dataset of 300 diverse board configurations. The set of geom properties consists of four shapes, sphere, pyramid, cube, and cylinder, and four colors, red, green, blue, and yellow, resulting in 16 unique combinations. VLM agents are tested on the same dataset for each modality, resulting in 900 episodes for each model.

4.2 Baselines

To contextualize agent performance and provide upper and lower bounds, we establish four baselines encompassing human and AI agents.

Human performance was evaluated with 30 participants using a web app GUI of the SGP, where participants interacted by prompting text commands over a command line, mirroring the interaction method of VLM agents. Baselines were provided for the 3D vision modality on the same dataset as the VLM agents.

AI baselines were introduced for two agents: an optimal agent executing shortest path solutions computed by A* (Hart et al., 1968), and a random agent performing uninformed but valid actions uniformly sampled from those leading to new board states. Algorithms for the AI agents are detailed in Appendix A.7.

4.3 Models

We evaluate a selection of open- and closed-source VLMs that scored high on OpenCompass[3] and which support multi-image inputs and a minimum context length of 800 tokens. Selected models are: Sonnet-3.5 (Claude Team, 2024), Gemini-2.0-flash (Gemini Team, 2024), GPT-4o (OpenAI et al., 2024), InternVL2.5-78B (Chen et al., 2024), LLaVA-OneVision-72B (Li et al., 2024a), Qwen2-72B (Wang et al., 2024d). For closed-source models, we rely on the official APIs and for open-source models, on the publicly available checkpoints. We use a temperature of 1.0, top-p of 0.95, and top-k of 50 for all open-source models. An overview of all models and their details can be found in the Appendix A.2.

4.4 Context-Aware Zero-Shot Reasoning

The models employ Chain-of-Thought (CoT) reasoning (Wei et al., 2022) to break down complex problems into smaller sub-tasks, enhancing accuracy and interpretability (Appendix A.1.3). We constrain VLMs' context windows to the past two steps, incorporating state representations alongside the model's action responses. This approach prioritizes extracting maximum value from limited experience to preserve the models' sequential coherence and minimize computational overhead. Operating within this context-aware zero-shot reasoning framework, the models interpret task requirements without examples, drawing exclusively from pretrained knowledge, task instructions, and limited past interactions.

4.5 Instruction Prompts

We avoided prompt engineering for any single model; the chosen template is the same for all systems and contains only the minimal information needed. Fixing one validated template provides a consistent basis for comparison and makes the benchmark easily reproducible. The visual and text prompts are isomorphic: the image placeholder is the only difference, so no modality receives extra hints. Our human-baseline study likewise found the final wording easy to follow. This supports our aim of testing spatial-reasoning ability itself, without relying on prompt engineering, so we use one clear, uniform template for all models.

---

[3] OpenCompass Official Rankings: `https://rank.opencompass.org.cn/leaderboard-multimodal`

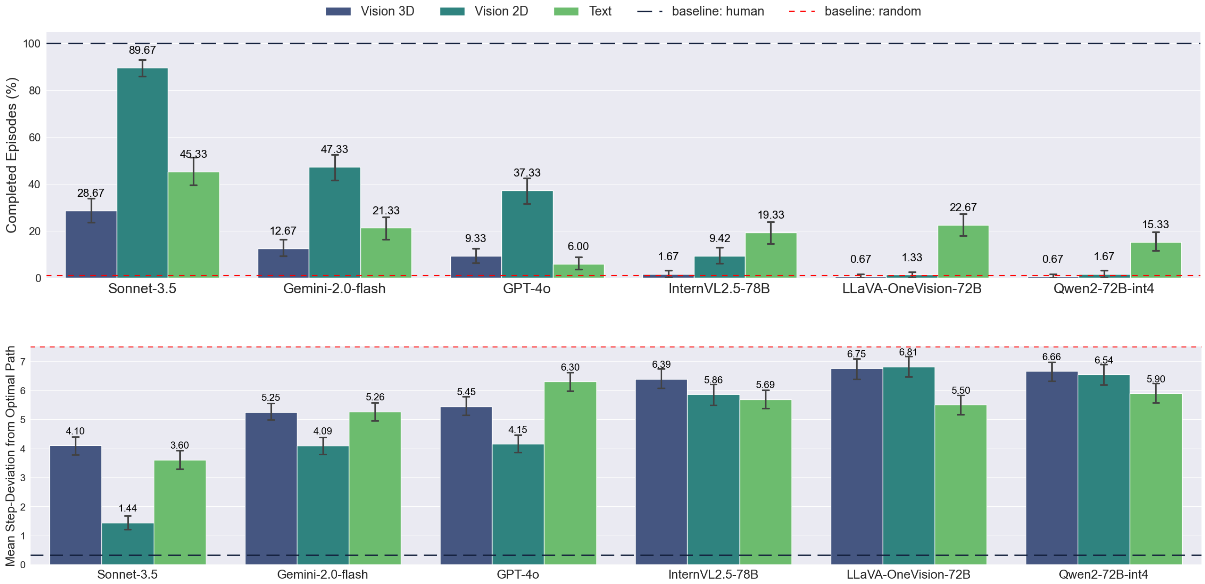


Figure 4: VLM evaluation on 900 episodes per model across all three modalities, with 95% confidence intervals. Baseline comparisons for human performance and random moves are shown. Top: VLMs' success rates of episodes completed with higher values denoting better performance. Bottom: VLMs' mean step deviation from the optimal path with lower values denoting better performance. Full numerical results are provided in Appendix A.4

4.6 Evaluation

Agent performance is evaluated through two primary metrics: the fraction of solved environments and mean step-deviation from the optimal path

Mean step-deviation from optimal path measures the deviation from optimal behavior during problem-solving. At each step t, the shortest path solution from the current board state to the goal, computed by A*, is used to assess efficiency. Formally,

$$R(t) = d(s_t, s^*) - \big[d(s_0, s^*) - t\big].$$

where $d(s, s^*)$ denotes the shortest path length from state s to the goal s^*. This metric quantifies how much further the agent is from the goal compared to an optimal agent after the same number of steps. A regret value of zero indicates that the agent follows an optimal trajectory, while positive regret reflects inefficiencies or unnecessary detours. By capturing performance even in unsolved environments, this approach provides insights into agent behavior under varying complexities.

To gain deeper insights, we analyze the most common error patterns exhibited by agents. This allows us to identify model weaknesses, recurring failure cases, and patterns of suboptimal decision-making.

4.7 Auxiliary Task

Additionally, we evaluate the models' ability to infer and represent board states from visual input across all 300 episodes. Given an image and accompanying instructions, each model is tasked with predicting the corresponding board configuration in text form, using the same format as the textual representation shown in Figure 2. This auxiliary task further enriches our understanding of the models' behavior and their capacity to interpret spatial information from visual inputs.

To analyze this task, we frame the comparison between the true and predicted board states as a set matching problem, solved using the Hungarian algorithm. A match is defined as any pair of geoms sharing at least either color or shape. Geoms that share neither are considered missed (if only present in the true state) or hallucinated (if only present in the prediction). Matched geoms may still contain mismatches in coordinates, color, or shape. Predicted elements that cannot be parsed into valid geom triplets are counted as format errors.

5 Results

We evaluated the spatial reasoning capabilities of VLMs in our SGP environment on 3D vision and

---

[*] Videos of exemplary agents' interactions with iVISPAR are available at: `https://youtu.be/Djis_xkgtW8`.

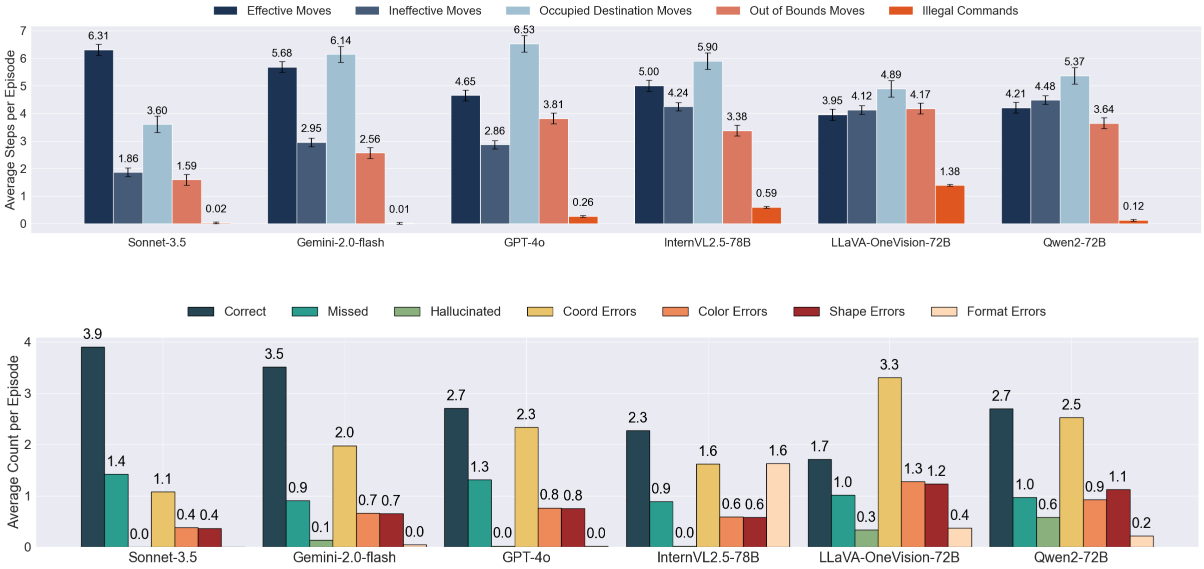


Figure 5: Error patterns showing average action counts per episode during SGP interaction (top) and average geoms per episode for the board state inference auxiliary task (bottom), both averaged across modalities (see Sections 5 and 4.7), each aggregated across modalities. Full numerical results are provided in Appendix A.4.

compared it to 2D vision and text-based modalities across 300 episodes each (see Figure 4). To standardize gameplay, the number of actions per episode was capped at 20.

Success rates: The percentage of episodes completed and the mean deviations of steps from the optimal path were measured for each modality and compared to human performance as well as random actions (Figure 4).

Action classification: We classified actions based on their effects on the board and calculated their average occurrence per episode to provide insights into the challenges VLMs face in efficiently completing episodes (see Figure 5 top). Effective and ineffective actions both result in valid new board states but, respectively, decrease or increase the path length to the goal state. Invalid moves, such as occupied destination and out-of-bounds actions, while illegal commands break the instructed action format, all of which leave the board state unchanged.

Auxiliary Task: For the board state inference task, we evaluate the number of geoms that were correctly inferred, missed, hallucinated, or contained a mismatch in coordinates, color, or shape. Format errors denote cases where the output failed to follow the expected structure (Figure 5, bottom).

Complexity scales: We evaluated the cumulative performance of VLMs across the three modalities using two complexity scales, the shortest path length required to solve an episode and the number of geoms on the board. Longer shortest paths demand a broader global planning horizon and consistent goal-directed progress, while higher geom counts require efficient local planning to optimize rearrangement order and manage free spaces. Figure 7 illustrates the performance of VLMs in 100 combinations of complexity, highlighting the average minimal distance to the goal state in 20 steps.

6 Discussion

6.1 Model Performance

All models show basic task understanding and spatial reasoning, progressing toward the goal state (see Figure 4). Performance, however, varies widely. Closed-source models outperform open-source ones: Sonnet-3.5 achieves the highest success rate at 89.7% in the 2D visual modality, followed by Gemini-2.0-Flash and GPT-4o. In contrast, open-source models such as InternVL2.5-78B, LLaVA-OneVision-72B, and Qwen2-72B perform near the random baseline. Human participants solve the tasks perfectly with near-optimal paths, setting a high benchmark.

Notably, even models solving fewer than 1% of tasks often produce more efficient paths than a random baseline (see Figure 4, bottom), indicating traces of goal-directed behavior despite overall failure. These task performances are also consistent

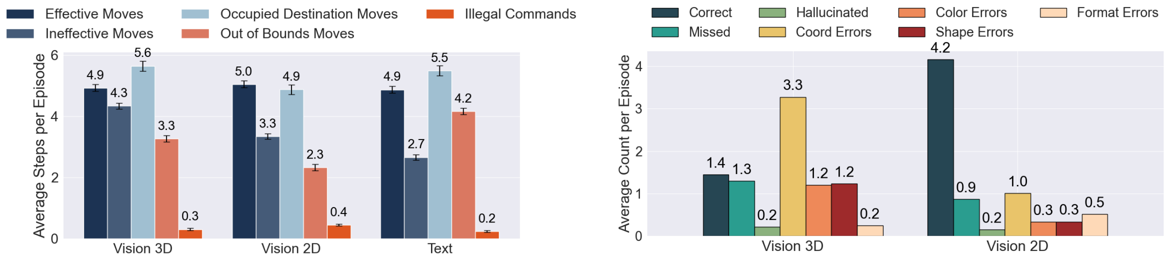


Figure 6: Error patterns showing average action counts per episode during SGP interaction (left; see Section 5) and average geoms per episode for the board state inference auxiliary task (right; see Section 4.7), shown per modality and aggregated across agents. Full numerical results are provided in Appendix A.4.

with the further analysis of the models' error types and their accuracy in the board state inference task, which we discuss in Section 6.2.

6.2 Error Patterns

We analyzed the types of mistakes models make during interaction with the simulator and evaluated their ability to infer board states from visual input. Overall, models rarely issue illegal commands or exhibit format errors (see Figure 5, top and bottom), suggesting that most VLMs understand how to follow instructions and interact with the environment appropriately.

However, board state inference accuracy reveals a sharp performance drop from 2D to 3D inputs: while models correctly identify an average of 4.2 objects in 2D, this number falls to 1.4 in the 3D setting (see Figure 6, right). This is primarily due to substantial increases in coordinate prediction errors, alongside moderate rises in color, shape mismatches, and missed detections. In contrast, hallucinations and format-related issues remain largely stable across both modalities.

These findings offer a clear explanation for the weaker performance in the 3D vision condition: precise localization of objects remains a critical challenge. As illustrated in Figure 5, this results in more ineffective moves, including frequent attempts to place objects out-of-bounds or onto already occupied cells.

6.3 Modality Impact

Despite being evaluated on identical tasks, model performance varied substantially across input modalities (see Figure 4). All closed-source models (Sonnet-3.5, Gemini-2.0-flash, GPT-4o) performed best on 2D vision, followed by text, and worst on 3D vision. This suggests that these models may have undergone more training on 2D visual inputs, which are more common in spatial benchmarks. Interestingly, text input, despite posing significant challenges for humans, ranked second, indicating some robustness in linguistic reasoning. In contrast, open-source models (InternVL2.5, LLaVA-OneVision, Qwen2) performed poorly across the board, with near-random scores on visual inputs. Their relatively stronger performance on text tasks may reflect a reliance on superficial pattern recognition rather than grounded spatial understanding. As shown in Figure 6 (left), error patterns for ineffective moves and collisions align with the overall performance ranking across modalities. Out-of-bounds errors are most frequent in the text condition, nearly twice as common as in 2D vision, indicating that understanding board dimensions was a primary challenge in the textual setting. Additional results from our board state inference task further support this view, showing that models, predict more correct objects on the board in Vision 2D compared to Vision 3D (Figure 6, right).

6.4 Complexity Scaling

We analyzed the correlation matrix between the number of objects on the board and the shortest path solution length to assess how different types of complexity affect model performance (see Figure 7, top). While performance consistently drops with increasing complexity in both dimensions, the heatmaps reveal modality-specific trends. Performance declines more steeply with increasing geom count (particularly in 3D), suggesting that sequential planning under visual conditions poses a major challenge. In contrast, in the text-only setting, the number of geoms seems to have little effect, with errors mostly determined by the length of the shortest path solution. This highlights limitations in spatial reference from language alone.

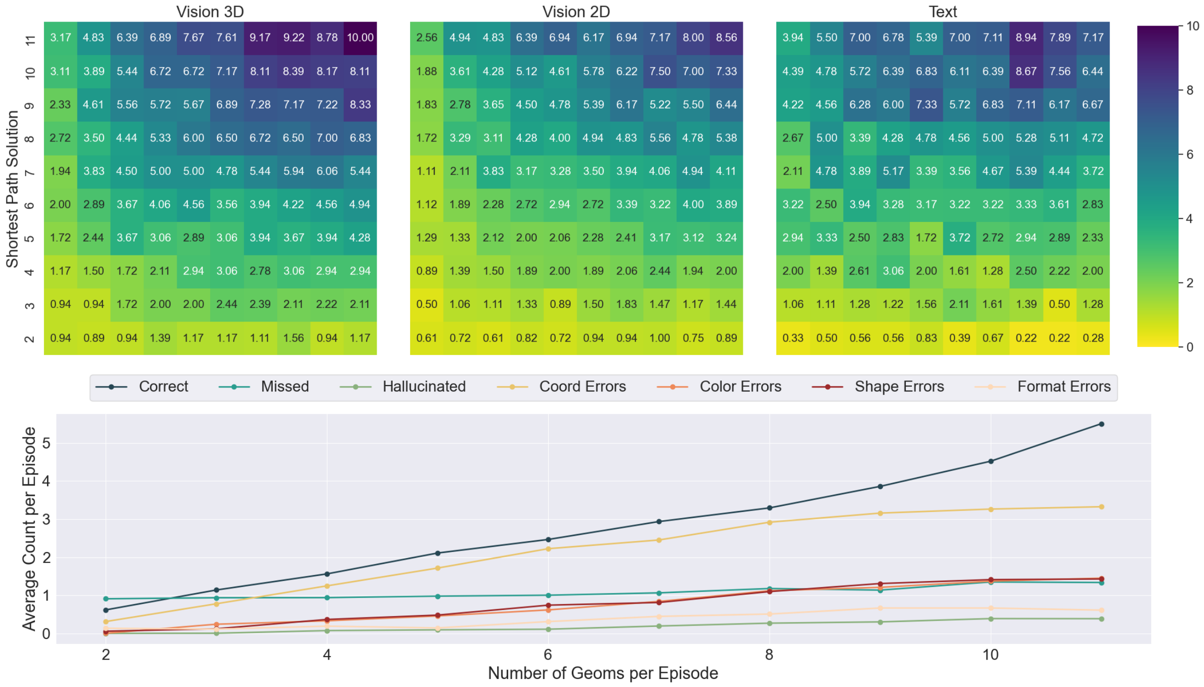


Figure 7: Cumulative graphs aggregated across agents. Top: Correlation matrix of remaining shortest-path lengths to the goal for tasks with optimal paths between 2–11 steps. Each run is capped at 20 actions, and the metric is computed at the agent's final state, either upon reaching the goal or, if unsolved, after the 20th action. Bottom: Error types in the board state inference auxiliary task over increasing number of geoms on the board.

Data from the auxiliary task of board state inference show that, while errors to predict the coordinates of geoms on the board increase with the number of geoms on the board, other error types remain relatively stable even for a higher number of geoms on the board (see Figure 7). Format errors and the number of hallucinated geoms is overall low, mismatches with colors and shapes increasing only slightly, and surprisingly the number of missed objects stays relatively stable as well.

7 Conclusion

We have introduced iVISPAR, a novel interactive multimodal benchmark designed to evaluate the spatial reasoning capabilities in 3D vision of VLMs acting as agents. The benchmark, centered on the Sliding Geom Puzzle, evaluates VLMs' abilities in logical planning, spatial awareness, and multi-step problem-solving, aiming to reflect real-world spatial reasoning. Our evaluation tested a suite of state-of-the-art open-source and closed-source VLMs on a dataset of board configurations, scaled across two levels of complexity. We compared them to baselines for human capabilities, optimal and random agents, providing insight into their performance under varying conditions.

Our findings demonstrate that VLMs struggle with spatial reasoning in 3D vision and that there are significant performance differences between the tested VLMs. While they understand the instructions and outperform random agents in simple spatial tasks, they struggle with more complex configurations and intricate problem properties. Interestingly, VLMs show stronger performance in 2D vision compared to 3D or text-based tasks. Our auxiliary board state inference task revealed that VLMs frequently miss geoms, misplace them on the board, or mismatch their colors or shapes, errors that occur more often with 3D vision input than with 2D. This suggests that visual alignment for 3D spatial reasoning continues to pose a significant challenge, underscoring persistent gaps in VLM capabilities and highlighting barriers to achieving human-level cognitive performance.

Future Work Looking ahead, we plan to expand the benchmark to incorporate additional tasks focused on scene understanding, as well as rotation and transformation challenges.

Resources For the most up-to-date results on state-of-the-art models and access to the leaderboard, please visit: `https://microcosm.ai/ivispar`.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) — 456666331, 321892712.

Limitations

We restricted the context window, limiting the number of images VLMs can process. Extended image inputs often disrupt VLMs' understanding of sequential coherence and increase computational demands and API costs. This contrasts with human participants, who recall each step of an episode and draw from past experiences.

Additionally, while some models are optimized for long-context reasoning or "deep thinking," their architecture and usage patterns are ill-suited for step-wise, interactive simulations. Their per-frame API costs are disproportionately higher, making them impractical for the interaction format used in our benchmark. This also limits direct comparisons to human participants, who recall previous steps and integrate episodic knowledge more efficiently.

Impact Statement

This paper contributes to advancements in vision-language models. While our work has potential applications in broader AI research, it does not introduce immediate ethical or societal risks beyond those already associated with the field. As our work is largely theoretical and not at a scale that could pose significant concerns, it does not raise specific risks of misuse or unintended consequences.

References

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. 2021. Can language models encode perceptual structure without grounding? a case study in color.

Mohamed Aghzal, Erion Plaku, and Ziyu Yao. 2024. Can large language models be good path planners? a benchmark and investigation on spatial-temporal reasoning. *Preprint*, arxiv:2310.03249 [cs].

Bernd Bohnet, Azade Nova, Aaron T. Parisi, Kevin Swersky, Katayoon Goshvadi, Hanjun Dai, Dale Schuurmans, Noah Fiedel, and Hanie Sedghi. 2024. Exploring and benchmarking the planning capabilities of large language models. *Preprint*, arxiv:2406.13094 [cs].

Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C. Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, Mark Ibrahim, Melissa Hall, Yunyang Xiong, Jonathan Lebensold, Candace Ross, Srihari Jayakumar, Chuan Guo, Diane Bouchacourt, Haider Al-Tahan, and 22 others. 2024. An introduction to vision-language modeling. *CoRR*, abs/2405.17247.

Declan Campbell, Sunayana Rane, Tyler Giallanza, Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven M. Frankland, Thomas L. Griffiths, Jonathan D. Cohen, and Taylor W. Webb. 2024. Understanding the limits of vision language models through the lens of the binding problem. *CoRR*, abs/2411.00238.

Declan Campbell, Sunayana Rane, Tyler Giallanza, Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven M. Frankland, Thomas L. Griffiths, Jonathan D. Cohen, and Taylor W. Webb. 2025. Understanding the limits of vision language models through the lens of the binding problem. *Preprint*, arxiv:2411.00238 [cs].

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. 2024. SpatialRGPT: Grounded spatial reasoning in vision language models. *Preprint*, arxiv:2406.01584 [cs]. Version: 3.

Claude Team. 2024. Introducing the next generation of claude. `https://www.anthropic.com/news/claude-3-family`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929.

Lin Duan, Yanming Xiu, and Maria Gorlatova. 2025. Advancing the understanding and evaluation of AR-generated scenes: When vision-language models shine and stumble. *Preprint*, arxiv:2501.13964 [cs].

Benjamin Estermann, Luca A. Lanzendörfer, Yannick Niedermayr, and Roger Wattenhofer. 2024. PUZZLES: A benchmark for neural algorithmic reasoning. *Preprint*, arxiv:2407.00401 [cs].

Yunhai Feng, Jiaming Han, Zhuoran Yang, Xiangyu Yue, Sergey Levine, and Jianlan Luo. 2025. Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation. *Preprint*, arxiv:2502.16707 [cs].

Gemini Team. 2024. Gemini 2.0 flash (experimental).

Marcus Gozon and Jingjin Yu. 2024. On computing makespan-optimal solutions for generalized sliding-tile puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38(9), pages 10288–10296.

Pranav Guruprasad, Harshvardhan Sikka, Jaewoo Song, Yangyue Wang, and Paul Pu Liang. 2024. Benchmarking vision, language, & action models on robotic learning tasks. *Preprint*, arxiv:2411.05821 [cs].

Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. 2023. Look before you leap: Unveiling the power of GPT-4v in robotic vision-language planning. *Preprint*, arxiv:2311.17842 [cs].

Gabriel Ilharco, Rowan Zellers, Ali Farhadi, and Hannaneh Hajishirzi. 2021. Probing contextual language models for common ground with visual representations. *Preprint*, arxiv:2005.00619 [cs].

Serwan Jassim, Mario Holubar, Annika Richter, Cornelius Wolff, Xenia Ohmer, and Elia Bruni. 2024. GRASP: A novel benchmark for evaluating language GRounding and situated physics understanding in multimodal language models. *Preprint*, arxiv:2311.09048.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2016. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *Preprint*, arxiv:1612.06890 [cs].

Amita Kamath, Jack Hessel, and Kai-Wei Chang. 2023a. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9161–9175. Association for Computational Linguistics.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. 2023b. What's "up" with vision-language models? investigating their struggle with spatial reasoning. *Preprint*, arxiv:2310.19785 [cs].

Alexander Kuhnle and Ann Copestake. 2017. Shape-World - a new test methodology for multimodal language understanding. *Preprint*, arxiv:1704.04517 [cs].

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

Kanxue Li, Baosheng Yu, Qi Zheng, Yibing Zhan, Yuhui Zhang, Tianle Zhang, Yijun Yang, Yue Chen, Lei Sun, Qiong Cao, Li Shen, Lusong Li, Dapeng Tao, and Xiaodong He. 2024b. MuEP: A multimodal benchmark for embodied planning with foundation models. In *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*, pages 129–138. International Joint Conferences on Artificial Intelligence Organization.

Zhuowan Li, Xingrui Wang, Elias Stengel-Eskin, Adam Kortylewski, Wufei Ma, Benjamin Van Durme, and Alan Yuille. 2023. Super-CLEVR: A virtual benchmark to diagnose domain robustness in visual reasoning. *Preprint*, arxiv:2212.00259 [cs].

Fangyu Liu, Guy Emerson, and Nigel Collier. 2023. Visual spatial reasoning. *Preprint*, arxiv:2205.00363 [cs].

Matteo G. Mecattaf, Ben Slater, Marko Tešić, Jonathan Prunty, Konstantinos Voudouris, and Lucy G. Cheke. 2024. A little less conversation, a little more action, please: Investigating the physical common-sense of LLMs in a 3d embodied environment. *Preprint*, arxiv:2410.23242 [cs].

Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. 2023. Linearly mapping from image to text space. *Preprint*, arxiv:2209.15162 [cs].

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjmashidi. 2021. SpartQA: : A textual question answering benchmark for spatial reasoning. *Preprint*, arxiv:2104.05832 [cs].

Bryan Lincoln Marques de Oliveira, Bruno Brandão, Murilo Lopes da Luz, Luana Guedes Barros Martins, Telma Woerle de Lima Soares, and Luckeciano Carvalho Melo. 2024. Sliding puzzles gym: A scalable benchmark for state representation in visual reinforcement learning. In *NeurIPS 2024 Workshop on Open-World Agents*.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Roma Patel and Ellie Pavlick. 2021. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*.

Navid Rajabi and Jana Kosecka. 2024a. GSR-BENCH: A benchmark for grounded spatial reasoning evaluation via multimodal LLMs. *Preprint*, arxiv:2406.13246 [cs]. Version: 2.

Navid Rajabi and Jana Kosecka. 2024b. Towards grounded visual spatial reasoning in multi-modal vision language models. *Preprint*, arxiv:2308.09778 [cs].

Md Imbesat Hassan Rizvi, Xiaodan Zhu, and Iryna Gurevych. 2024. SpaRC and SpaRP: Spatial reasoning characterization and path generation for understanding spatial reasoning capability of large language models. *Preprint*, arxiv:2406.04566 [cs].

Denisa Roberts and Lucas Roberts. 2024. Smart vision-language reasoners. *Preprint*, arxiv:2407.04212 [cs].

Ying Su, Zhan Ling, Haochen Shi, Jiayang Cheng, Yauwai Yim, and Yangqiu Song. 2024. ActPlan-1k: Benchmarking the procedural planning ability of visual language models in household activities. *Preprint*, arxiv:2410.03907 [cs].

Yihong Tang, Ao Qu, Zhaokai Wang, Dingyi Zhuang, Zhaofeng Wu, Wei Ma, Shenhao Wang, Yunhan Zheng, Zhan Zhao, and Jinhua Zhao. 2024. Sparkle: Mastering basic spatial capabilities in vision language models elicits generalization to composite spatial reasoning. *Preprint*, arxiv:2410.16162 [cs].

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, and Neel Joshi. 2024a. Is A picture worth A thousand words? delving into spatial reasoning for vision language models. *CoRR*, abs/2406.14852.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. 2024b. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *Preprint*, arxiv:2406.14852 [cs].

Jingquan Wang, Harry Zhang, Huzaifa Mustafa Unjhawala, Peter Negrut, Shu Wang, Khailanii Slaton, Radu Serban, Jin-Long Wu, and Dan Negrut. 2024c. SimBench: A rule-based multi-turn interaction benchmark for evaluating an LLM's ability to generate digital twins. *Preprint*, arxiv:2408.11987 [cs].

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024d. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Sida I. Wang, Percy Liang, and Christopher D. Manning. 2016. Learning language games through interaction. *Preprint*, arxiv:1606.02447 [cs].

Xingrui Wang, Wufei Ma, Zhuowan Li, Adam Kortylewski, and Alan Yuille. 2023. 3d-aware visual question answering about parts, poses and occlusions. *arXiv preprint*.

Xinyu Wang, Bohan Zhuang, and Qi Wu. 2025. Are large vision language models good game players? *Preprint*, arxiv:2503.02358 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Qiucheng Wu, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. 2024. VSP: Assessing the dual challenges of perception and reasoning in spatial planning tasks for VLMs. *Preprint*, arxiv:2407.01863 [cs].

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, and 1 others. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Yutaro Yamada, Yihan Bao, Andrew K. Lampinen, Jungo Kasai, and Ilker Yildirim. 2024. Evaluating spatial understanding of large language models. *Preprint*, arxiv:2310.14540 [cs].

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Fanlong Zeng, Wensheng Gan, Yongheng Wang, Ning Liu, and Philip S Yu. 2023. Large language models for robotics: A survey. *arXiv preprint arXiv:2311.07226*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *Preprint*, arXiv:2303.15343.

Wenqi Zhang, Zhenglin Cheng, Yuanyu He, Mengna Wang, Yongliang Shen, Zeqi Tan, Guiyang Hou, Mingqian He, Yanna Ma, Weiming Lu, and Yueting Zhuang. 2024. Multimodal self-instruct: Synthetic abstract image and visual reasoning instruction using language model. *Preprint*, arxiv:2407.07053 [cs].

Baining Zhao, Ziyou Wang, Jianjie Fang, Chen Gao, Fanhang Man, Jinqiang Cui, Xin Wang, Xinlei Chen, Yong Li, and Wenwu Zhu. 2025. Embodied-r: Collaborative framework for activating embodied spatial reasoning in foundation models via reinforcement learning. *Preprint*, arxiv:2504.12680 [cs].

A Appendix

A.1 Episode Details

A.1.1 System Prompt Instructions

Interactive Sliding Geom Puzzle Game

You are a highly intelligent AI solving a shape puzzle on a 4x4 grid. The board has two states: the current active state and the goal state. Your task is to generate valid actions that transform the current state into the goal state along the shortest path.

Steps:
(1) Analyze current state.
(2) Compare to goal.
(3) Check past actions.
(4) Propose next move.

Movement Rules: Each object occupies one tile. Objects cannot leave the grid or overlap.

Action Format: `move <color> <shape> <direction>`

Use only the following:
Colors: green, red, blue, yellow
Shapes: cube, sphere, pyramid, cylinder
Directions: up, down, left, right

Examples: `move green cube down`, `move red pyramid left`

Important: No coordinates. Each action must change the state. Invalid if blocked or out of bounds.

Explain Reasoning: Before suggesting an action, explain why. End with:
`action: move <color> <shape> <direction>`
(no extra characters after `action: ...`)

Visual Input:
Current: `{text_snippet_active}`;
Goal: `{text_snippet_goal}`;
Past: `{text_snippet_past}`.

Final Requirement: Always end your output with:
`description:<your object coordinate list>`
Do not add characters after the word `description`.

Board State Inference Auxiliary Task

You are a highly intelligent AI with exceptional spatial reasoning skills, and you are given the following task:

## Task Overview

1. You are provided with an input image of colored geometric objects on a 4 × 4 board.
2. Analyze the current board state and locate the position of all objects on the board.
3. Respond with a list of the chess-style coordinates and their objects.

## Board Overview

The board has labeled columns, rows, and fields

- Columns a–d run from left to right in the image.
- Rows 1–4 run from bottom to top in the image.

## Object Overview

- On the board are various objects, uniquely defined by their color and shape:
  - **Colors:** green, red, blue, yellow
  - **Shapes:** cube, sphere, pyramid, cylinder

## Solution Format

- Start your solution with ‘`Solution: `’ and list each object in any order, separated by a comma and a single space (‘, ‘).
- Your solution for each object must follow this exact format: `<coordinate> <object_color> <object_shape>`
  - `coordinate` must use a letter a–d followed by a digit 1–4.
  - `object_color` must be exactly one of: green, red, blue, yellow.
  - `object_shape` must be exactly one of: cube, sphere, pyramid, cylinder.
- Only list coordinates that contain an object; do not mention empty squares.
- Do not use quotation marks or angle brackets < > in your action.
- Do not include any extra text, reasoning, or punctuation after the formatted list.

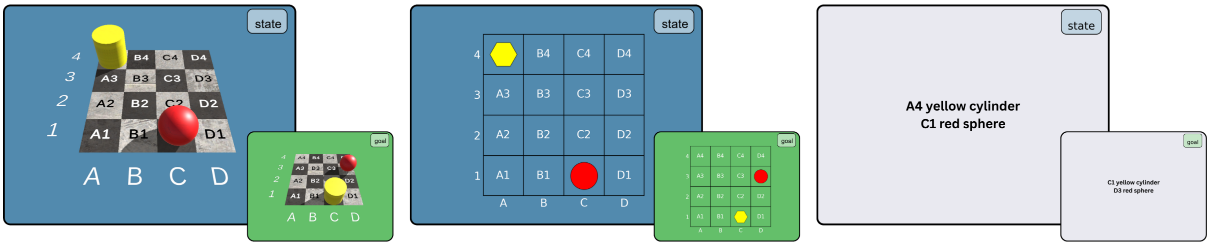
## Example

Solution: a3 green sphere, d1 blue cylinder, b4 yellow cube, c2 red pyramid

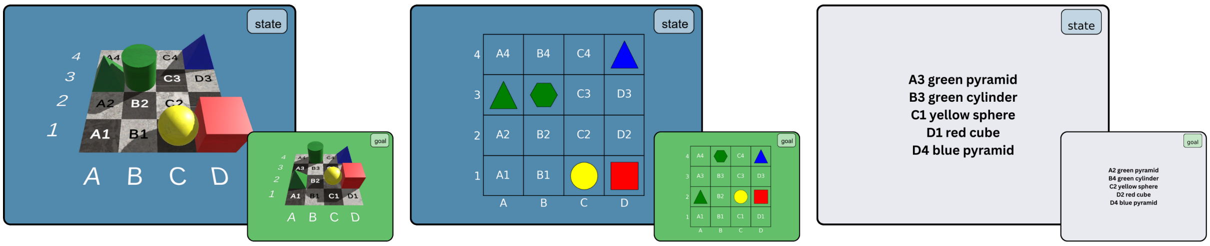
## Validation

- No two objects share the same coordinate.
- Every listed object uses one of the four allowed colors and shapes.

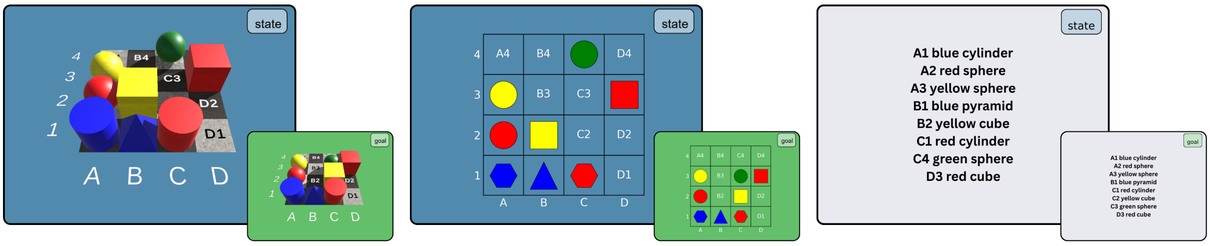
A.1.2 Observations of Scaling Episode Complexity



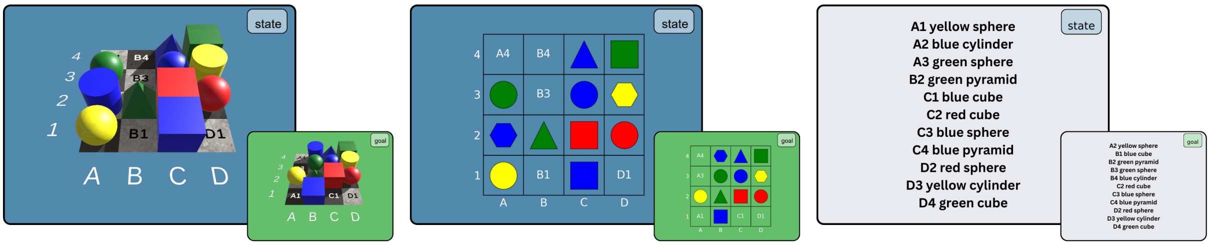
Sample from the **Geom Board Environment** in our dataset (ds25a), showing a state (blue) and the goal (green) at each step of an episode on a **4×4 board**, with **2 geoms** and an optimal **path length of 8**. All modalities are shown.



Sample from the **Geom Board Environment** in our dataset (ds25a), showing a state (blue) and the goal (green) at each step of an episode on a **4×4 board**, with **5 geoms** and an optimal **path length of 4**. All modalities are shown.



Sample from the **Geom Board Environment** in our dataset (ds25a), showing a state (blue) and the goal (green) at each step of an episode on a **4×4 board**, with **8 geoms** and an optimal **path length of 2**. All modalities are shown.

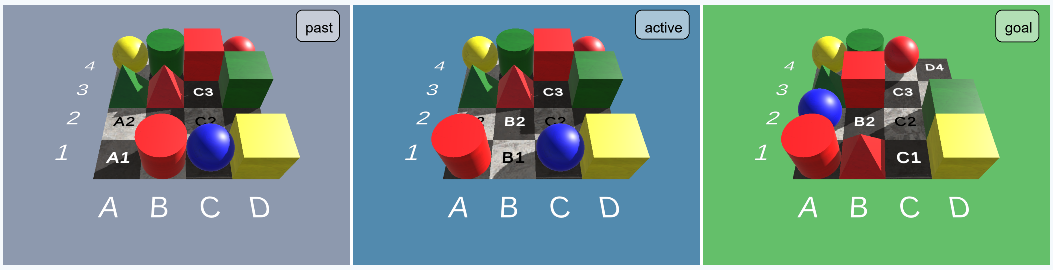


Sample from the **Geom Board Environment** in our dataset (ds25a), showing a state (blue) and the goal (green) at each step of an episode on a **4×4 board**, with **11 geoms** and an optimal **path length of 6**. All modalities are shown.

Figure 8: Examples of VLMs' observations for a state (blue) and the goal (green) at each step during an episode of the Sliding Geom Puzzle environment, on a 4×4 board with 2, 5, 8 and 11 geoms and an optimal path length of 8, 4, 2, and 6 respectively. Left to right, each tested modality: vision 3D, vision 2D, and text-based representation.

A.1.3 Interaction Example

Input Example



## Analyze the Images
You can view your current active board state in the last image marked with the label *active* in the image and a blue background. Study this image and the objects with their positions carefully.

Your goal is to match the goal state, shown in the image marked with the label *goal* in the image and a green background. Study this image and the objects with their positions carefully.

## Additionally, you are provided with:

- The previous state image(s) marked with the label *past* in the image and a grey background.
- Your previous suggested action
- Use this information by comparing it to your current active state to determine your next action.

## Invalid Actions:

- No Overlap: You are not allowed to position two objects in the same tile.
- If the suggested action does not move any objects, it is invalid (e.g., blocked by another object or out of bounds).
- Use the previous image(s) and action to understand why it failed and suggest a different action.

It is of most importance you always end your response with this exact format:

```
action: move <object color> <object shape> <direction>
```

where you replace `<object color> <object shape> <direction>` with the valid move action based on your reasoning and do not add any characters after your action.

Output Example

To reach the goal state I need to compare the current active state with the goal state.

Current active state

- In the current active state the green sphere is on top blocking movements of objects below it.
- The yellow sphere is on the bottom row B1.

Goal state

- In the goal state the green sphere is two steps down in column B.
- The red cylinder is in the bottom left, two steps down.
- The yellow sphere is in position B2, two steps up.

Analyzing the images I see that the yellow spheres path is blocked by the green cylinder I'll first move another object to clear the way. Moving the blue cube rightward will help clear the blocked pathway for the green cylinder.

Action: Move the blue cube right

A.1.4 Episode Progression Example

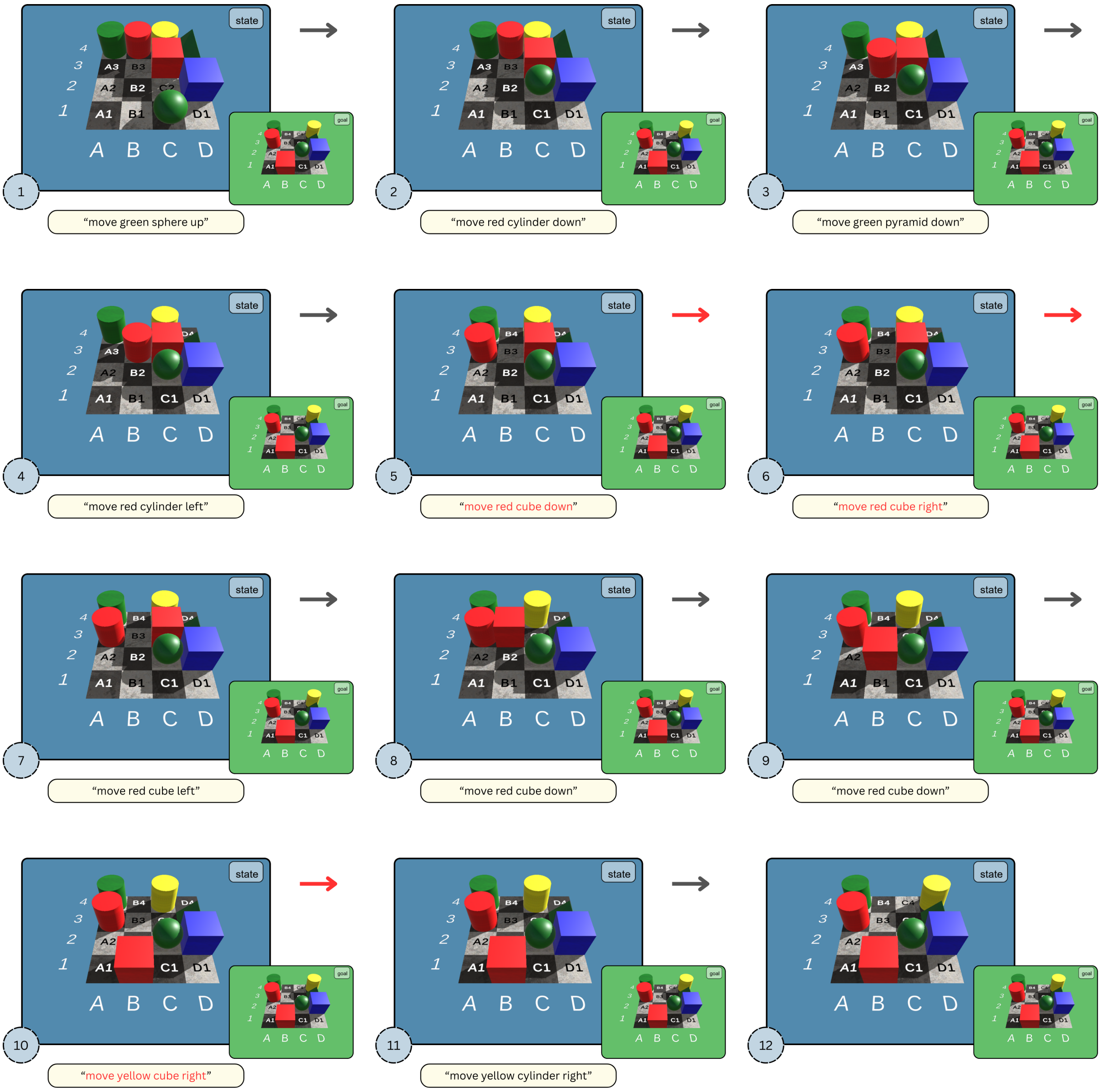


Figure 9: Example of an episode progression for an environment in vision 3D (other modalities progress analogously) with an optimal path length of 9, showing steps 1 to 12 in order, including 3 mistakes (red action text).

A.2 Models

Name	LLM	Vision Encoder	Model Size
Closed Source Models			
Sonnet-3.5 (Claude Team, 2024)	–	–	–
Gemini-2.0-flash (Gemini Team, 2024)	–	–	–
GPT-4o (OpenAI et al., 2024)	–	–	–
Open Source Models			
InternVL 2.5 (Chen et al., 2024)	Qwen 2.5 (Yang et al., 2024b)	InternViT (Chen et al., 2024)	78.4B
LLaVA OneVision (Li et al., 2024a)	Qwen 2 (Yang et al., 2024a)	SigLIP (Zhai et al., 2023)	73.2B
Qwen 2 VL (Wang et al., 2024d)	Qwen 2 (Yang et al., 2024a)	ViT (Dosovitskiy et al., 2020)	73.4B

Table 1: Overview of evaluated models. – indicates unavailable information.

A.3 Sliding Tile Puzzle

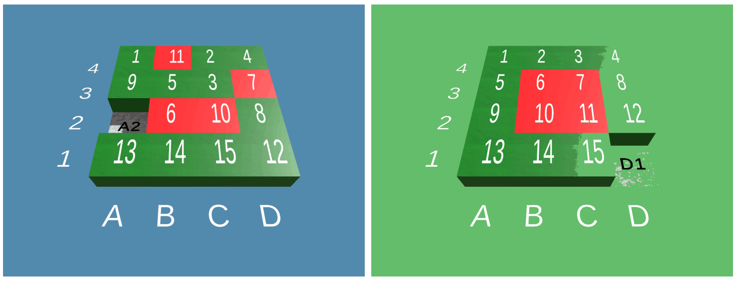


Figure 10: Visualization of a current state and the goal state in a classic 15-tile Sliding Tile Puzzle (STP) on a 4×4 board, playable by agents within the iVISPAR benchmark.

The sequential generalized sliding-tile puzzle (SGSTP) is a generalization of the classic 15-Tile Sliding Tile Puzzle (STP), see Figure 10. In the SGSTP, a set of $n < m_1 \times m_2$ tiles, each uniquely labeled $1, \ldots, n$, are placed on a rectangular grid of size $m_1 \times m_2$, denoted by $G = (V, E)$. The grid has $m_1 \times m_2 - n$ empty positions that allow tile movement.
A configuration of tiles is represented as an injective mapping from the set $\{1, \ldots, n\}$ to positions $V = \{(v_x, v_y) : 1 \leq v_x \leq m_2, 1 \leq v_y \leq m_1\}$. Each tile must be repositioned from an arbitrary initial configuration $S = \{s_1, \ldots, s_n\}$ to a specified goal configuration $G = \{g_1, \ldots, g_n\}$, such as an ordered row-major layout.
Let the movement path of tile i, where $1 \leq i \leq n$, be expressed as $p_i : \mathbb{N}_0 \rightarrow V$. The puzzle seeks a set of feasible paths $P = \{p_1, \ldots, p_n\}$ that satisfy the following conditions for all $1 \leq i, j \leq n$ with $i \neq j$, and for all time steps $t \geq 0$:
Incremental Movement: $p_i(t+1) = p_i(t)$ or $(p_i(t+1), p_i(t)) \in E$. Tiles move to adjacent, unoccupied positions or stay still.
Goal Achievement: $p_i(0) = s_i$ and $p_i(T) = g_i$ for some $T \geq 0$. Each tile must start at s_i and reach g_i.
Exclusive Occupancy: $p_i(t) \neq p_j(t)$ for all $i \neq j$. Two tiles cannot occupy the same position at the same time.
In this sequential version, tiles move one at a time. Therefore, the head-on collision and corner-following constraints found in the generalized sliding-tile puzzle are omitted, as simultaneous tile movements are not permitted.

A.4 Detailed Results

A.4.1 Performance Results

Model	**Metric**	**Avg**	**3D**	**2D**	**Text**
Closed Source Models					
Sonnet-3.5	Completed episodes	54.56	28.67	89.67	45.33
	Optimal path deviation	3.05	4.10	1.44	3.60
	Board state inference	60.00	35.38	84.62	–
Gemini-2.0-flash	Completed episodes	27.11	12.67	47.33	21.33
	Optimal path deviation	4.87	5.25	4.09	5.26
	Board state inference	54.08	28.67	79.49	–
GPT-4o	Completed episodes	17.56	9.33	37.33	6.00
	Optimal path deviation	5.30	5.45	4.15	6.30
	Board state inference	41.67	19.49	63.85	–
Open Source Models					
InternVL2.5-78B	Completed episodes	10.16	1.67	9.42	19.33
	Optimal path deviation	5.98	6.39	5.86	5.69
	Board state inference	34.95	16.51	53.38	–
LLaVA-OneVision-72B	Completed episodes	8.22	0.67	1.33	22.67
	Optimal path deviation	6.35	6.75	6.81	5.50
	Board state inference	26.36	14.72	38.00	–
Qwen2-72B	Completed episodes	5.89	0.67	1.67	15.33
	Optimal path deviation	6.37	6.66	6.54	5.90
	Board state inference	41.54	18.77	64.31	–
Aggregate Averages					
Average	Completed episodes	20.59	7.04	26.68	21.83
	Optimal path deviation	5.32	5.76	4.41	5.32
	Board state inference	43.10	22.26	63.94	–

Table 2: Evaluation of models across three modalities. Each row shows average episode completion rate (%), mean deviation from the optimal path (see Section 4.6), and board state inference accuracy (%).

A.4.2 Error Counts for the Geom Puzzle

Model	Metric	Avg	3D	2D	Text
Closed Source Models					
Sonnet-3.5	EM	6.31	6.51	6.20	6.21
	IM	1.86	3.34	0.21	2.03
	OD	3.60	4.77	2.29	3.75
	OB	1.59	1.95	0.04	2.79
	IC	0.02	0.07	0.00	0.00
Gemini-2.0-flash	EM	5.68	5.80	6.34	4.91
	IM	2.95	3.87	2.35	2.63
	OD	6.14	6.83	5.51	6.08
	OB	2.56	2.25	1.11	4.33
	IC	0.01	0.01	0.00	0.03
GPT-4o	EM	4.65	5.50	5.95	2.51
	IM	2.86	4.03	2.36	2.19
	OD	6.53	6.36	5.51	7.71
	OB	3.81	2.69	1.85	6.90
	IC	0.26	0.24	0.52	0.03
Open Source Models					
InternVL2.5-78B	EM	5.00	4.94	5.74	4.39
	IM	4.24	5.39	4.80	2.59
	OD	5.90	6.06	5.70	5.92
	OB	3.38	3.16	2.52	4.38
	IC	0.59	0.21	0.29	1.26
LLaVA-OneVision-72B	EM	3.95	3.41	3.22	5.23
	IM	4.12	4.55	4.42	3.40
	OD	4.89	4.58	4.74	5.36
	OB	4.17	4.62	4.46	3.44
	IC	1.38	1.90	2.19	0.07
Qwen2-72B	EM	4.07	3.88	3.96	4.85
	IM	4.61	4.81	4.67	3.89
	OD	5.39	5.55	5.21	5.25
	OB	3.72	4.05	3.17	3.83
	IC	0.10	0.07	0.06	0.26
Aggregate Averages					
Average	EM	4.82	4.72	5.04	4.68
	IM	3.61	4.45	3.34	2.79
	OD	5.40	5.66	4.87	5.68
	OB	3.28	3.35	2.33	4.28
	IC	0.35	0.33	0.45	0.28

Table 3: Evaluation of models across three modalities. Each row shows average steps per episode that were effective moves (EM), ineffective moves (IM), occupied destination moves (OD), out of bounds moves (OB) and illegal commands (IC).

A.4.3 Error Counts for the Auxiliary Task

Model	**Metric**	**Avg**	**3D**	**2D**	**Text**
Closed Source Models					
Sonnet-3.5	Correct	3.90	2.30	5.50	–
	Missed	1.42	1.84	1.00	–
	Hallucinated	0.00	0.00	0.00	–
	Coord Errors	1.08	2.16	0.00	–
	Color Errors	0.38	0.76	0.00	–
	Shape Errors	0.37	0.74	0.00	–
	Format Errors	0.00	0.00	0.00	–
Gemini-2.0-flash	Correct	3.52	1.86	5.17	–
	Missed	0.91	1.02	0.80	–
	Hallucinated	0.14	0.13	0.14	–
	Coord Errors	1.98	3.48	0.48	–
	Color Errors	0.66	1.14	0.18	–
	Shape Errors	0.65	1.14	0.16	–
	Format Errors	0.05	0.00	0.09	–
GPT-4o	Correct	2.71	1.27	4.15	–
	Missed	1.31	1.67	0.95	–
	Hallucinated	0.03	0.01	0.04	–
	Coord Errors	2.34	3.33	1.35	–
	Color Errors	0.77	1.18	0.35	–
	Shape Errors	0.75	1.18	0.32	–
	Format Errors	0.02	0.04	0.00	–
Aggregate Averages					
Average	Correct	3.37	1.81	4.94	–
	Missed	1.21	1.51	0.92	–
	Hallucinated	0.06	0.05	0.06	–
	Coord Errors	1.80	2.99	0.61	–
	Color Errors	0.60	1.03	0.18	–
	Shape Errors	0.59	1.02	0.16	–
	Format Errors	0.02	0.01	0.03	–

Table 4: Error analysis for the auxiliary position inference task across vision modalities (closed source models).

Model	Metric	Avg	3D	2D	Text
Open Source Models					
InternVL2.5-78B	Correct	2.27	1.07	3.47	–
	Missed	0.89	1.00	0.77	–
	Hallucinated	0.03	0.04	0.01	–
	Coord Errors	1.62	2.92	0.32	–
	Color Errors	0.59	1.11	0.08	–
	Shape Errors	0.58	1.08	0.08	–
	Format Errors	1.63	1.30	1.97	–
LLaVA-OneVision-72B	Correct	1.71	0.96	2.47	–
	Missed	1.02	1.18	0.86	–
	Hallucinated	0.34	0.31	0.37	–
	Coord Errors	3.30	3.95	2.65	–
	Color Errors	1.28	1.58	0.97	–
	Shape Errors	1.23	1.57	0.90	–
	Format Errors	0.37	0.09	0.65	–
Qwen2-72B	Correct	2.70	1.22	4.18	–
	Missed	0.97	1.08	0.85	–
	Hallucinated	0.58	0.81	0.36	–
	Coord Errors	2.52	3.80	1.24	–
	Color Errors	0.93	1.42	0.43	–
	Shape Errors	1.12	1.67	0.58	–
	Format Errors	0.22	0.06	0.38	–
Aggregate Averages					
Average	Correct	2.23	1.08	3.37	–
	Missed	0.96	1.09	0.83	–
	Hallucinated	0.32	0.39	0.25	–
	Coord Errors	2.48	3.56	1.40	–
	Color Errors	0.93	1.37	0.49	–
	Shape Errors	0.98	1.44	0.52	–
	Format Errors	0.74	0.48	1.00	–

Table 5: Error analysis for the auxiliary position inference task across vision modalities (open source models).

A.5 Supplementary Graphs

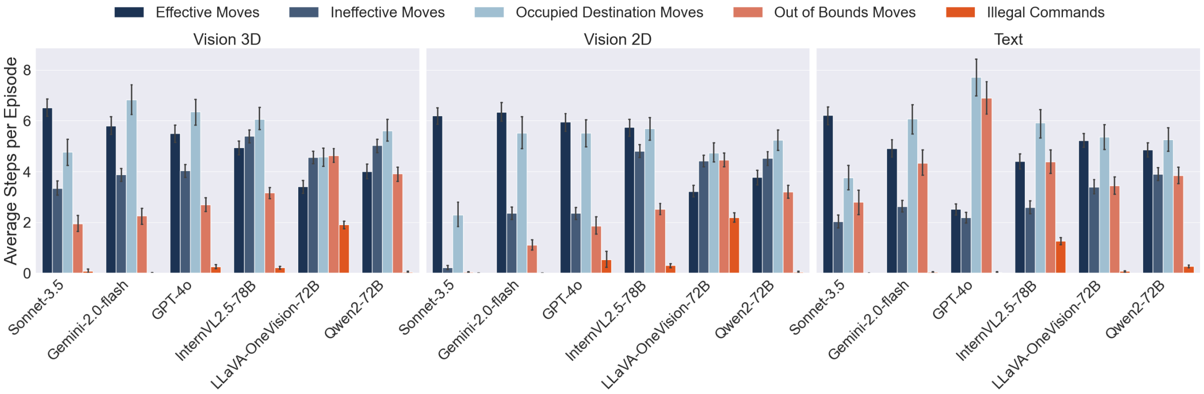


Figure 11: VLMs' average action counts per episode by category for each modality. Number of actions per episode is capped at 20. Effective / ineffective actions respectively decrease / increase the path length to the goal state. Occupied destination and out-of-bounds are invalid moves, while illegal commands break the instructed action format, all of which leave the board state unchanged.

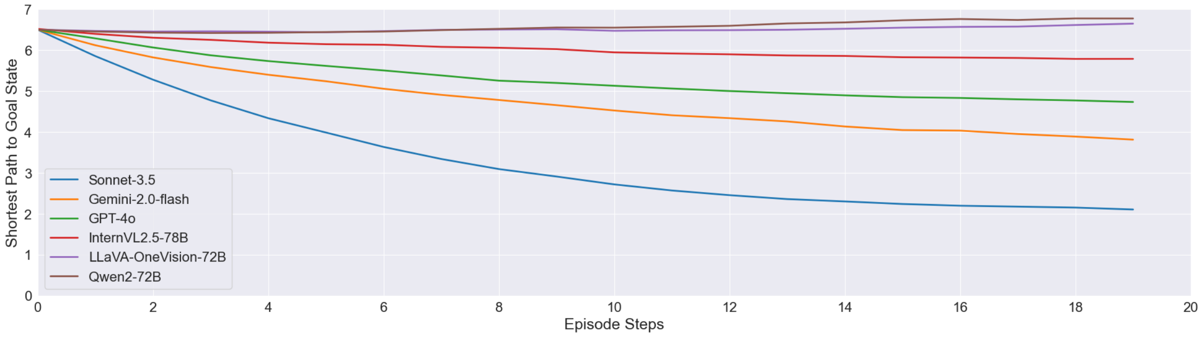


Figure 12: VLMs' average shortest path to the goal state across all modalities. Number of actions per episode is capped at 20.

A.6 Additional Agent Interaction Data

A.6.1 Systematic Formatting Errors

Unless noted otherwise, the numeral in parentheses after a model name is the *count of formatting errors* for that category. Notably, Sonnet-3.5 is not listed since it did not make any format errors, explaining its high benchmarking score.

(E1) Empty-cell mentions (N = 280)
The most common violation is the explicit listing of empty grid cells, even though instructions forbid any mention of empties. Surface forms vary widely, even within a single model:

> a4 empty

> c3 blank

> b1 no object

> a1 none none

Gemini-2.0-flash (24), InternVL-2.5-78B (21), LLaVA-OneVision-72B (105), Qwen2-72B (130).

(E2) Missing attributes (N = 88)
Some lines list an object but drop one of its required attributes (colour or shape):

> c1 none pyramid

> b2 sphere

Gemini-2.0-flash (2), LLaVA-OneVision-72B (86).

(E3) Illegal attributes (N = 21)
Entries introduce colours or shapes outside the predefined vocabulary, or mis-name legitimate ones:

> b2 black cone

> b3 red block

Gemini-2.0-flash (2), GPT-4o (9), LLaVA-OneVision-72B (9), Qwen2-72B (1).

(E4) Attribute order inversion (N = 21)
Colour–shape order is sometimes reversed:

> b4 cylinder yellow

LLaVA-OneVision-72B (21)

(E5) Multiple attributes per object (N = 3)
A single coordinate is assigned more than one colour or a compound colour phrase:

> d4 blue and red cube

> d1 multicoloured cube

> a2 a3 green pyramid

GPT-4o (1), InternVL-2.5-78B (1), Qwen2-72B (1).

(E6) Out-of-bounds coordinates (N = 3)
Objects occasionally appear outside the 4 × 4 grid, e.g. likely due to OCR misreads:

> f4 yellow pyramid

> c5 green pyramid

Gemini-2.0-flash (1), InternVL-2.5-78B (1), LLaVA-OneVision-72B (1).

(E7) Non-list answers (N = 47)
A few generations present free-form summaries or spurious URLs instead of the required flat list:

> d2 blue cylinder summary a4 yellow cylinder b4 blue sphere . . .

> d3 green sphere https user images githubusercontent com /.../f45e3211ecb93a6b429446.png

> obj in objectsitems if obj none output

> yellow objectshape must be exactly one of cube

> we will create the solution

InternVL-2.5-78B (47).

(E8) Solution not found (N = 147)
Prompts that did not begin their solution with "Solution: " are not picked up by the solution parser. GPT-4o (1), InternVL-2.5-78B (145), Qwen2-72B (1).

A.6.2 Chain of Thoughts

Sonnet-3.5 CoT Example

Looking at the current state and goal state:

1. The blue cube is correctly positioned in A2 in both states.
2. The yellow pyramid needs to move from D1 to C1.
3. The red cylinder is in D2 and needs to be in D3.

The yellow pyramid needs to be moved to make space for other movements. Looking at the past states we tried moving the yellow pyramid right and the red cylinder up. Now we need to move the yellow pyramid to its goal position. I can see that the yellow pyramid should move left to reach position C1 as shown in the goal state.

Qwen2-72B CoT Example

After analyzing the images, it is clear that moving the red sphere down was an invalid action because it resulted in no change to the board state. The red sphere is still in the same position, and the green cylinder has not moved.

To progress, I need to find a valid action that will move one of the objects to match the goal state.

Upon further inspection, I notice that the red sphere needs to be moved to the top-right corner of the board to match the goal state. However, the current state shows the red sphere in the bottom-right corner, which means I need to move it up to get it closer to the goal position.

GPT-4o CoT Example

To reach the goal state I need to compare the current active state with the goal state.

In the current active state the green sphere is on top blocking movements of objects below it. The yellow sphere is on the bottom row B1.

In the goal state the green sphere is two steps down in column B. The red cylinder is in the bottom left, two steps down. The yellow sphere is in position B2, two steps up.

Analyzing the images I see that the yellow spheres path is blocked by the green cylinder I'll first move another object to clear the way. Moving the blue cube rightward will help clear the blocked pathway for the green cylinder.

InternVL-2.5-78B CoT Example

to reach the goal state, let's analyze the current board and the objects that need to be moved:

current state (active)

- the red cylinder is on tile b4.
- the yellow pyramid is on tile d4.
- the blue cylinder is on tile c3.
- the yellow cube is on tile d2.

goal state (goal)

- the red cylinder needs to be on tile a4.
- the blue cylinder needs to be on tile b4.
- the yellow cube needs to be on tile d3.
- the yellow pyramid needs to be on tile c4.

step-by-step plan

1. move the red cylinder to a4:
- action: move the red cylinder left.

2. move the yellow pyramid to c4:
- action: move the yellow pyramid down.

3. move the yellow cube to d3:
- action: move the yellow cube down.

4. move the blue cylinder to b4:
- action: move the blue cylinder up.

first action

the first step is to move the red cylinder from tile b4 to tile a4.

Gemini-2.0-flash CoT Example

The green sphere needs to move from B2 to C3. It is currently at B2, moving up would move the green sphere to B3 where it is blocked by the red pyramid.

I need to move the green sphere to the right, then up. Its already been moved right once, so my next action is to move the green sphere up.

LLaVA-OneVision-72B CoT Example

The blue cube on `D2` needs to move to the position of the red cylinder on `D1`.

Since the red cylinder cannot be displaced, the blue cube must be moved elsewhere.

Moving the blue cube left to `C2` does not change the board state because another blue cube already occupies that square.

Moving it up to `D1` would push the red cylinder out of bounds, which is not allowed.

Therefore, the blue cube on `D2` should move left to `D3`.

A.7 Algorithms

A.7.1 Optimal Agent

Algorithm 1 A* Algorithm for Pathfinding

```
Require: start, goal
Ensure: Path from start to goal or failure
openSet ← {start}
cameFrom ← empty map
gScore[start] ← 0
fScore[start] ← heuristic(start, goal)
while openSet not empty do
    current ← node in openSet with lowest fScore
    if current = goal then
        return ReconstructPath(cameFrom, current)
    end if
    Remove current from openSet
    for each neighbor of current do
        tentativeGScore ← gScore[current] + d(current, neighbor)
        if tentativeGScore < gScore[neighbor] or neighbor not in gScore then
            cameFrom[neighbor] ← current
            gScore[neighbor] ← tentativeGScore
            fScore[neighbor] ← gScore[neighbor] + heuristic(neighbor, goal)
            if neighbor not in openSet then
                Add neighbor to openSet
            end if
        end if
    end for
end while
return failure
```

A.7.2 Random Agent

Algorithm 2 Generate Random Valid Path for Sliding Tile Puzzle

```
Require: $n$ (board size), initial_state, max_steps
Ensure: Path from initial to final state
path ← [initial_state]
current_state ← initial_state
for step = 1 to max_steps do
    neighbors ← get_neighbors(current_state, $n$)
    current_state ← random choice from neighbors
    Append current_state to path
end for
    return path
```