ASPIRE: Bridging the Gap Between Visual Perception and Spatial Agency

Anonymous ACL submission

Abstract

While Multimodal Large Language Models (MLLMs) have demonstrated impressive capabilities in passive visual perception and question answering, their ability to *actively* manipulate and design spatial environments remains a critical open challenge. As generative agents transition from text-based tasks to complex GUI and whiteboard operations, the distinction between "seeing" a spatial layout and effectively "acting" upon it becomes paramount. To bridge this gap, we present ASPIRE, a benchmark for Agentic Spatial Performance in Whiteboard Environments. Unlike previous benchmarks that rely on static multiple-choice questions, ASPIRE evaluates agents on open-ended state manipulation, requiring them to create, update, and organize visual elements to satisfy spatial constraints. Our extensive evaluation of state-of-the-art models reveals a fundamental dichotomy: while agents excel at discrete, structured tasks (e.g., maze navigation, graph coloring) where visual data can be mapped to symbolic logic, they struggle significantly with continuous, intuitive reasoning (e.g., visual balance, angular rotation). Furthermore, our ablation studies uncover a "scaffolding paradox": providing visual aids such as grids or polar plots often degrades performance, suggesting that current MLLMs rely heavily on semantic metadata rather than robust visual-spatial grounding.

1 Introduction

The rapid advancement of Multimodal Large Language Models (MLLMs), exemplified by state-of-the-art systems such as Claude 4.5 Opus (Anthropic, 2025), GPT-5 (OpenAI, 2025), and Gemini 3.0 Pro (Deepmind, 2025), has catalyzed a fundamental shift in artificial intelligence: the transition from passive chatbots and analysis tools to active generative agents. By leveraging sophisticated control structures and tool use, these agents have graduated from toy environments (Park et al., 2023) to tackling complex, high-stakes tasks. Today's agents are being integrated into operating systems to navigate GUIs (Gou et al., 2024; Xie et al., 2024), deployed as autonomous software engineers (Zhang et al., 2025), and utilized as educational assistants in virtual laboratories (Chu et al., 2025; Liu et al., 2025). Key to the functionality of these systems is their ability not just to process information, but to evaluate, understand, and operationally modify complex environments.

As model input modalities expand to include high-resolution visual input, opportunities for more complex agentic integrations arise. However, this expansion brings a critical need for scrutiny. When considering real-world agentic tasks, such as designing a presentation, organizing a dashboard, or illustrating a concept, spatial reasoning plays a paramount role. Beyond the literal understanding of geometric relationships, spatial reasoning extends into the domain of *spatial metaphor*. Humans use space to structure and interact with abstract data, imbuing deeper meaning through proximity, grouping, and visual flow (Pitt and Casasanto, 2022). Consequently, for an AI agent to be a true collaborative partner, it must possess "Spatial Agency": the ability to actively leverage visual spatial metaphors to create robust, interpretable visualizations.

While there has been extensive study of MLLM spatial reasoning, with models showing potential in navigational tasks and static information extraction (Li et al., 2024b; Liu et al., 2023; Wang et al., 2024a; Shiri et al., 2024; Yang et al., 2025b; Hao et al., 2025; Hu et al., 2024), there is a distinct lack of fine-grained agentic study. Current benchmarks primarily evaluate *passive perception* by asking models to select the correct answer from a multiple-choice list or describe a static image. They fail to effectively evaluate how an agent uses visualization tools to *actively understand and alter* a spatial environment. An agent that can correctly answer "Is the circle to the left of the square?" may still fail to "Draw a circle to the left of the square in a visually balanced way".

To address this deficiency, we present **ASPIRE**: A Benchmark for **A**gentic **S**patial **P**erformance in **I**nteractive **R**easoning **E**nvironments. We propose that a digital whiteboard represents the ideal spatial sandbox for evaluating this skill. Whiteboards, from browser-based tools like tldraw (tldraw, 2025) and Excalidraw (excalidraw, 2025) to tablet-based solutions (Microsoft, 2025), are the native medium for brainstorming and diagramming. Their operational structures (create, update, delete) align perfectly with the agentic tool paradigm, allowing us to isolate spatial reasoning from the noise of complex operating systems.

Through ASPIRE, we examine how agents act on open-ended spatial problems. Our extensive evaluation reveals an interesting phenomenon: models excel at tasks that can be serialized into discrete mathematical logic (e.g., graphs, mazes) but struggle significantly with tasks requiring continuous visual intuition (e.g., balance, rotation). Furthermore, our analysis of input modalities and prompting techniques uncovers a *scaffolding paradox*, where explicit visual aids often degrade rather than enhance performance.

In summary, our contributions are as follows:

- We introduce ASPIRE, a novel benchmark suite for MLLM Agent Spatial Understanding that moves beyond passive Q&A to active state manipulation.
- We test agentic reasoning through *open-ended* questions, evaluating performance across fundamental spatial building blocks, visualization tasks, and visual understanding.
- We analyze the effect of image-only input versus multimodal state, revealing critical dependencies in how agents process spatial data.
- We evaluate the impact of Chain-of-Thought (CoT) prompting and reasoning-heavy inference modes, identifying trade-offs between token efficiency and spatial accuracy.

2 Background and Motivation

2.1 Multimodal LLMs

As the emergent capabilities of LLMs have expanded, the integration of non-text modalities has become a central focus of research. While early attempts to process visual data involved converting images into descriptive text with limited success (Wu et al., 2023a), modern MLLMs focus on encoding input into a common feature space. This allows for more nuanced handling of multimodal data and enables effective attention mechanisms between different modalities (Zhang et al., 2024a).

However, utilizing a common feature space does not guarantee effective reasoning. The process of extracting specific spatial information from dense image inputs remains critical. To address this, recent research has emphasized prompting strategies that help models focus on relevant visual regions (Shao et al., 2024; Yang et al., 2023). A growing trend to support reasoning is the addition of *visual scaffolds*, overlays that explicitly add information such as relevant labels, coordinates, and angles to the visual field (Wu et al., 2024; Lei et al., 2025; Hu et al., 2024). While these methods have shown promise in passive recognition tasks, it remains an open question whether they aid or hinder agentic performance where the model must actively generate spatial actions.

2.2 Generative Agents

Generative agents have evolved significantly from small toy models in controlled environments (Park et al., 2023) to complex systems capable of navigating high-level interfaces. By leveraging control structures, planning, memory, and tool functions, these agents can now operate within complex problem classes (Wang et al., 2024b; Li et al., 2024c). Recent work has demonstrated agents navigating GUI systems and interacting with computers through the human interface (Gou et al., 2024; Rawles et al., 2024; Zhang et al., 2025; Xie et al., 2024), as well as tackling domain-specific problems such as spatial biology (Wang et al., 2025) and educational roles like teaching and laboratory assistants (Chu et al., 2025; Schmidgall et al., 2025; Liu et al., 2025).

Due to the complexity of these systems, agentic understanding must be extended through thoughtful, in-depth benchmarking (Gioacchini et al., 2024; Wu et al., 2023b). With existing agents leveraging broad foundational models, efforts to build better agent-focused foundation models utilize comprehensive benchmarks to improve accuracy (Yang et al., 2025a). Furthermore, comprehensive design of spatial benchmarks can benefit model accuracy across spatial design domains when used for training (Song et al., 2025; Feng et al., 2025; Li et al., 2025). However, because effective tool use and action generation are paramount, benchmarks must move beyond static evaluation to consider how tools are leveraged within the application context.

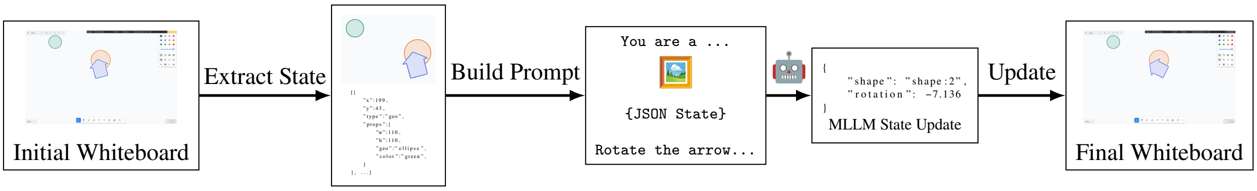


Figure 1: The ASPIRE Agent Pipeline

2.3 The Gap: Passive Perception vs. Active Agency

Spatial understanding poses unique challenges for MLLMs. For text-only LLMs, understanding must arise from information extracted via natural language. While models show potential in navigating spatial structures (Feng et al., 2025), the quality of this understanding is heavily affected by context length, complexity, and information ordering (Yamada et al., 2023; Shi et al., 2022). Specialized grammars can improve performance (Mirzaee et al., 2021), yet establishing effective context in complex environments remains a failure point for reasoning over real-world spatial tasks (Xing et al., 2024).

In the multimodal domain, spatial reasoning is often included as a sub-component of visual understanding benchmarks (Fu et al., 2024; Kil et al., 2024; Hao et al., 2025; Li et al., 2024a; Liao et al., 2024). Yet, adding vision capabilities does not inherently improve spatial reasoning without careful guidance. Performance is highly sensitive to perspective and reference frames, with models performing better when queried from a camera perspective rather than a subject perspective (Wang et al., 2024a; Liu et al., 2023; Shiri et al., 2024). While techniques like Chain-of-Thought (CoT) and cognitive map construction can boost results (Li et al., 2024b; Yang et al., 2025b), models still struggle with reasoning compared to simple recognition and localization (Li et al., 2024b; Hao et al., 2025). Crucially, these existing works primarily focus on *passive* evaluation, using multiple choice questioning or static information extraction. They do not evaluate *active spatial agency*, where an agent must understand and alter a scene to satisfy open-ended constraints.

2.4 Digital Whiteboards as Spatial Sandboxes

To address this deficiency, we propose utilizing digital whiteboards for evaluation. Tools like tldraw (tldraw, 2025) and Excalidraw (excalidraw, 2025), as well as tablet-based applications like Microsoft Whiteboard (Microsoft, 2025), are compelling spaces for testing agentic spatial reasoning. Whiteboards require inherent spatial skills; effective diagrams rely on utilizing spatial relationships and metaphors, such as proximity, containment, layering, and visual balance, to structure information (Pitt and Casasanto, 2022).

As illustrated in our pipeline (Figure 1), this environment allows us to build a robust loop of observation, reasoning, and action. By interfacing with the JSON APIs of these tools, we can evaluate how an agent extracts state, builds a prompt, infers spatial relationships, and executes updates. This environment provides a clear space to break down how an agent leverages spatial reasoning with tool use for updating a scene, creating the ideal foundation for ASPIRE.

3 ASPIRE

Rather than evaluating spatial understanding in a rigid environment, our benchmark focuses on agentic reasoning with open-ended problem solving. To accomplish this, we build a robust interactive whiteboard. We minimize environmental restrictions, allowing agents to operate on the whiteboard with the same actions as human users. This allows us to directly evaluate the problem solving ability in the spatial domain.

3.1 Metrics

A common limitation in generative benchmarking is the reliance on "LLM-as-a-Judge" metrics, which can struggle without clear grounding and often fail on difficult visual problems (Krumdick et al., 2025). To address this, we build metrics that are concrete, deterministic, and directly quantifiable. These metrics measure whiteboard state to determine if spatial relationships were updated according to the task. As such, each metric is task-specific. Each of our metrics extends from the following categories:

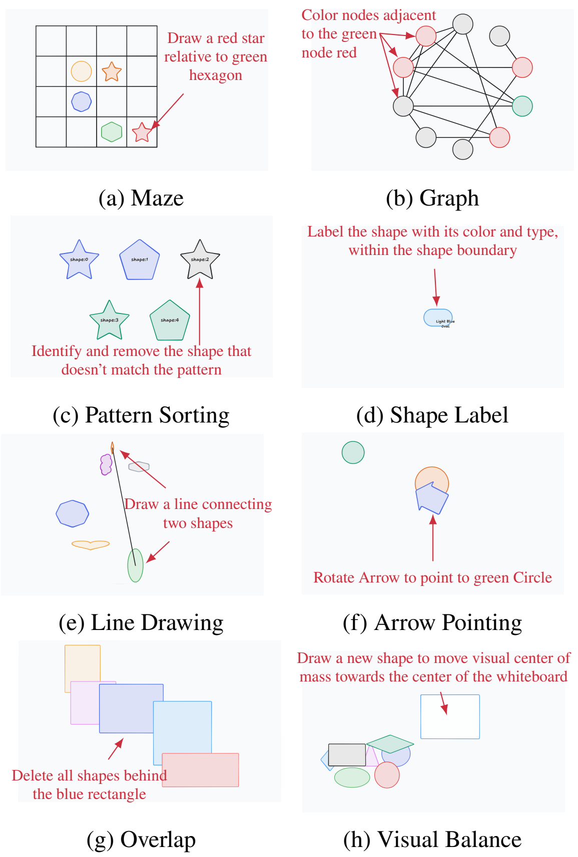
- Distance Metrics – evaluating object placement using linear distance or angular measurements.
- Grouping Scores – evaluating model performance by evaluating set modification using classification

metrics.
- Visual analysis – evaluating performance through image analysis.

These metrics are specialized to each test, and aim to provide a concrete baseline for evaluation. Where possible, distance metrics are normalized to look at *percent improvement* within the context of the metric. As each test is introduced in this section, we will outline the exact metrics used.

3.2 Interactive Whiteboard Implementation

To evaluate agentic whiteboard interaction, we leverage tldraw's digital whiteboard. Building an interface for applying agent actions to the tldraw editor allows an agent to control whiteboard state through create, update, and delete operations. This system also allows us to capture images of the whiteboard, allowing dynamic prompt construction with controls for input modalities and inference types.



(a) Maze (b) Graph (c) Pattern Sorting (d) Shape Label (e) Line Drawing (f) Arrow Pointing (g) Overlap (h) Visual Balance

Figure 2: The 8 tests comprising ASPIRE. Final states shown where possible. (c) and (g) indicate intended target shape on initial state.

3.3 The ASPIRE Test Suite

To evaluate the full spectrum of spatial agency, we developed a suite of 8 distinct tests. These tests are visualized in **Figure 2**. Each test focuses on a specific "spatial primitive," ranging from discrete, rule-based logic to continuous, intuitive visual reasoning. Each test is designed dynamically, allowing us to generate 25 unique, randomized scenarios per test type to ensure statistical robustness. Detailed implementation notes can be found in Appendix A.

3.3.1 Structured & Discrete Tasks

These tests evaluate the agent's ability to follow explicit logical rules in a structured environment. Success here depends on mapping visual elements to discrete symbolic representations.

Maze Navigation (Fig 2a): Evaluating how an agent understands spatial direction in a concrete environment, the maze test creates a 4x4 grid populated with 4 randomly selected shapes. The agent is asked to draw a new shape relative to an existing anchor (e.g., "East of the green hexagon"). We score the agent based on the Euclidean distance of the new shape from the center of the target grid square.

Graph Connectivity (Fig 2b): For a more in-depth test of connectivity, we utilize a test that queries the agent to perform spatial tasks relative to graph topology. Highlighting a single node of a random graph green, we ask the agent to color all neighboring nodes red. This is scored with an F1 score between the true neighbor set and the agent's selection, testing the ability to trace visual edges.

Pattern Sorting (Fig 2c): Similar to the graph test, the Pattern Sorting test asks the agent to analyze the scene and alter it based on visual properties. We generate 5 shapes where 4 follow a specific pattern (color or geometry) and one is an outlier. The agent must identify and delete the "odd-one-out". We score this with a binary value indicating if the correct shape was removed.

3.3.2 Semantic & Relational Tasks

These tests evaluate how agents associate disparate objects to create meaning, a key requirement for diagramming and educational tasks.

Shape Labeling (Fig 2d): Spatial proximity of labels is vital for visual clarity. To test this, we draw a random shape on the whiteboard and prompt the agent to label the shape with its color and type. We score the percentage of the text's pixels that are placed *inside* the shape boundary, penalizing labels that drift into empty space.

Line Drawing (Fig 2e): Many whiteboard diagrams represent information flow through connecting lines. Our line drawing test evaluates how well the agent can indicate connectivity. We ask the agent to connect two randomly selected shapes from a set of six. We score precision based on how far the line endpoints deviate from the visual centers of the target shapes.

3.3.3 Unstructured & Intuitive Tasks

These tests represent the “frontier” of spatial agency, requiring intuitive estimation in continuous space without clear references.

Arrow Pointing (Fig 2f): Building on directional understanding from the maze test, this test removes the implicit constraints of a grid. We place a green target circle, an orange distractor, and a blue arrow with a random initial rotation. The agent must rotate the arrow to point directly at the green circle. We score the percentage of the initial angular error that is corrected.

Overlap & Depth (Fig 2g): While the whiteboard is 2D, it encodes 3D relationships through layering (Z-axis). Given a sequence of overlapping rectangles, the agent is asked to delete all shapes “behind” or “in front of” a target. This tests the agent’s ability to infer depth from occlusion cues. We score performance using an F1 score on the deleted set.

Visual Balance (Fig 2h): Creating compelling visualizations requires an understanding of visual weight. We skew existing shapes into one quadrant and ask the agent to add a single new shape to restore the visual center-of-mass to the center of the canvas. This is scored based on the percent improvement of the center-of-mass toward the geometric center, requiring a holistic “feeling” of the scene rather than simple calculation.

3.4 Model Selection

For our evaluation, we focus on three closed-weight state-of-the-art models: `Gemini-3.0-Pro-Preview` (Google), `Claude-4.5-Opus` (Anthropic), and `GPT-5.1` (OpenAI). We also evaluate one open-weight model, `Qwen-3-VL-235B`. All models are multimodal and offer control over reasoning windows. Each model is selected to be highly performant within its model family.

For all models, the *default* inference method is a zero-shot approach with reasoning capabilities disabled or minimized. To test the impact of inference methods, we also evaluate:

- **Chain-of-Thought (CoT):** Appending “Think step-by-step” to the prompt.

Model	Settings	Score	Tokens
Gemini-3-Pro-Preview	Default	0.83	1940.38
	CoT	0.82	2071.98
	Reasoning	0.85	4808.56
Claude-Opus-4-5	Default	0.81	300.59
	CoT	0.82	471.20
	Reasoning	0.85	760.00
GPT-5.1	Default	0.42	112.59
	CoT	0.44	112.02
	Reasoning	0.64	421.87
Qwen-3-VL-235b	Default	0.36	75.35
	CoT	0.29	93.26
	Reasoning	0.64	2660.19

Table 1: Per-model average scores and token usage across ASPIRE tests

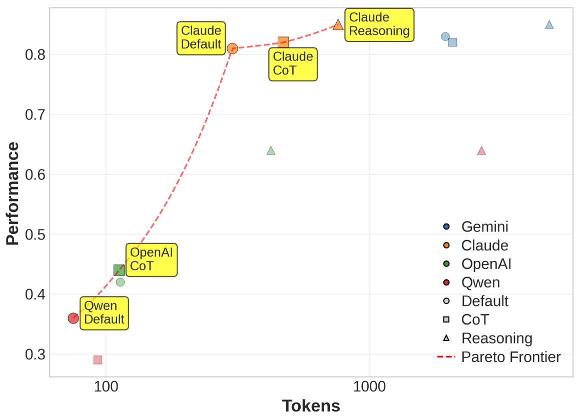


Figure 3: Average Performance across ASPIRE

- **Reasoning Mode:** Enabling model-specific reasoning features, such as setting reasoning context, or enabling levels of “thinking”.

4 Results

The evaluations of our 8 tests are run with 25 static scenarios per test, so that each variation of model and inference mode uses the same tests. This reduces problem variability between models and modes. Model inference is run through each respective API, with `Qwen-3-VL` inference through OpenRouter. All results are the mean of all runs for each test configuration, unless stated otherwise. Per-model average performance averages the normalized test results into a single score. This means that overall average performance should not be compared to individual test scores, but can serve to compare model performance against each other.

Testing over a variety of spatial tasks and environments outlines the complex nature of how agents evaluate and operate on a 2D whiteboard. Overall, `Gemini-3.0-Pro` and `Claude-4.5-Opus` demonstrate the best performance across all categories, reflecting their state-of-the-art capabilities. `GPT-5.1` and `Qwen-3-VL` provide moderate results when reasoning is enabled, though ASPIRE reveals that both architectures share similar limitations in understanding complex spatial scenes.

Performance varies dramatically across task types, revealing distinct strengths and weaknesses in spatial understanding. Models consistently excel at highly structured tasks—achieving near-perfect performance on maze navigation, pattern identification, and graph connectivity problems where clear patterns can be extracted and leveraged. Shape labeling and line drawing tasks also yield strong results, with top models achieving or approaching perfect performance. However, all models, struggle significantly with tasks requiring nuanced spatial reasoning without clear structure. The arrow pointing test proves particularly challenging, as it demands extraction of angular relationships from positional or visual data without explicit patterns to follow. Visual balance presents another difficulty, though `Gemini-3.0-Pro` shows notable advantages here, suggesting architecture-specific strengths in understanding visual weight and center-of-mass. Model reference frame also significantly impacts performance. We observed in the overlap test that prompts using "in front/behind" outperform "over/under" for identical layering tasks. Similarly, in the overlap test agents can focus on immediate relationships well, but fail to identify transitive relationships required to complete the task completely.

With regards to token usage and inference styles, reasoning consistently provides the best accuracy on spatial problems at the cost of increased token usage. CoT prompting often provides modest improvements for comparably small token increases, though `Qwen-3-VL` notably fails to benefit from CoT despite increased token usage.

4.1 Ablation

To build better understanding of how agent actions are taken, we perform an ablation study to control model input directly. For our ablation, we focus on the Arrow, Balance, Graph, and Overlap tests. For the ablation, we run these tests while providing minimal JSON state (update and delete tests require model ids to be known by the agent) – ensuring that the model must primarily work from the image input. We then extend the ablation to leverage basic visual scaffolds for the Arrow, Balance, and Graph tests to help with our understanding of how spatial information is inferred and extracted from image input modalities. We omit a scaffold test for the overlap test as it does not have a clear scaffold to leverage. For the Arrow test, the scaffold provided is a polar plot placed behind the arrow. For balance we provide a Cartesian grid. The graph test adds labels to each node. Each scaffold is selected to provide visual data to supplement what is lost in removing the JSON state. As such, scaffold test are also run with minimal JSON state. Details of the ablation implementation and scaffolds can be found in appendix B. Full results are provided in table 2.

4.1.1 Arrow

Through the ablation study, the arrow test clearly shows limitations in visual understanding. When provided with the full input, agents are capable of extracting and inferring some angle information through shape coordinates. However, in the absence of such information models loose the ability to reliably extract *any* useful information. When inspecting model output, we see that when provided only image input and shape ids, the models choose to estimate shape positioning followed by angle calculations using those positions, or to estimate angles. Unexpectedly, when provided with detailed angle information in the input image through the polar plot scaffold, the models become more confused and exhibit less understanding. Adding this scaffold encourages the models to rely more heavily on visual angle estimations, even when incorrect, rather than more reliable coordinate based calculations.

4.1.2 Balance

Unlike the arrow test, ablating the balance test shows a case where visual reasoning proves to be stronger than text-based reasoning. This reflects the complex encoding of visual weight, and its separation from data like coordinates and shape sizes. By forcing the model to only leverage the image input for evaluating where visual weight is placed in an image, the agent can more readily adjust visual balance. However, when the Cartesian grid scaffold is added, overall output quality goes down. While model output is minimal and cannot be analyzed, we theorize this is due to the model being encouraged to leverage coordinates, rather than direct visual evaluation.

4.1.3 Graph

In analyzing the graph test ablation, we can see the importance of properly embedding information visually, and the inability for current models to correctly leverage that information. When provided

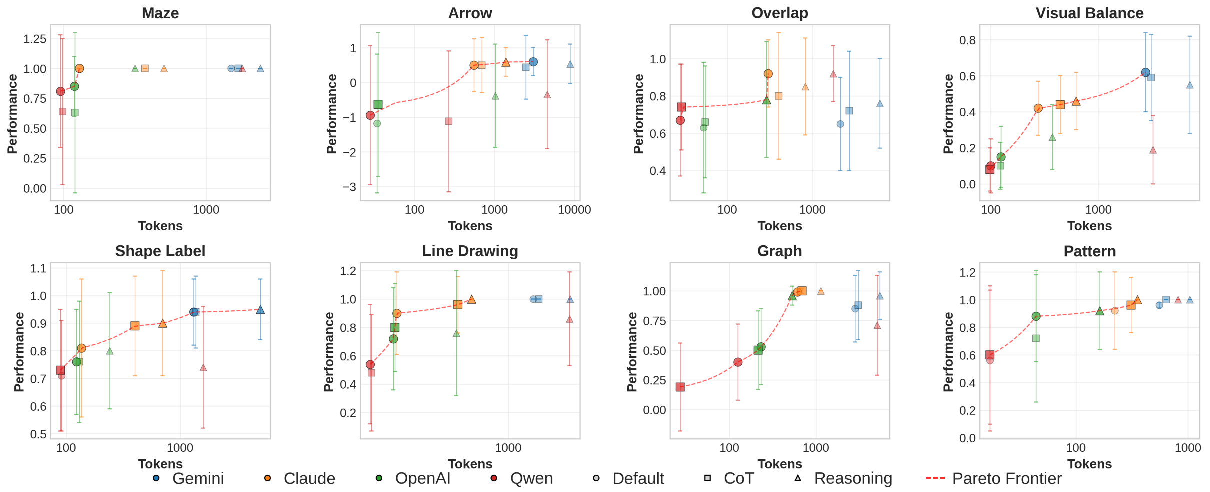


Figure 4: Individual Test Results Including Standard Error

minimal JSON input, the model has no reference for node ids, and cannot reliably identify nodes, or follow edges well. Meanwhile, when we provide the lost data through a labeling scaffold, performance does not return to the high level present with full input. In the limited model output, we see that the model can correctly identify the initial highlighted node, but fails to trace the edges effectively, hallucinating edges and proving ineffective for the task.

4.1.4 Overlap

While unable to provide scaffold analysis, we can see that reducing overlap input to image only has a minimal impact on agent performance, except in the case of `Claude-4.5-Opus`. While the JSON state does encode layered z-axis data, modern models are capable of of extracting the color and shape information required to confidently identify how the shapes are layered. Due to the minimal change in performance, we can infer that the agents still struggle with transitive relationships in this environment.

5 Analysis

ASPIRE provides an insightful window into how MLLM agents see, analyze, and act in spatial environments. Modern MLLM architectures are capable of correctly solving and evaluating spatial problems, when the problem is defined with a clear structure. We can see this with the high performance in the Maze, Graph, Line Drawing, and Shape Labeling tests, which all have clear, well-defined structures. Meanwhile, in the Arrow, Overlap, and Visual Balance tests, with loose or undefined structures, performance drops off consider-

Test	Model	Inference	Full Input	Minimal Input	Scaffold
Arrow	Gemini	Default	0.60	0.14	0.00
		CoT	0.44	0.35	0.23
		Reasoning	0.54	0.46	0.21
	Claude	Default	0.50	-0.37	-0.77
		CoT	0.50	-0.38	-0.75
		Reasoning	0.59	-0.49	-0.80
	OpenAI	Default	-1.18	-1.06	-0.61
		CoT	-0.63	-1.17	-0.97
		Reasoning	-0.38	-0.53	-0.57
	Qwen	Default	-0.94	-0.68	-0.90
		CoT	-1.12	-0.69	-0.83
		Reasoning	-0.34	-0.43	-0.85
Balance	Gemini	Default	0.62	0.68	0.54
		CoT	0.59	0.76	0.58
		Reasoning	0.55	0.70	0.59
	Claude	Default	0.42	0.46	0.45
		CoT	0.44	0.47	0.46
		Reasoning	0.46	0.52	0.48
	OpenAI	Default	0.15	0.36	0.27
		CoT	0.10	0.31	0.27
		Reasoning	0.26	0.39	0.37
	Qwen	Default	0.10	0.16	0.16
		CoT	0.08	0.16	0.14
		Reasoning	0.19	0.21	0.23
Graph	Gemini	Default	0.85	0.44	0.66
		CoT	0.88	0.50	0.70
		Reasoning	0.96	0.53	0.66
	Claude	Default	0.99	0.31	0.60
		CoT	1.00	0.40	0.53
		Reasoning	1.00	0.30	0.57
	OpenAI	Default	0.53	0.34	0.53
		CoT	0.50	0.33	0.50
		Reasoning	0.96	0.36	0.72
	Qwen	Default	0.40	0.12	0.17
		CoT	0.19	0.12	0.17
		Reasoning	0.71	0.31	0.52
Overlap	Gemini	Default	0.65	0.69	N/A
		CoT	0.72	0.70	
		Reasoning	0.76	0.56	
	Claude	Default	0.92	0.62	
		CoT	0.80	0.54	
		Reasoning	0.85	0.77	
	OpenAI	Default	0.63	0.75	
		CoT	0.66	0.76	
		Reasoning	0.78	0.74	
	Qwen	Default	0.67	0.77	
		CoT	0.74	0.77	
		Reasoning	0.92	0.71	

Table 2: Ablation Results for Arrow, Balance, Graph, and Overlap tests

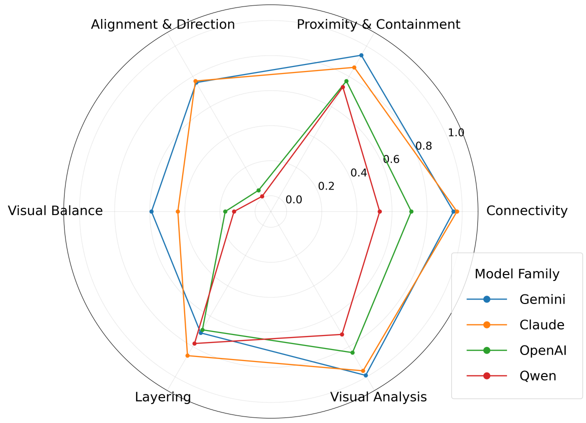


Figure 5: Average model family performance across ASPIRE spatial task categories

ably. This may be caused by breakdowns in spatial understanding, or deficiencies in propagating transitive relationships, preventing spatial understanding on a broad level.

Some of an agents ability to leverage highly structured environments may be tied to its ability to extract and leverage patterns. The pattern test shows agents can identify visual patterns by selecting the "odd one out". Performance in this test is mirrored closely in other tests that have well defined structures. This indicates that for structured tasks agents may be following implicit patterns defined in the spatial structures.

In well structured tests without clear patterns, such as the graph test, agents may be leveraging existing understanding to extract and comprehend the spatial concepts required. LLMs have show to have limited understanding of graph data (Zhang et al., 2024b), and simple problems such as finding neighboring nodes should be within this understanding. We can see how the models leverage this existing knowledge when JSON state data is provided and graph structure can be extracted. Once structures that encode graph data are removed, the models struggle to visually understand and follow edge relationships.

Agent efficiency is an important consideration in selecting models for spatial tasks. We can see that for models such as `Claude-4.5-Opus` and `Gemini-3-pro`, there is relatively little performance difference between default, CoT, and reasoning inference modes. For these models, default inference may be satisfactory. Lower performance models such as `GPT-5.1` and `Qwen-3-VL` have a larger performance delta between default and reasoning modes, meaning that selecting and enabling higher token use reasoning modes may be important to achieve desired performance. In terms of per-token performance efficiency, `Claude-4.5-Opus` is the highest performing model to achieve Pareto efficiency, making it a compelling choice for approaching spatial problems.

While multimodal input has become a common trait for high-end LLM models, performance impacts of image input may be overstated. In some cases, such as the visual balance test, image input can help encode abstract concepts such as visual weight better than text-based JSON encodings. However, for complex structures that may be represented with mathematical representations, JSON state provides a better, more interpretable representation. Not providing JSON state can exaggerate discrepancies in whiteboard, image, and token space for these tasks (Fei et al., 2024).

This behavior is counter intuitive, creating a *scaffolding paradox*, where important environmental data cannot be recovered from visual data. Visual aids, while intuitive to humans, may not provide adequate information to agents, or may encourage an agent to draw conclusions that it may not have considered without these added elements. Scaffolds should be selected with careful consideration as to complement methods that the agent naturally selects for spatial problem solving. Encouraging alternative approaches does not always produce a strong benefit.

6 Conclusion

ASPIRE provides a novel approach to benchmarking agentic spatial understanding. By focusing extending benchmarking beyond passive perception, we open the door for nuanced evaluation of agentic tools. Through this direct evaluation we can better understand how an agent approaches open-ended spatial problems. Our results show that current models provide a strong backbone for agents that are operating within well defined, patterned structures, or that extend clearly-defined concepts. However, agents struggle where they must evaluate and operate within looser structures that rely more on visual data. If desired, visual scaffolds should be carefully selected to complement architectural strengths. To leverage the full potential of multimodal input, agentic systems and foundation models need to work together to improve visual extraction, encoding of abstract concepts, and balancing mathematical strength with visual weaknesses.

7 Limitations

Creating a comprehensive, agentic, spatial benchmarking suite poses many challenges, and we inherently had to make decisions to limit the scope of our benchmark design and evaluation. While our tests aim to encompass a wide variety of spatial concepts, some restrictions arise from our choice to leverage a 2D whiteboard environment. As a result, our tests are more specific to 2D spaces, and to visualization techniques.

Prompting and presenting data to a MLLM is a complex field. There is no absolute truth in how to present this data, nor how to determine the best prompt. As a result, we acknowledge that our choice to leverage simple prompts may underperform highly tuned input for the same problems. However, we feel that in leveraging our benchmark as a standalone, end-to-end comparison between models and inference techniques, our data is still the basis for reliable comparison.

We also recognize the abstract nature of analyzing and drawing conclusions from black-box systems like commercial, closed-weight MLLMs. Our benchmark helps highlight correlations in how MLLMs process and handle spatial data, and aims to encourage further discourse in MLLM agentic benchmarking, MLLM model architecture, and agentic control system design.

8 Acknowledgements

Portions of this paper were revised and rewritten using LLM AI tools. This use was limited to summarizing and condensing low-novelty text, and assistance with plotting tools. LLM AI tools were also used in construction of the benchmarking tools used in this work.

References

Anthropic. 2025. System card: Claude opus 4.5. Technical report, Anthropic PBC.

Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jinheng Ye, Aoxiao Zhong, Xuming Hu, Jing Liang, Philip S Yu, and 1 others. 2025. Llm agents for education: Advances and applications. *arXiv preprint arXiv:2503.11733*.

Google Deepmind. 2025. Gemini 3 pro model card. Technical report, Google.

excalidraw. 2025. Excalidraw.

Hao Fei, Shengqiong Wu, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. 2024. Vitron: A unified pixel-level vision llm for understanding, generating, segmenting, editing. *Advances in neural information processing systems*, 37:57207–57239.

Jie Feng, Tianhui Liu, Yuwei Du, Siqi Guo, Yuming Lin, and Yong Li. 2025. Citygpt: Empowering urban spatial cognition of large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 591–602.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer.

Luca Gioacchini, Giuseppe Siracusano, Davide Sanvito, Kiril Gashteovski, David Friede, Roberto Bifulco, and Carolin Lawrence. 2024. Agentquest: A modular benchmark framework to measure progress and improve llm agents. *arXiv preprint arXiv:2404.06411*.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. 2024. Navigating the digital world as humans do: Universal visual grounding for gui agents. *arXiv preprint arXiv:2410.05243*.

Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. 2025. Can mllms reason in multimodality? emma: An enhanced multimodal reasoning benchmark. *Preprint*, arXiv:2501.05444.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379.

Jihyung Kil, Zheda Mai, Justin Lee, Arpita Chowdhury, Zihe Wang, Kerrie Cheng, Lemeng Wang, Ye Liu, and Wei-Lun Harry Chao. 2024. Mllm-compbench: A comparative reasoning benchmark for multimodal llms. *Advances in Neural Information Processing Systems*, 37:28798–28827.

Michael Krumdick, Charles Lovering, Varshini Reddy, Seth Ebner, and Chris Tanner. 2025. No free labels: Limitations of llm-as-a-judge without human grounding. *arXiv preprint arXiv:2503.05061*.

Xuanyu Lei, Zonghan Yang, Xinrui Chen, Peng Li, and Yang Liu. 2025. Scaffolding coordinates to promote vision-language coordination in large multimodal models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2886–2903.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024a. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13299–13308.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*.

Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2024b. Topviewrs: Vision-language models as top-view spatial reasoners. *arXiv preprint arXiv:2406.02537*.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, and 1 others. 2024c. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*.

Ruotong Liao, Max Erler, Huiyu Wang, Guangyao Zhai, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2024. Videoinsta: Zero-shot long video understanding via informative spatial-temporal reasoning with llms. *arXiv preprint arXiv:2409.20365*.

Chang Liu, Loc Hoang, Andrew Stolman, Rene F. Kizilcec, and Bo Wu. 2025. Understanding student engagement with large language model-powered course assistants. In *Artificial Intelligence in Education*, pages 3–10, Cham. Springer Nature Switzerland.

Fangyu Liu, Guy Emerson, and Nigel Collier. 2023. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651.

Microsoft. 2025. Microsoft whiteboard.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. 2021. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598, Online. Association for Computational Linguistics.

OpenAI. 2025. Gpt-5.1 instant and gpt-5.1 thinking system card addendum. Technical report, OpenAI.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Benjamin Pitt and Daniel Casasanto. 2022. Spatial metaphors and the design of everyday things. *Frontiers in Psychology*, 13:1019957.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, and 1 others. 2024. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642.

Zhengxiang Shi, Qiang Zhang, and Aldo Lipani. 2022. Stepgame: A new benchmark for robust multi-hop spatial reasoning in texts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11321–11329.

Fatemeh Shiri, Xiao-Yu Guo, Mona Far, Xin Yu, Reza Haf, and Yuan-Fang Li. 2024. An empirical analysis on spatial reasoning capabilities of large multimodal models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21440–21455.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. 2025. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15768–15780.

tldraw. 2025. tldraw, the infinite canvas sdk.

Hanchen Wang, Yichun He, Paula P Coelho, Matthew Bucci, Abbas Nazir, Bob Chen, Linh Trinh, Serena Zhang, Kexin Huang, Vineethkrishna Chandrasekar, and 1 others. 2025. Spatialagent: An autonomous ai agent for spatial biology. *bioRxiv*, pages 2025–04.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Sharon Li, and Neel Joshi. 2024a. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *Advances in Neural Information Processing Systems*, 37:75392–75421.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024b. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. 2023a. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE.

Yixuan Wu, Yizhou Wang, Shixiang Tang, Wenhao Wu, Tong He, Wanli Ouyang, Philip Torr, and Jian Wu. 2024. Dettoolchain: A new prompting paradigm to unleash detection ability of mllm. In *European Conference on Computer Vision*, pages 164–182. Springer.

Yue Wu, Xuan Tang, Tom M Mitchell, and Yuanzhi Li. 2023b. Smartplay: A benchmark for llms as intelligent agents. *arXiv preprint arXiv:2310.01557*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, pages 52040–52094. Curran Associates, Inc.

Mingzhe Xing, Rongkai Zhang, Hui Xue, Qi Chen, Fan Yang, and Zhen Xiao. 2024. Understanding the weakness of large language model agents within a complex android environment. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6061–6072.

Yutaro Yamada, Yihan Bao, Andrew K Lampinen, Jungo Kasai, and Ilker Yildirim. 2023. Evaluating spatial understanding of large language models. *arXiv preprint arXiv:2310.14540*.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, and 1 others. 2025a. Magma: A foundation model for multimodal ai agents. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14203–14214.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2025b. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2025. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. 2024a. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*.

Yizhuo Zhang, Heng Wang, Shangbin Feng, Zhaoxuan Tan, Xiaochuang Han, Tianxing He, and Yulia Tsvetkov. 2024b. Can llm graph reasoning generalize beyond pattern memorization? *arXiv preprint arXiv:2406.15992*.

A Implementation Architecture

We leverage tldraw for the digital whiteboard environment that agents can work in. To interface with tldraw, we wrap a small tldraw instance in the following control architecture.

A.1 ASPIRE Execution Pipeline

Each ASPIRE test is a subclass of an overarching ASPIRE test class. This guarantees that each test follows the same initialization, update, and scoring pattern. As such, each test can be independently, agonistically executed by our runtime environment. While each test may have specific settings that are applied for image gathering and prompt construction, in general execution is as follows:

1. Generate initial shapes and evaluation criteria
2. Update tldraw editor with initial shapes
3. Export tldraw editor state and whiteboard image
4. Build prompt with state, image, and task
5. Query MLLM
6. Extract and apply changes from MLLM response
7. Use updated tldraw state and model response for scoring

Scoring is task dependent, and may use model response, tldraw state, and/or images of the whiteboard to determine model performance. For separation and stability, tldraw is run in a standalone puppeteer container. Editor updates, state information, and images are to and from the container, which is reset between tests.

A.2 Model Output

To achieve accurate output, models are provided with a JSON schema requesting a response with `createShapes : []`, `deleteShapes : []`, and `updateShapes : []` objects containing objects compatible with the corresponding tldraw functions. To ensure stable batched execution, the model response is processed minimally until its JSON is compatible. This includes rich text handling, as well as arithmetic solving.

A.3 ASPIRE Tests

ASPIRE is built as a flexible architecture ready for extension. For this work, we have implemented 8 tests over a variety of tasks.

A.3.1 Maze

The maze test draws a 4x4 grid on the whiteboard. 4 shapes are randomly generated and placed in unique squares on this grid. We select one of the shapes X, and then select an adjacent, empty grid square T. Therefore, T has a cardinal or ordinal descriptor D From this, we can build a prompt:

```
Draw a red star to the D of the
T.color T.shape
```

For the example given in Figure 2a, the prompt would be:

```
Draw a red star to the east of the
green hexagon.
```

Evaluation of the maze test utilizes a distance metric. Let N be the center of the new target shape, T be the center of the target square, and W be the grid width. Using euclidean distance function d, then the score $S = 1 - \frac{d(N,T)}{W/2}$. This gives a score of 100% when the target shape is placed directly in the center of the target square, decreasing as the shape is placed off-center.

A.3.2 Graph

For the graph test, we generate a random Erdös-Rènyi Graph G with 10 nodes and an edge probability of 0.3. Selecting a node $n \in G$ we draw the graph such that n is green, and all other nodes are black. We then prompt:

```
Color all neighboring nodes to the
green node red. Remember, neighboring
nodes are connected with an edge in
the graph.
```

Given the set T is the set of neighbors of n, and given that the model gave the set D nodes to be colored red, we can compute the F1 score between these sets to score the graph test.

A.3.3 Pattern

For the pattern test, 5 shapes S are randomly generated such that 4 of them match a pattern based on their colors and shapes. That is, given S, $\exists_{P \subset S}$ such that P fits a prescribed pattern and $|P| = 4$. The pattern is one of two options:

- Color: For the color problem, We choose two shapes, and three colors. All of the five generated shapes are of the two selected shapes, and in two of the selected colors. This leaves a pattern-breaking shape on the whiteboard that matches the shape pattern, but is a unique color.
- Shape: This pattern is an inversion of the color pattern, with 3 shapes and 2 colors.

We prompt the agent with:

```
On the whiteboard there are 5
shapes. Remove the shape that does
not belong.
```

Evaluation for this test is a simple binary value. The agent is scored 0 if it deleted the wrong shape, and 1 if it deletes the correct shape.

A.3.4 Shape Label

The shape label test initializes a single shape S on the whiteboard. We then prompt:

```
Label the shape on the canvas with
its color and type. To label, place
a text box entirely within the shape.
Do not let the text extend outside of
the shape. Adjust the text size and
add newlines as needed. Remember that
the textAlign property only accepts
'start', 'middle', and 'end'. Do not
use 'left', 'center', or 'right'.
```

We use image analysis to evaluate the placement of the label. When capturing images for evaluation, we capture a PNG of just the label P_L, and a PNG of just the shape P_S. For $p_l \in P_L$, if p_l is part of the label (if it is non alpha), and if $p_s \in P_S$ such that p_s and p_l have the same location and p_s is part of the shape, we increment a count of correctly placed label pixels. We track incorrectly placed (not over the shape) pixels in the same manner. We can then calculate the percentage of pixels that are correctly placed over the initial shape.

A.3.5 Line Drawing

For the Line Drawing test, we initialize the whiteboard with 6 random shapes S of unique color-shape pairings. We then select a random pair (A, B) such that $A \in S \wedge B \in S$. We then prompt:

```
Draw a line from the center of
A.color A.shape to the center of
B.color B.shape.
```

For example, in figure 2e the prompt would be:

```
Draw a line from the center of
orange diamond to the center of green
ellipse
```

To evaluate the agent on this test, we measure how far off the ends of the line are from the center of the target shapes. Let L_1 and L_2 be the endpoints of the line that the agent draws. Let C_A and C_B be the centers of the target shapes. With euclidean distance function d, $D = d(C_A, C_B)$. Then, score $S = 1 - (\frac{d(L_1,C_A)}{D} + \frac{d(L_2,C_B)}{D})/2$. That is, the score is the average of each ends percentage error in terms of the whole line distance.

A.3.6 Arrow Pointing

The arrow test places 3 shapes on the whiteboard. G an green circle, O an orange circle, and B a blue arrow. The green circle is a reference target, the orange circle is a confusion element, and the blue arrow is what the agent is tasked with rotating. We prompt the agent to:

`Rotate the arrow so that it is pointed at the green circle. Give your rotation in radians to rotate in a clockwise direction from the current position. You can use a negative rotation to rotate counterclockwise.`

The response from the LLM is used to rotate the arrow with the `rotateShapesBy()` function for the tldraw editor.

We use an angular distance metric to evaluate this test. Let θ_i be the starting angle between B and the G, and θ_f be the final angle, after the rotation is applied. Then the score $S = \frac{\theta_i - \theta_f}{\theta_i}$. That is, the score is the percentage of the angle that has been corrected.

A.3.7 Overlap

The overlap test creates a sequence of uniquely colored rectangles R of length n such that for $R_i \in R$, R_{i+1} overlaps some part of R_i. This guarantees a sequence of overlapping rectangles. Then, we select some $R_x \in R$ such that $R_x \neq R_0 \wedge R_x \neq R_{n-1}$. Selecting a direction D that is either "in front of" or "behind", we can then build out prompt:

`Delete all shapes` D `the` R_x`.color rectangle. Do not change any of the other shapes.`

For example if figure 2g the prompt would be:

`Delete all shapes behind the blue rectangle`

We score this test with an F1 score. Let R_T be the target set that should be deleted given the selection of D and R_x. Let R_D be the set that was actually deleted by the agent response. We can then compute the F1 score between these sets.

A.3.8 Visual Balance

The visual balance test selects a random quadrant of the whiteboard within a 1400×800 bounding rectangle. In that quadrant we draw 7 random shapes of various colors. Then, we prompt:

`Given the current state of the whiteboard, add 1 large shape to create a more visually balanced scene. After adding the new shape, the visual weight of the whiteboard should be focused at the center of the frame. Adjust the size, type, and rotation of the shape to best reflect where new visual weight should be added to balance the existing shapes. Do not add more than one shape. Do not delete or update any shapes.`

We score the agent based on how far the visual center-of-mass is moved towards the center as a percentage improvement. Let C_i be the initial distance of the visual center of mass from the center, and C_f be the final center of mass distance . Then, the score $S = \frac{C_i - C_f}{C_f}$ evaluate how far the visual center of mass has moved towards the center.

To calculate visual center of mass, we create a simple weightmap from a PNG of the whiteboard. To create this weightmap we use the alpha channel of the PNG. We can then average the weights to find a visual center of mass.

B Ablation

Ablation is important, especially with MLLM models when trying to understand the impact of visual input. Our ablation focuses on 4 tests: Arrow, Balance, Graph, and Overlap. When considering the ablation, we need to limit text input to favor image input. However, not all tests can tolerate complete removal of JSON state. While tests based on the create operation don't require knowledge of existing shapes, in order to update or delete objects on the whiteboard the agent must know the object ids. As such, for balance test, which leverages the create operation, we omit all JSON state. For the other tests, we still provide a minimal list of shape ids.

For example, with the overlap test we go from the prompt:


```
Initial whiteboard state (TLDraw JSON):
```json
  [
   {
    "id": "shape:light-violet",
    "type": "geo",
    "x": 16,
    "y": 201,
    "props": {
        "geo": "rectangle",
        "color": "light-violet",
        "w": 327,
        "h": 219,
        "fill": "solid"
        }
    },
    {
     "id": "shape:yellow",
     "type": "geo",
     "x": 173,
     "y": 410,
```
```

```
        "props": {
            "geo": "rectangle",
            "color": "yellow",
            "w": 454.5,
            "h": 483,
            "fill": "solid"
            }
      },
      {
        "id": "shape:blue",
        "type": "geo",
        "x": 515.5,
        "y": 585,
        "props": {
            "geo": "rectangle",
            "color": "blue",
            "w": 464,
            "h": 382.5,
            "fill": "solid"
        }
      },
      {
        "id": "shape:red",
        "type": "geo",
        "x": 674.5,
        "y": 275,
        "props": {
            "geo": "rectangle",
            "color": "red",
            "w": 586.25,
            "h": 393,
            "fill": "solid"
        }
      },
      {
        "id": "shape:orange",
        "type": "geo",
        "x": 355.5,
        "y": 439,
        "props": {
            "geo": "rectangle",
            "color": "orange",
            "w": 455,
            "h": 344.5,
            "fill": "solid"
        }
      }
    ]
    ```

Task:
Delete all shapes behind the yellow rectangle.
Do not change any of the other shapes.

Output actions to update the whiteboard scene.
Your output should be in a JSON block from
tldraw using the following schema: ...
```

To ablate this prompt, we remove all information save for the shape ids. This leads to the ablated prompt:

```
Initial whiteboard state (TLDraw JSON):
```json
[
  {
   "id": "shape:light-violet"
  },
  {
    "id": "shape:yellow"
  },
  {
    "id": "shape:blue"
  },
  {
    "id": "shape:red"
  },
  {
    "id": "shape:orange"
  }
]
```
Task:
Delete all shapes behind the yellow rectangle.
Do not change any of the other shapes.

Output actions to update the whiteboard scene.
Your output should be in a JSON block from
tldraw using the following schema: ...
```

Both prompts still are sent with the companion image input.

### B.1 Scaffolds

To see if the information that is removed through ablation can be supplemented with visual input, we also test Arrow, Balance, and Graph with visual scaffolds. The goal of testing with scaffolds is to highlight what aspects of visual scene can be digested by modern MLLM models. Examples of each scaffold can be found in figure 6.

For the Arrow test, we add a polar plot behind the blue arrow. This is intended to provide a visual reference of current and relative angles. For the balance test we provide a Cartesian grid that can help quantify relative locations without exact coordinate data. Finally for the graph test we add a basic label system to that shape id can be correlated with visual objects. All of the scaffolds are added in a way to minimize visual impact, while still providing the relevant information. We do not change any aspect of the prompts when providing the scaffolds, allowing us to see how the agents interpret these scaffolds. When providing scaffolds, we use the ablated JSON state, so that they model must rely on the visual input.

#### B.1.1 Arrow Responses

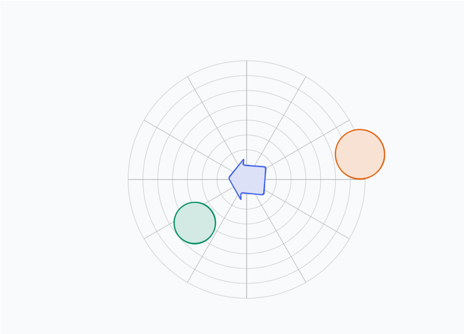
Here we provide an example of how model response changes between full, minimum, and scaffold tests on the arrow test ablation, for `Claude-4.5-Opus` with reasoning enabled.

Full Input:

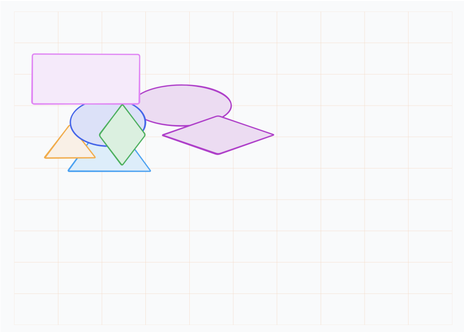
```
Looking at the image, I need to analyze the
    current position and orientation of the
    arrow and determine how to rotate it to
    point at the green circle.

Current state:
- Blue arrow is at position (374, 231) with
    rotation 3.583 radians
- Green circle center is approximately at
    (400 + 183/2, 216 + 183/2) = (491.5,
    307.5)
- Arrow center is approximately at (374 +
    123/2, 231 + 148/2) = (435.5, 305)

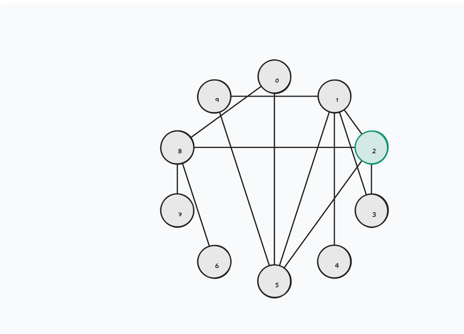
Direction calculation:
```
```



(a) Polar Plot



(b) Cartesian Grid



(c) Labels

Figure 6: Scaffold Structures

```
- Vector from arrow to green circle: (491.5 -
     435.5, 307.5 - 305) = (56, 2.5)
- This angle from positive x-axis: atan2(2.5,
     56) ≈ 0.045 radians

An "arrow-up" shape points upward by default
     (negative y direction). To point at the
     green circle (which is to the right and
     slightly below), the arrow needs an
     absolute rotation of approximately π/2 ≈
     1.571 radians.

**Required rotation change:**
- Target rotation: ~1.571 radians
- Current rotation: 3.583 radians
- Change needed: 1.571 - 3.583 = -2.012
     radians (counterclockwise)

```json
{
  "shape": "shape:blue-arrow",
  "rotation": -2.01
}
```
```

Minimum Input:

```
Looking at the whiteboard, I can see:
- A blue arrow in the upper-left area
     pointing diagonally down and to the right
- A green circle in the center-left of the
     canvas
- An orange circle to the right of the green
     circle

The arrow currently appears to be pointing at
      approximately a 45-degree angle (around
     -0.8 radians from horizontal). The green
 circle is positioned below and slightly to
      the right of the arrow's base.

To point directly at the green circle, I need
      to calculate the angle from the arrow's
     position to the green circle's center. T
he green circle appears to be at a steeper
     downward angle from the arrow than where
     it's currently pointing.

Estimating the positions:
- Arrow is approximately at (370, 180)
- Green circle center is approximately at
     (500, 370)

The required angle to point at the green
     circle would be steeper (more vertical)
     than the current orientation. This means
     rota
ting clockwise slightly to point more
     downward toward the green circle.

The rotation needed is approximately 0.25
     radians clockwise.

```json
{
  "shape": "shape:blue-arrow",
  "rotation": 0.25
}
```
```

Scaffold Input:

```
Looking at the image, I need to analyze the
     current orientation of the blue arrow and
      determine how much to rotate it to point
 at the green circle.

The blue arrow is currently positioned in the
      upper-left area and appears to be
     pointing downward and to the left (
     roughly sou
thwest direction). The green circle is
     located below and to the right of the
     arrow's position.

To make the arrow point at the green circle:
- Current arrow direction: approximately
     pointing down-left (around -45° from
     vertical, or roughly π-/4 from straight
     down)
- Target direction: The green circle is
     roughly down and to the right of the
     arrow

To rotate from pointing down-left to pointing
      down-right toward the green circle, I
     need to rotate clockwise by approximately
     π
/2 radians (90 degrees).

```json
{
  "shape": "shape:blue-arrow",
  "rotation": 1.5708
```
```

~~~
}
```
~~~