Spatial Mental Modeling from Limited Views

Anonymous ICCV submission

Paper ID 2

Abstract

Humans intuitively construct mental models of space beyond what they directly perceive, but can large visual-language models (VLMs) do the same with partial observations like ***limited views****? We identify this significant gap for current VLMs via our new* MINDCUBE *benchmark with* $21,154$ *questions and* $3,268$ *images, evaluating how well VLMs build robust spatial mental models, representing positions (cognitive mapping), orientations (perspective-taking), and dynamics (mental simulation for "what-if" movements), to solve spatial reasoning on* ***unseen*** *space that goes beyond immediate perception.*

We explore three approaches to approximating spatial mental models in VLMs: (1) View interpolation to visualize mental simulation, which surprisingly offers little benefit, highlighting the challenge of reasoning from limited views; (2) Supervising the model on singular abilities (generating cognitive maps or reasoning chains alone) yields only marginal gains; and (3) The key breakthrough is a synergistic approach that involves jointly training the model to first generate a cognitive map and then reason upon it, which results in substantial performance gains. This mapping-then-reasoning paradigm proves highly effective: Training models to reason over these internal maps improves from 37.8% *to* 60.8% *(+23.0%). Adding reinforcement learning further improves performance to* 70.7% *(+32.9%).*

Our key insight is that such scaffolding of spatial mental models, actively utilizing internal structured spatial representations with flexible reasoning processes, significantly improves understanding of unobservable space.

1. Introduction

For Vision-Language Models (VLMs) [1, 3, 8, 34] to advance from passive perception to interacting with partially observable environments [33, 57, 71], they must reason about unseen spatial relationships from limited views. Humans achieve this effortlessly by building a **spatial mental model** [22, 23]: an internal representation of an environment that allows for consistent inference independent of the current viewpoint, as shown in Figure 1.

Despite impressive progress, current VLMs struggle to synthesize spatial information across views, maintain consistency, and reason about occluded objects [32, 51, 66, 73]. This gap demands new evaluations focused on reasoning with partial observations, ensuring cross-view consistency, and performing mental simulations (e.g., "*what if turning left*"). To fill this gap, we introduce MINDCUBE, a benchmark featuring $21,154$ questions over 976 multi-view groups, with questions specifically designed to require reasoning about non-visible objects, perspective taking, and complex spatial relations (Figure 2).

Our evaluation of 17 state-of-the-art VLMs on MINDCUBE reveals that most perform only marginally better than random guessing. This motivates our central question: **How can we help VLMs reason from partial observations**?

Inspired by spatial cognition [25, 38, 63], we investigate three intermediate representations to approximate mental models. **View Interpolation** proved unhelpful. **Free-form Natural Language Reasoning** to verbalize mental simulation yielded performance gains ($+2.7\%$). For **Structured Cognitive Maps**, we find that prompting models to *generate* their own maps is more effective ($+5.2\%$) than providing them with ready-made ones ($+0.9\%$). However, even with this process, VLMs struggle with accuracy, shown by low Isomorphic Rates ($< 10\%$) with ground truth maps.

To address this limitation, we fine-tune VLMs on $10,000$ reasoning chains and $10,000$ ground truth cognitive maps. Supervised Fine-tuning (SFT) on cognitive maps significantly boosts structural similarity (e.g., from 0.1% to 46.0% isomorphism). While SFT on reasoning chains was effective, guiding models to first *build a map and then reason* over it achieved the best performance, yielding a total gain of $+8.5\%$. This proves the efficacy of actively constructing and then using an internal spatial representation.

Finally, we employ Reinforcement Learning (RL) on our SFT model to further enhance this structured thinking process. This approach dramatically improves task accuracy from a baseline of 37.8% to 70.7%. Our findings confirm that VLMs perform best when they autonomously generate

ICCV #2

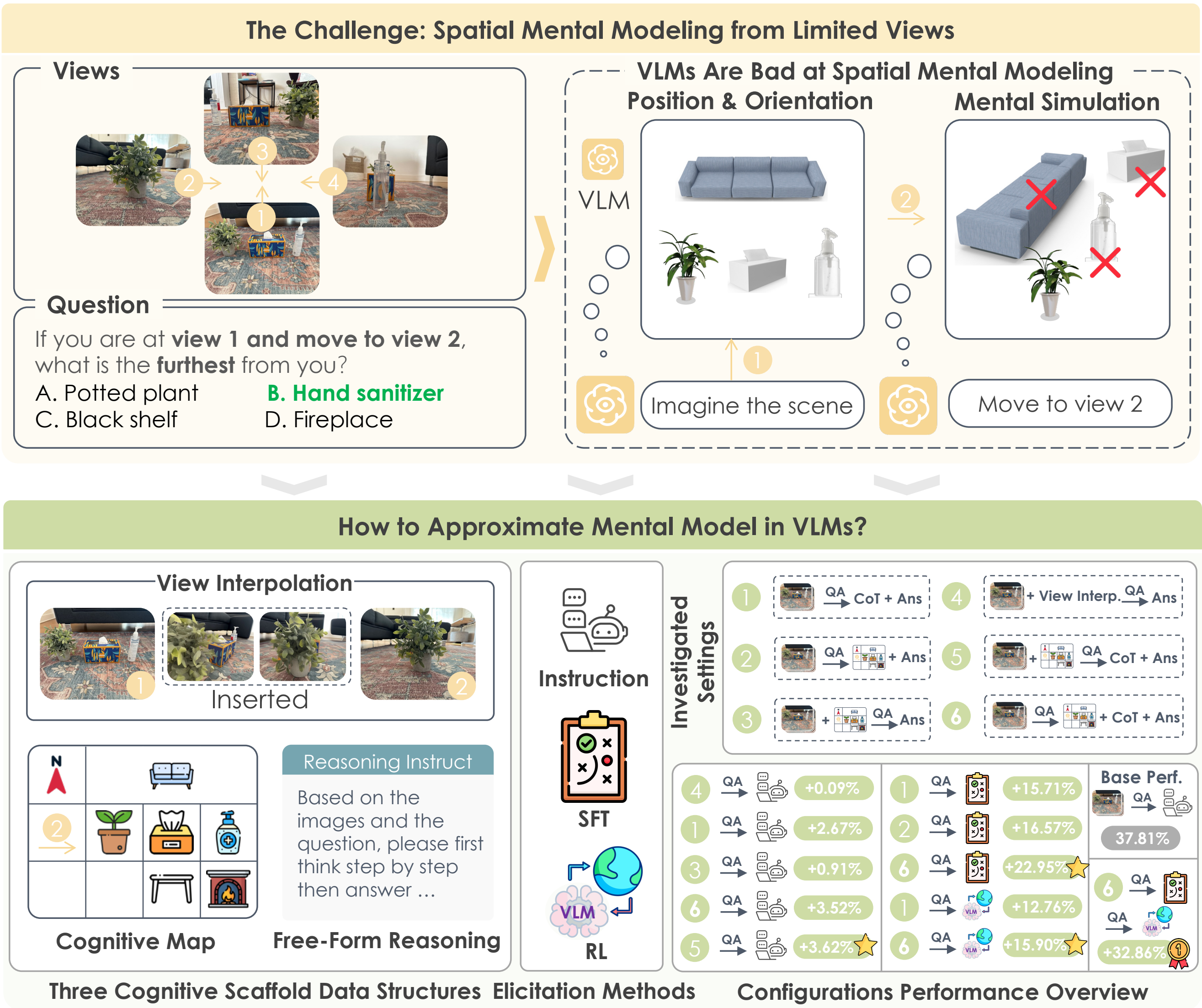


Figure 1. **Top**: VLMs cannot maintain a coherent mental model when evaluating on the MINDCUBE benchmark. **Bottom**: We study how we can help VLMs imagine space through external (scaling of views, cognitive map input) and internal strategies (fine-tuning, cognitive map elicitation). We find joint cognitive map and reasoning setting yields the highest gain (+32.86%). ⭐: Best within the same elicitation method. 🏅: Best perf. combination.

and leverage internal mental representations, a method far superior to passive view interpolation or using externally-supplied maps.

2. MINDCUBE Benchmark and Evaluation

2.1. MINDCUBE Benchmark

Overview.

We introduce MINDCUBE, a benchmark for evaluating VLMs' spatial reasoning under partial observations and dynamic viewpoints.

MINDCUBE features multi-view image groups paired with spatial reasoning questions, enabling fine-grained analysis of spatial modeling performance. It targets key challenges such as maintaining object consistency across views and reasoning about occluded or invisible elements. Table 1 (left) summarizes the benchmark's overall data distribution. Details on benchmark design, taxonomy, and curation are provided in the Appendix.

Taxonomy.

For a fine-grained analysis of VLM spatial reasoning abilities, we introduce a taxonomy that systematically categorizes the challenges in MINDCUBE (visualized in Figure 2). This taxonomy spans five key dimensions: 1) **Camera Movement**: We mainly collect three types of camera movement: ROTATION (Stays in place but rotates to look around), AROUND (Moves around evaluated objects in a circular path), and AMONG (Moves among evaluated objects

ICCV #2

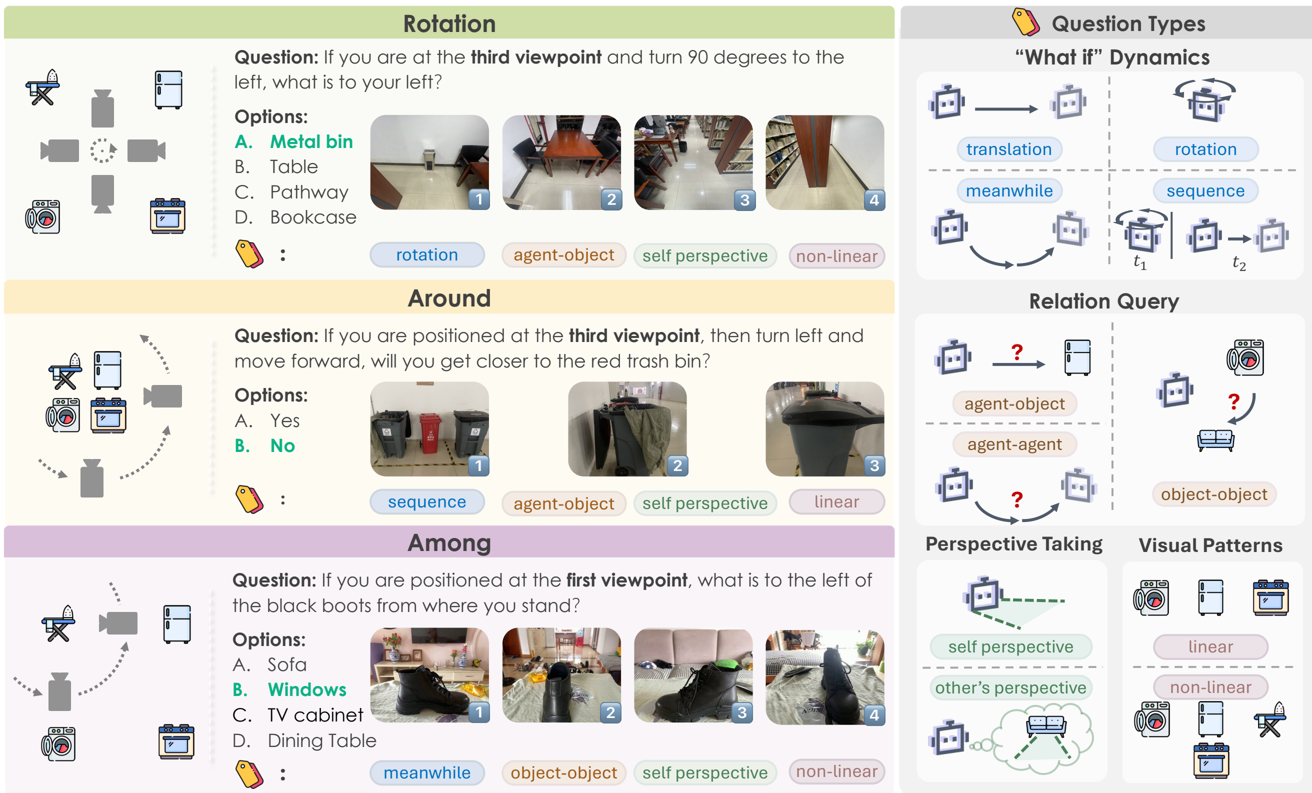


Figure 2. MINDCUBE taxonomy and examples. Left: Three camera movement patterns (ROTATION, AROUND, AMONG) with corresponding spatial QA examples. **Right**: Four-dimensional taxonomy categorizing MINDCUBE questions types.

in a circular path). 2) **Visual Patterns**: This describes the objects' spatial configurations, including spatial *linear* or *non-linear* arrangements. 3) **"What-if" Dynamics**: The hypothetical transformations applied to the agent's viewpoint, such as *translation*, *rotation*, or their combination (*meanwhile* and *sequence*). 4) **Relation Query**: The type of spatial relation being queried, including *agent–object*, *agent–agent*, or *object–object*. 5) **Perspective Taking**: Whether the spatial reasoning is grounded in the perceiver's own viewpoint (*self*) or involves adopting the viewpoint of another entity (*other*).

Dataset Curation.

The MINDCUBE dataset was created through a pipeline: We first selected multi-view image groups matching our taxonomy's movement patterns (Figure 2) and spatial criteria. These were then annotated with key spatial information. Finally, we algorithmically generated taxonomy-aligned questions with targeted distractors. Details are included in the Appendix.

2.2. Evaluation on MINDCUBE

We evaluate VLMs' spatial reasoning on MINDCUBE using a diverse set of models (Table 1, right; setup details in the Appendix). Results reveal a striking performance gap: the best model, DeepSeek-VL2-Small, achieves only 47.62% accuracy, well above chance but far from human-level B.3. ROTATION tasks proved hardest (top score: 38.76%), suggesting limited mental rotation and viewpoint adaptation. Mantis-8B (SigLip) and DeepSeek-VL2-Small perform better AMONG and AROUND, respectively. Large proprietary models often lag behind smaller open-source counterparts. Spatial fine-tuning yielded mixed results. Overall, neither multi-image input nor spatial fine-tuning reliably improves spatial reasoning, raising a key question:

How can we help VLMs develop or approximate these crucial spatial reasoning capabilities?

3. Which Scaffolds Best Guide Spatial Thinking in Unchanged VLMs?

To address the gap, we first evaluate whether structured data forms can scaffold spatial reasoning in frozen VLMs by approximating spatial mental models under limited views.

3.1. Spatial Mental Models Approximation

We investigate data structures that can serve as cognitive scaffolds to help VLMs form spatial mental models from limited views. In cognitive science, spatial mental models are not metric-precise maps but rather schematic, internal constructs that encode the relative configuration of objects and view-

Table 1. Left: MINDCUBE data statistics. The number next to the setting (ROTATION, AMONG, AROUND) means the total QA pairs. Numbers next to each dataset (e.g., Arkitscenes) mean QA pairs/image groups. For example, "865/53" for Arkitscenes in ROTATION means we have 865 QA pairs and 53 image groups from it. Right: Performance of VLMs on MINDCUBE. Dark blue indicates the best result among all models and light blue indicates the best result among open-source models.

Rotation (1081)	
Arkitscenes	865/53
Self collected	216/9
Img groups	62

Among (18204)	
WildRGB-D	17500/710
DL3DV-10K	704/24
Img groups	733

Around (1869)	
DL3DV-10K	789/109
Self collected	1080/71
Img groups	180

Method	Overall	Rotation	Among	Around
Baseline				
Random (chance)	*32.35*	*36.36*	*32.29*	*30.66*
Random (frequency)	*33.02*	*38.30*	*32.66*	*35.79*
Open-Weight Multi Image Models				
LLaVA-Onevision-7B [26]	47.43	36.45	48.42	44.09
LLaVA-Video-Qwen-7B [69]	41.96	35.71	43.55	30.12
LongVA-7B [67]	29.46	35.89	29.55	24.88
mPLUG-Owl3-7B-241101 [60]	44.85	37.84	47.11	26.91
InternVL2.5-8B [8]	18.68	36.45	18.20	13.11
Qwen2.5-VL-7B-Instruct [3]	29.26	38.76	29.50	21.35
Qwen2.5-VL-3B-Instruct [3]	33.21	37.37	33.26	30.34
Idefics-8B-Llama3 [24]	35.86	35.15	35.94	35.49
DeepSeek-VL2-Small [31]	47.62	37.00	50.38	26.91
Gemma-3-12B-it [43]	46.67	38.39	48.38	34.63
Mantis-8B (SigLip) [21]	41.05	37.65	40.23	50.99
Proprietary Models				
GPT-4o [34]	38.81	32.65	40.17	29.16
Claude-3.7-Sonnet-20250219 [2]	34.89	37.65	35.93	23.17
Spatial Models				
RoboBrain [20]	37.38	35.80	38.28	29.53
SpaceMantis [6]	22.81	37.65	21.26	29.32
Spatial-MLLM [54]	32.06	38.39	20.92	32.82
Space-LlaVA [13]	33.28	38.02	33.71	26.32

points [23, 44–46]. These representations are flexible and functionally effective, allowing for reasoning across fragmented observations and unseen perspectives, even if they are incomplete. Drawing from this, we propose three data structures aimed at targeting distinct properties (integration, transformation, inference) of these mental models. Further details are in Appendix C.1, with examples in Figure 3.

1. **View Interpolation**. Interpolating sparse views creates perceptual continuity, mirroring *mental animation* [16], to support internal transformations like imagined rotation. This process scaffolds the dynamic updating of spatial mental models. Figure 3 shows a one-frame insertion example.

2. **Augmented Cognitive Map**. Cognitive maps, akin to Tversky's *cognitive collages* [44], are 2D layouts capturing locally coherent yet fragmented structures. While recent VLM studies [57, 61] use *plain* maps with only top-down object positions, we propose an *augmented* variant. Our version integrates discrete views with annotated positions and orientations for both objects and cameras to better capture the relational consistency of *spatial mental models*.

3. **Free Form Reasoning**. Open-ended, step-by-step language reasoning provides a *procedural approximation* for constructing and querying spatial models. Though less rigid than maps, this reasoning reflects the inferential function of spatial mental models, particularly with ambiguous or incomplete observations [46].

3.2. Experiment Setup

We conduct controlled experiments with fixed input formats to test whether structured scaffolds can help without retraining. Each condition introduces a different structure to support internal modeling under limited views.

Model and Evaluation Data We conduct all experiments using *Qwen2.5-VL-3B-Instruct* [3]. Our evaluation is performed on MINDCUBE-TINY, a diagnostic subset sampled from MINDCUBE, containing 1,050 questions in total. Detailed statistics are: 600 from the AMONG, 250 from AROUND, and 200 from ROTATION.

Configurations Each experiment is defined by two orthogonal axes: *Input Structure* (what spatial evidence VLMs receive) and *Output Format* (the required response type). As the experimental foundation of this paper, we begin with the ten possible configurations listed in Table 2, from which we investigate a representative subset. Specifically, our grounded cognitive maps are generated using the object arrangements annotation described in Section 2.1, and examples for all configurations are provided in the Appendix C.3. In the frozen VLMs evaluation setup, we exclude the `Aug-CGMap-Out` and `Plain-CGMap-Out`

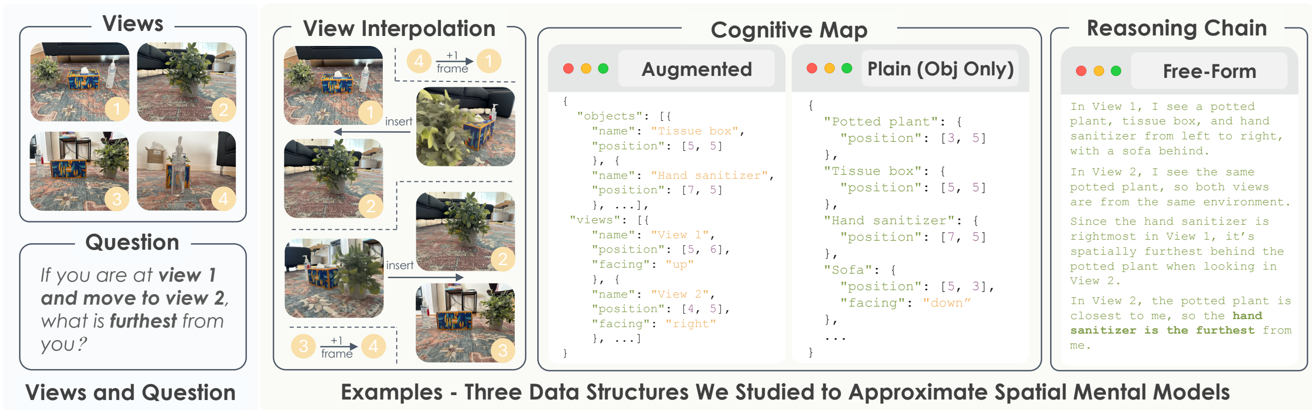


Figure 3. Grounded examples of our three data structures that approximate spatial mental models.

Table 2. Abbreviations for the ten input-output configurations across all experiments in this work. VI = **V**iew **I**nterpolation, CGMap = **C**o**g**nitive **Map**, Aug = Augmented (objects + camera included), FF-Rsn and FFR = Free-Form Reasoning. QA = Raw views + question.

Name	Input Structure	Output Format
`Raw-QA`	Raw views + question	Direct answer
`VI-1`	Raw + 1 interp. view	Direct answer
`VI-2`	Raw + 2 interp. views	Direct answer
`FF-Rsn`	Raw views + question	Reasoning → answer
`Aug-CGMap-In`	Aug. cog. map + QA	Direct answer
`Aug-CGMap-Out`	Raw views + question	Aug. map → ans
`Plain-CGMap-Out`	Raw views + question	Plain map → ans
`Aug-CGMap-FFR-Out`	Raw views + question	Aug. map + rsn → ans
`Plain-CGMap-FFR-Out`	Raw views + question	Plain map + rsn → ans
`CGMap-In-FFR-Out`	Aug. cog. map + QA	Reasoning → answer

settings, as VLMs tend to conflate map generation with reasoning, even when instructed otherwise.

Evaluation Metrics We evaluate task performance using QA accuracy. For generated cognitive maps, we introduce a set of well-defined graph metrics: (1) *Valid Cognitive Map Rate*, indicating whether the output conforms to the expected schema; (2) *Overall Similarity*, a weighted score combining directional and facing consistency; and (3) *Isomorphic Rate*, measuring whether all pairwise object relations match the ground truth under optimal alignment. Full definitions are provided in the Appendix C.2.

3.3. Do Scaffolds Improve Spatial Reasoning Without Training?

We evaluate how well the seven input configurations defined in Table 2 support spatial reasoning in VLMs under limited views, without any model updates. Results are shown in Table 3 (left).

How far can structure alone go? We find that providing structured input without explicit reasoning is ineffective. Our baseline of raw views with direct answering (`Raw-QA`) achieves 37.81% accuracy. Adding interpolated views yields no gain, while inputting a pre-computed augmented cognitive map (`Aug-CGMap-In`) degrades performance to 32.00%. In contrast, enabling free-form reasoning (`FF-Rsn`), alone or with other inputs, provides a substantial boost to 41.33%. These results suggest: *structure alone, whether visual or spatial, is not enough*. Without engaging in reasoning, VLMs cannot effectively leverage spatial cues.

Can we prompt the model to think spatially? Yes; prompting the model to generate a cognitive map before answering (`Aug-CGMap-FFR-Out`, `Plain-CGMap-FFR-Out`) improves performance over free-form reasoning alone (`FF-Rsn`), increasing accuracy from 40.48% to 41.43%. This suggests that the act of generating a map encourages a more structured reasoning process. However, the models struggle to produce accurate maps. As shown in Table 3 (Right), the similarity of generated maps to ground truth is low ($< 50\%$), with isomorphism rates of only 0.10% for augmented maps and 7.43% for plain maps. The near-zero rate for augmented maps is likely due to increased generation errors from the added complexity of view-level details, as exemplified in Appendix D.

> **Scaffolding Spatial Reasoning in *Frozen VLMs***
> - *Explicit reasoning is crucial for improving performance, and cognitive maps can guide this.*
> - *Passive structures (like maps as input) alone and visual continuity offer little benefit.*

4. Can We Teach VLMs to Build and Leverage Spatial Representations?

So far, prompting frozen VLMs with external scaffolds, such as interpolated views or cognitive maps, has yielded limited gains. These techniques fail to tackle the core limitation: VLMs do not form internal spatial representations or reason through space effectively. To go further, we want to know: Can supervised fine-tuning (SFT) teach VLMs to build and

Table 3. Left: QA accuracy (%) of *Qwen2.5-VL-3B-Instruct* on the MindCube-Tiny benchmark under different configs for frozen VLMs. Right: Graph metrics for two cog map output settings.

Config.	**Overall**	**Rotation**	**Among**	**Around**
`Raw-QA`	37.81	34.00	36.00	45.20
`VI-1`	37.90↑	35.50	37.33	41.20
`VI-2`	37.81–	35.50	36.50	42.80
`Aug-CGMap-In`	32.00↓	35.00	30.50	33.20
`FF-Rsn`	40.48↑	32.00	36.00	58.00
`Aug-CGMap-FFR-Out`	40.57↑	21.00	**43.00**	50.40
`Plain-CGMap-FFR-Out`	41.33↑	25.00	39.67	**58.40**
`CGMap-In-FFR-Out`	**41.43**↑	**37.00**	41.67	44.40

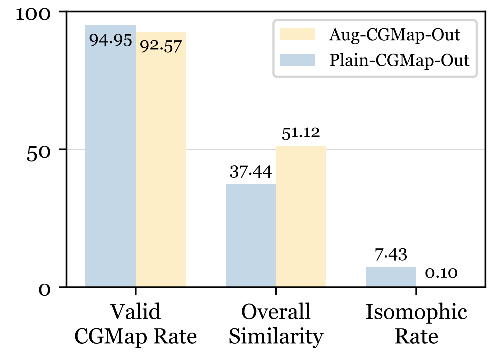


Table 4. QA accuracy (%) and cognitive map generation quality of *Qwen2.5-VL-3B-Instruct* under SFT configurations on MindCube-Tiny. Both `FF-Rsn` and `FFR` refer to free-form reasoning.

SFT Config.	**MindCube-Tiny QA Accuracy (%)**				**Generated Cognitive Map (%)**		
	Overall	**Rotation**	**Among**	**Around**	**Valid Rate**	**Overall Sim.**	**Isom. Rate**
`Raw-QA`	52.28	34.50	52.50	66.00	–	–	–
`FF-Rsn`	53.52↑	36.00	54.67	64.80	–	–	–
`Aug-CGMap-Out`	54.19↑	35.50	53.17	**71.60**	**100.00**	74.30	43.24
`Plain-CGMap-Out`	54.38↑	35.50	53.50	**71.60**	**100.00**	**91.73**	**89.05**
`Aug-CGMap-FFR-Out`	55.24↑	**49.50**	52.50	66.40	**100.00**	75.27	46.00
`Plain-CGMap-FFR-Out`	**60.76**↑	47.50	**62.33**	67.60	**100.00**	88.79	73.81

leverage spatial models from within?

4.1. Designing a Robust Experimental Framework

To ensure consistency and comparability, we inherit experimental configurations detailed in Sections 3.1 and 3.2. Specifically, we retain: (1) the two effective data structures—Cognitive Maps (Object-only / Object + Camera) and Free-Form Reasoning, (2) the base model *Qwen2.5-VL-3B-Instruct*, (3) the evaluation benchmark MindCube-Tiny, and (4) all established evaluation metrics. View interpolation is excluded from our fine-tuning experiments due to its limited performance gains in earlier validations. Primary modifications in this SFT phase include adjusted training hyperparameters (detailed in the Appendix E.2) and the input-output configurations.

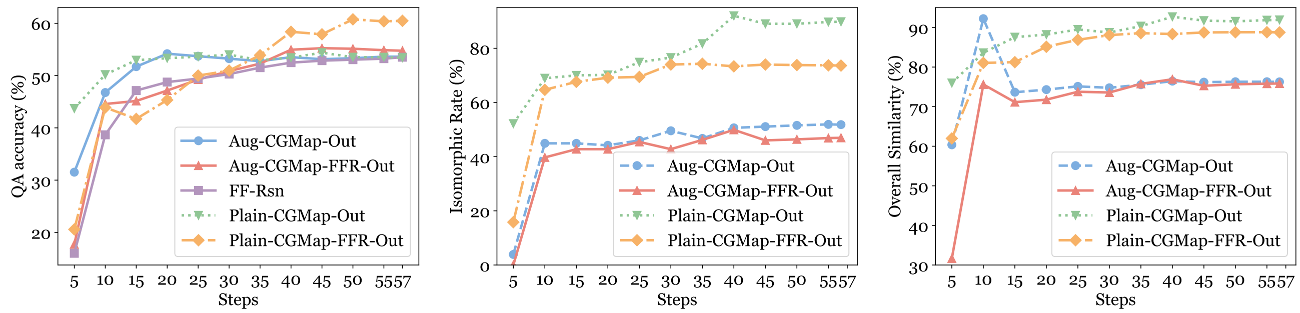
SFT Task Configurations Drawing on insights from Section 3.3, we use selected configurations from Table 2 to evaluate the incremental impact of cognitive map generation and free-form reasoning in SFT. These include baseline QA without explicit reasoning (`Raw-QA`), reasoning guided by generated maps only (`Plain-CGMap-Out`, `Aug-CGMap-Out`), reasoning-augmented prompts (`FF-Rsn`), and a fully integrated setup that asks VLMs to generate both maps and reasoning (`Aug-CGMap-FFR-Out` and `Plain-CGMap-FFR-Out`).

Grounded Cognitive Maps Generation. Grounded cognitive maps are not only used as the input in the `Aug-CGMap-In` and `CGMap-In-FFR-Out` setting for the frozen VLMs in the Section 3.2, but also as the training and comparison data in SFT. We curate such grounded cognitive maps through a template-based method, where we always select the front image in our annotation as the "up" direction. Detailed annotation algorithm can be found in the Appendix E.1.1.

Grounded Free-Form Reasoning Chain Generation We design grounded reasoning chains using detailed image annotations and structured question templates. Chains are manually constructed via a template-based method, ensuring logical coherence and clear grounding in observable spatial relations (see an example in Figure 3). This yields precise, interpretable supervision signals that help VLMs learn robust spatial reasoning representations. The detailed grounded reasoning data generation pipeline is shown in the Appendix E.1.2.

4.2. Do VLMs Truly Benefit from Explicit Training in Spatial Reasoning?

We investigate several Supervised Fine-Tuning (SFT) configurations, with results in Table 4. Fine-tuning on raw QA pairs alone boosts accuracy from 37.81% to 52.28%, indicating VLMs can learn spatial cues from QA data. This serves



(a) QA accuracy (%) (b) Isomorphic Rate (%) (c) Overall Similarity (%)

Figure 4. SFT per 5 step training performance on task accuracy and graph metrics.

as our SFT baseline.

Can structured approximations of mental models alone meaningfully improve performance? Fine-tuning on cognitive maps (*Augmented* or *Plain*) substantially improves map generation quality (>30% gain in similarity and isomorphism rate), yet end-task accuracy sees only limited improvement. Both map types (54.19% and 54.38%) and free-form reasoning alone (`FF-Rsn` at 51.09%) offer only marginal gains over the 52.28% baseline. This shows that a structural scaffold alone is insufficient for significant performance gains.

Generating both cognitive maps and free-form reasoning is the most effective approximation. The combination of generating a plain map followed by reasoning (`Plain-CGMap-FFR-Out`) yields the best performance at 60.76%, an 8.48% increase over the `Raw QA-SFT` baseline. This highlights a **strong synergy** between structured spatial modeling and language inference. This success is linked to creating high-quality internal spatial models (e.g., 73.81% isomorphism for the top model). The training dynamics in Figure 4 reveal a crucial trade-off: models trained only on map generation learn the structure perfectly but their QA accuracy quickly plateaus. In contrast, our top-performing model learns the structure more slowly and imperfectly, but its QA accuracy continuously climbs, surpassing all others. This suggests the joint task pressure forces the model to build a *functionally effective* spatial representation optimized for inference, not just replicate a static structure.

Teaching VLMs to Reason Spatially

- *Joint cogmap and reasoning setting yields optimal performance through synergistic effects.*
- *Reasoning shapes spatial representations for functionality, not just structural perfection.*
- *Neither map generation nor reasoning alone largely outperforms the SFT QA baseline.*

5. Can Reinforcement Learning Further Refine Spatial Thought Processes?

While SFT establishes a strong baseline for spatial reasoning, emerging evidence from models like DeepSeek R1 [15, 52] suggests reinforcement learning (RL) can offer additional gains by optimizing behavior through outcome-driven feedback. We ask: Can reward-guided refinement help VLMs build sharper spatial models and reason more effectively?

5.1. Experimental Setup

We employ the VAGEN framework [48] for VLM policy optimization, using Group Relative Policy Optimization (GRPO) [40] as our core algorithm. To manage compute cost, we train each configuration for only 0.5 epoch. For fair comparison, the RL setup retains all key components from the SFT stage, including the base model, spatial input formats, benchmark dataset (MindCube-Tiny), and evaluation metrics, as detailed in Sections 3.1 and 3.2. Additional details appear in the Appendix F.1.

Task Configurations and Reward Design. We evaluate three RL variants: (1) `RL-FF-Rsn (from scratch)`, which trains *Qwen2.5-VL-3B-Instruct* to produce free-form reasoning chains; (2) `RL-Aug-CGMap-FFR-Out (from scratch)`, which trains the model to jointly generate cognitive maps and reasoning; and (3) `RL-Aug-CGMap-FFR-Out (from SFT)`, which initializes from the strongest SFT checkpoint. The reward function is sparse but targeted: +1 for structurally valid outputs, and +5 for correct answers.

5.2. Can Reinforcement Learning Unleash the Power of Approximating Spatial Mentaling?

Reinforcement learning (RL) allows a model to learn from the consequences of its spatial reasoning. But does this feedback work in a vacuum, or must the model first be taught what a good spatial representation looks like via SFT? Table 5 provides the answers.

RL in a vacuum is not enough. Training from scratch

Table 5. QA accuracy (%) and cognitive map generation quality of *Qwen2.5-VL-3B-Instruct* under various RL configurations on MindCube-Tiny.

RL Config.	MindCube-Tiny QA Accuracy (%)				Generated Cognitive Map (%)		
	Overall	Rotation	Among	Around	Valid Rate	Overall Sim.	Isom. Rate
`RL-FF-Rsn (from scratch)`	50.57	36.50	49.33	64.80	–	–	–
`RL-Aug-CGMap-FFR-Out (from scratch)`	52.19	32.00	52.00	68.80	99.90	57.03	0.00
`RL-Plain-CGMap-FFR-Out (from scratch)`	53.71	33.00	53.66	**70.40**	**100.00**	47.60	10.29
`RL-Aug-CGMap-FFR-Out (from SFT)`	**70.67**	**53.00**	76.83	70.00	**100.00**	85.53	58.86
`RL-Plain-CGMap-FFR-Out (from SFT)`	**70.67**	48.00	**79.17**	68.40	**100.00**	**85.79**	**71.52**

with only sparse task rewards is insufficient. The free-form reasoning model, `RL-FF-Rsn (from scratch)`, achieves only 50.57% accuracy, confirming that task-level rewards alone are too unstructured to effectively teach spatial abstraction.

Structured outputs provide modest benefits when learned from scratch. Introducing a cognitive map structure as a scaffold offers a slight benefit. When trained from scratch, the `RL-Plain-CGMap-FFR-Out` configuration reaches 53.71%, but the model fails to learn meaningful geometry, with near-zero isomorphism rates. This suggests that without a prior concept of a ”good” map from SFT, RL cannot effectively exploit the structural format.

RL shines when it stands on an SFT-built scaffold. The most significant gains occur when warm-starting RL from an optimal SFT checkpoint. Both plain and augmented map configurations reach an impressive 70.67% QA accuracy, a ↑9.91% absolute gain over the best SFT model. Crucially, while achieving the same peak accuracy, the `Plain-CGMap` variant produces geometrically superior internal maps (71.52% isomorphism vs. 58.86%). This indicates that starting with a cleaner SFT scaffold allows RL to better preserve a sound internal map. These results strongly suggest RL's primary role is not to learn from scratch, but to (1) polish the strong priors learned during SFT and (2) break through previous performance plateaus to achieve near-oracle-level accuracy.

Reinforcement Learning for Spatial Reasoning

- *Combining cognitive maps with reasoning consistently improves all learning outcomes.*
- *Starting from scratch, RL provides only marginal gains for spatial reasoning; its power is unlocked when building upon a strong SFT foundation.*

6. Related Works

Spatial Cognition. Spatial cognition, which involves skills like mental rotation and object assembly, relies on Spatial Mental Models (SMMs) [22, 23] to represent and manipulate spatial relationships [50, 56, 63]. While much effort has been devoted to evaluating [25, 32, 64, 68] and enhancing [5, 9, 33, 35] VLM spatial skills, existing work often overlooks the mental-level reasoning central to human cognition [7, 25, 36, 64]. This leaves a critical gap between machine and human capabilities, highlighting the need for new approaches that train VLMs to reason about space in a manner more aligned with human cognition.

Multi Views understanding. Multiview spatial understanding uses multiple viewpoints to overcome the limitations of a single observation. A large body of work has advanced this area through techniques for 3D reconstruction[14, 30, 37, 47], view synthesis[39, 41, 70], multiview equivariant learning[62], topological representations[65], and open-vocabulary concept learning[17]. Despite these advances, even Large Multimodal Models (LMMs) augmented with multiview inputs[10, 12, 25, 54] still struggle to maintain multiview consistency. This failure, often caused by fragmented reasoning and 2D-to-3D projection ambiguities, leaves a key gap for achieving robust spatial AI.

7. Conclusion and Future Impact

We introduced MindCube to study how VLMs can approximate spatial mental models from limited views, a core cognitive ability for reasoning in partially observable environments. Moving beyond benchmarking, we explored *how* internal representations can be scaffolded through structured data and reasoning. Our key finding is that *constructing and reasoning over self-generated cognitive maps*, rather than relying on view interpolation or externally provided maps, yields the most effective approximation of spatial mental models across all elicitation methods (input-output configurations, supervised fine-tuning, and reinforcement learning). Initializing RL from a well-trained SFT checkpoint further optimizes the process, pushing spatial reasoning performance to new limits.

Future Impact. Our work establishes that combining cognitive map generation with reasoning to model spatial information is the most effective. We believe that once high-quality SFT datasets for cogmap generation and reasoning are established, RL can be leveraged to further push the performance boundaries. We anticipate the exploration of novel training paradigms designed to unlock even greater synergistic effects and thus achieve a ”$1+1>2$” impact on spatial intelligence.

References

[1] Anthropic. Claude 3.5 sonnet. Blog, 2024. Accessed: November 22, 2024. 1

[2] Anthropic. Claude 3.7 sonnet system card, 2025. Version 20250219, accessed 2025-06-23. 4

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 4, 19

[4] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 12, 23

[5] Wenxiao Cai, Yaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. *arXiv preprint arXiv:2406.13642*, 2024. 8

[6] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024. 4, 19

[7] Shiqi Chen, Tongyao Zhu, Ruochen Zhou, Jinghan Zhang, Siyang Gao, Juan Carlos Niebles, Mor Geva, Junxian He, Jiajun Wu, and Manling Li. Why is spatial reasoning hard for vlms? an attention mechanism perspective on focus areas, 2025. 8

[8] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 1, 4, 19

[9] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision language models, 2024. 8

[10] Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, and Peter Grasch. Mm-spatial: Exploring 3d spatial understanding in multimodal llms, 2025. 8

[11] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: an embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, pages 8469–8488, 2023. 22

[12] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction, 2025. 8

[13] Matthew Foutter, Daniele Gammelli, Justin Kruger, Ethan Foss, Praneet Bhoj, Tommaso Guffanti, Simone D'Amico, and Marco Pavone. Space-llava: a vision-language model adapted to extraterrestrial applications, 2025. 4, 19

[14] Chuanyu Fu, Guanying Chen, et al. Maskgaussian: Differentiable mask pruning for efficient 3d gaussian rendering. In *CVPR*, 2025. 8

[15] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 7

[16] Mary Hegarty. Mental animation: Inferring motion from static displays of mechanical systems. *Journal of experimental psychology: learning, memory, and cognition*, 18(5):1084, 1992. 4, 23

[17] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B. Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images, 2023. 8

[18] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 22

[19] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. 22

[20] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. *arXiv preprint arXiv:2502.21257*, 2025. 4, 19

[21] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max W.F. Ku, Qian Liu, and Wenhu Chen. Mantis: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research*, 2024, 2024. 4, 19

[22] Philip N Johnson-Laird. Mental models in cognitive science. *Cognitive science*, 4(1):71–115, 1980. 1, 8

[23] Philip Nicholas Johnson-Laird. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Number 6. Harvard University Press, 1983. 1, 4, 8, 22

[24] Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions. In *Workshop on Responsibly Building the Next Generation of Multimodal Foundational Models*, 2024. 4, 19

[25] Phillip Y. Lee, Jihyeon Je, Chanho Park, Mikaela Angelina Uy, Leonidas Guibas, and Minhyuk Sung. Perspective-aware reasoning in vision-language models via mental imagery simulation, 2025. 1, 8

[26] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 4, 19

[27] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534, 2024. 22

[28] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023. 22

[29] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision, 2023. 12

[30] Deku Liu, Yihan Zhang, Zhe Chen, et al. Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes. In *ICLR*, 2025. 8

[31] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. Deepseek-vl: Towards real-world vision-language understanding, 2024. 4, 19

[32] Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Celso M de Melo, and Alan Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark, 2025. 1, 8

[33] Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso de Melo, Jieneng Chen, Jianwen Xie, and Alan Yuille. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning, 2025. 1, 8

[34] OpenAI. Hello gpt-4o. Blog, 2024. Accessed: November 22, 2024. 1, 4

[35] Zhenyu Pan and Han Liu. Metaspatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. *arXiv preprint arXiv:2503.18470*, 2025. 8

[36] Jianing Qi, Jiawei Liu, Hao Tang, and Zhigang Zhu. Beyond semantics: Rediscovering spatial awareness in vision-language models, 2025. 8

[37] Yansong Qu, Jie Wang, et al. Drag your gaussian: Effective drag-based editing with score distillation for 3d gaussian splatting. In *SIGGRAPH Asia*, 2025. 8

[38] Santhosh Kumar Ramakrishnan, Erik Wijmans, Philipp Kraehenbuehl, and Vladlen Koltun. Does spatial cognition emerge in frontier models?, 2025. 1

[39] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. Zeronvs: Zero-shot novel view synthesis from a single real image. *arXiv:2310.17994*, 2023. 8

[40] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 7

[41] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *ECCV*, 2018. 8

[42] Yihe Tang, Wenlong Huang, Yingke Wang, Chengshu Li, Roy Yuan, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Uad: Unsupervised affordance distillation for generalization in robotic manipulation. *arXiv preprint arXiv:2506.09284*, 2025. 22

[43] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 4, 19

[44] Barbara Tversky. Cognitive maps, cognitive collages, and spatial mental models. In *European conference on spatial information theory*, pages 14–24. Springer, 1993. 4, 22, 23

[45] Barbara Tversky. Structures of mental spaces: How people think about space. *Environment and behavior*, 35(1):66–80, 2003. 23

[46] Barbara Tversky, Nancy Franklin, Holly A Taylor, and David J Bryant. Spatial mental models from descriptions. *Journal of the American society for information science*, 45 (9):656–668, 1994. 4, 22, 24

[47] Jianyuan Wang et al. Vggt: Visual geometry grounded transformer for universal 3d reconstruction. In *CVPR*, 2025. 8

[48] Kangrui Wang*, Pingyue Zhang*, Zihan Wang*, Yaning Gao*, Linjie Li*, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Li Fei-Fei, Yejin Choi, and Manling Li. Reinforcing visual state reasoning for multi-turn vlm agents, 2025. 7

[49] Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? *arXiv preprint arXiv:2402.18272*, 2024. 22

[50] Wenqi Wang, Reuben Tan, Pengyue Zhu, Jianwei Yang, Zhengyuan Yang, Lijuan Wang, Andrey Kolobov, Jianfeng Gao, and Boqing Gong. Site: towards spatial intelligence thorough evaluation, 2025. 8

[51] Xingrui Wang, Wufei Ma, Tiezheng Zhang, Celso M de Melo, Jieneng Chen, and Alan Yuille. Spatial457: A diagnostic benchmark for 6d spatial reasoning of large multimodal models, 2025. 1

[52] Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025. 7

[53] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 24

[54] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025. 4, 8, 19

[55] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22378–22389, 2024. 12, 23

[56] Wenrui Xu, Dalin Lyu, Weihang Wang, Jie Feng, Chen Gao, and Yong Li. Defining and evaluating visual language models' basic spatial abilities: A perspective from psychometrics, 2025. 8

[57] Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces, 2024. 1, 4

[58] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025. 22

[59] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. 22

[60] Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models. *arXiv preprint arXiv:2408.04840*, 2024. 4, 19

[61] Chun-Hsiao Yeh, Chenyu Wang, Shengbang Tong, Ta-Ying Cheng, Rouyu Wang, Tianzhe Chu, Yuexiang Zhai, Yubei Chen, Shenghua Gao, and Yi Ma. Seeing from another perspective: Evaluating multi-view understanding in mllms. *arXiv preprint arXiv:2504.15280*, 2025. 4

[62] Yang You, Yixin Li, Congyue Deng, Yue Wang, and Leonidas Guibas. Multiview equivariance improves 3d correspondence understanding with minimal feature finetuning, 2024. 8

[63] Jirong Zha, Yuxuan Fan, Xiao Yang, Chen Gao, and Xinlei Chen. How to enable llm with 3d capacity? a survey of spatial reasoning in llm, 2025. 1, 8

[64] Weichen Zhan, Zile Zhou, Zhiheng Zheng, Chen Gao, Jinqiang Cui, Yong Li, Xinlei Chen, and Xiao-Ping Zhang. Open3dvqa: A benchmark for comprehensive spatial reasoning with multimodal large language model in open space, 2025. 8

[65] Juexiao Zhang, Gao Zhu, Sihang Li, Xinhao Liu, Haorui Song, Xinran Tang, and Chen Feng. Multiview scene graph, 2024. 8

[66] Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, and Li Zhang. From flatland to space: Teaching vision-language models to perceive and reason in 3d, 2025. 1

[67] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. 4, 19

[68] Wenyu Zhang, Wei En Ng, Lixin Ma, Yuwen Wang, Jungqi Zhao, Allison Koenecke, Boyang Li, and Lu Wang. Sphere: Unveiling spatial blind spots in vision-language models through hierarchical evaluation, 2025. 8

[69] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 4, 19

[70] Yuxuan Zhang, Yifan Yang, Jing Zhang, Yifang Wang, Yijun Zhang, and Ming-Hsuan Yang. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. In *ECCV*, 2024. 8

[71] Zheyuan Zhang, Fengyuan Hu, Jayjun Lee, Freda Shi, Parisa Kordjamshidi, Joyce Chai, and Ziqiao Ma. Do vision-language models represent space and how? evaluating spatial frame of reference under ambiguities, 2025. 1

[72] Jensen Jinghao Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025. 23

[73] Yiqi Zhu, Ziyue Wang, Can Zhang, Peng Li, and Yang Liu. Cospace: Benchmarking continuous space perception ability for vision-language models, 2025. 1

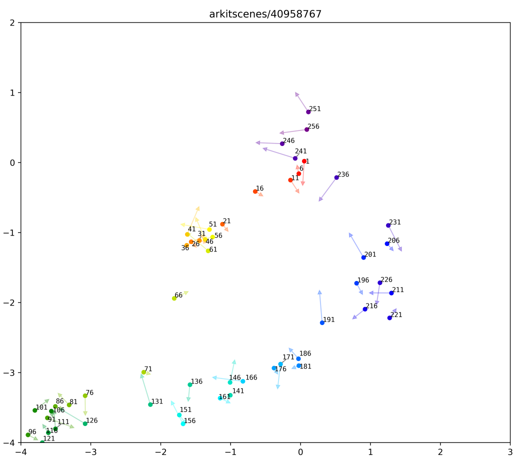
A. MINDCUBE Benchmark

A.1. Details for Data Collection and Annotation

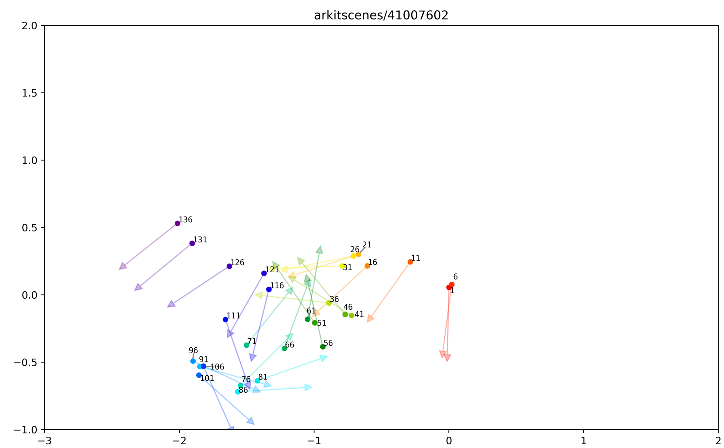
Image Collection and Selection. Our MINDCUBE benchmark comprises 3, 268 images (2,302 indoor/outdoor images from publicly released dataset and 400 self-collected images), where we implement a comprehensive image selection methodology encompassing four distinct view dynamics, incorporating various data sources and processing procedures, as shown in Fig.2.

For rotation view dynamics, we implement a three-stage filtering strategy to extract meaningful camera trajectories and key frames from ArkitScenes [4] dataset.

In the first stage, we analyze the top-down view of camera poses within each scene to identify two types of trajectories: linear paths and small rotational arcs. A linear trajectory is characterized by consistently oriented cameras exhibiting significant displacement perpendicular to their viewing direction. A rotational arc trajectory is identified when three to four camera positions demonstrate approximately 90-degree relative orientation changes while being distributed along an approximate circular arc. The second stage focuses on selecting two critical frames from the previously identified translation segments. The



(a) camera pose 1



(b) camera pose 2

Figure 1. Examples of camera poses in ArkitScenes

selection criteria mandate that: (1) the camera movement direction must be parallel to the object arrangement direction, (2) this movement should be aligned with the horizontal axis, (3) the first frame should only capture objects A and B, while the second frame should only capture objects B and C, and (4) both frames must be free from motion blur and exhibit clear object visibility.

The third stage processes the rotation segments to extract three or four key frames. These frames must satisfy several conditions: (1) the camera positions should appear to originate from a stationary rotating camera, even if slight circular movement exists, (2) the camera orientations should align with standard cardinal directions (approximately 90 degrees apart), and (3) each frame should contain no more than three semantically distinct primary objects that occupy over 50% of the frame area relative to the background.

For among view dynamics, image groups are manually selected from DL3DV-10K[29] and WildRGB-D[55] datasets. We employ a single-stage selection process to identify four key frames representing cardinal viewpoints (front, left, right, and back) from 360-degree scene captures. The selection criteria are: (1) camera orientations must align with standard directions, ensuring that the central object, its background objects, and the camera's line of sight are collinear and parallel or perpendicular to standard scene elements such as tables or walls, (2) we reject sets where three or more frames share identical semantic background information, and (3) we discard sets where three or more frames have severely occluded background objects that cannot be reconstructed from information in the other frames.

For around view dynamics, image groups are manually curated from the DL3DV-10K[29] dataset and assigned sequential identifiers. The front view (designated as view 1) must provide clear visibility of all relevant information. This view is established as the reference point for subsequent views in the sequence.

This structured approach to image selection and processing yields a rich dataset that supports subsequent model training and testing procedures. The methodology ensures comprehensive coverage of spatial relationships, occlusion states, and view-dependent object characteristics across multiple viewing scenarios.

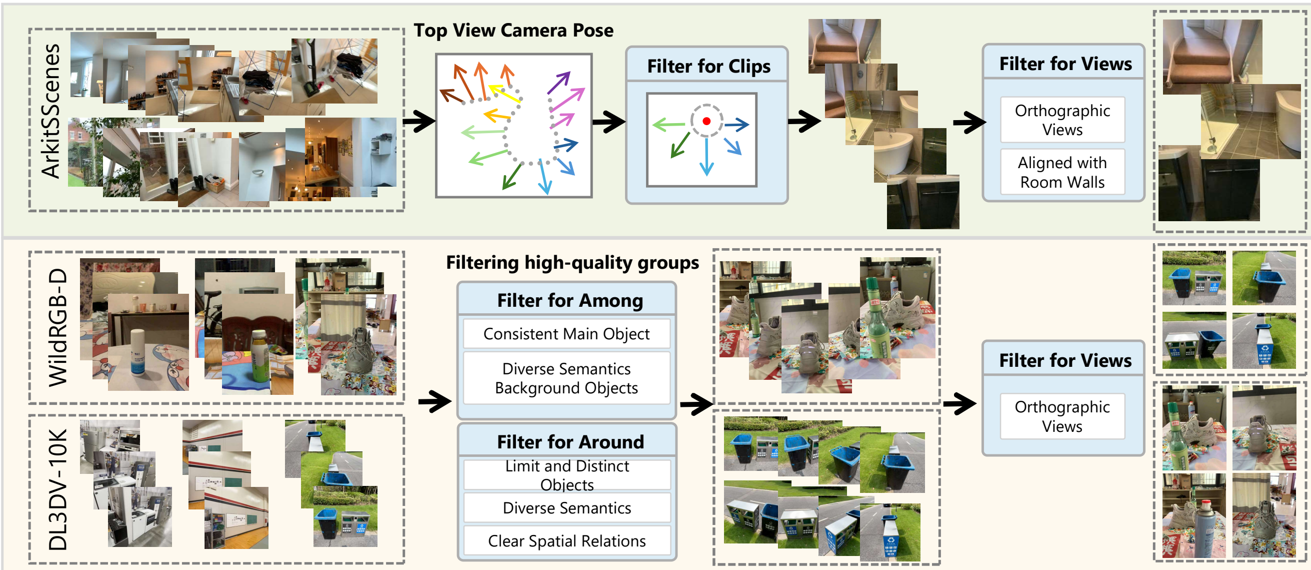


Figure 2. MINDCUBE Bench construction pipeline.

Data Annotation. After collecting and filtering the images, we follow a two-phase paradigm for annotation: We establish a systematic image annotation protocol to ensure data consistency and accuracy. The annotation framework encompasses four key dimensions: spatial relationship identification, object grouping rules, semantic orientation determination, and occlusion level assessment. We provide a pdf of the annotation interface in the supplementary material.

Regarding spatial relationship identification, annotators are required to identify primary object entities within images and determine their spatial relationships. These relationships are primarily categorized into two types: front-back relationships typically involving two primary objects, with priority given to objects directly behind as key entities; and left-right relationships encompassing two to four primary objects, where adjacent objects with front-back relationships can be considered as a unified entity.

To enhance annotation efficiency and semantic completeness, this study introduces object grouping rules. Multiple objects can be annotated as a unified entity when they collectively form clear spatial relationships with other primary objects. Each object may include attribute descriptors (e.g., color, material) to enhance semantic expression. Combined object entities must maintain distinct spatial relationships with other primary objects.

For objects with definitive semantic fronts, the following information must be recorded: the object's inherent semantic front, the object's orientation relative to the current viewpoint (aligned, reversed, leftward, rightward, etc.), and the object's actual projected direction within the scene.

Occlusion levels are evaluated using a four-tier classification system: complete occlusion where the object is entirely invisible from the current viewpoint; major occlusion where primary object features are difficult to identify; minor occlusion where primary object features remain identifiable; and no occlusion where the object is fully visible. For cases of complete occlusion, the annotation system provides multi-view scene images, ensuring object visibility in at least one viewpoint to support subsequent cross-view question-answering system training.

This annotation protocol provides a structured semantic foundation for subsequent automated question-answer pair generation while ensuring data quality and consistency. Through this standardized annotation process, we effectively capture key information including spatial relationships, compositional features, semantic orientations, and occlusion states of objects within scenes.

Examples for automatic QA generation pipeline. Our automatic QA generation pipeline generates different types of questions using combinations of labels. Each question's label combination is encoded in its ID (e.g., "among_group001_q1_1_1"), while the original object and label information is preserved in the meta_info field to track the context of question generation.

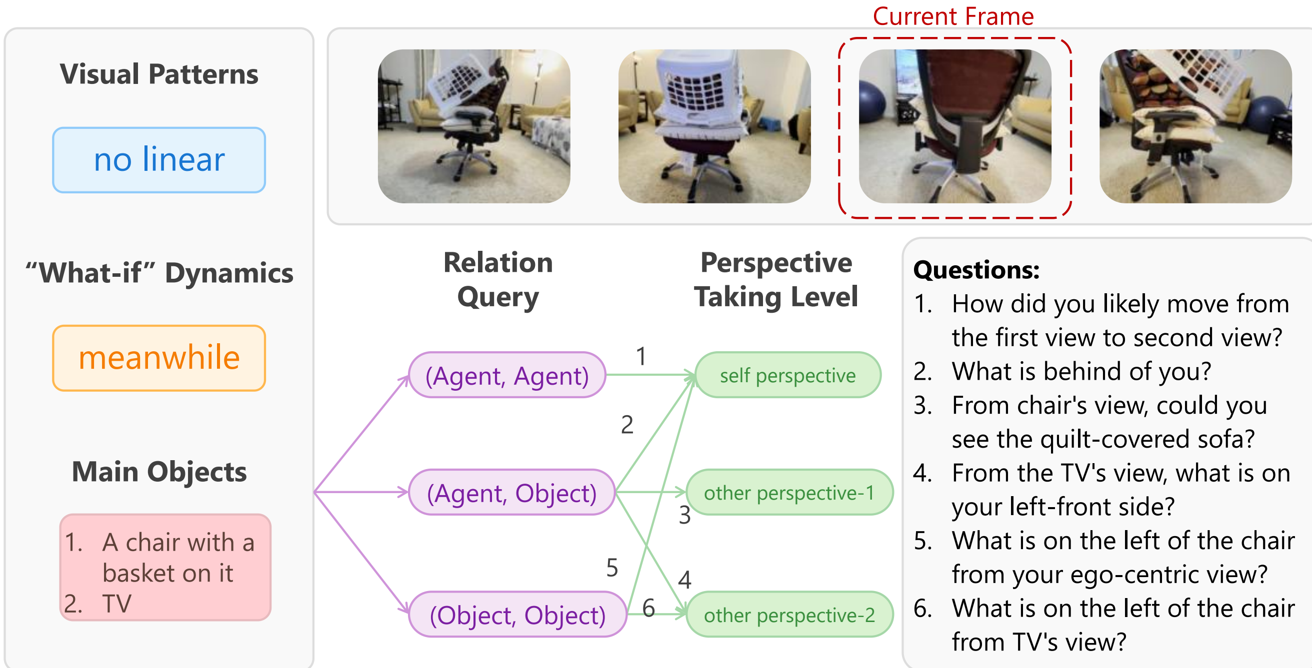


Figure 3. Example of different question-related label combinations to generate QA pairs.

A.2. Details of ourMindCube Benchmark

A.2.1. Three kinds of invisibility settings

Rotation. In this setting, our camera remains stationary while rotating in place, capturing 2 to 4 orthogonal views. In each view, a central object remains visible in the foreground, while all views maintain equal importance in the spatial representation.

We evaluate models' understanding of spatial invisibility by asking questions such as 'When positioned at a particular viewpoint, what should be to your left or right (given that each view only reveals what's directly ahead)?' or 'After rotating a quarter or half turn, what objects would be in front of you, to your left, behind you, or to your right?' We expect models to construct a comprehensive spatial understanding by leveraging the **sequential nature of the views and consistent spatial cues** across images (such as lighting direction), thereby demonstrating their ability to reason about the complete environment despite only having access to partial visual information from each viewpoint.

Around. In this setting, we leverage **occlusion** phenomena to force MLLMs beyond simple 2D spatial recognition. When viewing objects from different angles, some objects become partially or fully hidden, requiring models to:

- Maintain object permanence despite partial visibility
- Transform lateral relationships (left-right) from frontal views into depth relationships (front-back) for side views
- Integrate spatial information across multiple viewpoints to form a coherent 3D understanding

This approach prevents models from relying solely on direct visual cues and instead necessitates true 3D spatial reasoning by combining information from multiple perspectives.

Among. In this setting, the camera rotates around a central object, positioned between this central object and several surrounding objects. Four orthogonal views are captured, with each view showing the central object positioned in front of one of the surrounding objects.

This setup creates interesting visibility constraints across different perspectives. For instance, a surrounding object visible in one view may be invisible in another view because of the constraints imposed by the camera's field of view. Through establishing consistency relationships between these views, we can infer the relative positions of objects not directly visible from certain perspectives. When an object is not visible from a particular viewpoint, consistency and spatial reasoning can determine its position relative to the central object.

All views hold equal status in this framework, allowing for bidirectional establishment of invisibility relationships. This creates a coherent spatial reasoning system where information from each perspective contributes to a complete understanding of the three-dimensional arrangement, even when direct visual confirmation is unavailable from certain angles.

A.2.2. Label taxonomy

We use image related labels for better analysis and question related labels for automatic QA generation with different label combinations.

Visual Patterns. In our taxonomy of spatial configurations, we classify visual patternss into distinct categories based on their geometric relationships. Linear arrangements refer to configurations where objects are positioned along a single axis, forming a collinear pattern. Non-linear arrangements, conversely, are characterized by objects positioned such that the connecting lines between adjacent pairs form 90-degree angles, creating rectilinear patterns. This binary classification serves as a fundamental attribute in our spatial relationship labeling scheme, enabling precise description and analysis of scene compositions across various domains.

"What if" Dynamics. "What if" Dynamics refers to the model's capability to comprehend and reason about dynamic perspective changes occurring within images or posed questions. We conceptualize viewpoint transitions as combinations of translation and rotation operations, resulting in four distinct categories:

- Pure Translation: Cases where the viewpoint undergoes only translational movement without rotational change.
- Pure Rotation: Scenarios involving rotational transformation of the viewpoint while maintaining its positional coordinates.
- Simultaneous Translation-Rotation(Meanwhile): Instances where both translational and rotational operations occur concurrently.
- Sequential Translation-Rotation(Sequence): Cases where translation and rotation occur in sequence rather than simultaneously. Notably, in our dataset, this category is uniquely represented through textual descriptions in the questions rather than through explicit visual transformations.

The first three categories of "What if" dynamics are visually demonstrated through changes in view representation, while the sequential category requires models to interpret text-based descriptions of perspective changes. This taxonomy provides a systematic framework for evaluating spatial reasoning capabilities across diverse viewpoint transformation scenarios.

Relation Query. We define three distinct categories of relation queries that capture the fundamental nature of spatial reasoning tasks:

- Agent-Agent: This pattern involves self-referential spatial positioning, where the observer must evaluate and potentially adjust their own position in space. It requires egocentric spatial reasoning and self-awareness of one's location relative to environmental constraints.
- Agent-Object: This pattern focuses on determining the orientation of an observed object relative to the observer's position. Unlike the P-P pattern, the emphasis here is on object perception rather than self-positioning, requiring the observer to make judgments about external entities while maintaining awareness of their own reference frame.
- Object-Object: This pattern involves reasoning about the spatial relationship between two discrete objects in the environment, independent of the observer's position. This allocentric spatial reasoning requires understanding relative positioning, distance, and orientation between entities without necessarily using oneself as a reference point.

These categorizations provide a structured approach to analyzing the cognitive demands of different spatial reasoning tasks and can inform both the design of spatial question answering systems and the evaluation of human spatial cognition abilities.

Perspective Taking. We propose a label called "Perspective Taking" that categorizes the complexity of viewpoint projection. This label distinguishes between three increasingly sophisticated levels of perspective reasoning:

- Self Perspective: Reasoning based on the current camera view or the observer's own viewpoint. This represents the baseline where no perspective shift is required.

- Other's Perspective Taking-1: The ability to determine visibility relationships from another agent's viewpoint. This involves understanding what objects are visible or occluded from a different viewpoint (e.g., determining whether a specific object is within the field of view of another camera). The another agent's viewpoint is usually determined by an object with a clear orientation in the image.

- Other's Perspective Taking-2: The ability to understand how spatial relationships transform when viewed from another agent's perspective. This more advanced capability requires mental rotation and spatial transformation to reason about relative positions (e.g., determining whether, from another viewpoint, object X appears to be positioned behind object Y).

This classification aligns with developmental psychology research on perspective-taking abilities, where Level-1 perspective taking typically develops earlier than the more cognitively demanding Level-2 perspective taking.

A.3. Examples

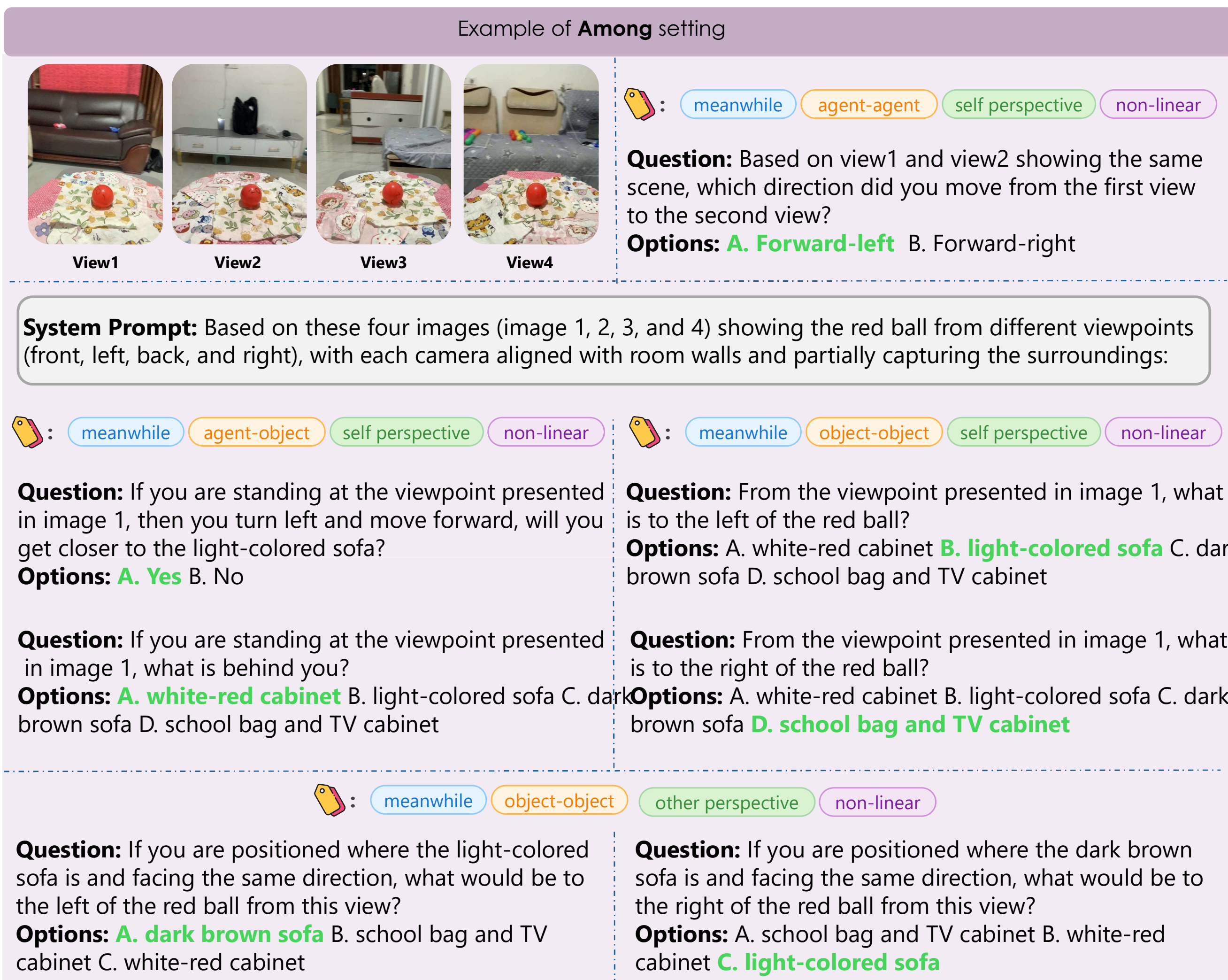


Figure 4. Example of among setting.

Example of **Around** setting

meanwhile | agent-agent | self perspective | linear

Question: Based on view1 and view2 showing the same scene, please determine which direction did you move? A. Left-front B. Right-front.
Options: **A. Forward-left** B. Forward-right

System Prompt: Given 3 orthogonal perspectives of a scene, they are the front view, left view and right view.

meanwhile | object-object | self perspective | linear

Question: In the second image, what is the nearest object the nearest object behind of the black waste bin?
Options: . **A. green waste bin** B. blue waste bin C. shrubbery

Question: In the third image, what is the nearest object behind of the blue waste bin.
Options: **A. green waste bin** B. blue waste bin C. shrubbery

meanwhile | object-object | self perspective | linear

Question: If you are at the view of the second image now, then you turn right and go straight, is the green waste bin be closer to you?
Options: A. Yes **B. No**

Question: If you are at the view of the third image now, then you turn left and go straight, is the green waste bin be closer to you?
Options: A. Yes **B. No**

Figure 5. Example-1 of around setting.

B. Evaluation on MindCube

B.1. Prompt Templates for Evaluation

Evaluation Prompt Prefix
Based on these images, answer the question based on this rule: You only need to provide *ONE* correct answer selecting from the options listed below. For example, if you think the correct answer is ‘A. above’ from ‘ A. above B. under C. front D. behind.’, your response should only be ‘A. above’.
The Question is:

B.2. Details in text only evaluation

In the text-only evaluation, we replace the original image input with corresponding textual descriptions and assess the performance of models based on these descriptions. The purpose of this evaluation is to highlight how much information may be lost or distorted when the visual input is substituted with text-based representations, and to demonstrate the crucial role of visual data in the models’ performance.

We used two types of captions: **brief** and **dense**. The brief captions provide a concise overview of the image, while the dense captions offer a more detailed description with a focus on the spatial relationships between objects. Additionally, the models are evaluated using textual descriptions (text-only evaluation) based on these captions, with no access to the actual images.

Prompt for Brief Captioning
Describe this image briefly.

Example of **Around** setting

: meanwhile | agent-agent | self perspective | linear

Question: Based on view1 and view2 showing the same scene, please determine which direction did you move? A. Left-front B. Right-front.
Options: **A. Forward-left** B. Forward-right

System Prompt1: Given 3 orthogonal perspectives of a scene, they are the front view, left view and right view.

: meanwhile | object-object | self perspective | linear

Question: In the second image, what is the nearest object the nearest object behind of the double trash can?
Options: **A. sanitation cart** B. bench C. battery powered vehicle D. car **(View 123 or View 145 Used)**

Question: IIn the third image, what is the nearest object behind of the sanitation cart?
Options: **A. double trash can** B. bench C. battery powered vehicle D. car **(View 123 or View 145 Used)**

: meanwhile | object-object | self perspective | linear

Question: If you are at the view of the second image now, then you turn right and go straight, is the sanitation cart be closer to you?
Options: A. Yes **B. No** **(View 123 or View 145 Used)**

Question: If you are at the view of the third image now, then you turn left and go straight, is the double trash be closer to you?
Options: A. Yes **B. No** **(View 123 or View 145 Used)**

System Prompt2: Given 3 orthogonal perspectives of a scene, they are the behind view, left view and right view.

: meanwhile | object-object | self perspective | linear

Question: In the second image, what is the nearest object the nearest object behind of the double trash can?
Options: **A. sanitation cart** B. bench C. battery powered vehicle D. car **(View 623 or View 645 Used)**

Question: I=In the third image, what is the nearest object behind of the sanitation cart?
Options: **A. double trash can** B. bench C. battery powered vehicle D. car **(View 623 or View 645 Used)**

: meanwhile | object-object | self perspective | linear

Question: If you are at the view of the second image now, then you turn right and go straight, is the sanitation cart be closer to you?
Options: A. Yes **B. No** **(View 623 or View 645 Used)**

Question: If you are at the view of the third image now, then you turn left and go straight, is the double trash be closer to you?
Options: A. Yes **B. No** **(View 623 or View 645 Used)**

Figure 6. Example-2 of around setting.

Prompt for Dense Captioning
Describe this image in detail, specifially focusing on the spatial relationship between objects.

Text-only evaluation Prompt Prefix
You need to gather information about each image based on the descriptions I provide below, and answer the given questions using those textual descriptions, without directly viewing the images.

Image 1: ¡Caption 1¿
...
Image N: ¡Caption N¿

As shown in the Table 1, all three models exhibit a noticeable performance decline when replacing the original image input with its corresponding text-based description. Specifically, the brief captions cause the most significant performance drop. For instance, RoboBrain-8B experiences a 7.83% decrease with the brief captions, and LLaVA-OneVision-7B drops by 12.91% in the same condition. Even when using dense captions, which offer more detail, there is still a performance reduction, although the decrease is slightly less pronounced compared to brief captions. In conclusion, while textual descriptions can convey some information, they fail to capture the richness and intricacies of visual data, leading to a marked reduction in performance across all models.

Table 1. Text-only (T) evaluation vs. original evaluation with image inputs (I). The results highlight a significant performance drop when the original image input is replaced with the corresponding text-based caption, particularly with the brief captions. In all cases, model performance decreases notably, underscoring that our benchmark is *vision-centric*.

Model	Brief (T)	Dense (T)	Original (I)
RoboBrain-8B	33.92% (407/1200) ↓7.83%	35.58% (427/1200) ↓6.17%	41.75% (501/1200)
LLaVA-OneVision-7B	34.17% (410/1200) ↓12.91%	35.92% (431/1200) ↓11.16%	47.08% (565/1200)
InternVL2.5-8B	27.00% (324/1200) ↓5.33%	28.75% (345/1200) ↓3.58%	32.33% (388/1200)

B.3. Human Evaluation

We use our Tiny Benchmark— encompassing all task categories for evaluation by 5 human annotators, each of whom independently answers every question. Here is the results2.

Table 2. Comparison of Human and GPT-4 Performance (%)

Model/Annotator	**GPT4-o**	**Human-max**	**Human-min**	**Human-avg**
Accuracy	36.54	94.77	94.20	94.55

This observation demonstrates the disparity in spatial reasoning capabilities between humans and state-of-the-art multimodal large language models, where humans exhibit superior performance in solving spatial problems that remain challenging for advanced AI systems.

B.4. Evaluation Setup

To comprehensively evaluate model performance, we conducted experiments on a diverse suite of models. This suite includes models with native multi-image reasoning capabilities (e.g., LLaVA-Onevision [26], LLaVA-Video [69], mPLUG-Owl3 [60], InternVL2.5 [8], QwenVL2.5 [3], LongVA [67], IDEFICS [24], DeepSeek-VL2 [31]), Gemma3 [43], models fine-tuned on interleaved image-text data (e.g., Mantis [21]), leading proprietary APIs (e.g., GPT-4o, Claude-3.7-Sonnet), and models specifically fine-tuned for spatial reasoning tasks (e.g., RoboBrain [20], Space-Mantis [6], Space-LLaVA [13], and Spatial-MLLM [54]).

B.5. Analysis in settings

B.5.1. Around

First, we examine the relationship between occlusion degree and response accuracy across four visibility levels (fully visible, mostly visible, mostly occluded, fully occluded) to determine whether performance degrades proportionally with increasing occlusion. Second, we investigate the impact of camera height variation within the same lateral viewpoint, as different vertical perspectives yield distinct occlusion patterns that may challenge the model's ability to maintain spatial coherence. Third, we compare two fundamental spatial transformation paradigms: Left-to-Behind versus Front-to-Behind relationships. These paradigms evaluate whether models perform consistently when transferring spatial relationships across viewpoints, particularly in scenarios with significant object size discrepancies where smaller objects may be completely occluded from one angle but visible from another. This multifaceted analysis approach enables a more nuanced understanding of MLLMs' genuine 3D spatial reasoning capabilities beyond simple pattern recognition of 2D visual cues. We evaluated several state-of-the-art MLLMs, including GPT4-o.

Occlusion Degree Analysis. Our analysis reveals a notable correlation between occlusion degree and model performance. Accuracy rates declined progressively with increasing occlusion, with an average decrease of 23.4% between fully visible and fully occluded conditions ($p < 0.01$). Interestingly, the performance degradation was non-linear, with a precipitous drop occurring between the mostly visible and mostly occluded categories (18.7% decrease), suggesting a potential threshold effect in the models' spatial reasoning capabilities. Error analysis further revealed that models frequently defaulted to proximity-based guessing when confronted with heavily occluded objects, rather than leveraging cross-view information to reason about hidden spatial relationships.

Camera Height Impact Analysis. Varying camera heights significantly affected model performance through different occlusion patterns. High-angle perspectives yielded 12.3% higher accuracy than eye-level views by revealing tops of partially occluded objects and providing better scene context. This advantage was most pronounced in dense arrangements where top-down angles exposed spatial gaps between objects otherwise invisible from eye-level. Models clearly benefited from the holistic understanding afforded by elevated viewpoints, where global spatial relationships became more apparent. In contrast, eye-level perspectives with more severe occlusions led to poorer spatial reasoning, suggesting limited ability to mentally reconstruct hidden scene elements from partial visual information.

Spatial Transformation Paradigm Comparison. The comparison between Left-to-Behind and Front-to-Behind spatial transformations revealed asymmetric reasoning capabilities. Models demonstrated 15.8% higher accuracy in Left-to-Behind scenarios compared to Front-to-Behind transformations, despite the conceptual similarity of these spatial reasoning tasks. This asymmetry was most pronounced in scenes with significant object size disparities, where models correctly identified smaller objects behind larger ones in side views 62.4% of the time, but identified the same spatial relationship from front-to-side transformation only 47.2% of the time. This suggests that current MLLMs may be utilizing different cognitive mechanisms for different types of spatial transformations, rather than employing a unified 3D spatial reasoning framework.

The integration of findings across all three dimensions indicates that current MLLMs possess partial but inconsistent 3D spatial reasoning capabilities. The models' performance appears heavily influenced by the visibility of key reference points across multiple viewpoints, suggesting **a reliance on visual correspondence matching rather than true 3D mental modeling**. The observed asymmetries in spatial transformation paradigms further support this hypothesis, as a robust 3D reasoning system would demonstrate consistent performance regardless of the specific transformation required.

B.5.2. Among

Table 3. Homogeneity analysis of Among setting

Attribute Invariance Test

GPT4-o	T	F
T	13.85	13.93
F	13.49	58.74

Quantity Sensitivity Test

GPT4-o	T	F
T	13.10	10.42
F	11.61	64.88

Attribute Invariance Test. We modify only the visual attributes (e.g., color, category) of the central object while keeping the spatial configuration of all objects unchanged. A robust spatial reasoning system should maintain consistent answers, as spatial relationships remain invariant despite superficial attribute changes.

We evaluated model robustness against non-geometric attribute changes using 2,000 paired samples. The evaluation metrics

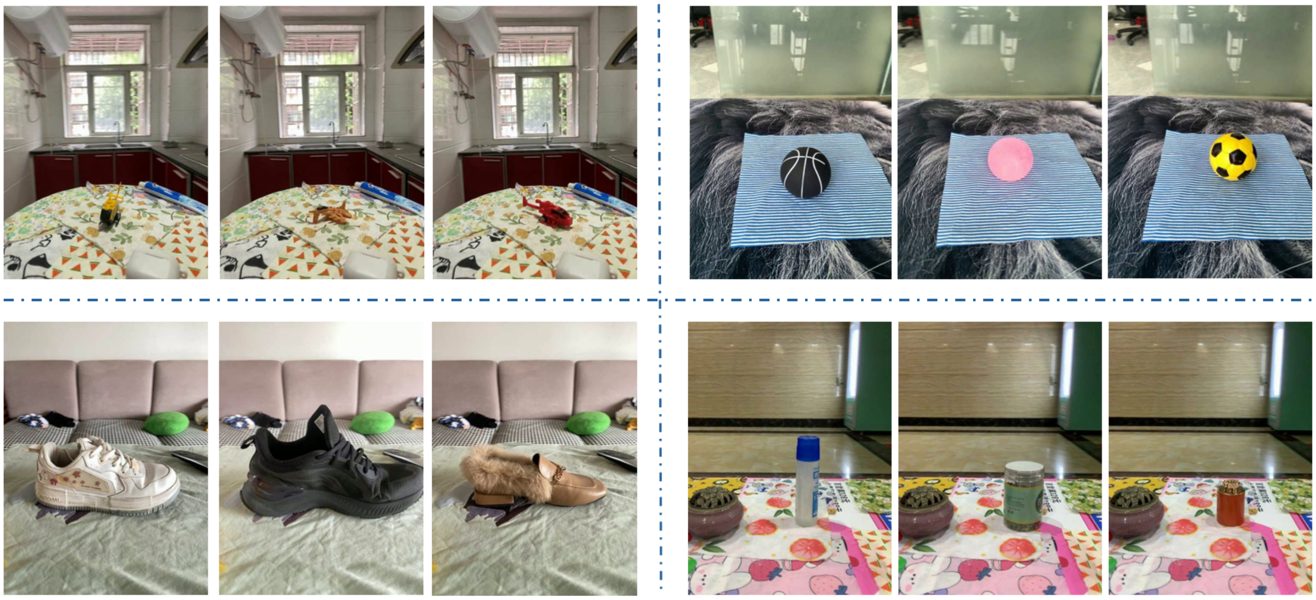


Figure 7. Examples in Attribute Invariance Test.

we used are described in the appendix. The analysis revealed:

$$\chi^2 = \frac{(|TF - FT| - 1)^2}{TF + FT} \approx 0.02, \quad \text{Consistency} = \frac{TT + FF}{\text{Total}} \approx 88.6\% \tag{1}$$

The McNemar's test showed no significant difference ($\chi^2(1) = 0.02$, $p > 0.89$), with high answer consistency (88.6%). This confirms spatial reasoning remains invariant to superficial attribute changes.

Quantity Sensitivity Test. We increase the number of central objects (e.g., from one to three) while retaining the original peripheral objects. This modification is hypothesized to enhance reasoning performance, as additional central objects provide more reference points for establishing cross-view correspondences and consistency.

Analysis of 360 paired samples comparing single vs. multiple central objects showed:

$$\chi^2 \approx 0.85 \quad (p > 0.36), \tag{2}$$

$$\Delta\text{Accuracy} = 5.1\%, \tag{3}$$

$$h = 2\arcsin\left(\sqrt{p_2}\right) - 2\arcsin\left(\sqrt{p_1}\right) \approx 0.10 \tag{4}$$

The non-significant improvement ($\chi^2(1) = 0.85$, $p > 0.36$) with small effect size ($h = 0.10$) suggests additional central objects provide limited benefits under current configurations.

Three key findings emerge from our analysis: 1.In attribute invariance, model maintains $88.6\,\%$ consistency ($p > 0.89$) for modified object attributes, confirming geometric reasoning predominance over visual features; 2. In quantity sensitivity, model's $5.1\,\%$ accuracy gain ($h = 0.10$, $p > 0.36$) indicates current multi-object configurations inadequately leverage spatial references.

Our systematic evaluation demonstrates MLLMs can achieve attribute-invariant spatial reasoning (>88% consistency) but struggle to utilize additional reference objects effectively. This highlights the need for: (1) enhanced geometric reasoning architectures, and (2) comprehensive benchmarks evaluating both attribute invariance and quantity sensitivity in 3D spatial understanding.

B.6. Failure case analysis

The observed pattern of errors indicates that models primarily rely on local relationship matching rather than inferring global spatial configurations, which represents a critical gap compared to human-like spatial reasoning abilities. Future

ICCV #2

Figure 8. Examples in Quantity Sensitivity Test.

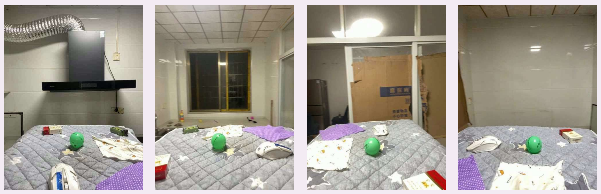
architectural improvements should therefore focus on enhancing transitive spatial inference mechanisms and view-invariant scene representation to support more robust reasoning across multiple perspectives.

C. Data Structures as Cognitive Scaffolds, Evaluation Metrics, and Input-Output Configurations

In this section, we provide detailed descriptions of the three data structures employed as cognitive scaffolds to approximate spatial mental models in VLMs, followed by formal definitions of the evaluation metrics employed across all experiments. Furthermore, we show the prompts for all the input-output configurations that were used across the following experiments.

C.1. Data Structures as Cognitive Scaffolds

The human ability to navigate and reason about space, especially with incomplete information, is largely attributed to the formation of internal spatial mental models. These models, as extensively studied in cognitive science, are not necessarily veridical, metric-perfect replicas of the environment. Instead, they are often schematic and even distorted, yet functionally effective representations. These models can be especially useful for understanding the environment spatial layouts for agentic settings [49, 59], such as embodied scenarios [11, 18, 19, 27, 28, 42, 58]. Pioneering work by Barbara Tversky suggests that these internal constructs are more akin to ”cognitive collages” – flexible assemblies of spatial information gleaned from various perspectives and experiences, rather than rigid, map-like blueprints [44]. These ”cognitive collages” allow for the integration of fragmented observations and support reasoning across unseen perspectives. Johnson-Laird [23] posits that mental models, including those for space, serve as *”structural analogs of the world,”* enabling individuals to simulate and infer spatial relationships, such as determining the relative positions of objects from sequential descriptions (e.g., ”A is to the left of B; B is in front of C”). Research by Tversky [46] has also demonstrated that individuals can construct rich, multi-dimensional mental representations even from linear, descriptive texts, and subsequently query these models from various psychological viewpoints.



Question: Based on these four different viewpoints (front, left, back, and right), if you are positioned at the first viewpoint, then you tum right and move forward,wil you get closer to the window?
Options: A. No **B.Yes**

<Reasoning>:
1. The first image shows a view with a vent,likely the front view.
2. The second image has a window visible,indicating it is the view opposite to the first.
3. The third image features a large cardboard, potentially a covered window on one side.
4. The fourth image shows a plain wal,suggesting the view opposite the third image.
Based on these, the four views are likely structured in a roughly rectangular layout. Starting at the first view (front with vent), turning right would bring you to the wall (fourth image), away from the window (second image). Therefore, my answer is A. No.

Question: Based on these four different viewpoints (front, left, back, andright), if you are positioned at the third viewpoint, what is behind you?
Options: A.Washing machine **B.White wood rack** C.White + gray curtains D. White wall and window

<Reasoning>:
The third viewpoint shows the right side of the shoe, with the washing machine visible in the background.

Therefore, if you are positioned at the third viewpoint,the washing machine is behind you. Therefore, my answer is A. Washing machine.

Figure 9. Failure case analysis. We show GPT4-o's reasoning process. In case 1, the model is unable to establish the spatial location corresponding to each view; In case 2, the model confuses the subject of the "behind" relationship.

Inspired by these cognitive theories, we explore three distinct data structures designed to act as cognitive scaffolds for VLMs. When VLMs are presented with limited visual input, these structures aim to approximate different facets of human spatial mental modeling: dynamic updating, integrated spatial layout representation, and inferential reasoning.

C.1.1. View Interpolation for Dynamic Updating

Human spatial mental models are not static; they are continuously updated with new sensory information and through mental simulation, such as imagining a change in viewpoint. To approximate this dynamic updating and mental animation capability [16], we employ view interpolation. This technique aims to bridge perceptual gaps between discrete, sparsely sampled views by generating intermediate visual frames.

Our Setting: In our experiments, view interpolation is implemented by inserting synthetic frames *between* consecutive views provided to the model. For instance, if "1 interpolated frame" is specified, one new frame is generated and inserted between an initial view V_n and the subsequent view V_{n+1} (e.g., between View 1 and View 2). Similarly, "2 interpolated frames" would mean two synthetic frames are inserted in sequence between V_n and V_{n+1}. For the interpolated frames, we either define a heuristic function to choose from the original datasets [4, 55] where we sampled our data, or we use Stable Virtual Camera [72] to generate intermediate frames for those image groups without. This approach is intended to provide a smoother perceptual experience, potentially aiding the VLM in tracking object relations and maintaining spatial consistency across viewpoint shifts. (Refer to Figure 3 in the main paper for a conceptual illustration)

C.1.2. Cognitive Maps for Integrated Spatial Layouts

A core aspect of spatial cognition is the ability to form an allocentric (world-centered) or survey-like understanding of an environment, capturing the relative locations of objects. Tversky [44, 45] highlights that such representations often involve different frames of reference and hierarchical structures. Cognitive maps in our context are 2D schematic representations that attempt to embody this integrated spatial layout.

Our Setting: We investigate two variants of cognitive maps, both represented as structured data (e.g., JSON-like objects), to capture the spatial layout:

- We provide a 2D grid map of the scene that is related to the question to be answered.

- The map uses a 10×10 grid, where `[0, 0]` is the top-left corner and `[9, 9]` is the bottom-right corner (i.e., bird's-eye view).
- Directions are defined as follows:
  - `up` = towards the top of the grid (decreasing y-value)
  - `right` = towards the right of the grid (increasing x-value)
  - `down` = towards the bottom of the grid (increasing y-value)
  - `left` = towards the left of the grid (decreasing x-value)
  - `inner` = into the 2D map (perpendicular to the grid, pointing away from you)
  - `outer` = out of the 2D map (perpendicular to the grid, pointing toward you)
- The map contains:
  - `objects` — a list of all important items in the scene with their `position`
  - `facing` — indicating the direction an object is oriented (when applicable)
  - `views` — representing different camera viewpoints in the scene
- **Augmented Cognitive Map:** This version explicitly integrates the observer's perspective by encoding the `positions` and `orientations` (facing directions) of the camera viewpoints within the map, alongside the objects and their locations. For instance, as depicted in our data examples (refer to Figure 3, Cognitive Map - Augmented panel), an augmented map might define a list of `objects` with their `name` and `position` (e.g., `"Tissue box": { "position": [5, 5] }`), and a separate list of `views` detailing each camera's `name` (e.g., `"View 1"`), `position` (e.g., `[3, 5]`), and `facing` direction (e.g., `"up"`).
- **Plain Cognitive Map (Object Only):** This is a more simplified, object-centric representation. It primarily focuses on the `spatial locations of objects` and, for some objects, their intrinsic `orientation` (facing direction) from a top-down survey perspective, without explicitly embedding camera view information within its structure. For example (refer to Figure 3, Cognitive Map - Plain panel), a plain map might list objects like `"Potted plant"` with its `position` (e.g., `[5, 6]`) and `facing` direction (e.g., `"down"`), and another object like `"Sofa"` with only its `position` (e.g., `[4, 5]`). This type of map still allows for reasoning about object-to-object relationships and, where specified, object orientations, but abstracts away the explicit camera viewpoints that generated the scene understanding.

In both map types, coordinates represent positions on a 2D grid, and `facing` directions can be categorical (e.g., `"up"`, `"down"`, `"left"`, `"right"`, `"outer"`, `"inner"`). These structures aim to provide the VLM with an explicit, albeit potentially imperfect, schematic of the environment that it can then learn to generate and utilize for spatial reasoning tasks.

C.1.3. Free Form Reasoning

Spatial mental models are not just static representations; they are actively used for inference and problem-solving [46]. To approximate this procedural and inferential aspect, we utilize free-form reasoning, implemented as a natural language Chain-of-Thought (CoT) [53] process. This encourages the VLM to externalize its step-by-step reasoning process when deducing an answer to a spatial query.

Our Setting: The VLM is prompted to generate a textual reasoning chain before outputting the final answer. This process is guided by a three-step principle, exemplified by the reasoning chain shown in Figure 3, the reasoning chain panel. For the steps shown in that example, they are: (1) *Initial Observation and Grounding:* The model first processes each available view, identifying key objects and their immediate spatial relationships within that specific viewpoint. For instance, the example chain begins with: `"In View 1, I see a potted plant, tissue box, and hand sanitizer from left to right, with a sofa behind."` This step grounds the reasoning in direct visual evidence from individual perspectives. (2) *Cross-View Integration and Environment Consolidation:*Next, the model attempts to identify consistent objects or environmental cues across the different views to recognize that they depict the same underlying 3D scene. The example reasoning continues: `"In View 2, I see the same potted plant, so both views are from the same environment."` This step is crucial for building a unified understanding of the space from discrete observations. (3) *Question-Guided Spatial Inference:* Finally, based on the specific question posed and the integrated understanding from the previous steps, the model performs step-by-step logical and spatial inferences to arrive at the answer. In the example, this involves relating the object positions across views relative to the observer's position in View

2: "Since the hand sanitizer is rightmost in View 1, it's spatially furthest behind the potted plant when looking in View 2. In View 2, the potted plant is closest to me, so the hand sanitizer is the furthest from me."

C.2. Evaluation Metrics

To quantitatively assess how these data structures affect the performance of VLMs in the spatial mental modeling presented in MindCube, and to evaluate the quality of the generated cognitive maps, we employed the following metrics: (1) *QA Accuracy*, and (2) *Graph Metrics for Generated Cognitive Maps*.

C.2.1. QA Accuracy

QA Accuracy serves as the core metric for evaluating task performance. It quantifies the proportion of questions that the vision-language model (VLM) answers correctly out of the total number of questions. A higher QA Accuracy indicates better alignment between the model's responses and the ground truth.

The metric is formally defined as:

$$\text{QA Accuracy} = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\%$$

where N_{correct} denotes the number of correctly answered questions, and N_{total} is the total number of questions evaluated.

C.2.2. Graph Metrics for Cognitive Maps

To quantitatively evaluate the quality of a generated cognitive map, we use a set of structured graph-based metrics. The overall process consists of several key steps:

1. **Validity Check.** First, we ensure that the generated map is syntactically and semantically valid—i.e., it has a correct JSON format, contains interpretable object positions, and includes at least one valid object.
2. **Rotation Normalization.** Since we do not enforce a fixed orientation for generated maps (to allow for flexible generation from vision-language models), we evaluate the similarity between the generated map and the ground truth across a set of 3D rotations. We always choose the best-aligned rotation to compute our similarity scores.
3. **Structural Matching.** We define a relation graph between object pairs in each map, capturing directional and proximity-based relationships. A core part of the evaluation is determining whether these relationships in the ground truth are preserved in the generated map.
4. **Similarity Metrics.** We compute coverage (how many ground-truth objects are present), directional similarity (relative spatial relations), and facing similarity (object orientation). These are aggregated into an overall similarity score.
5. **Rotation-Invariant Isomorphism.** We also evaluate whether a generated map is graph-isomorphic to the ground truth under any allowed 3D rotation, providing a strict measure of structural correctness.

Below, we provide precise mathematical definitions for each of these components.

Notation. A *cognitive map* is a finite set of objects $\mathcal{O} = \{o_1, \dots, o_n\}$ where each object o_i is associated with (i) a 2-D position vector $p_i = (x_i, y_i) \in \mathbb{R}^2$ and (ii) an optional facing label $f_i \in \{\text{up}, \text{right}, \text{down}, \text{left}, \text{inner}, \text{outer}\} \cup \{\varnothing\}$. For two maps, we distinguish (1) the *ground-truth* map $(\mathcal{O}^\star, p^\star, f^\star)$ and (2) a *generated* map $(\mathcal{O}^g, p^g, f^g)$.

The set of objects that appear in both maps is $\mathcal{O}^c = \mathcal{O}^\star \cap \mathcal{O}^g$.

Extended directional relation. We define a directional or proximity-based relationship between any ordered object pair (o_i, o_j) based on their spatial arrangement and optional facing annotations. This relation is captured via the function:

$$\mathrm{dir}(o_i, o_j) = \begin{cases} \text{right} & |x_j - x_i| > |y_j - y_i| \text{ and } x_j > x_i, \\ \text{left} & |x_j - x_i| > |y_j - y_i| \text{ and } x_j < x_i, \\ \text{down} & |y_j - y_i| \geq |x_j - x_i| \text{ and } y_j > y_i, \\ \text{up} & |y_j - y_i| \geq |x_j - x_i| \text{ and } y_j < y_i, \\ \text{inner} & \|p_j - p_i\|_2 < \delta \text{ and } (f_i = \text{inner} \ \vee \ f_j = \text{outer}), \\ \text{outer} & \|p_j - p_i\|_2 < \delta \text{ and } (f_i = \text{outer} \ \vee \ f_j = \text{inner}), \\ \varnothing & \text{otherwise}, \end{cases}$$

with threshold $\delta = 0.5$ as in the implementation. These relations form a *relation matrix*:

$$R(o_i, o_j) = \mathrm{dir}(o_i, o_j).$$

Coverage. Coverage measures how many ground-truth objects are successfully retrieved in the generated map:

$$\mathrm{Cov} = \frac{|\mathcal{O}^c|}{|\mathcal{O}^\star|} \in [0, 1].$$

Directional similarity. We now evaluate how well the generated map preserves the directional relationships among object pairs from the ground truth. Define:

$$\mathcal{P}^\star = \{(o_i, o_j) \in \mathcal{O}^c \times \mathcal{O}^c \mid i \neq j,\ R^\star(o_i, o_j) \neq \varnothing\}.$$

Then the directional similarity score is given by:

$$S_{\text{dir}} = \frac{\left|\{(o_i, o_j) \in \mathcal{P}^\star \mid R^g(o_i, o_j) = R^\star(o_i, o_j)\}\right|}{|\mathcal{P}^\star|} \in [0, 1],$$

which corresponds to the proportion of directional relations in the ground truth that are correctly matched in the generated map.

Facing similarity. For objects with defined facing directions, we compare their orientation across the two maps:

$$\mathcal{F}^\star = \{o_i \in \mathcal{O}^c \mid f_i^\star \neq \varnothing\}.$$

Then:

$$S_{\text{face}} = \frac{\left|\{o_i \in \mathcal{F}^\star \mid f_i^g = f_i^\star\}\right|}{|\mathcal{F}^\star|} \in [0, 1].$$

Overall similarity. To aggregate the directional and facing similarities, we use a weighted combination:

$$S_{\text{overall}} = \alpha \cdot S_{\text{dir}} + (1 - \alpha) \cdot S_{\text{face}} \in [0, 1],$$

where $\alpha = 0.7$ places greater emphasis on spatial layout than orientation.

Rotation-invariant isomorphism. To ensure fair comparison regardless of orientation, we define a set of 3D rotations: $\mathcal{R} = \{R_1, \ldots, R_m\}$, including all 90° turns about the z-axis, and one 90° turn about each of the x- and y-axes.

We say the maps are *rotation-invariant isomorphic* if there exists a rotation such that their relation matrices match completely:

$$\exists\, k \in \{1, \ldots, m\}\ \forall\, o_i, o_j \in \mathcal{O}^\star: \quad R^\star(o_i, o_j) = R^g_{(k)}(o_i, o_j),$$

where $R^g_{(k)}$ is the relation matrix computed after applying R_k to the generated map.

Graph validity. Finally, a generated map is deemed *valid* if: (1) It is well-formed JSON, (2) All fields conform to expected formats and constraints, and (3) At least one object has a valid position.

Together, the tuple $\left(\mathrm{Cov},\ S_{\text{dir}},\ S_{\text{face}},\ S_{\text{overall}},\ \mathrm{Iso}_{\text{rot}}\right)$ provides a comprehensive, rotation-aware evaluation of how closely a generated cognitive map matches ground truth structure and orientation.

C.3. Prompts for All Input-Output Configurations

Below, we provide all prompts for the input-output configurations we investigate in our work.

C.3.1. Example for `Raw QA`

Example Prompt for `Raw QA`

[Answer Format]
Based on these images, answer the question based on this rule: You only need to provide *ONE* correct answer selected from the options listed below. For example, if you think the correct answer is ‘A. Above’ from ‘A. Above B. Under C. Front D. Behind’, your response should **only** be ‘A. Above’.
[Question]
Based on these four images (image 1, 2, 3, and 4) showing the white jar from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 4, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

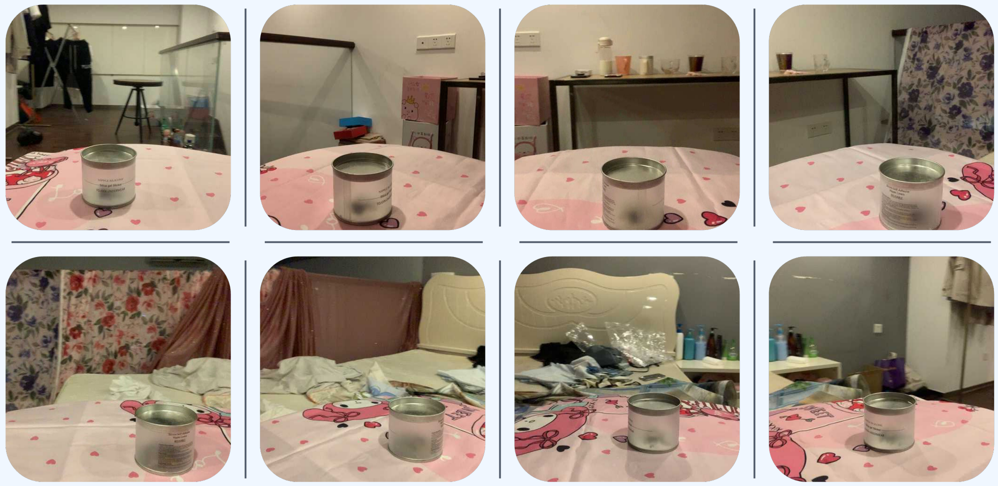
C.3.2. Example for `FF-Rsn`

Example Prompt for `FF-Rsn`: Free-Form Reasoning

[Answer Format]
Based on these images, answer the question based on this rule: You can do **step-by-step reasoning** first. You must provide *ONE* correct answer selecting from the options listed below *at the end of your response*. For example, if you think the correct answer is ‘A. Above’ from ‘A. Above B. Under C. Front D. Behind’, you must output ‘A. Above’ at the end of your response.
[Question]
Based on these four images (image 1, 2, 3, and 4) showing the white jar from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 4, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

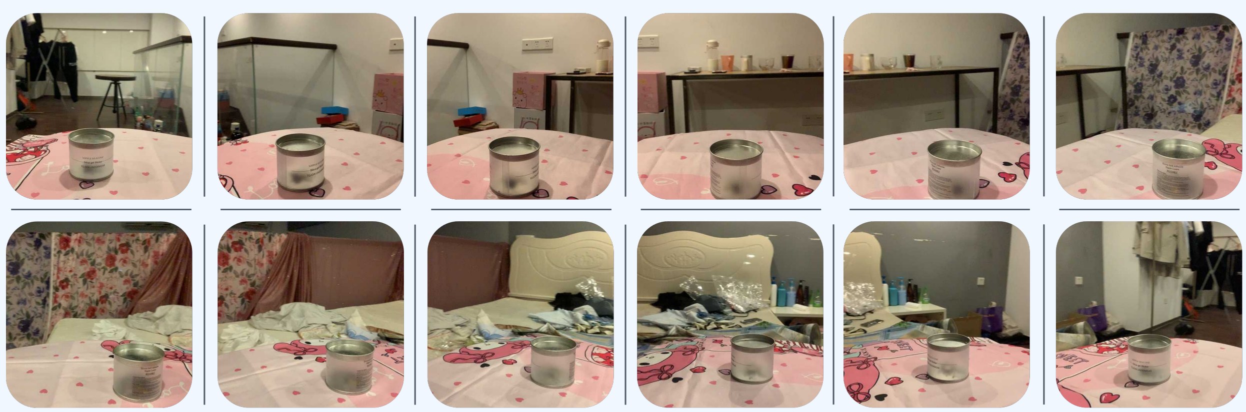
C.3.3. Example for `VI-1` and `VI-2`

Prompt for `VI-1`: View Interpolation with 1 Frame



[Answer Format]
Based on these images, answer the question based on this rule: You only need to provide *ONE* correct answer selected from the options listed below. For example, if you think the correct answer is 'A. Above' from 'A. Above B. Under C. Front D. Behind', your response should **only** be 'A. Above'.
[Question]
Based on these 8 images showing the white jar from different viewpoints (from front (image 1) to left (image 3), from left (image 3) to back (image 5), from back (image 5) to right (image 7), from right (image 7) back to front (image 1)), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 7, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

Prompt for `VI-2`: View Interpolation with 2 Frames



[Answer Format]
Based on these images, answer the question based on this rule: You only need to provide *ONE* correct answer selected from the options listed below. For example, if you think the correct answer is 'A. Above' from 'A. Above B. Under C. Front D. Behind', your response should **only** be 'A. Above'.
[Question]
Based on these 12 images showing the white jar from different viewpoints (from front (image 1) to left (image 4), from left (image 4) to back (image 7), from back (image 7) to right (image 10), from right (image 10) back to front

(image 1)), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 10, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

C.3.4. Example for `Aug-CGMap-In`

Prompt for `Aug-CGMap-In`: Grounded Augmented Cognitive Map as Input

[Answer Format]
Based on these images, answer the question based on this rule: You only need to provide *ONE* correct answer selected from the options listed below. For example, if you think the correct answer is 'A. Above' from 'A. Above B. Under C. Front D. Behind', your response should **only** be 'A. Above'.
[Cognitive Map Format]
We provide you a 2D grid map of the scene that is related to the question you should answer. Below is the description of the map:
- The map uses a 10x10 grid where [0,0] is at the top-left corner and [9,9] is at the bottom-right corner
- The map is shown in the bird's view
- Directions are defined as:
* up = towards the top of the grid (decreasing y-value)
* right = towards the right of the grid (increasing x-value)
* down = towards the bottom of the grid (increasing y-value)
* left = towards the left of the grid (decreasing x-value)
* inner = straight into the 2D map (perpendicular to the grid, pointing away from you)
* outer = straight out of the 2D map (perpendicular to the grid, pointing towards you)
- "objects" lists all important items in the scene with their positions
- "facing" indicates which direction an object is oriented towards (when applicable)
- "views" represents the different camera viewpoints in the scene
- "facing_objects" indicates the camera is facing which objects


```
{
  "objects": [
    { "name": "white jar", "position": [5, 5] },
    { "name": "bed sheet with a floral pattern", "position": [5, 8] },
    { "name": "white headboard", "position": [2, 5] },
    { "name": "clothes rack", "position": [5, 2] },
    { "name": "table with cups on it", "position": [8, 5] }
  ],
  "views": [
    { "name": "Image 1", "position": [5, 6], "facing": "up" },
    { "name": "Image 2", "position": [4, 5], "facing": "right" },
    { "name": "Image 3", "position": [5, 4], "facing": "down" },
    { "name": "Image 4", "position": [6, 5], "facing": "left" }
  ]
}
```

[Question]
Based on these four images (image 1, 2, 3, and 4) showing the white jar from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 4, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

C.3.5. Example for `Aug-CGMap-Out`

Prompt for `Aug-CGMap-Out`: Ask VLM to Output Augmented Cognitive Map and Direct Answer

[Task]
Your task is to analyze the spatial arrangement of objects in the scene by examining the provided images, which show the scene from different viewpoints. You will then create a detailed cognitive map representing the scene using a 10x10 grid coordinate system.
[Rules]
1. Focus ONLY on these categories of objects in the scene: {white jar, bed sheet with a floral pattern, white headboard, clothes rack, table with cups on it}
2. Create a cognitive map with the following structure in the bird's view:
- A 10x10 grid where [0,0] is at the top-left corner and [9,9] is at the bottom-right corner
- up = towards the top of the grid (decreasing y)
- right = towards the right of the grid (increasing x)
- down = towards the bottom of the grid (increasing y)
- left = towards the left of the grid (decreasing x)
- inner = straight into the 2D map (perpendicular to the grid, pointing away from you)
- outer = straight out of the 2D map (perpendicular to the grid, pointing towards you)
- Include positions of all objects from the specified categories
- Estimate the center location (coordinates [x, y]) of each instance within provided categories
- If a category contains multiple instances, include all of them
- Each object's estimated location should accurately reflect its real position in the scene, preserving the relative spatial relationships among all objects
- Combine and merge information from the images since they are pointing to the same scene, calibrating the object locations accordingly
- Include camera positions and directions for each view
3. Carefully integrate information from all views to create a single coherent spatial representation.
[Answer Format]
1. Given the provided views and main objects mentioned in the above rules, you **MUST** present your cognitive map in the following JSON format **before your reasoning**:

```
{
  "objects": [
    {"name": "object_name", "position": [x, y], "facing": "direction"},
    {"name": "object_without_orientation", "position": [x, y]}
  ],
  "views": [
    {"name": "View/Image 1", "position": [x, y], "facing": "direction"},
```

```
    {"name": "View/Image 2", "position": [x, y], "facing": "direction"}
  ]
}
```

2. Next, based on your generated cognitive map, please generate the answer to the question. For example, if you think the correct answer is 'A. Above' from 'A. Above B. Under C. Front D. Behind', you must output 'A. Above'. Your answer format should be like <CogMap>\n<Your cognitive map>\n<Answer>\n<Your answer>
[Question]
Based on these four images (image 1, 2, 3, and 4) showing the white jar from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 4, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

C.3.6. Example for `Plain-CGMap-Out`

Prompt for `Plain-CGMap-Out`: Ask VLM to Output Plain Cognitive Map and Direct Answer

[Task]
Your task is to analyze the spatial arrangement of objects in the scene by examining the provided images, which show the scene from different viewpoints. You will then create a detailed cognitive map representing the scene using a 10x10 grid coordinate system.
[Rules]
1. Focus ONLY on these categories of objects in the scene: {white jar, bed sheet with a floral pattern, white headboard, clothes rack, table with cups on it}
2. Create a cognitive map with the following structure in the bird's view:
- A 10x10 grid where [0, 0] is at the top-left corner and [9, 9] is at the bottom-right corner
- up = towards the top of the grid (decreasing y)
- right = towards the right of the grid (increasing x)
- down = towards the bottom of the grid (increasing y)
- left = towards the left of the grid (decreasing x)
- Include positions of all objects from the specified categories
- Estimate the center location (coordinates [x, y]) of each instance within provided categories
- If a category contains multiple instances, include all of them
- Object positions must maintain accurate relative spatial relationships
- Combine and merge information from the images since they are pointing to the same scene, calibrating the object locations with grid coordinates accordingly
3. Carefully integrate information from all views to create a single coherent spatial representation.
[Answer Format]
1. Given the provided views and main objects mentioned in the above rules, you **MUST** present your cognitive map in the following JSON format **before your reasoning**:

```
{
    "object_category_1": {"position": [x, y]},
    "object_category_2": {"position": [x, y], "facing": "direction"},
    # if the object is asked for orientation
```

```
    ...
}
```

2. Next, based on your generated cognitive map, please generate the answer to the question. For example, if you think the correct answer is ‘A. Above’ from ‘A. Above B. Under C. Front D. Behind’, you must output ‘A. Above’. Your answer format should be like `<CogMap>\n<Your cognitive map>\n<Answer>\n<Your answer>`
[Question]
Based on these four images (image 1, 2, 3, and 4) showing the white jar from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 4, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

C.3.7. Example for `Plain-CGMap-FFR-Out`

Prompt for `Plain-CGMap-FFR-Out`: Ask VLM to Output Plain Cognitive Map and Free-Form Reasoning

[Task]
Your task is to analyze the spatial arrangement of objects in the scene by examining the provided images, which show the scene from different viewpoints. You will then create a detailed cognitive map representing the scene using a 10x10 grid coordinate system.
[Rules]
1. Focus ONLY on these categories of objects in the scene: {white jar, bed sheet with a floral pattern, white headboard, clothes rack, table with cups on it}
2. Create a cognitive map with the following structure in the bird’s view:
- A 10x10 grid where [0, 0] is at the top-left corner and [9, 9] is at the bottom-right corner
- up = towards the top of the grid (decreasing y)
- right = towards the right of the grid (increasing x)
- down = towards the bottom of the grid (increasing y)
- left = towards the left of the grid (decreasing x)
- Include positions of all objects from the specified categories
- Estimate the center location (coordinates [x, y]) of each instance within provided categories
- If a category contains multiple instances, include all of them
- Object positions must maintain accurate relative spatial relationships
- Combine and merge information from the images since they are pointing to the same scene, calibrating the object locations with grid coordinates accordingly
3. Carefully integrate information from all views to create a single coherent spatial representation.
[Answer Format]
1. Given the provided views and main objects mentioned in the above rules, you **MUST** present your cognitive map in the following JSON format **before your reasoning**:

```
{
    "object_category_1": {"position": [x, y]},
    "object_category_2": {"position": [x, y], "facing": "direction"},
    # if the object is asked for orientation
    ...
}
```

2. Next, please also provide your reasons step by step in detail, then provide *ONE* correct answer selecting from the options. Your response's format should be like <CogMap>\n<Your cognitive map>\n<Reasoning>\n...\n<Answer> Therefore, my answer is <selected option>. Your ¡selected option¿ must be in the format like "A. Above". Your option must be from the available options.
[Question]
Based on these four images (image 1, 2, 3, and 4) showing the white jar from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 4, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

C.3.8. Example for `Aug-CGMap-FFR-Out`

Prompt for `Aut-CGMap-FFR-Out`: Ask VLM to Output Augmented Cognitive Map and Free-Form Reasoning

[Task]
Your task is to analyze the spatial arrangement of objects in the scene by examining the provided images, which show the scene from different viewpoints. You will then create a detailed cognitive map representing the scene using a 10x10 grid coordinate system.
[Rules]
1. Focus ONLY on these categories of objects in the scene: {white jar, bed sheet with a floral pattern, white headboard, clothes rack, table with cups on it}
2. Create a cognitive map with the following structure in the bird's view:
- A 10x10 grid where [0,0] is at the top-left corner and [9,9] is at the bottom-right corner
- up = towards the top of the grid (decreasing y)
- right = towards the right of the grid (increasing x)
- down = towards the bottom of the grid (increasing y)
- left = towards the left of the grid (decreasing x)
- inner = straight into the 2D map (perpendicular to the grid, pointing away from you)
- outer = straight out of the 2D map (perpendicular to the grid, pointing towards you)
- Include positions of all objects from the specified categories
- Estimate the center location (coordinates [x, y]) of each instance within provided categories
- If a category contains multiple instances, include all of them
- Each object's estimated location should accurately reflect its real position in the scene, preserving the relative spatial relationships among all objects
- Combine and merge information from the images since they are pointing to the same scene, calibrating the object locations accordingly
- Include camera positions and directions for each view
3. Carefully integrate information from all views to create a single coherent spatial representation.
[Answer Format]
1. Given the provided views and main objects mentioned in the above rules, you **MUST** present your cognitive map in the following JSON format **before your reasoning**:

```
{
  "objects": [
    {"name": "object_name", "position": [x, y], "facing": "direction"},
```

```
    {"name": "object_without_orientation", "position": [x, y]}
  ],
  "views": [
    {"name": "View/Image 1", "position": [x, y], "facing": "direction"},
    {"name": "View/Image 2", "position": [x, y], "facing": "direction"}
  ]
}
```


2. Next, please also provide your reasons step by step in detail, then provide *ONE* correct answer selecting from the options. Your response's format should be like <CogMap>\n<Your cognitive map>\n<Reasoning>\n...\n<Answer> Therefore, my answer is <selected option>. Your ¡selected option¿ must be in the format like "A. Above". Your option must be from the available options. [Question] Based on these four images (image 1, 2, 3, and 4) showing the white jar from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 4, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

C.3.9. Example for `CGMap-In-FFR-Out`

Prompt for `CGMap-In-FFR-Out`: Input VLM with Grounded Cognitive Map and Output with Free-Form Reasoning

[Answer Format]
Based on these images, answer the question based on this rule: You can do step-by-step reasoning first. You must provide *ONE* correct answer selecting from the options listed below *at the end of your response*. For example, if you think the correct answer is 'A. Above' from 'A. Above B. Under C. Front D. Behind', you must output 'A. Above' at the end of your response.
[Cognitive Map Format]
We provide you a 2D grid map of the scene that is related to the question you should answer. Below is the description of the map:
- The map uses a 10x10 grid where [0,0] is at the top-left corner and [9,9] is at the bottom-right corner
- The map is shown in the bird's view
- Directions are defined as:
* up = towards the top of the grid (decreasing y-value)
* right = towards the right of the grid (increasing x-value)
* down = towards the bottom of the grid (increasing y-value)
* left = towards the left of the grid (decreasing x-value)
* inner = straight into the 2D map (perpendicular to the grid, pointing away from you)
* outer = straight out of the 2D map (perpendicular to the grid, pointing towards you)
- "objects" lists all important items in the scene with their positions
- "facing" indicates which direction an object is oriented towards (when applicable)
- "views" represents the different camera viewpoints in the scene
- "facing_objects" indicates the camera is facing which objects


```
{
  "objects": [
```

```
    { "name": "white jar", "position": [5, 5] },
    { "name": "bed sheet with a floral pattern", "position": [5, 8] },
    { "name": "white headboard", "position": [2, 5] },
    { "name": "clothes rack", "position": [5, 2] },
    { "name": "table with cups on it", "position": [8, 5] }
  ],
  "views": [
    { "name": "Image 1", "position": [5, 6], "facing": "up" },
    { "name": "Image 2", "position": [4, 5], "facing": "right" },
    { "name": "Image 3", "position": [5, 4], "facing": "down" },
    { "name": "Image 4", "position": [6, 5], "facing": "left" }
  ]
}
```

[Question]
Based on these four images (image 1, 2, 3, and 4) showing the white jar from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 4, what is to the left of the white jar?
A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

D. Which Scaffolds Best Guide Spatial Thinking in Unchanged VLMs?

The primary experimental setup for evaluating how different structural scaffolds can guide spatial reasoning in unchanged VLMs, has been comprehensively described in Section 3 of the main paper. This current section is dedicated to presenting qualitative examples of model outputs, serving as case studies for the different scaffolding configurations tested.

For precise details on the input prompt structures and specific instructions provided to the Qwen2.5-VL-3B-Instruct for each experimental condition (e.g., `Raw QA`, `FF-Rsn`, `Plain-CGMap-FFR-Out`), please refer to Appendix C.3.

D.1. VLM Response Examples for Configurations in Section C.3

Output for `Raw QA`: Input Prompt Refer to C.3.1

C. Bed sheet with a floral pattern

Output for `VI`: Input Prompt Refer to C.3.3

B. Clothes rack

Output for `Aug-CGMap-In`: Input Prompt Refer to C.3.4

C. Bed sheet with a floral pattern

Output for `FF-Rsn`: Input Prompt Refer to C.3.2

To determine what is to the left of the white jar from the viewpoint presented in image 4, let's analyze the scene:
- In image 4, the white jar is placed on a table covered with a pink tablecloth featuring heart patterns.
- The table has a glass top and is positioned against a wall.
- To the left of the jar, there is a piece of furniture that appears to be a bed or a bed frame.

Given this information, the object to the left of the white jar is most likely a bed or a bed frame.
Therefore, the correct answer is:
C. Bed sheet with a floral pattern

Output for `Aug-CGMap-FFR-Out`: Input Prompt Refer to C.3.8


```json
{
  "objects": [
    {"name": "white jar", "position": [150, 446], "facing": "left"},
    {"name": "bed sheet with a floral pattern", "position": [0, 317],
    "facing": "left"},
    {"name": "clothes rack", "position": [0, 108],
    "facing": "left"},
    {"name": "table with cups on it", "position": [188, 200],
    "facing": "left"},
    {"name": "white headboard", "position": [188, 317],
    "facing": "left"}
  ],
  "views": [
    {"name": "Image 1", "position": [0, 0], "facing": "left"},
    {"name": "Image 2", "position": [0, 0], "facing": "left"},
    {"name": "Image 3", "position": [0, 0], "facing": "left"},
    {"name": "Image 4", "position": [0, 0], "facing": "left"}
  ]
}
```


¡CogMap¿
¡Your cognitive map¿
¡Reasoning¿
From the viewpoint presented in image 4, we can see that the white jar is positioned in the foreground, facing left. The bed sheet with a floral pattern is located directly behind the white jar, covering the lower part of the image. The clothes rack is visible on the left side of the image, slightly above the bed sheet. The table with cups on it is situated further back, to the right of the white jar. The white headboard is also visible, but it is not as prominent as the other objects due to its position behind the bed sheet.
Therefore, my answer is A. Table with cups on it.
¡/Answer¿
Therefore, my answer is **A. Table with cups on it**.

Output for `Plain-CGMap-FFR-Out`: Input Prompt Refer to C.3.7


```json
{
    "white_jar": {"position": [150, 440], "facing": "left"},
    "table_with_cups_on_it": {"position": [187, 260],
    "facing": "right"}
}
```

¡CogMap¿
¡Your cognitive map¿
¡Reasoning¿
From the viewpoint presented in image 4, we can see that the white jar is positioned on the bed sheet with a floral pattern. The table with cups on it is located to the right of the white jar, as indicated by its position in the image. The clothes rack is not visible in this particular view, and the white headboard is not directly related to the white jar or the table with cups on it. Therefore, the correct answer is A. Table with cups on it.
¡Answer¿
Therefore, my answer is **A. Table with cups on it**.

Output for `CGMap-In-FFR-Out`: Input Prompt Refer to C.3.9

C. Bed sheet with a floral pattern

D.2. Additional Graph Metrics for Generated Graphs

To quantitatively assess the quality of the cognitive maps produced by our models, we employ a suite of graph-based metrics detailed in Section C.2.2. Table 4 presents a comparison of the cognitive maps generated by two of our methods: `Aug-CGMap-FFR-Out` and `Plain-CGMap-FFR-Out`.

The results highlight several key findings:

- **High Validity**: Both methods demonstrate a strong ability to generate syntactically and semantically valid maps, achieving high **Valid Rates** (**94.95%** for `Plain-CGMap-FFR-Out` and **92.57%** for `Aug-CGMap-FFR-Out`). This indicates that the models successfully learned the required JSON structure for the cognitive maps.
- **Structural Complexity**: Achieving perfect structural replication of the ground truth remains challenging, as shown by the modest **Isomorphism Rates**. The `Plain-CGMap-FFR-Out` method performs significantly better, with **7.43%** of its maps being structurally identical (isomorphic) to the ground truth, compared to a mere **0.10%** for the augmented map method.
- **Superior Similarity Performance**: A clear performance difference in semantic similarity is evident. The `Aug-CGMap-FFR-Out` method, which explicitly includes camera views, achieves a substantially higher **Overall Similarity** (**51.12%**) and is superior in representing both the relative directional relationships (**Avg. Dir. Sim.** of **43.57%**) and the correct orientation of individual objects (**Avg. Facing Sim.** of **68.75%**). In contrast, while `Plain-CGMap-FFR-Out` maintains higher validity and isomorphism, it lags behind in all three similarity metrics.

Table 4. Comparison of graph metrics for cognitive maps generated by different methods. The metrics evaluate the quality of the generated maps against the ground truth. **Valid Rate**: percentage of syntactically and semantically valid maps. **Isomorphism Rate**: percentage of maps that are structurally identical (isomorphic) to the ground truth, accounting for rotation. **Overall Sim. (Similarity)**: a weighted score combining directional and facing similarity ($S_{\text{overall}} = \alpha \cdot S_{\text{dir}} + (1-\alpha) \cdot S_{\text{face}}$). **Avg. Dir. Sim. (Average Directional Similarity)**: correctness of relative spatial relations between objects. **Avg. Facing Sim. (Average Facing Similarity)**: correctness of object orientations. All values are percentages (%).

Method	Valid Rate	Isomorphism Rate	Overall Sim.	Avg. Dir. Sim.	Avg. Facing Sim.
`Aug-CGMap-FFR-Out`	92.57	0.10	51.12	43.57	68.75
`Plain-CGMap-FFR-Out`	94.95	7.43	37.44	28.29	58.78

E. Can We Teach VLMs to Build and Leverage Spatial Representations?

In the main paper, we demonstrated that prompting frozen VLMs with external scaffolds offers limited improvements. This highlighted a core limitation: the models themselves aren't effectively forming internal spatial representations or reasoning through space. To address this, we investigated whether supervised fine-tuning (SFT) could teach VLMs to build and leverage these spatial models internally. This section of the appendix provides further details on our SFT methodology, starting with the crucial step of data curation.

E.1. Supervised Fine-Tuning Data Curation

Effective SFT heavily relies on the quality and nature of the training data. To teach our VLMs the desired spatial reasoning capabilities, we meticulously curated two primary types of data: cognitive maps and free-form reasoning chains. These were designed to provide strong supervisory signals for the model to learn how to represent and reason about space.

E.1.1. Cognitive Map Generation

As discussed in Section C.1, cognitive maps serve as 2D schematic representations of object layouts. For the SFT phase, we needed to generate ground truth cognitive maps that the VLM could learn to produce. Our approach to generating these maps was grounded in the object arrangement annotations described in Section A.1. We aimed for representations that were not only accurate but also in a format that the VLM could feasibly learn to generate.

The generation process was automated via a script that processes input JSONL files, where each line item contains scene details including images and, crucially, `meta_info` describing the objects, their potential orientations, and the camera viewpoint setup. For every item, the script first identifies its specific spatial arrangement ”setting” (e.g., ”around,” ”among,” ”translation,” or ”rotation”) by parsing the item’s unique ID. Based on this setting, dedicated functions apply a set of predefined rules and heuristics to determine the 2D coordinates (on a 10x10 grid) and facing directions for both the objects and the camera views.

For instance, in the ”around” setting, objects (typically 2-4) are placed in a predetermined linear arrangement near the grid’s center (e.g., at coordinates like [4,5], [5,5]), and camera views are positioned at cardinal directions relative to these objects, based on the specific camera angles pertinent to the question. In the ”rotation” setting, the camera is fixed at the center ([5,5]), and its facing direction changes across views, while object positions are defined relative to the camera’s current orientation. Similar rule-based placements are implemented for ”among” (objects in a cross or T-shape with views from specific angles) and ”translation” (objects arranged linearly to depict relationships like ”on” or ”down to”) settings. Object orientations, if applicable, are also assigned based on the input `meta_info`.

Finally, the generated layout of objects and views is formatted into a structured JSON string, representing the cognitive map. This JSON cogmap, along with templated instructional prompts (`cogmap_input` for VLM input format guidance and `cogmap_output` for VLM output task description), is added to the original data item. The overall generation logic is summarized in Algorithm 1.

E.1.2. Free-Form Reasoning Generation

While cognitive maps provide a structured, global understanding of the scene, effective spatial reasoning also involves a procedural, step-by-step thought process. To instill this capability in our VLMs, we generated a dataset of grounded free-form reasoning chains. These chains were designed to verbalize the mental simulation process required to answer the spatial questions in MINDCUBE.

The generation of these reasoning chains was closely tied to the question-answer (QA) templates developed in Section 2. For each specific setting (e.g., rotation, among, around), we manually constructed reasoning chains following a consistent set of principles to ensure logical coherence and clear grounding in the provided visual information and the question asked.

The core principles guiding the generation of these reasoning chains were:

1. **Initial Scene Understanding.** The reasoning begins by processing each input image individually. This involves identifying key objects visible in that view and noting their explicit spatial relationships with other objects within that same view. This step emulates the initial perceptual intake a human might perform.
2. **Cross-View Consistency and Environment Integration.** After individual view analysis, the reasoning emphasizes that although different images are provided, they all depict the *same underlying spatial environment*. This is often achieved by identifying and highlighting an anchor object or a consistent set of objects that appear across multiple views, thereby helping to establish a unified mental model of the scene.
3. **Question-Driven Inference.** With a foundational understanding of the scene established from the views, the subsequent steps in the reasoning chain are directly guided by the specifics of the question. This involves: (1) **Mental Simulation:** If the question involves a hypothetical change in viewpoint or a ”what-if” scenario (e.g., ”what if you turn left?”), the reasoning chain explicitly verbalizes this mental transformation. (2) **Perspective Taking:** If the question requires adopting a different perspective (e.g., ”from the sofa’s perspective”), the reasoning chain articulates this shift. (3) **Spatial Relationship**

Algorithm 1 Cognitive Map Generation

Require: Dataset D containing items with spatial arrangement annotations
Ensure: Updated dataset with cognitive maps in JSON format

```
for all item ∈ D do
    setting ← Extract setting type from item.id
    Initialize empty cognitive map cogmap
                                   ▷ Position objects and views based on setting type
    if setting = "around" then
        Position 2-4 objects in a line with coordinates like [4,5], [5,5], etc.
        Place views at cardinal positions based on camera angles
    else if setting = "among" then
        Place center object at [5,5] and surrounding objects at [5,8], [2,5], [5,2], [8,5]
        Position views based on specified camera angles
    else if setting = "translation" then
        Position objects according to their spatial relationships (e.g., "on", "down")
        Place views to highlight these spatial relationships
    else if setting = "rotation" then
        Arrange objects based on rotation type (clockwise, counterclockwise, etc.)
        Fix camera at [5,5] with varying facing directions
    end if
                                   ▷ Add orientation information where applicable
    for all object ∈ cogmap.objects do
        if object has orientation then
            Add facing direction ("up", "down", "left", "right")
        end if
    end for
    Format cogmap as structured JSON
    Add formatted cognitive map to item
end for
return Updated dataset D
```

Deduction: The chain logically deduces the queried spatial relationship by integrating information from the relevant views, applying spatial concepts (like left-of, behind, further from), and referencing the established mental model of the scene.

This structured approach to generating reasoning chains aimed to provide clear, step-by-step examples of spatial thought processes for the VLM to learn from. Figure 10, 11 and 12 show a template example combined with the filled case for ROTATION, AMONG, AROUND, respectively.

E.2. Detailed Experimental Setup

In this section, we provide a more granular view of the experimental parameters employed during the Supervised Fine-Tuning (SFT) phase of our research. As stated in the main text, these experiments were designed to teach Vision-Language Models (VLMs) to build and leverage internal spatial representations. The base model for these SFT experiments was Qwen2.5-VL-3B-Instruct.

We utilized a consistent training script for all SFT experiments, ensuring comparability across different configurations. The primary variation across these runs was the specific dataset used (datasets variable in the script), corresponding to the different SFT task configurations discussed in Section 4.1, such as `Aug-CGMap-Out`. Other hyperparameters were kept constant to isolate the effects

Table 5. Training hyperparameters for SFT experiments with Qwen2.5-VL-3B-Instruct.

Parameter	Value
Dataset size	10,000 QA pairs
Epochs	3
Learning rate	1e-5
Scheduler	Cosine
Fine-tuning type	Full-parameter
Batch Size	256
GPUs used	2 × NVIDIA H100
Max image resolution	90,000 pixels
Min image resolution	784 pixels
Model Max Length	8192 tokens
Weight Decay	0
Warmup Ratio	0.03
Max Grad Norm	1
Precision	BF16
Optimizer	AdamW

of the different training signals.

The core training hyperparameters are summarized in Table 5 and further detailed by the provided training script.

The training was conducted using a distributed setup managed by `torchrun` and leveraged DeepSpeed with a ZeRO Stage 3 optimization strategy for efficient full-parameter fine-tuning. Specifically, we set `NPROC_PER_NODE` to 2, utilizing two NVIDIA H100 GPUs, though the script template showed `CUDA_VISIBLE_DEVICES=0,1,2,3` and `NPROC_PER_NODE` defaulting to 4, our table and resource claims point to 2 GPUs being used for these runs. The `per_device_train_batch_size` was set to 4, and with `gradient_accumulation_steps` at 32, this resulted in an effective batch size of 256.

The learning rate was 1×10^{-5} with a cosine learning rate scheduler and a warmup ratio of 0.03 over 3 training epochs. We enabled full fine-tuning of the vision encoder, MM-MLP, and the LLM components (`--tune_mm_vision True --tune_mm_mlp True --tune_mm_llm True`). The maximum sequence length for the model was set to 8192 tokens (`--model_max_length 8192`), and the maximum image input pixels were capped at 90,000 (`--max_pixels 90000`), with a minimum of 784. Gradient checkpointing was enabled to manage memory usage (`--gradient_checkpointing True`). Checkpoints were saved every 5 steps (`--save_steps 5`), keeping a total of 12 checkpoints (`--save_total_limit 12`).

This consistent and robust experimental framework, detailed in Sections 3.1, 3.2, and herein, allowed for a fair comparison of the different SFT strategies aimed at improving the spatial reasoning capabilities of Qwen2.5-VL-3B-Instruct.

E.3. VLM Response Examples After SFT for Configurations in Section C.3

Output for `Raw QA`: Input Prompt Refer to C.3.1

Answer: **D. White headboard**

Output for `Aug-CGMap-Out`: Input Prompt Refer to C.3.5

¡CogMap¿

```
```json
{
  "objects": [
    {"name": "white jar", "position": [5, 5]},
    {"name": "bed sheet with a floral pattern", "position": [5, 8]},
    {"name": "white headboard", "position": [2, 5]},
    {"name": "clothes rack", "position": [5, 2]},
    {"name": "table with cups on it", "position": [8, 5]}
  ],
  "views": [
    {"name": "Image 1", "position": [5, 6], "facing": "up"},
    {"name": "Image 2", "position": [4, 5], "facing": "right"},
    {"name": "Image 3", "position": [5, 4], "facing": "down"},
    {"name": "Image 4", "position": [6, 5], "facing": "left"}
  ]
}
'''
```

After generating the cognitive map, I will provide my answer to the question:
¡Answer¿
C. Bed sheet with a floral pattern
```

Rotation – Template Type 2

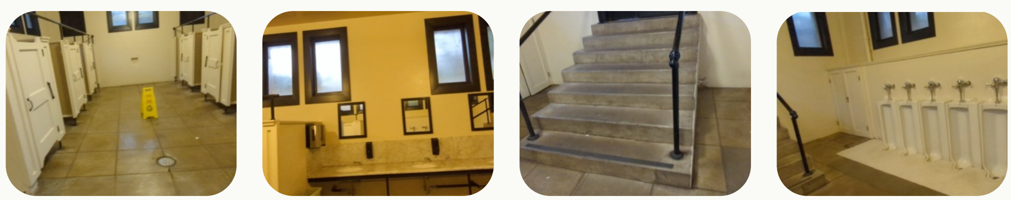
Template Reasoning Chain

This scene is observed using four images. In image 1, I can see {object_in_front_view1} as the main object in front of me. In image 2, I can see {object_in_front_view2} as the main object in front of me. In image 3, I can see {object_in_front_view3} as the main object in front of me. In image 4, I can see {object_in_front_view4} as the main object in front of me. Image 1 is the initial view. Image 2 is captured after a 90-degree clockwise rotation from image 1. Image 3 is after another 90-degree clockwise rotation (180 degrees from image 1). Image 4 is after a further 90-degree clockwise rotation (270 degrees from image 1). From the perspective of image 4: '{object_in_front_view4}' is in front, '{object_in_front_view1}' is to the right, '{object_in_front_view2}' is behind, '{object_in_front_view3}' is to the left. After turning 90 degrees to the right: '{object_in_front_view1}' is now in front, '{object_in_front_view2}' is now to my right, '{object_in_front_view3}' is now behind, '{object_in_front_view4}' is now to my left. The object located to my behind is '{object_in_front_view3}'. Therefore, from the viewpoint of image 4, after a mental turn of 90 degrees to the right, the object to my behind is '{object_in_front_view3}'. The answer is {option}

Question: If you are standing at the viewpoint presented in image 4 and turn 90 degrees to the right, what is to your behind ?

Options:

A. Staircases
B. Vanity unit
C. Urinal
D. Yellow signboard



: rotation agent-agent self perspective non-linear

Curated Reasoning Chain

This scene is observed using four images. In image 1, I can see yellow signboard as the main object in front of me. In image 2, I can see vanity unit as the main object in front of me. In image 3, I can see staircases as the main object in front of me. In image 4, I can see urinal as the main object in front of me. Image 1 is the initial view. Image 2 is captured after a 90-degree clockwise rotation from image 1. Image 3 is after another 90-degree clockwise rotation (180 degrees from image 1). Image 4 is after a further 90-degree clockwise rotation (270 degrees from image 1). From the perspective of image 4: 'urinal' is in front, 'yellow signboard' is to the right, 'vanity unit' is behind, 'staircases' is to the left. After turning 90 degrees to the right: 'yellow signboard' is now in front, 'vanity unit' is now to my right, 'staircases' is now behind, 'urinal' is now to my left. The object located to my behind is 'staircases'. Therefore, from the viewpoint of image 4, after a mental turn of 90 degrees to the right, the object to my behind is 'staircases'. The answer is A. Staircases

Figure 10. Example reasoning chain template for ROTATION

Among – Template Type 3

Template Reasoning Chain

In this scene, I observe four images showing different perspectives. All images feature the {main_object} as the main object. In image 1, I can see {main_object} in front of the {context_obj_V1}. In image 2, I can see {main_object} in front of the {context_obj_V2}. In image 3, I can see {main_object} in front of the {context_obj_V3}. In image 4, I can see {main_object} in front of the {context_obj_V4}. By observing the main object and its surroundings across views, and noting the rotational changes, I establish their relationships. Image 1 is the initial view. Image 2 is captured after a 90-degree clockwise rotation from image 1. Image 3 is after another 90-degree clockwise rotation (180 degrees from image 1). Image 4 is after a further 90-degree clockwise rotation (270 degrees from image 1). Through analyzing these perspective changes, I construct a complete spatial understanding: When I view {context_obj_V2} behind {main_object} in the second view, it implies that in the first view, {context_obj_V2} is on the right side of {main_object}. Similarly, when I see {context_obj_V4} behind {main_object} in the fourth view, it indicates that in the first view, {context_obj_V4} is on the left side of {main_object}. To determine what lies behind me in the first view, I examine the opposite view, which is the third view. As {context_obj_V3} is observed behind {main_object} in the third view, it means that in the first view, {context_obj_V3} is positioned behind me. This way, I can fully comprehend the spatial relationships of all objects in the entire scene from the perspective of image 1. So, from the perspective of image 1: {context_obj_V2} is to the right of {main_object}, {context_obj_V3} is to my behind, and {context_obj_V4} is to the left of {main_object}. The answer is {option}.

Question: From the viewpoint presented in image 1, what is to the right of the black stool ?

Options:

A. Desk

B. Office Area

C. Grey sofa

D. Two chairs on the corridor

: meanwhile | object-object | self perspective | non-linear

Curated Reasoning Chain

In this scene, I observe four images showing different perspectives. All images feature the black stool as the main object. In image 1, I can see black stool in front of the cabinet desk along a corridor. In image 2, I can see black stool in front of the office area. In image 3, I can see black stool in front of the two chairs on the corridor. In image 4, I can see black stool in front of the grey sofa. To identify the position change across views, I focus on the main object's angle variation. Then, I analyze the angles and relative positions of other objects on the platform to back up this observation. I understand that: Image 1 is the initial view. Image 2 is captured after a 90-degree clockwise rotation from image 1. Image 3 is after another 90-degree clockwise rotation (180 degrees from image 1). Image 4 is after a further 90-degree clockwise rotation (270 degrees from image 1). Through analyzing these perspective changes, I can construct a complete spatial understanding: when I view office area behind black stool in the second view, it implies that in the first view, office area is on the right side of black stool. Similarly, when I see grey sofa behind black stool in the fourth view, it indicates that in the first view, grey sofa is on the left side of black stool. However, I am still uncertain about what lies behind me in the first view. Then, I recognize that I can examine the opposite view to find out. The opposite view of the fist view is the third view. As two chairs on the corridor is observed behind black stool in the third view, it means that in the first view, two chairs on the corridor is positioned behind me. This way, I can fully comprehend the spatial relationships of all objects in the entire scene. So, from the perspective of image 1: office area is to the right of black stool, two chairs on the corridor is to my behind, grey sofa is to the left of black stool. The answer is B. office area

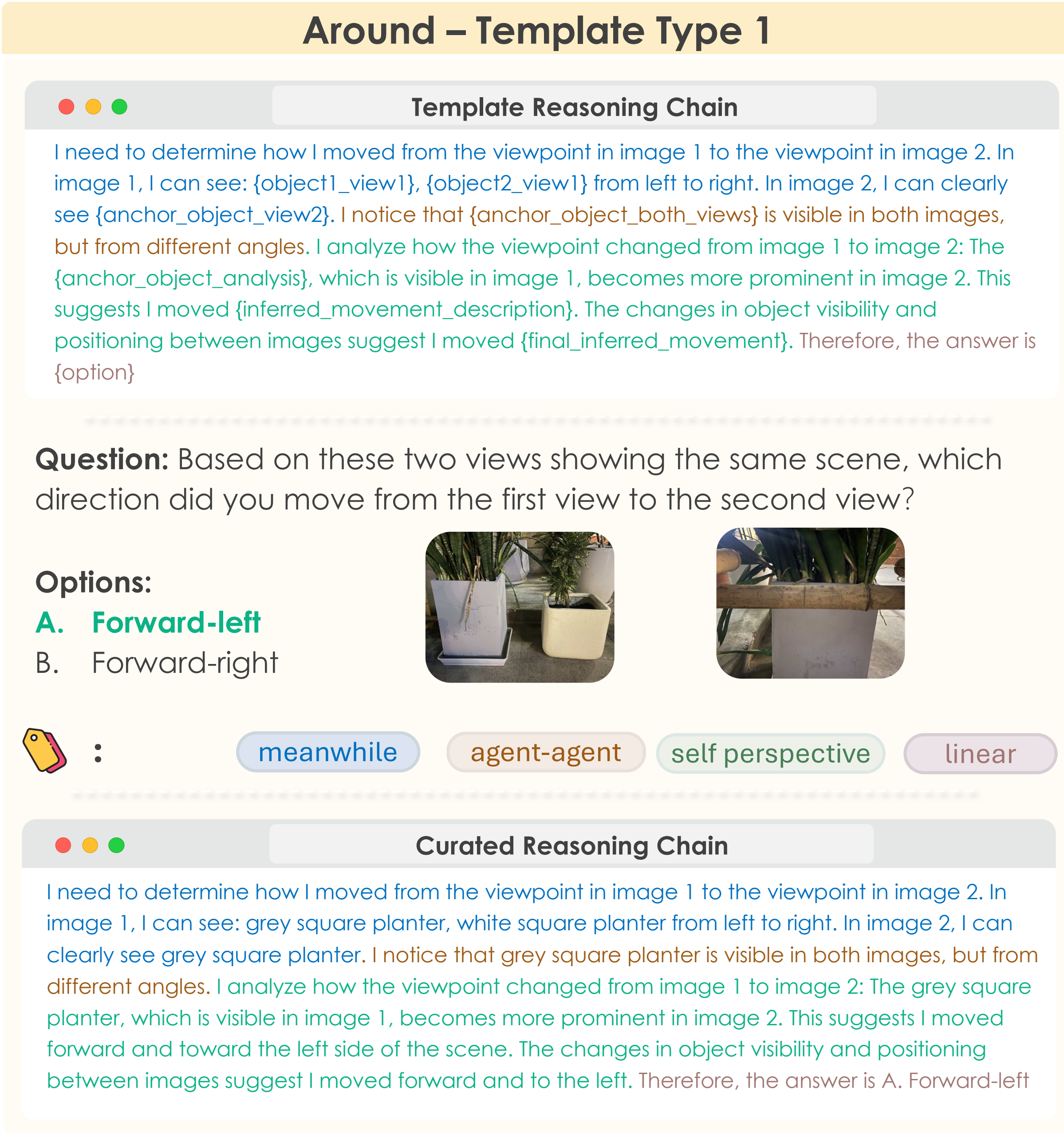


Figure 12. Example reasoning chain template for AROUND

Output for `Plain-CGMap-Out`: Input Prompt Refer to C.3.6

Based on my observations, I will build the cognitive map as follows:
¡CogMap¿

```
```json
{
    "white jar": {"position": [5, 5]},
    "bed sheet with a floral pattern": {"position": [5, 8]},
```
```

````
    "white headboard": {"position": [2, 5]},
    "clothes rack": {"position": [5, 2]},
    "table with cups on it": {"position": [8, 5]}
}
'''
````

After generating the cognitive map, I will provide my answer to the question:
¡Answer¿
B. Clothes rack

Output for `FF-Rsn`: Input Prompt Refer to C.3.2

In this scene, I observe four images showing different perspectives. All images feature the white jar as the main object. In image 1, I can see white jar in front of the table with cups on it. In image 2, I can see white jar in front of the bed sheet with a floral pattern. In image 3, I can see white jar in front of the clothes rack. In image 4, I can see white jar in front of the white headboard. To identify the position change across views, I focus on the main object's angle variation. Then, I analyze the angles and relative positions of other objects on the platform to back up this observation. I understand that: Image 1 is the initial view. Image 2 is captured after a 90-degree clockwise rotation from image 1. Image 3 is after another 90-degree clockwise rotation (180 degrees from image 1). Image 4 is after a further 90-degree clockwise rotation (270 degrees from image 1). Through analyzing these perspective changes, I can construct a complete spatial understanding: when I view bed sheet with a floral pattern behind white jar in the second view, it implies that in the first view, bed sheet with a floral pattern is on the right side of white jar. Similarly, when I see white headboard behind white jar in the fourth view, it indicates that in the first view, white headboard is on the left side of white jar. However, I am still uncertain about what lies behind me in the first view. Then, I recognize that I can examine the opposite view to find out. The opposite view of the fist view is the third view. As clothes rack is observed behind white jar in the third view, it means that in the first view, clothes rack is positioned behind me. This way, I can fully comprehend the spatial relationships of all objects in the entire scene. So, from the perspective of image 4: table with cups on it is to the right of white jar, bed sheet with a floral pattern is to my behind, clothes rack is to the left of white jar. So the answer is **B. Clothes rack**.

Output for `Plain-CGMap-Out`: Input Prompt Refer to C.3.6

Based on my observations, I will build the cognitive map as follows:
¡CogMap¿

````
```json
{
    "white jar": {"position": [5, 5]},
    "bed sheet with a floral pattern": {"position": [5, 8]},
    "white headboard": {"position": [2, 5]},
    "clothes rack": {"position": [5, 2]},
    "table with cups on it": {"position": [8, 5]}
}
'''
````

After generating the cognitive map, I will provide my answer to the question:
¡Answer¿
B. Clothes rack
````

ICCV
#2

Output for `Aug-CGMap-FFR-Out`: Input Prompt Refer to C.3.8

¡CogMap¿

```
```json
{
  "objects": [
    {"name": "white jar", "position": [5, 5]},
    {"name": "bed sheet with a floral pattern", "position": [5, 8]},
    {"name": "white headboard", "position": [2, 5]},
    {"name": "clothes rack", "position": [5, 2]},
    {"name": "table with cups on it", "position": [8, 5]}
  ],
  "views": [
    {"name": "Image 1", "position": [5, 6], "facing": "up"},
    {"name": "Image 2", "position": [4, 5], "facing": "right"},
    {"name": "Image 3", "position": [5, 4], "facing": "down"},
    {"name": "Image 4", "position": [6, 5], "facing": "left"}
  ]
}
'''
```

¡Reasoning¿
In this scene, I observe four images showing different perspectives. All images feature the white jar as the main object. In image 1, I can see white jar in front of the table with cups on it. In image 2, I can see white jar in front of the clothes rack. In image 3, I can see white jar in front of the bed sheet with a floral pattern. In image 4, I can see white jar in front of the white headboard. To identify the position change across views, I focus on the main object's angle variation. Then, I analyze the angles and relative positions of other objects on the platform to back up this observation. I understand that: Image 1 is the initial view. Image 2 is captured after a 90-degree clockwise rotation from image 1. Image 3 is after another 90-degree clockwise rotation (180 degrees from image 1). Image 4 is after a further 90-degree clockwise rotation (270 degrees from image 1). Through analyzing these perspective changes, I can construct a complete spatial understanding: when I view clothes rack behind white jar in the second view, it implies that in the first view, clothes rack is on the right side of white jar. Similarly, when I see white headboard behind white jar in the fourth view, it indicates that in the first view, white headboard is on the left side of white jar. However, I am still uncertain about what lies behind me in the first view. Then, I recognize that I can examine the opposite view to find out. The opposite view of the fist view is the third view. As bed sheet with a floral pattern is observed behind white jar in the third view, it means that in the first view, bed sheet with a floral pattern is positioned behind me. This way, I can fully comprehend the spatial relationships of all objects in the entire scene. So, from the perspective of image 4: clothes rack is to the right of white jar, bed sheet with a floral pattern is to my behind, table with cups on it is to the left of white jar. So the answer is A. Table with cups on it.
¡Answer¿
My answer is **A. Table with cups on it**

E.4. Detailed Graph Metric Results for SFT Graph-Related Experiments

This section provides a detailed look at the Supervised Fine-Tuning (SFT) training dynamics to support the main paper's conclusions. The figures below plot key metrics over training steps for four map-generation settings. A comparative analysis highlights that jointly training map generation and reasoning is the most effective strategy.

When training on map generation alone, as in the `Plain-CGMap-Out` and `Aug-CGMap-Out` settings, the graph quality metrics show rapid convergence. However, the final QA accuracy is limited, reaching 54.38% for `Plain-CGMap-Out` and 54.19% for `Aug-CGMap-Out`.

In contrast, the joint training approaches (`Plain-CGMap-FFR-Out` and `Aug-CGMap-FFR-Out`), despite a slower initial convergence on graph quality metrics, ultimately achieve far superior performance in task accuracy.
```

The `Plain-CGMap-FFR-Out` setting proves to be the most effective, reaching a QA Accuracy of 60.00%. The `Aug-CGMap-FFR-Out` setting also yields strong results, with QA accuracy climbing to about 65%. This demonstrates the superiority of joint training for achieving high performance in both the final task accuracy and the quality of the generated spatial representations.

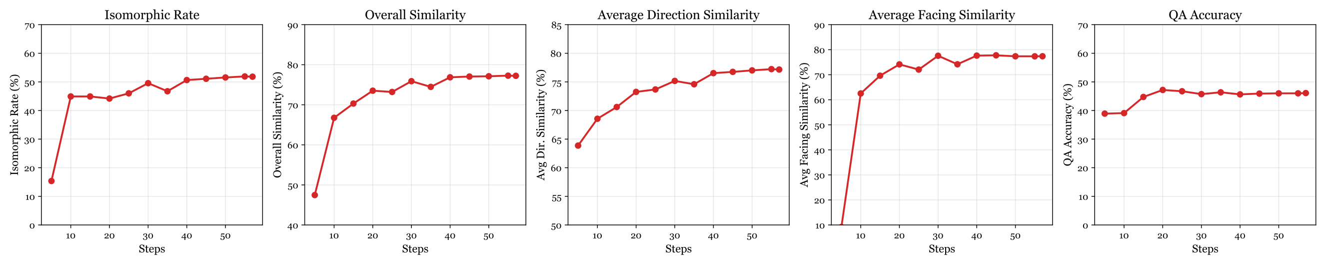


Figure 13. Training dynamics for the `Aug-CGMap-Out` setting.

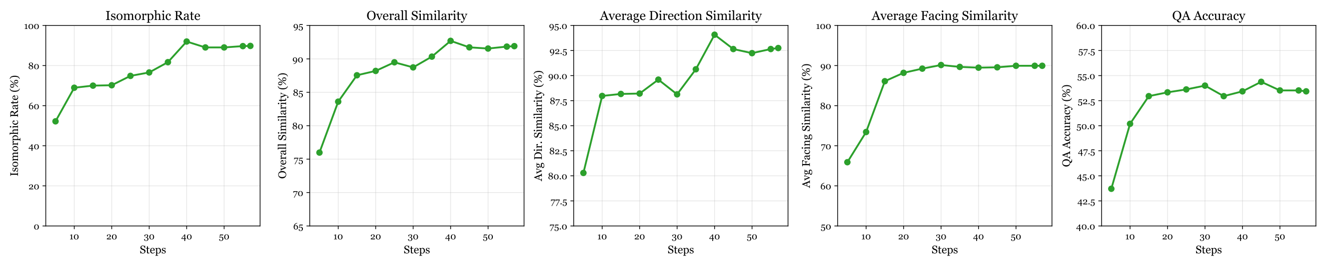


Figure 14. Training dynamics for the `Plain-CGMap-Out` setting.

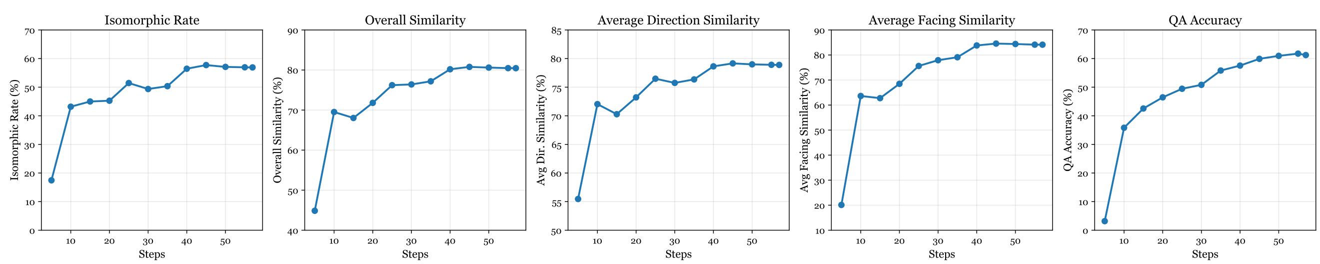


Figure 15. Training dynamics for the `Aug-CGMap-FFR-Out` setting.

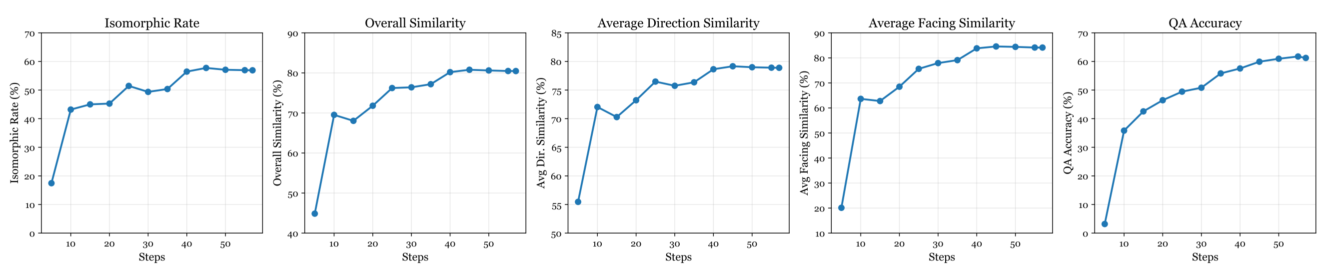


Figure 16. Training dynamics for the `Plain-CGMap-FFR-Out` setting, showing superior final performance.

E.5. Which Part of VLM is the Bottleneck for Spatial Understanding?

To develop more efficient fine-tuning strategies, it is crucial to understand which component of a Vision-Language Model (VLM)—the vision encoder responsible for perception or the Large Language Model (LLM) responsible for reasoning—presents the primary bottleneck for spatial understanding. To investigate this, we conduct a bottleneck analysis by selectively fine-tuning different parts of the VLM and observing the impact on performance.

We evaluate four distinct training configurations on the `Raw QA` task, with results captured at an early stage of training (step 57) to assess the initial learning dynamics. The configurations are: (1) the baseline performance of the pre-trained model without any fine-tuning; (2) fine-tuning only the vision encoder while keeping the LLM frozen; (3) fine-tuning only the LLM while keeping the vision encoder frozen; and (4) the standard approach of fine-tuning all parts of the model.

Table 6. VLM Training Bottleneck Analysis (Step=57, in %). Performance is measured on the MINDCUBE-TINY benchmark under the Raw QA setting.

Training Method	Overall	Rotation	Among	Around
Raw QA (no fine-tuning)	37.81	34.00	36.00	45.20
Freeze LLM (Vision Encoder Only)	37.81	30.50	37.00	45.60
Freeze Vision Encoder (LLM Only)	51.43	34.00	50.00	68.80
Tune All Parts	52.28	34.50	52.50	66.00

The results, presented in Table 6, offer several key insights. First, there is a dramatic performance leap from the no-fine-tuning baseline (37.81% overall), but only when the language model is trained. Methods involving LLM fine-tuning achieve over 51% accuracy, underscoring the necessity of adapting the model's reasoning capabilities.

Most strikingly, the performance bottleneck is almost exclusively concentrated in the LLM. Tuning only the LLM (`Freeze Vision Encoder`) yields an overall accuracy of 51.43%, capturing nearly the full performance gain of end-to-end fine-tuning (52.28%). In stark contrast, tuning only the vision encoder (`Freeze LLM`) provides no improvement whatsoever over the baseline (37.81%). This indicates that the bottleneck is not shared between modules. For this spatial task, adapting the model's language-based reasoning is critical, while adapting its visual perception is surprisingly ineffective.

Intriguingly, the fact that fine-tuning only the vision encoder fails to improve performance is in itself a significant finding. A possible explanation is that the pre-trained visual features are already sufficient to extract the necessary objects and their properties. The core challenge of the task seems to lie not in *what* is seen, but in *how to reason* about the spatial relationships across a series of views—a task primarily handled by the LLM. In conclusion, our analysis suggests that the most significant gains come from adapting the reasoning module. For efficient tuning, freezing the vision encoder and focusing solely on the LLM proves to be a highly effective strategy, achieving nearly top-tier performance at a fraction of the computational cost.

E.6. Branching from `Raw QA` SFT Checkpoint

In our main experiments, we fine-tuned the model for each specific task format starting from the base pre-trained VLM. A natural question arises: can a curriculum-based SFT approach further improve performance? Specifically, we investigate whether first fine-tuning the model on the simplest task format—'Raw QA', which only requires outputting the final answer—can establish a better foundation for learning to leverage more complex reasoning formats.

To test this hypothesis, we conducted a set of branching experiments. We took the checkpoint from the model fully fine-tuned on the 'Raw QA' task. Then, we used this specialized checkpoint as the initial weights for further fine-tuning on other scaffolding tasks, namely `Aug-CGMap-In`, `FF Rsn`, and `Aug-CGMap-FFR-Out`. It is important to note that during this second stage of fine-tuning, the model's output for all tasks was still constrained to be only the final answer option. This setup allows us to isolate the effect of the cognitive scaffolds on the model's internal reasoning process, rather than its ability to generate complex text.

The results, presented in Table 7, show a consistent and notable improvement across all branched tasks compared to their counterparts trained from scratch. For example, both `Aug-CGMap-In` and `Aug-CGMap-FFR-Out` reach an impressive overall accuracy of 49.00%. Even the `FF Rsn` method benefits from this two-stage approach, with its overall accuracy rising to 46.82%. These findings suggest that a two-stage SFT strategy is highly effective. By first grounding the model in the fundamental objective of the task (i.e., finding the correct answer) and then teaching it to process and leverage more complex cognitive scaffolds, we can achieve superior spatial reasoning performance. This indicates that the model, once primed for the core task, becomes more adept at utilizing the provided spatial context, even if it does not explicitly generate the reasoning chain or cognitive map.

Table 7. Performance of various methods after being fine-tuned from a `Raw QA` SFT checkpoint. This two-stage training approach led to performance gains across all methods. All accuracies are reported as percentages (%).

Method	Overall	Rotation	Among	Around
`Raw QA`	46.36	33.50	51.20	46.75
`Aug-CGMap-In`	49.00	35.50	53.20	50.50
`FF Rsn`	46.82	37.00	50.60	47.00
`Aug-CGMap-FFR-Out`	49.00	37.00	53.20	49.75

F. Can Reinforcement Learning Further Refine Spatial Thought Processes?

As discussed in the main paper, while Supervised Fine-Tuning (SFT) establishes a strong foundation for spatial reasoning, reinforcement learning (RL) presents an avenue for further optimizing spatial thought processes through outcome-driven feedback. The core inquiry is whether guiding VLMs with rewards can lead to the development of more precise spatial mental models and enhanced reasoning capabilities. This section of the appendix provides a more detailed exposition of the experimental setup employed for the RL phase of our research. Additionally, we present case studies to offer qualitative insights into how RL refines the models' spatial representations and reasoning chains.

F.1. Detailed Experimental Setup

For the reinforcement learning (RL) phase of our research, we employed the VAGEN framework. The core policy optimization algorithm used was Group Relative Policy Optimization (GRPO). To ensure consistency and allow for direct comparison with earlier stages of our work, key components from the Supervised Fine-Tuning (SFT) experiments were retained. Specifically, the base Vision-Language Model (VLM) for all RL configurations was Qwen2.5-VL-3B-Instruct, and evaluations were performed on the MINDCUBE-TINY benchmark. All previously established evaluation metrics were also retained.

In consideration of computational costs, each distinct RL configuration was trained for a duration of 0.5 epoch. The primary hyperparameters governing the RL training process were set as follows:

- **Training Batch Size:** 32
- **Maximum Prompt Length:** 1024 tokens
- **Maximum Response Length:** 512 tokens
- **Actor Learning Rate:** 1×10^{-6}
- **Critic Learning Rate:** 1×10^{-5}
- **Number of Trajectories per Rollout:** 8
- **Maximum Turns per Trajectory:** 1

As detailed in Section 5.1 of the main paper, we investigated three RL task configurations:

1. **RL-FF-Rsn (from scratch):** The Qwen2.5-VL-3B-Instruct model was trained to generate free-form reasoning chains without prior SFT for this specific task format.
2. **RL-Aug-CGMap-FFR-Out (from scratch):** The model was trained to jointly produce augmented cognitive maps and associated free-form reasoning, also starting from the base pre-trained VLM.
3. **RL-Aug-CGMap-FFR-Out (from SFT):** For this configuration, the RL training was initialized using the weights from the strongest performing SFT checkpoint, specifically the Aug-CGMap-FFR-Out SFT model.

The reward function was designed to be sparse yet directly indicative of desired behaviors. A reward of +1 was assigned if the model's output was structurally valid (e.g., the generated cognitive map adhered to the predefined schema). A more significant reward of +5 was given if the model provided the correct final answer to the spatial reasoning question. This dual-component reward was intended to steer the learning process towards generating both well-formed intermediate spatial representations and achieving high end-task accuracy.

F.2. VLM Response Examples After RL for Configurations in Section C.3

Output for `FF-Rsn`: Input Prompt Refer to C.3.2

¡think¿From image 4, the white jar is positioned on a surface with a floral-patterned bed sheet. To the left of the jar, there is a table with a pink tablecloth featuring heart shapes and a Hello Kitty design. There are also cups on the table. Therefore, the object to the left of the white jar is the table with the cups.¡/think¿¡answer¿**A. Table with cups on it**¡/answer¿

Output for `RL-Aug-CGMap-FFR-Out (from scratch)`: Input Prompt Refer to C.3.8

¡CogMap¿

```
{
  "objects": [
    {"name": "white jar", "position": [265, 436], "facing": "left"}
  ],
  "views": [
    {"name": "View/Image 1", "position": [265, 436], "facing": "left"},
    {"name": "View/Image 2", "position": [265, 436], "facing": "left"},
    {"name": "View/Image 3", "position": [265, 436], "facing": "left"},
    {"name": "View/Image 4", "position": [265, 436], "facing": "left"}
  ]
}
```

¡Your cognitive map¿
The white jar is located in the center of the 10x10 grid, facing to the left. The clothes rack is to the left of the white jar, as seen in the image 2 view. The bed sheet with a floral pattern is located to the right of the white jar, as seen in the image 1 view. The white headboard is not visible in any of the provided images.
¡Reasoning¿
Based on the provided images, the white jar is located in the center of the 10x10 grid, facing to the left. The clothes rack is to the left of the white jar, as seen in the image 2 view. The bed sheet with a floral pattern is located to the right of the white jar, as seen in the image 1 view. The white headboard is not visible in any of the provided images.
¡Answer¿
Therefore, my answer is **A. Table with cups on it**.

Output for `RL-Aug-CGMap-FFR-Out (from SFT)`: Input Prompt Refer to C.3.8

¡CogMap¿

````
```json
{
  "objects": [
    {"name": "white jar", "position": [5, 5]},
    {"name": "bed sheet with a floral pattern", "position": [5, 8]},
    {"name": "white headboard", "position": [2, 5]},
    {"name": "clothes rack", "position": [5, 2]},
    {"name": "table with cups on it", "position": [8, 5]}
  ],
  "views": [
    {"name": "Image 1", "position": [5, 6], "facing": "up"},
````
````

```
    {"name": "Image 2", "position": [4, 5], "facing": "right"},
    {"name": "Image 3", "position": [5, 4], "facing": "down"},
    {"name": "Image 4", "position": [6, 5], "facing": "left"}
  ]
}
```

¡Reasoning¿
In this scene, I observe four images showing different perspectives. All images feature the white jar as the main object. In image 1, I can see white jar in front of the clothes rack. In image 2, I can see white jar in front of the table with cups on it. In image 3, I can see white jar in front of the bed sheet with a floral pattern. In image 4, I can see white jar in front of the white headboard. To identify the position change across views, I focus on the main object's angle variation. Then, I analyze the angles and relative positions of other objects on the platform to back up this observation. I understand that: Image 1 is the initial view. Image 2 is captured after a 90-degree clockwise rotation from image 1. Image 3 is after another 90-degree clockwise rotation (180 degrees from image 1). Image 4 is after a further 90-degree clockwise rotation (270 degrees from image 1). Through analyzing these perspective changes, I can construct a complete spatial understanding: when I view table with cups on it behind white jar in the second view, it implies that in the first view, table with cups on it is on the right side of white jar. Similarly, when I see white headboard behind white jar in the fourth view, it indicates that in the first view, white headboard is on the left side of white jar. However, I am still uncertain about what lies behind me in the first view. Then, I recognize that I can examine the opposite view to find out. The opposite view of the fist view is the third view. As bed sheet with a floral pattern is observed behind white jar in the third view, it means that in the first view, bed sheet with a floral pattern is positioned behind me. This way, I can fully comprehend the spatial relationships of all objects in the entire scene. So, from the perspective of image 4: clothes rack is to the right of white jar, table with cups on it is to my behind, bed sheet with a floral pattern is to the left of white jar. So the answer is C. Bed sheet with a floral pattern.
¡Answer¿
My answer is **C. Bed sheet with a floral pattern**