
Spatial Mental Modeling from Limited Views

Supplementary Materials

Anonymous Author(s)

Affiliation

Address

email

1 Contents

2	A MINDCUBE Benchmark	3
3	A.1 Details for Data Collection and Annotation	3
4	A.2 Details of ourMINDCUBE Benchmark	5
5	A.2.1 Three kinds of invisibility settings	5
6	A.2.2 Label taxonomy	6
7	A.3 Examples	7
8	B Evaluation on MINDCUBE	8
9	B.1 Prompt Templates for Evaluation	8
10	B.2 Details in text only evaluation	8
11	B.3 Human Evaluation	10
12	B.4 Evaluation Setup	10
13	B.5 Analysis in settings	10
14	B.5.1 Around	10
15	B.5.2 Among	11
16	B.6 Failure case analysis	13
17	C Data Structures as Cognitive Scaffolds, Evaluation Metrics, and Input-Output Configurations	13
18	C.1 Data Structures as Cognitive Scaffolds	13
19	C.1.1 View Interpolation for Dynamic Updating	14
20	C.1.2 Cognitive Maps for Integrated Spatial Layouts	14
21	C.1.3 Free Form Reasoning	15
22	C.2 Evaluation Metrics	15
23	C.2.1 QA Accuracy	15
24	C.2.2 Graph Metrics for Cognitive Maps	16
25	C.3 Prompts for All Input-Output Configurations	17
26		

27	C.3.1	Example for Raw QA	17
28	C.3.2	Example for FF-Rsn	18
29	C.3.3	Example for VI-1 and VI-2	18
30	C.3.4	Example for Aug-CGMap-In	20
31	C.3.5	Example for Aug-CGMap-Out	21
32	C.3.6	Example for Plain-CGMap-Out	22
33	C.3.7	Example for Plain-CGMap-FFR-Out	23
34	C.3.8	Example for Aug-CGMap-FFR-Out	24
35	C.3.9	Example for CGMap-In-FFR-Out	25
36	D	Which Scaffolds Best Guide Spatial Thinking in Unchanged VLMs?	26
37	D.1	VLM Response Examples for Configurations in Section C.3	26
38	D.2	Additional Graph Metrics for Generated Graphs	28
39	E	Can We Teach VLMs to Build and Leverage Spatial Representations?	28
40	E.1	Supervised Fine-Tuning Data Curation	28
41	E.1.1	Cognitive Map Generation	28
42	E.1.2	Free-Form Reasoning Generation	29
43	E.2	Detailed Experimental Setup	30
44	E.3	VLM Response Examples After SFT for Configurations in Section C.3	34
45	E.4	Detailed Graph Metric Results for SFT Graph-Related Experiments	36
46	E.5	Which Part of VLM is the Bottleneck for Spatial Understanding?	37
47	E.6	Branching from Raw QA SFT Checkpoint	37
48	F	Can Reinforcement Learning Further Refine Spatial Thought Processes?	38
49	F.1	Detailed Experimental Setup	38
50	F.2	VLM Response Examples After RL for Configurations in Section C.3	39

51 A MINDCUBE Benchmark

52 A.1 Details for Data Collection and Annotation

53 **Image Collection and Selection.** Our MINDCUBE benchmark comprises 2,919 images (2,302
54 outdoor images from publicly released dataset and 616 self-collected images), where we imple-
55 ment a comprehensive image selection methodology encompassing four distinct view dynamics,
56 incorporating various data sources and processing procedures, as shown in Fig.2.

57 For rotation view dynamics, we implement a three-stage filtering strategy to extract meaningful
58 camera trajectories and key frames from ArkitScenes[1] dataset.

59 In the first stage, we analyze the top-down view of camera poses within each scene to identify two
60 types of trajectories: linear paths and small rotational arcs. A linear trajectory is characterized by
61 consistently oriented cameras exhibiting significant displacement perpendicular to their viewing
62 direction. A rotational arc trajectory is identified when three to four camera positions demonstrate
63 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

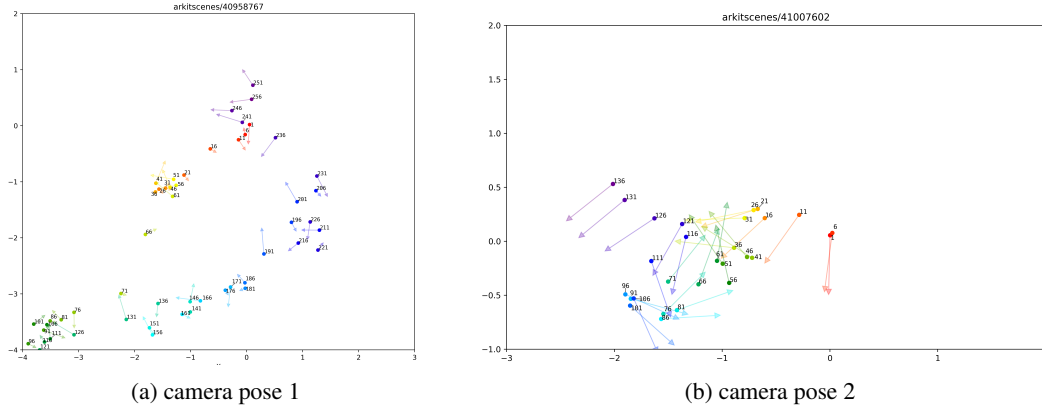


Figure 1: Examples of camera poses in ArkitScenes

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

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

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

85 For around view dynamics, image groups are manually curated from the DL3DV-10K[2] dataset and
86 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.

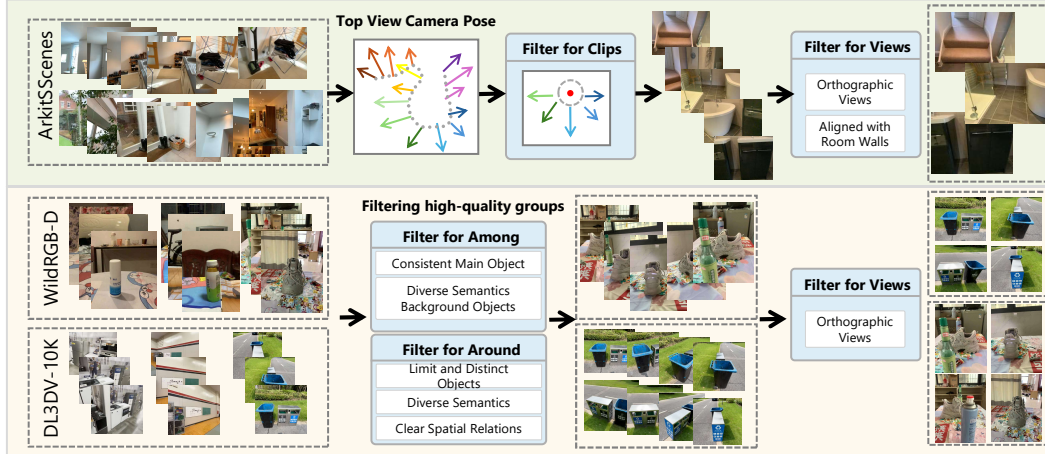


Figure 2: MINDCUBE Bench construction pipeline.

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

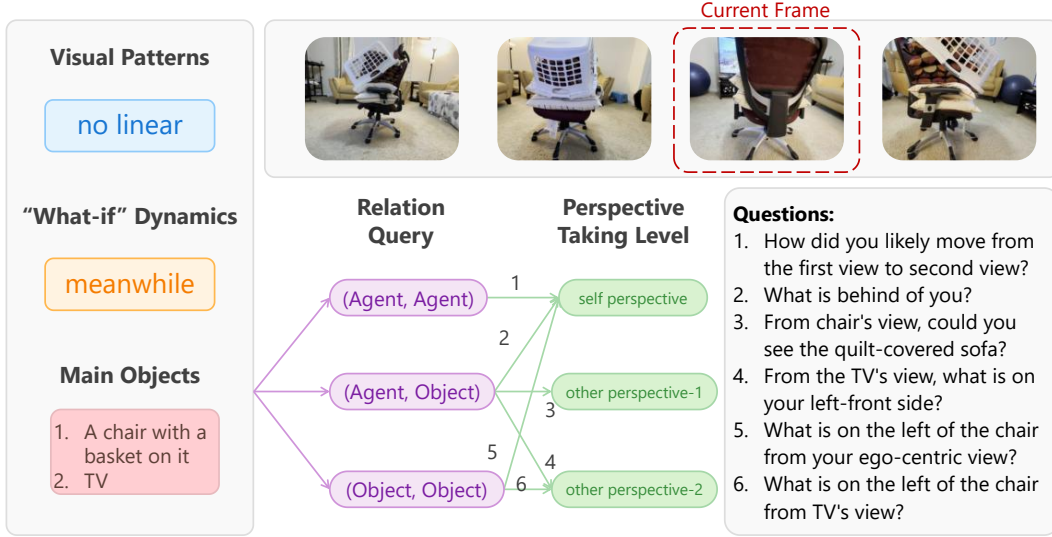


Figure 3: Example of different question-related label combinations to generate QA pairs.

123 encoded in its ID (e.g., "among_group001_q1_1_1"), while the original object and label information
 124 is preserved in the meta_info field to track the context of question generation.

125 A.2 Details of our MINDCUBE Benchmark

126 A.2.1 Three kinds of invisibility settings

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

130 We evaluate models' understanding of spatial invisibility by asking questions such as 'When
 131 positioned at a particular viewpoint, what should be to your left or right (given that each view
 132 only reveals what's directly ahead)?' or 'After rotating a quarter or half turn, what objects would
 133 be in front of you, to your left, behind you, or to your right?' We expect models to construct
 134 a comprehensive spatial understanding by leveraging the **sequential nature of the views and**
 135 **consistent spatial cues** across images (such as lighting direction), thereby demonstrating their ability
 136 to reason about the complete environment despite only having access to partial visual information
 137 from each viewpoint.
 138

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

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

146 This approach prevents models from relying solely on direct visual cues and instead necessitates true
 147 3D spatial reasoning by combining information from multiple perspectives.
 148

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

152 This setup creates interesting visibility constraints across different perspectives. For instance, a
 153 surrounding object visible in one view may be invisible in another view because of the constraints

154 imposed by the camera’s field of view. Through establishing consistency relationships between these
155 views, we can infer the relative positions of objects not directly visible from certain perspectives.
156 When an object is not visible from a particular viewpoint, consistency and spatial reasoning can
157 determine its position relative to the central object.

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

162 A.2.2 Label taxonomy

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

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

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

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

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

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

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

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

206 Perspective Taking.



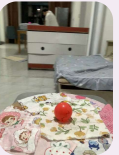

207 We propose a label called "Perspective Taking" that categorizes the complexity of viewpoint projection.
 208 This label distinguishes between three increasingly sophisticated levels of perspective reasoning:

- 209 • Self Perspective: Reasoning based on the current camera view or the observer's own
 210 viewpoint. This represents the baseline where no perspective shift is required.
- 211 • Other's Perspective Taking-1: The ability to determine visibility relationships from another
 212 agent's viewpoint. This involves understanding what objects are visible or occluded from a
 213 different viewpoint (e.g., determining whether a specific object is within the field of view of
 214 another camera). The another agent's viewpoint is usually determined by an object with a
 215 clear orientation in the image.
- 216 • Other's Perspective Taking-2: The ability to understand how spatial relationships transform
 217 when viewed from another agent's perspective. This more advanced capability requires men-
 218 tal rotation and spatial transformation to reason about relative positions (e.g., determining
 219 whether, from another viewpoint, object X appears to be positioned behind object Y).

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

223 A.3 Examples

Example of **Among** setting

Question: Based on view1 and view2 showing the same scene, which direction did you move from the first view to the second view?
Options: A. Forward-left B. Forward-right

System Prompt: Based on these four images (image 1, 2, 3, and 4) showing the red ball from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings:

Question: If you are standing at the viewpoint presented in image 1, then you turn left and move forward, will you get closer to the light-colored sofa?
Options: A. Yes B. No

Question: If you are standing at the viewpoint presented in image 1, what is behind you?
Options: A. white-red cabinet B. light-colored sofa C. dark brown sofa D. school bag and TV cabinet

Question: From the viewpoint presented in image 1, what is to the left of the red ball?
Options: A. white-red cabinet B. light-colored sofa C. dark brown sofa D. school bag and TV cabinet

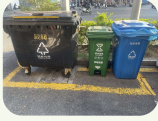


Question: From the viewpoint presented in image 1, what is to the right of the red ball?
Options: A. white-red cabinet B. light-colored sofa C. dark brown sofa D. school bag and TV cabinet

Question: If you are positioned where the light-colored sofa is and facing the same direction, what would be to the left of the red ball from this view?
Options: A. dark brown sofa B. school bag and TV cabinet C. white-red cabinet

Question: If you are positioned where the dark brown sofa is and facing the same direction, what would be to the right of the red ball from this view?
Options: A. school bag and TV cabinet B. white-red cabinet C. light-colored sofa

Figure 4: Example of among setting.

Example of **Around** setting

View1 (Front)
View2 (Left)
View3 (Right)

: 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.

Prompt for Dense Captioning

Describe this image in detail, specifically focusing on the spatial relationship between objects.

8

Example of **Around** setting

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.

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: 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 123 or View 145 Used)**

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.

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: 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)**

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.

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>

239

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 are 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 [4], LLaVA-Video [5], mPLUG-Owl3 [6], InternVL2.5 [7], QwenVL2.5 [8], LongVA [9], IDEFICS [10], DeepSeek-VL2 [11]), models fine-tuned on interleaved image-text data (e.g., Mantis [12]), leading proprietary APIs (e.g., GPT-4o, Claude-3.7-Sonnet), and models specifically fine-tuned for spatial reasoning tasks (e.g., RoboBrain [13], Space-Mantis [14], Space-LLaVA [15]).

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			Quantity Sensitivity Test		
GPT4-o	T	F	GPT4-o	T	F
T	13.85	13.93	T	13.10	10.42
F	13.49	58.74	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 we used are described in the appendix. The analysis revealed:

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

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.

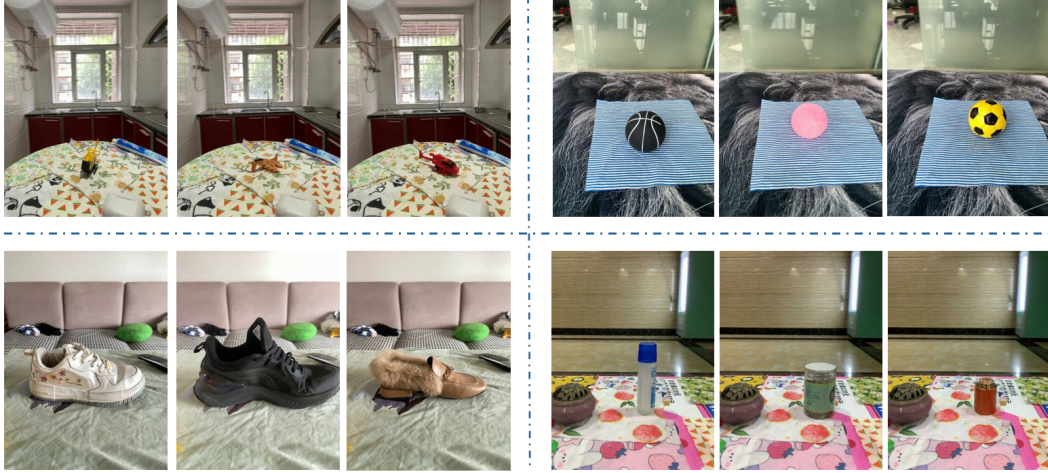


Figure 7: Examples in Attribute Invariance Test.



Figure 8: Examples in Quantity Sensitivity Test.

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

$$\chi^2 \approx 0.85 \quad (p > 0.36), \quad \Delta \text{Accuracy} = 5.1\%, \quad h = 2 \arcsin(\sqrt{p_2}) - 2 \arcsin(\sqrt{p_1}) \approx 0.10 \quad (2)$$

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

330 Three key findings emerge from our analysis: 1. In attribute invariance, model maintains 88.6 %
 331 consistency ($p > 0.89$) for modified object attributes, confirming geometric reasoning predominance
 332 over visual features; 2. In quantity sensitivity, model's 5.1 % accuracy gain ($h = 0.10, p > 0.36$)
 333 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 architectural improvements should therefore focus on enhancing transitive spatial inference mechanisms and view-invariant scene representation to support more robust reasoning across multiple perspectives.

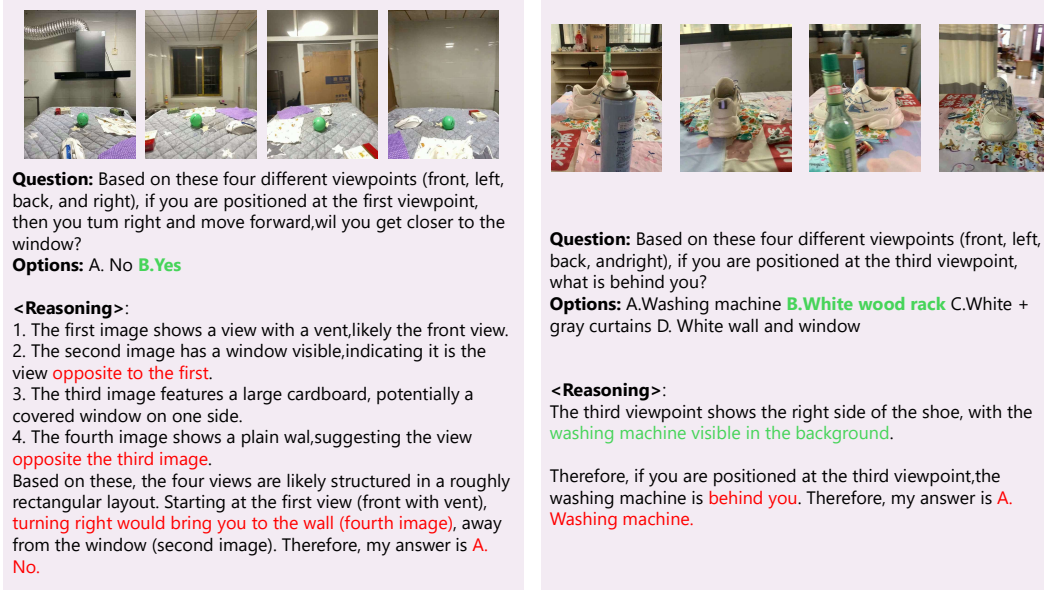


Figure 9: Failure case analysis. We show GPT4-o’s reasoning process. In case 1, the model is unable to establish the spatial location corresponding to each view; In case 2, the model confuses the subject of the “behind” relationship.

343

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. 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 [16]. These “cognitive collages” allow for the integration of fragmented observations and support reasoning across unseen perspectives. Johnson-Laird [17] 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 [18] 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.

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 [19], 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 [1, 3] where we sampled our data, or we use Stable Virtual Camera [20] 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 [16, 21] 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
 - facing_objects — indicating which objects a given camera is facing
- **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 [18]. To approximate this procedural and inferential aspect, we utilize free-form reasoning, implemented as a natural language Chain-of-Thought (CoT) [22] 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

461 questions. A higher QA Accuracy indicates better alignment between the model’s responses and the
 462 ground truth.

463 The metric is formally defined as:

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

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

466 C.2.2 Graph Metrics for Cognitive Maps

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

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

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

486 **Notation.** A *cognitive map* is a finite set of objects $\mathcal{O} = \{o_1, \dots, o_n\}$ where each object o_i
 487 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
 488 $f_i \in \{\text{up, right, down, left, inner, outer}\} \cup \{\emptyset\}$. For two maps, we distinguish (1) the *ground-truth*
 489 map $(\mathcal{O}^*, p^*, f^*)$ and (2) a *generated* map $(\mathcal{O}^g, p^g, f^g)$.

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

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

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

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

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

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

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

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

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

499 Then the directional similarity score is given by:

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

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

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

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

504 Then:

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

505 **Overall similarity.** To aggregate the directional and facing similarities, we use a weighted combi-
 506 nation:

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

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

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

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

$$\exists k \in \{1, \dots, m\} \forall o_i, o_j \in \mathcal{O}^* : R^*(o_i, o_j) = R_{(k)}^g(o_i, o_j),$$

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

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

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

518 C.3 Prompts for All Input-Output Configurations

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

520 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

521

522 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]

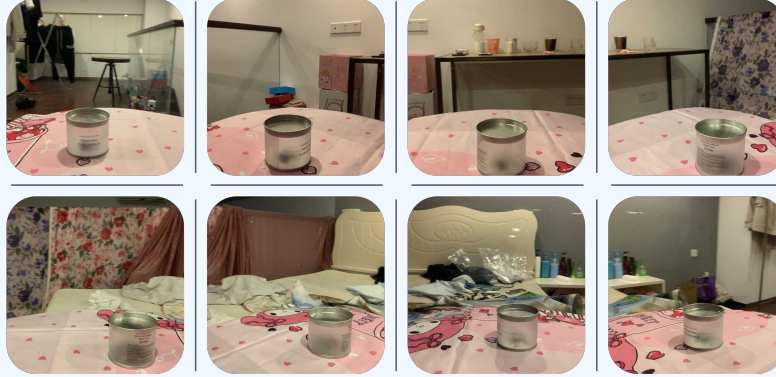
Based on these four images (image 1, 2, 3, and 4) showing the white jar from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: From the viewpoint presented in image 4, what is to the left of the white jar?

A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

523

524 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'.

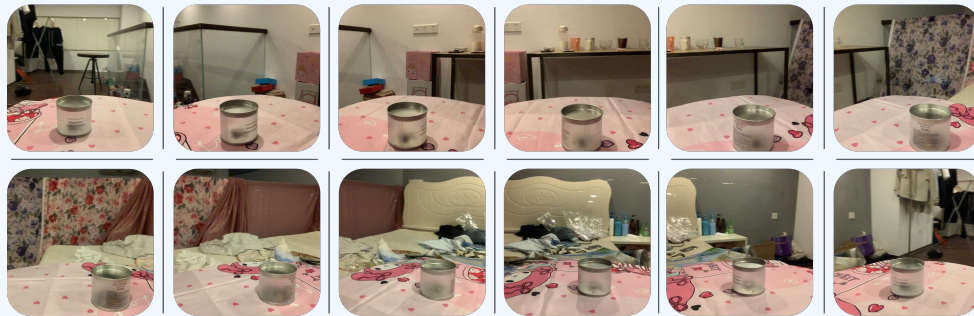
[Question]

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

525

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

526

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" }
  ]
}
```

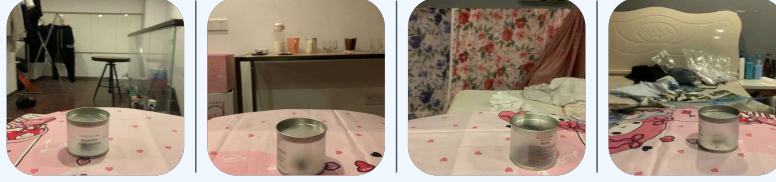
[Question]

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

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

531



[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

532

left of the white jar?

A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

533

534 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

535

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

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

540



[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

541

left of the white jar?

A. Table with cups on it B. Clothes rack C. Bed sheet with a floral pattern D. White headboard

542

543 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

544

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],
"facings": "left"},
{"name": "table with cups on it", "position": [188, 200],
"facings": "left"},
{"name": "white headboard", "position": [188, 317],
"facings": "left"}
],
"views": [
{"name": "Image 1", "position": [0, 0], "facings": "left"},
{"name": "Image 2", "position": [0, 0], "facings": "left"},
{"name": "Image 3", "position": [0, 0], "facings": "left"},
{"name": "Image 4", "position": [0, 0], "facings": "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.**

560

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

```

'''json
{
  "white_jar": {"position": [150, 440], "facings": "left"},
  "table_with_cups_on_it": {"position": [187, 260],
"facings": "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.**

561

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

C. Bed sheet with a floral pattern

562

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 near-perfect ability to generate syntactically and semantically valid maps, achieving a **99.00% Valid Rate**. 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 is challenging, as shown by the modest **Isomorphism Rates**. The Plain-CGMap-FFR-Out method performs slightly better, with 16.64% of its maps being structurally identical (isomorphic) to the ground truth, compared to 11.45% for the augmented map method.
- **Similarity Trade-off:** There is a clear trade-off between capturing object relationships and object orientations. The Aug-CGMap-FFR-Out method, which explicitly includes camera views, achieves a higher **Overall Similarity** (41.14%) and is notably better at representing the relative directional relationships between objects (**Avg. Dir. Sim.** of 40.53%). In contrast, the Plain-CGMap-FFR-Out method is significantly more adept at identifying the correct orientation of individual objects, with an **Avg. Facing Sim.** of 50.92%.

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	99.00	11.45	41.14	40.53	42.56
Plain-CGMap-FFR-Out	99.00	16.64	36.86	30.84	50.92

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.

599 The generation process was automated via a script that processes input JSONL files, where each
 600 line item contains scene details including images and, crucially, `meta_info` describing the objects,
 601 their potential orientations, and the camera viewpoint setup. For every item, the script first identifies
 602 its specific spatial arrangement "setting" (e.g., "around," "among," "translation," or "rotation") by
 603 parsing the item's unique ID. Based on this setting, dedicated functions apply a set of predefined
 604 rules and heuristics to determine the 2D coordinates (on a 10x10 grid) and facing directions for both
 605 the objects and the camera views.
 606 For instance, in the "around" setting, objects (typically 2-4) are placed in a predetermined linear
 607 arrangement near the grid's center (e.g., at coordinates like [4,5], [5,5]), and camera views are
 608 positioned at cardinal directions relative to these objects, based on the specific camera angles
 609 pertinent to the question. In the "rotation" setting, the camera is fixed at the center ([5,5]), and its
 610 facing direction changes across views, while object positions are defined relative to the camera's
 611 current orientation. Similar rule-based placements are implemented for "among" (objects in a cross
 612 or T-shape with views from specific angles) and "translation" (objects arranged linearly to depict
 613 relationships like "on" or "down to") settings. Object orientations, if applicable, are also assigned
 614 based on the input `meta_info`.
 615 Finally, the generated layout of objects and views is formatted into a structured JSON string, rep-
 616 resenting the cognitive map. This JSON cogmap, along with templated instructional prompts
 617 (`cogmap_input` for VLM input format guidance and `cogmap_output` for VLM output task descrip-
 618 tion), is added to the original data item. The overall generation logic is summarized in Algorithm 1.

Algorithm 1 Cognitive Map Generation

Require: Dataset D containing items with spatial arrangement annotations
Ensure: Updated dataset with cognitive maps in JSON format
 1: **for all** $item \in D$ **do**
 2: $setting \leftarrow$ Extract setting type from $item.id$
 3: Initialize empty cognitive map $cogmap$ ▷ Position objects and views based on setting type
 4: **if** $setting = \text{"around"}$ **then**
 5: Position 2-4 objects in a line with coordinates like [4,5], [5,5], etc.
 6: Place views at cardinal positions based on camera angles
 7: **else if** $setting = \text{"among"}$ **then**
 8: Place center object at [5,5] and surrounding objects at [5,8], [2,5], [5,2], [8,5]
 9: Position views based on specified camera angles
 10: **else if** $setting = \text{"translation"}$ **then**
 11: Position objects according to their spatial relationships (e.g., "on", "down")
 12: Place views to highlight these spatial relationships
 13: **else if** $setting = \text{"rotation"}$ **then**
 14: Arrange objects based on rotation type (clockwise, counterclockwise, etc.)
 15: Fix camera at [5,5] with varying facing directions
 16: **end if** ▷ Add orientation information where applicable
 17: **for all** $object \in cogmap.objects$ **do**
 18: **if** $object$ has orientation **then**
 19: Add facing direction ("up", "down", "left", "right")
 20: **end if**
 21: **end for**
 22: Format $cogmap$ as structured JSON
 23: Add formatted cognitive map to $item$
 24: **end for**
 25: **return** Updated dataset D

E.1.2 Free-Form Reasoning Generation

620 While cognitive maps provide a structured, global understanding of the scene, effective spatial
 621 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 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 of the different training signals.

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

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

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

Rotation – Template Type 2



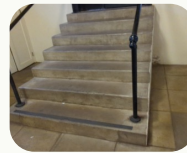
Template Reasoning Chain

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

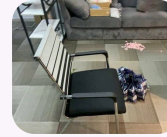
Template Reasoning Chain

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 first 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 11: Example reasoning chain template for AMONG

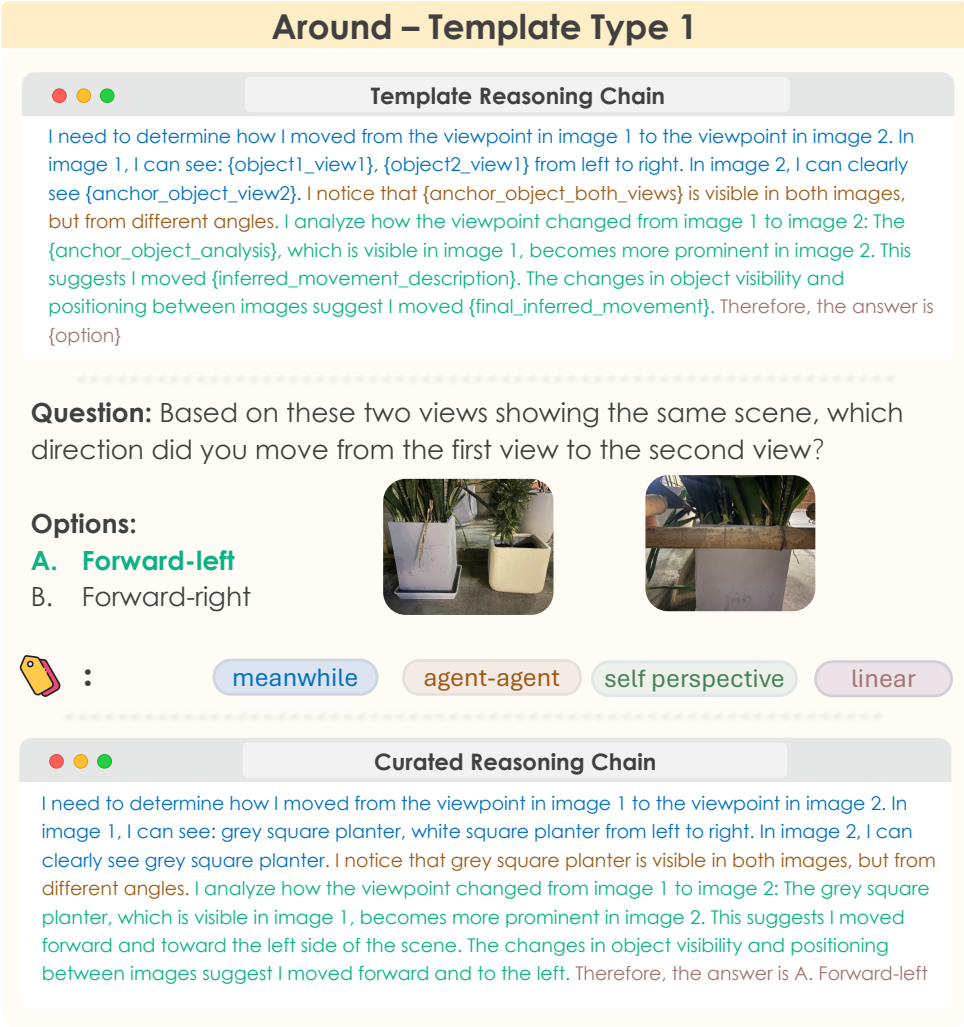


Figure 12: Example reasoning chain template for AROUND

675 claims point to 2 GPUs being used for these runs. The `per_device_train_batch_size` was set to
 676 4, and with `gradient_accumulation_steps` at 32, this resulted in an effective batch size of 256.

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

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

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

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

690 Answer: **D. White headboard**

689 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

691 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

692 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 first 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**.

694

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

695

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"}
  ]
}
```

696

}
, , ,

<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 first 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**

697

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

699 This section provides a detailed look at the Supervised Fine-Tuning (SFT) training dynamics to
700 support the main paper's conclusions. The figures below plot key metrics over training steps for three
701 map-generation settings. A comparative analysis highlights that jointly training map generation and
702 reasoning (Aug-CGMap-FFR-Out) is the most effective strategy. While training on map generation
703 alone leads to rapid convergence of graph quality metrics, the final QA accuracy remains limited. In
704 contrast, the joint training approach, despite a slower initial convergence, ultimately achieves superior
705 performance in both task accuracy and the quality of the generated spatial representations.

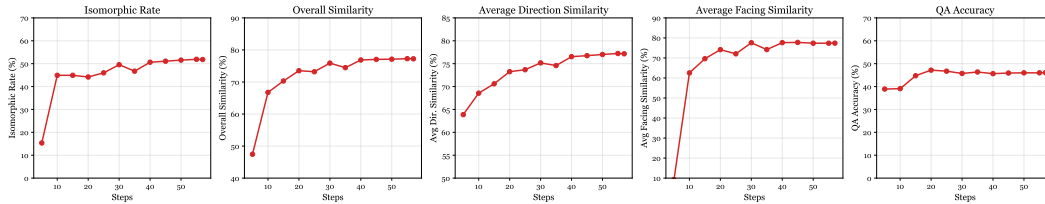


Figure 13: Training dynamics for the Aug-CGMap-Out setting.

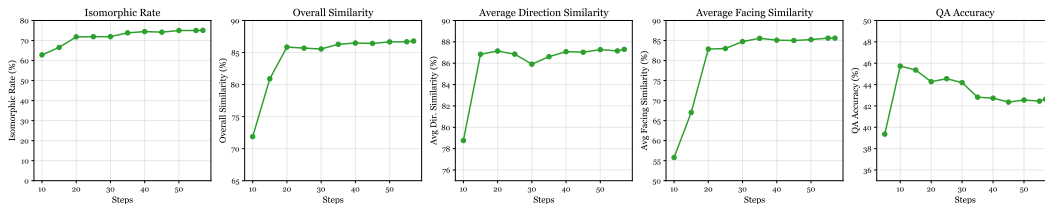


Figure 14: Training dynamics for the Plain-CGMap-Out setting.

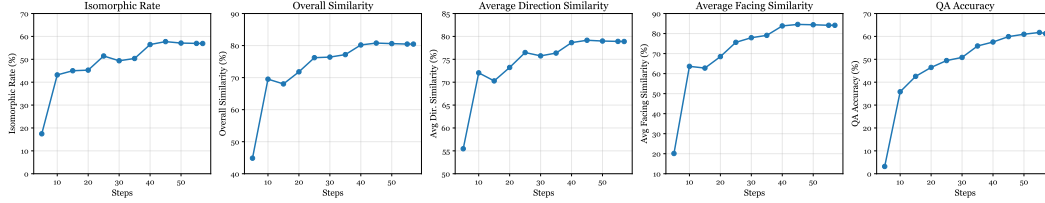


Figure 15: Training dynamics for the Aug-CGMap-FFR-Out setting, showing superior final performance.

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 19) 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=19, 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)	17.42	19.50	14.80	22.25
Freeze LLM (Vision Encoder Only)	46.83	33.50	51.20	47.50
Freeze Vision Encoder (LLM Only)	46.33	33.50	51.20	46.75
Tune All Parts	46.33	33.50	51.20	46.75

The results, presented in Table 6, offer several key insights. First, there is a dramatic performance leap from the baseline (17.42% overall) across all fine-tuning methods (all achieving >46% accuracy), underscoring the necessity of task-specific adaptation.

Most strikingly, training either the vision encoder alone or the LLM alone is sufficient to achieve nearly the full performance gain of end-to-end fine-tuning. Specifically, tuning only the vision encoder yields an overall accuracy of 46.83%, while tuning only the LLM reaches 46.33%. This indicates that the bottleneck is not confined to a single module. Both adapting the model’s visual perception to the spatial domain and aligning its language-based reasoning capabilities are almost equally critical.

Intriguingly, fine-tuning only the vision encoder slightly outperforms tuning all parts of the model (46.83% vs. 46.33%). A possible explanation is that this approach effectively adapts the visual features to the nuances of the spatial task while preserving the powerful, general-purpose reasoning capabilities of the frozen, pre-trained LLM, which might be slightly compromised during full fine-tuning. In conclusion, our analysis suggests that the most significant gains in spatial reasoning come from adapting both the perceptual and reasoning modules. For efficient tuning, focusing on adapting the vision encoder while leveraging a frozen LLM proves to be a highly effective strategy.

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}

- 777 • **Critic Learning Rate:** 1×10^{-5}
- 778 • **Number of Trajectories per Rollout:** 8
- 779 • **Maximum Turns per Trajectory:** 1

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

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

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

794 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>

795

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**.

796

<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 first 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**

797

798 References

- 799 [1] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz,
800 Tal Dimry, Brandon Joffe, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor
801 scene understanding using mobile rgb-d data. In *Thirty-fifth Conference on Neural Information Processing*
802 *Systems Datasets and Benchmarks Track (Round 1)*. 3, 14
- 803 [2] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu,
804 Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong,
805 Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. D13dv-10k: A large-scale scene dataset for
806 deep learning-based 3d vision, 2023. 3

- [3] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgb-d 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. 3, 14
- [4] 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. 10
- [5] 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. 10
- [6] 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. 10
- [7] 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. 10
- [8] 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. 10
- [9] 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. 10
- [10] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023. 10
- [11] 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. 10
- [12] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max W.F. Ku, Qian Liu, and Wenhui Chen. Mantis: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research*, 2024, 2024. 10
- [13] 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. 10
- [14] 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. 10
- [15] 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. 10
- [16] Barbara Tversky. Cognitive maps, cognitive collages, and spatial mental models. In *European conference on spatial information theory*, pages 14–24. Springer, 1993. 13, 14
- [17] Philip Nicholas Johnson-Laird. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Number 6. Harvard University Press, 1983. 13
- [18] 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. 14, 15
- [19] Mary Hegarty. Mental animation: Inferring motion from static displays of mechanical systems. *Journal of experimental psychology: learning, memory, and cognition*, 18(5):1084, 1992. 14
- [20] 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. 14

- 858 [21] Barbara Tversky. Structures of mental spaces: How people think about space. *Environment and behavior*,
859 35(1):66–80, 2003. [14](#)
- 860 [22] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny
861 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural*
862 *information processing systems*, 35:24824–24837, 2022. [15](#)