HIPPOSPATIAL: PERSISTENT SPATIAL SCENE GRAPHS FOR MULTI-VIEW SPATIAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-view spatial reasoning requires integrating partial observations from multiple viewpoints into a coherent global understanding of object positions, relations, and perspectives, a task on which Vision-Language Models (VLMs) consistently perform near random chance. Existing training-free methods either estimate object positions without metric grounding or re-derive geometry from scratch at every reasoning step, leaving no persistent spatial representation that spans all available views. Inspired by two principles of hippocampal spatial memory: pairwise relational encodings between objects enabling inference about pairs never directly co-observed, and the strict separation between allocentric long-term memory and egocentric working memory combined only through an explicit transform at retrieval time, we propose **HIPPOSPATIAL**, a training-free framework for multi-view spatial reasoning. Given a question and its candidate answers, HIPPOSPATIAL uses perception tools to segment relevant objects, recover their metric 3D positions in a shared world frame, and encode pairwise spatial relations in a Spatial Scene Graph, while separately maintaining camera geometry for each input view. At query time, a VLM performs visual grounding to resolve object references to graph nodes, then dispatches deterministic geometric tools that combine allocentric object positions with egocentric viewpoint information, delegating all numerical computation away from LLM reasoning. Experiments on MIND-CUBE, MMSI-BENCH, and SPAR-BENCH demonstrate that HIPPOSPATIAL outperforms prior training-free methods in most backbone-benchmark settings, with gains of up to 7.0 points over the best prior method.

1 INTRODUCTION

Spatial reasoning across multiple viewpoints is a basic requirement for any agent that acts in the physical world. A warehouse robot must locate a pallet seen by one camera but occluded in another; an autonomous vehicle must account for a pedestrian visible to a roadside camera but hidden from its own; an augmented-reality headset must anchor instructions to objects the user saw moments earlier from a different angle (Song et al., 2025; Qiao et al., 2026). In each case, the agent must fuse partial observations into one coherent model of the scene and answer questions such as *how far apart are two objects never seen in the same frame*, or *what lies to my left from another viewpoint*. Despite rapid progress in visual understanding, vision–language models (VLMs) perform close to chance on dedicated multi-view spatial reasoning benchmarks (Yang et al., 2026). The failure holds across model families and scales, pointing to a structural rather than a capacity limitation: VLMs tend to match local visual patterns within individual images instead of maintaining a globally consistent spatial model built from all views (Wang et al., 2026a; Yang et al., 2026; 2025).

Existing approaches fall into two groups. *Training-based* methods fine-tune VLMs with explicit 3D supervision, either by distilling depth annotations into instruction-tuning data (Chen et al., 2024; Cheng et al., 2024) or by adding geometric modules to the architecture (Fan et al., 2026; Hu et al., 2026). They improve spatial perception but require curated 3D data, substantial retraining compute, and re-adaptation whenever the task distribution or perception module changes. *Training-free* methods avoid these costs but have weaknesses of their own. *Cognitive-map* methods have the VLM sketch a 2D grid of the object layout before reasoning (Wang et al., 2026a; Yang et al., 2025); because positions are estimated without depth or a shared coordinate frame, these maps lack metric grounding and are prone to hallucination, and making them reliable has so far required supervised

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

fine-tuning or reinforcement learning. *Tool-augmented agents* equip VLMs with depth estimators and segmentation models (Cho et al., 2026; Luo et al., 2026; Chen et al., 2026; Marsili et al., 2025), gaining metric perception, but re-derive geometry from scratch through ad-hoc code or tool calls at every step: errors accumulate across calls, and objects observed only in non-active views are frequently lost. What both lines of training-free work lack is a *persistent, metric, and globally consistent* representation of the scene, built from all views before reasoning begins.

The brain offers a blueprint for exactly this kind of representation. Human spatial memory relies on two largely distinct systems (Burgess, 2006; Byrne et al., 2007): an *allocentric* store in the hippocampus that encodes where things are in viewpoint-independent world coordinates (O'keefe & Nadel, 1978), and an *egocentric* representation in parietal cortex that encodes the scene from the observer's current position. Rather than being fused, the two are kept apart and combined through an explicit coordinate transformation only when a spatial judgment is required (Byrne et al., 2007). This separation is what allows us to recall the layout of a room we have walked through and reason about it from a viewpoint we never occupied.

We turn this blueprint into **HIPPOSPATIAL**, a training-free neurosymbolic framework built on a simple principle: *build the map once, then reason over it*. HIPPOSPATIAL uses off-the-shelf segmentation and multi-view reconstruction to place every object from every view into a single metric world frame, merging repeated observations of the same object across views. The result is a *Spatial Scene Graph* (SSG), an allocentric memory whose nodes are objects with world coordinates and whose edges store the precomputed distance and direction between every pair of objects, including pairs never visible in the same image. Camera poses are held in a separate egocentric *View Registry*, mirroring the hippocampal – parietal division. To answer the question, the VLM is relieved of everything it does poorly: it only grounds the objects and viewpoints named in the question to entries in the graph and selects a deterministic geometric tool, which combines allocentric edges with egocentric camera geometry to compute the answer exactly. Every answer is therefore grounded in the same globally consistent map, objects seen only in other views remain a single lookup away, and no spatial arithmetic is left to the language model.

In summary, our contributions are as follows:

- We identify the absence of a persistent, metric, globally consistent scene representation as the shared limitation of training-free multi-view spatial reasoning, and propose HIPPOSPATIAL, which addresses it with a neuroscience-inspired Spatial Scene Graph that separates allocentric object memory from egocentric viewpoints and precomputes pairwise metric relations, without fine-tuning or task-specific adaptation.
- We evaluate HIPPOSPATIAL on MINDCUBE, MMSI-BENCH, and SPAR-BENCH across multiple VLM backbones, exceeding strong spatial-agent baselines in 6 of 9 backbone–benchmark combinations, with gains of up to 7.0 points over the best prior method.
- Through ablation and failure-mode analysis, we show that scene-graph construction and graph-guided reasoning are cleanly separable, and that accuracy scales with the quality of the reconstruction backbone without retraining or architectural change, so HIPPOSPATIAL improves directly as perception models improve.

2 RELATED WORK

2.1 SPATIAL REASONING IN VISION-LANGUAGE MODELS

Spatial reasoning remains a persistently limited capability of vision-language models despite broad progress in visual understanding (Chen et al., 2024; Cheng et al., 2024; Song et al., 2025). Early benchmarks probe single-image spatial relations such as relative depth, distance, and object placement (Ma et al., 2025; Fu et al., 2024; Li et al., 2026), revealing systematic failures that persist even in the largest models. A common response is to fine-tune VLMs with spatial supervision, either through scalable 3D annotation pipelines (Chen et al., 2024; Cheng et al., 2024) or by augmenting architectures with explicit geometry modules that fuse depth and point cloud features into the backbone (Fan et al., 2026; Hu et al., 2026). These approaches improve spatial perception but require curated 3D training data, substantial compute, and re-adaptation whenever the task distribution or perception modules change. More recent benchmarks reveal that even fine-tuned models fail on

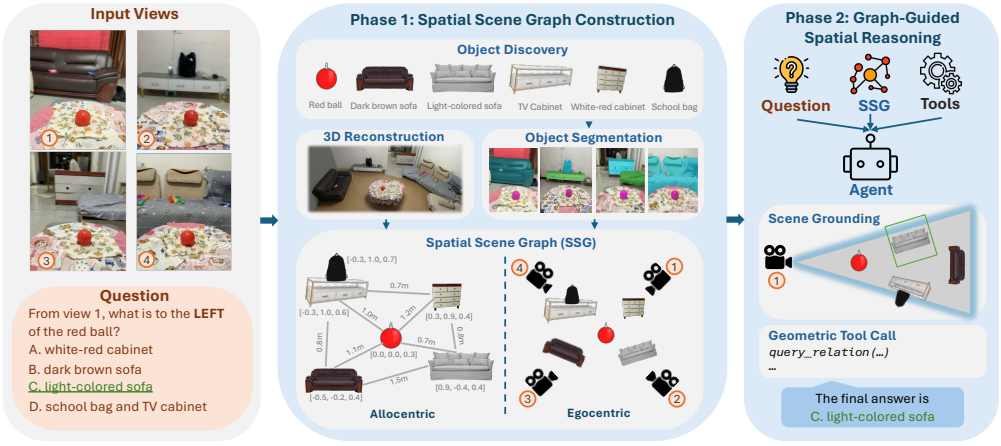


Figure 1: **The HIPPOSPATIAL framework.** Phase 1 constructs a Spatial Scene Graph from all input views, encoding object positions and pairwise spatial relations in an allocentric world frame with camera poses stored separately in an egocentric registry. Phase 2 grounds the question against the graph and dispatches deterministic geometric tools to produce the final answer.

multi-view settings that require integrating observations across viewpoints (Yang et al., 2026; Zhang et al., 2026), exposing a structural gap that neither single-view supervision nor architecture augmentation addresses. A training-free alternative prompts the VLM to estimate object positions on a 2D layout and reason over it before answering (Yang et al., 2025). Without metric depth estimation and without a mechanism for establishing a shared coordinate frame across views, however, these position estimates are unreliable, and the approach requires supervised fine-tuning and reinforcement learning to become usable in practice (Wang et al., 2026a).

2.2 TOOL-AUGMENTED SPATIAL AGENTS

Tool-augmented agents extend VLMs with specialist perception modules such as detectors, segmenters, and depth estimators to produce perceptual outputs that VLMs would otherwise have to approximate from visual appearance alone (Gupta & Kembhavi, 2023; Surís et al., 2023; Shen et al., 2023). Early work synthesizes a complete program in a single pass before executing it (Gupta & Kembhavi, 2023; Surís et al., 2023), enabling flexible tool composition but requiring a full analysis strategy to be committed before any intermediate result is observed. Subsequent work dispatches tool calls through structured command interfaces (Shen et al., 2023), offering a cleaner interface but constraining composition to pre-registered operations. In the spatial domain, pySpatial (Luo et al., 2026) and VADAR (Marsili et al., 2025) compose reconstruction and novel-view synthesis into single-pass visual programs, while SpaceTools (Chen et al., 2026) fine-tunes a VLM to coordinate perception tools through reinforcement learning. SpatialClaw (Cho et al., 2026) adopts code as the action interface in a persistent kernel, letting the VLM write one executable cell per step conditioned on prior outputs, and demonstrates that flexible code generation outperforms both single-pass programs and structured tool calls. HIPPOSPATIAL shares the insight that persistent state is valuable, but departs from prior work by pre-building a structured semantic SSG before reasoning, replacing LLM-written geometry with deterministic tools that operate directly on the graph.

3 METHODOLOGY

3.1 NEUROSCIENCE MOTIVATION

Human spatial cognition provides a principled blueprint for multi-view spatial reasoning. Neuroscience establishes that the brain maintains two strictly separate spatial systems for encoding and retrieving spatial information (Byrne et al., 2007; Burgess, 2006; O'keefe & Nadel, 1978). The hippocampus encodes an *allocentric* long-term memory, representing object locations in a viewpoint-independent world frame that remains stable regardless of where the observer stands or looks. The parietal cortex maintains an *egocentric* working memory, representing spatial relationships relative

to the observer's current perspective. These two systems are never merged into a single structure; they are connected only through an explicit transform at the moment a viewpoint-specific judgment is required (Byrne et al., 2007). Empirical evidence supports the importance of this separation: providing a map that mixes object positions with camera information degrades spatial reasoning compared to a map that encodes object positions alone (Wang et al., 2026a).

Beyond viewpoint separation, the hippocampus organizes spatial knowledge as *pairwise relational encodings* between objects rather than as isolated coordinates (Eichenbaum et al., 1999). When two objects appear together in a single view, the brain encodes their relative distance and direction as a pairwise association. Critically, this structure enables *transitive spatial inference*: if Object A is co-observed with Object B in one view, and Object B is co-observed with Object C in a different view, the system can compute the spatial relationship between A and C through intermediate node B without ever seeing A and C together in a single image. These twin principles, allocentric/egocentric decoupling and pairwise relational encoding, form the foundation of HIPPOSPATIAL.

3.2 OVERVIEW

HIPPOSPATIAL takes as input a set of images capturing the same scene from different viewpoints, together with a natural-language question Q about the scene. The goal is to produce a correct answer A by reasoning over all views, including about objects that are not visible from the viewpoint the question refers to.

The method runs in two phases, illustrated in Figure 1. In **Phase 1**, we build a Spatial Scene Graph $\mathcal{G} = (\mathcal{O}, \mathcal{E}, \mathcal{R})$ from all views before the question is read. The graph has three parts: object nodes $\mathcal{O}$, each placing one physical object at a fixed position in a shared world frame (an allocentric representation); relation edges $\mathcal{E}$, storing the distance and direction between every pair of objects; and a View Registry $\mathcal{R}$, recording where each camera was and which way it was facing (an egocentric representation), kept strictly separate from the objects. In **Phase 2**, a VLM agent links the objects and viewpoints mentioned in Q to entries of $\mathcal{G}$, optionally calls deterministic geometric tools from a fixed set $\mathcal{F}$ that query the graph, and states the result as the answer A.

The central idea is a clean division of labour. The graph gives the agent a structured picture of the scene, so it can work out which objects and viewpoints matter for the question. The tools carry out every spatial computation, such as distances, directions, and viewpoint-dependent relations, which would otherwise require the kind of numerical reasoning at which language models are error-prone. This leaves the VLM to focus on language and vision understanding, with all spatial computation offloaded to the tools.

3.3 PHASE 1: SPATIAL SCENE GRAPH CONSTRUCTION

Phase 1 builds $\mathcal{G}$ in three steps: perception, which detects objects and estimates their 3D positions in each view; cross-view association, which recognises when observations from different views belong to the same physical object; and graph construction, which assembles the object nodes, pairwise relation edges, and a registry of camera viewpoints.

3.3.1 PERCEPTION

Object discovery. Given the question and its candidate answer options, we ask a VLM with all views at once to identify the objects relevant to the scene. This produces a focused vocabulary of object labels shared across every view, so that the same object is named consistently throughout the pipeline.

Object segmentation. For every view and every label in the vocabulary, we pass the label to SAM3 (Carion et al., 2026) to obtain a pixel mask for each matching object instance. We refer to the j-th instance detected in view i as observation (i, j); it carries a label $\ell_{i,j}$ and the confidence $c_{i,j} \in [0, 1]$ that SAM3 assigns to its mask. A label may yield no instances in a view where the object is not visible, or several where more than one instance appears.

Metric 3D reconstruction. We pass all views jointly to Depth Anything 3 (Lin et al., 2026) for metric 3D reconstruction. From the visual overlap between views, it estimates the camera pose of

each view and a dense depth map, which together give the 3D world position of every pixel. Because the poses are estimated jointly, all views are expressed in one shared metric world frame without any external calibration.

Object position estimation. The 3D position $\mathbf{x}_{i,j} \in \mathbb{R}^3$ of observation (i, j) is the mean world-frame position of the pixels inside its mask. This gives one position estimate per object instance per view in which it appears.

3.3.2 Cross-view Association

The same physical object is often visible in several views, producing several separate position estimates. Cross-view association decides which of these observations refer to the same object and merges them into a single node.

Two observations are considered for merging only if they carry the same label and come from different views. For each such pair (i, j) and (i', j'), we compute a score that combines how close their position estimates are in the world frame with how similar the two objects look:

$$s = \alpha \exp\left(-\frac{\|\mathbf{x}_{i,j} - \mathbf{x}_{i',j'}\|_2^2}{\sigma^2}\right) + (1 - \alpha)\, \cos(\boldsymbol{\phi}_{i,j},\, \boldsymbol{\phi}_{i',j'})\,, \tag{1}$$

where $\boldsymbol{\phi}_{i,j}$ is the CLIP (Radford et al., 2021) embedding of the masked image crop of observation (i, j), $\cos(\cdot, \cdot)$ is cosine similarity, σ sets how quickly the geometric term decays with distance, and $\alpha \in [0, 1]$ balances geometry against appearance. Pairs scoring above a threshold τ are linked, and each connected group of linked observations becomes one object o_k, with $\mathcal{S}_k$ denoting its set of observations. Values of α, σ, and τ are provided in Appendix B.

Each object then receives a single canonical position, the average of its individual estimates weighted by segmentation confidence:

$$\mathbf{x}_k = \frac{\sum_{(i,j)\in\mathcal{S}_k} c_{i,j}\, \mathbf{x}_{i,j}}{\sum_{(i,j)\in\mathcal{S}_k} c_{i,j}}. \tag{2}$$

Combining several noisy depth-based estimates in this way yields a more reliable position than any single view provides.

3.3.3 Graph Construction

Object nodes (allocentric). Each merged object o_k becomes one node in $\mathcal{O}$. A node stores the object's label, its canonical position $\mathbf{x}_k$ from Equation 2, its mean detection confidence, and the views in which it was observed. Because positions are stored in the shared world frame rather than relative to any camera, each node describes *where the object is*, independently of *who is looking*.

Relation edges. For every ordered pair of distinct objects $(o_k, o_{k'})$, we precompute an edge $e_{kk'} \in \mathcal{E}$ storing their distance and the direction from one to the other, both in the world frame:

$$d_{kk'} = \|\mathbf{x}_{k'} - \mathbf{x}_k\|_2, \qquad \hat{\mathbf{u}}_{kk'} = \frac{\mathbf{x}_{k'} - \mathbf{x}_k}{d_{kk'}}. \tag{3}$$

These edges turn the graph from a list of isolated positions into a relational structure: every pair of objects carries an explicit distance and direction, *even if the two never appear together in the same image*, giving the VLM a direct spatial picture of the scene to reason over.

View Registry (egocentric). Separately from the objects, the View Registry $\mathcal{R}$ records, for each view i, the camera's position, its unit heading (forward) vector $\hat{\mathbf{h}}_i$, and its unit right vector $\hat{\mathbf{w}}_i$, all taken from the camera pose estimated during reconstruction. The registry is deliberately kept apart from $\mathcal{O}$ and $\mathcal{E}$, mirroring the allocentric/egocentric separation motivating our design, and is combined with object positions only when a spatial query requires it.

3.4 Phase 2: Graph-Guided Spatial Reasoning

Given Q, the graph $\mathcal{G}$, and the tool set $\mathcal{F}$, Phase 2 produces the answer in three steps: scene grounding, an optional geometric tool call, and answer generation.

3.4.1 SCENE GROUNDING

The VLM receives Q together with $\mathcal{G}$ serialised as structured text, with the allocentric object nodes and edges presented separately from the egocentric View Registry. The graph plays two roles here. First, it gives the agent a complete, structured view of the scene: which objects exist, where they lie relative to one another, and from which views they were seen. For simple questions that only require existence or counting judgements, this is enough for the agent to answer directly, without calling any tool. Second, the graph provides the exact object labels and view identifiers needed to write valid tool calls. The agent links each object mentioned in Q to a node in $\mathcal{O}$, and each mentioned viewpoint to an entry in $\mathcal{R}$.

3.4.2 GEOMETRIC TOOL CALL

When the question requires spatial computation, such as deciding whether one object is to the left of, right of, in front of, or behind another from a given viewpoint, or identifying which object lies closest to a reference, the agent calls a tool from $\mathcal{F}$. For questions requiring more than one geometric fact, the agent can iteratively call additional tools, up to a predefined maximum number of tool calls. Handing these computations to dedicated tools avoids the arithmetic and vector-manipulation errors that language models make when performing them directly.

To illustrate, `query_relation(anchor, target, viewpoint, relation)` is where the allocentric and egocentric parts of the graph meet. To decide whether object $o_{k'}$ lies to the left or right of anchor o_k as seen from view i, it projects the world-frame direction between them onto the camera's right vector:

$$\text{side}_i(k, k') = \begin{cases} \text{right} & \text{if } \hat{\mathbf{u}}_{kk'} \cdot \hat{\mathbf{w}}_i > 0, \\ \text{left} & \text{otherwise,} \end{cases} \tag{4}$$

with front/behind relations resolved analogously using the heading vector $\hat{\mathbf{h}}_i$. The full tool set and its implementation are given in Appendix C.

3.4.3 ANSWER GENERATION

Finally, the VLM receives the tool's structured output and produces the answer A. It interprets the spatial facts returned by the tool in the light of Q and expresses them in the required format, whether a multiple-choice option, a number, or a free-form description.

4 EXPERIMENTS

4.1 SETUP

Datasets. We evaluate HIPPOSPATIAL on three multi-view spatial reasoning benchmarks: MIND-CUBE (Wang et al., 2026a), covering egocentric perspective-taking and cross-view inference across indoor scenes; MMSI-BENCH (Yang et al., 2026), spanning positional relationships, attributes, and motion across real-world image pairs; and SPAR-BENCH (Zhang et al., 2026), covering twenty spatial tasks at three difficulty levels, evaluated on its multi-view subset. Full dataset descriptions and statistics are provided in Appendix A.

Baselines. We compare HIPPOSPATIAL against four groups of methods. **Direct VLM baselines** answer spatial questions directly from input images with no spatial augmentation. **Spatial mental models** (Wang et al., 2026a) prompt the VLM to construct an internal spatial representation before answering, including Chain-of-Thought, View Interpolation, and Cognitive Map. **Tool-augmented spatial agents** use external tools or reconstruction to support spatial computation: PYSPATIAL (Luo et al., 2026) composes 3D reconstruction and novel-view synthesis into a single-pass program; and SPATIALCLAW (Cho et al., 2026) uses code as the action interface in a persistent kernel with fixed perception tools. **Proprietary models** include Claude Haiku 4.5 (Anthropic, 2025) and Gemini 3.6 Flash (Google, 2026) as reference points representing capable general-purpose VLMs without spatial augmentation.

Table 1: Performance on three multi-view spatial reasoning benchmarks. Best results per backbone group are **bolded**, second-best underlined. Proprietary models are listed as reference and are not grouped by backbone. *Rot.*: Rotation, *Amo.*: Among, *Aro.*: Around.

Backbone	Method	MindCube				MMSI-Bench	SPAR-Bench
		Rot.	*Amo.*	*Aro.*	**Avg.**		
Proprietary Models							
	Claude Haiku 4.5	50.1	56.4	50.4	52.3	39.1	53.2
	Gemini 3.6 Flash	54.2	60.8	54.2	56.4	43.3	56.1
Qwen3.5-9B	*Direct VLM Baselines*						
	Direct VLM	43.5	49.2	44.7	45.8	31.1	43.6
	Spatial Mental Models (Wang et al., 2026a)						
	Chain-of-Thought	46.2	52.1	47.2	48.5	33.5	45.8
	View Interpolation	47.8	54.0	48.8	50.2	35.2	47.3
	Cognitive Map	49.6	55.9	50.8	52.1	37.1	49.1
	Tool-augmented Spatial Agents						
	pySpatial (Luo et al., 2026)	60.1	57.4	47.8	55.1	34.8	48.3
	SpatialClaw (Cho et al., 2026)	56.2	62.4	54.2	57.6	38.8	**55.4**
	HIPPOSPATIAL (Ours)	**61.4**	**69.8**	**61.7**	**64.3**	**45.8**	54.4
Gemma-4-31B	*Direct VLM Baselines*						
	Direct VLM	54.1	60.3	55.4	56.6	37.2	53.1
	Spatial Mental Models (Wang et al., 2026a)						
	Chain-of-Thought	56.8	63.1	57.7	59.2	40.1	55.8
	View Interpolation	58.6	65.2	59.5	61.1	42.3	57.4
	Cognitive Map	60.8	67.5	61.9	63.4	44.8	59.6
	Tool-augmented Spatial Agents						
	pySpatial (Luo et al., 2026)	65.2	67.1	67.3	66.5	**54.2**	58.4
	SpatialClaw (Cho et al., 2026)	66.8	68.1	**72.4**	69.1	49.9	61.9
	HIPPOSPATIAL (Ours)	**74.0**	**79.3**	69.9	**74.4**	53.8	**65.7**
GPT-5-mini	*Direct VLM Baselines*						
	Direct VLM	52.1	58.2	53.5	54.6	38.3	52.2
	Spatial Mental Models (Wang et al., 2026a)						
	Chain-of-Thought	54.6	61.0	55.7	57.1	40.2	54.1
	View Interpolation	56.2	62.8	57.4	58.8	42.1	55.8
	Cognitive Map	58.1	64.8	59.2	60.7	44.3	57.6
	Tool-augmented Spatial Agents						
	pySpatial (Luo et al., 2026)	58.8	68.6	**70.3**	65.9	48.3	56.7
	SpatialClaw (Cho et al., 2026)	67.2	**75.1**	60.2	67.5	51.8	**60.4**
	HIPPOSPATIAL (Ours)	**71.8**	72.4	68.5	**70.9**	**55.4**	59.8

Implementation Details. We evaluate HIPPOSPATIAL with three backbone VLMs to assess generalization across model families and scales: **Qwen3.5-9B** (Team, 2026), **Gemma-4-31B** (Team et al., 2026), and **GPT-5-mini** (Singh et al., 2025). All baselines across all groups are evaluated with the same backbones where applicable for fair comparison. For perception, we use **SAM3** (Carion et al., 2026) with multi-prompt mask selection and **Depth Anything 3 Giant** (Lin et al., 2026) for metric 3D reconstruction. Full implementation details are provided in Appendix B.

4.2 MAIN RESULTS

Table 1 presents results of HIPPOSPATIAL against all baseline groups across three benchmarks and three backbone VLMs. HIPPOSPATIAL achieves the highest MindCube average accuracy across all three backbone settings and sets the overall best SPAR-Bench score with Gemma-4-31B, surpassing all baselines across all backbones. On MMSI-Bench, HIPPOSPATIAL leads in two of three backbone settings, with gains consistent across model families and scales.

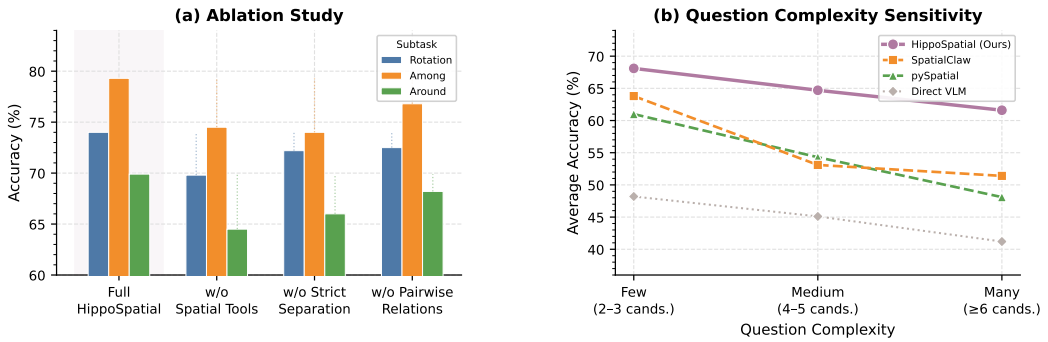


Figure 2: **(a) Ablation study** on MindCube with Gemma-4-31B, removing one component at a time from the full pipeline. **(b) Question complexity sensitivity** on MindCube with Qwen3.5-9B, binned by the number of objects detected during the object discovery step.

HIPPOSPATIAL demonstrates consistent gains over tool-augmented spatial agents. Spatial mental models already improve over Direct VLM by 3–6 points on MindCube, and tool-augmented agents advance further, with SpatialClaw reaching 57.6 on MindCube with Qwen3.5-9B compared to Cognitive Map at 52.1. HIPPOSPATIAL advances beyond prior tool-augmented agents by strictly separating allocentric world coordinates from egocentric viewpoint information in the SSG, giving the agent a frame-consistent scene description to ground spatial queries against rather than reconciling competing per-view cues on the fly. In contrast, agents such as SpatialClaw must invoke multiple geometric tools at query time and synthesize their outputs through VLM reasoning, a process that accumulates errors as scene complexity grows. HIPPOSPATIAL leads SpatialClaw by 6.7 points on MindCube with Qwen3.5-9B and 5.3 points with Gemma-4-31B, with the larger gain under the smaller backbone suggesting that pre-computed spatial structure compensates most when the backbone has limited capacity for on-the-fly geometric reasoning.

Generalization to diverse benchmarks. On MMSI-Bench, HIPPOSPATIAL leads with Qwen3.5-9B (45.8) and GPT-5-mini (55.4), and remains second with Gemma-4-31B at 53.8 against pySpatial at 54.2. On SPAR-Bench, HIPPOSPATIAL achieves its largest margin with Gemma-4-31B, outperforming SpatialClaw by 3.8 points (65.7 vs. 61.9), while remaining within one point of SpatialClaw in the Qwen3.5-9B and GPT-5-mini settings (54.4 vs. 55.4 and 59.8 vs. 60.4), demonstrating that the persistent spatial memory generalizes beyond the controlled indoor setting of MindCube to diverse real-world multi-image spatial scenarios.

Comparison to proprietary models. With Gemma-4-31B, HIPPOSPATIAL (MindCube Avg 74.4) substantially surpasses both Claude Haiku 4.5 (52.3) and Gemini 3.6 Flash (56.4) evaluated without spatial augmentation, showing that an open-source backbone within HIPPOSPATIAL can match or exceed capable closed-source VLMs on structured spatial tasks. HIPPOSPATIAL with GPT-5-mini as backbone (70.9) also consistently improves over Direct GPT-5-mini (54.6), confirming the gains extend to proprietary backbones as well.

4.3 ANALYSIS

Ablation study. Figure 2(a) isolates the contribution of each core component by removing it one at a time from the full pipeline on MindCube, holding the Gemma-4-31B backbone fixed. Replacing deterministic geometric tools with VLM-estimated answers (*w/o Spatial Tools*) causes the largest average drop (-4.2 on Rotation, -4.8 on Among, -5.4 on Around), confirming that offloading metric computation to external tools is critical for accurate spatial reasoning. Mixing allocentric and egocentric representations (*w/o Strict Separation*) disproportionately hurts Among and Around (-5.3 and -3.9 pts respectively) while leaving Rotation relatively intact (-1.8 pts), consistent with cross-view queries being most sensitive to coordinate-frame confusion. Removing pre-computed relational edges (*w/o Pairwise Relations*) produces the smallest overall drop (-1.5 on Rotation, -2.5 on Among, -1.7 on Around), suggesting that the agent can partially recover by calling tools to compute spatial relations from raw 3D positions when explicit edges are absent.

Question complexity sensitivity. Figure 2(b) groups MindCube queries by the number of objects detected during the object discovery step and tracks accuracy across bins with Qwen3.5-9B. All methods degrade as the candidate set grows, but with markedly different patterns. HIPPOSPATIAL drops only 6.5 points across the full range, declining gradually and consistently, as the pre-built SSG gives the agent a structured scene reference to ground against even when many objects must be disambiguated. SpatialClaw loses 10.7 points in the few-to-medium transition then nearly plateaus (-1.7 pts to many), suggesting its tool calls saturate at moderate candidate density. pySpatial degrades more evenly across both transitions (-6.7 and -6.2 pts), consistent with single-pass program synthesis scaling poorly with more interchangeable objects.

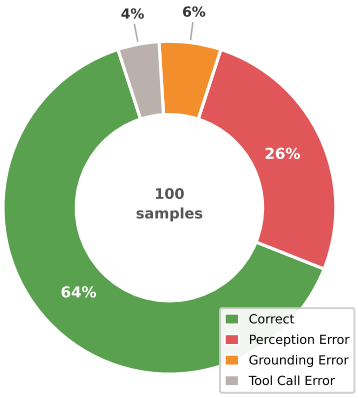


Figure 3: Failure case analysis over 100 randomly sampled cases from MindCube.

Failure mode analysis. We randomly sample 100 cases from MindCube under the Qwen3.5-9B backbone and manually categorise each outcome, with results shown in Figure 3. Among the remaining failures, perception noise accounts for 26% of all cases, originating from imprecise depth estimates or segmentation masks produced by DA3 and SAM3 that cause the SSG to be built with incorrect object positions or missing nodes. Grounding errors, where the VLM fails to map the question to the correct graph nodes or viewpoints, account for 6%, and malformed tool calls for the remaining 4%. The low incidence of grounding and tool call errors confirms that the graph structure is reliable once perception succeeds, and that advances in depth and segmentation models represent the most direct path to higher overall accuracy.

3D reconstruction backbone sensitivity. Figure 4 evaluates HIPPOSPATIAL under three reconstruction backbones while holding graph querying and spatial reasoning fixed. Even the weakest backbone, VGGT (Wang et al., 2025), gains $+13.6$, $+12.9$, and $+9.3$ points over Direct VLM on MindCube, MMSI-Bench, and SPAR-Bench respectively, confirming that HIPPOSPATIAL generalises across backbones of varying quality. DA3-Giant (Lin et al., 2026) is strongest, outperforming VGGT by up to 4.2 points and Pi3 (Wang et al., 2026b) by up to 2.8 points, with the inter-backbone gap remaining uniform across all three benchmarks as a direct consequence of depth quality differences during SSG construction.

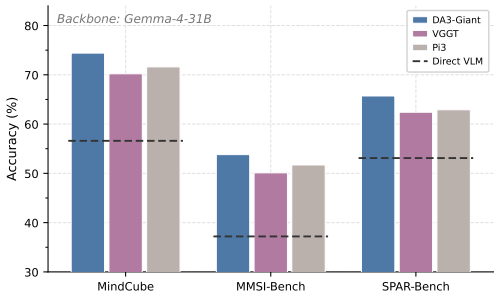


Figure 4: Accuracy under three 3D reconstruction backbones.

5 CONCLUSION

In this work, we present **HIPPOSPATIAL**, a training-free framework for multi-view spatial reasoning that constructs a persistent, metric Spatial Scene Graph from all input views prior to inference. By grounding the representation design in neuroscientific principles of allocentric/egocentric separation and pairwise relational memory, and by delegating all geometric computation to deterministic tools, HIPPOSPATIAL addresses the dominant failure modes of prior tool-augmented agents. Experiments on MINDCUBE, MMSI-BENCH, and SPAR-BENCH show consistent improvements over prior training-free baselines across three backbone VLMs. More broadly, HIPPOSPATIAL suggests that structured, neuroscience-inspired spatial memory is not merely a convenient engineering choice but a principled foundation for spatial reasoning in vision-language agents.

AI USE STATEMENT

Large language models were not used in the design of our approach or experiments. They were used solely for grammar and style editing of this manuscript. All technical decisions, experimental runs, and analyses were conducted by the authors, who bear full responsibility for the content.

ETHICS STATEMENT

This work relies exclusively on publicly released benchmarks (MINDCUBE, MMSI-BENCH, SPAR-BENCH) and open-weight perception models; no new data was collected from human subjects, and no personal or identifiable information is introduced by our method. The indoor scene images underlying these benchmarks originate from their respective public releases and are used under the terms provided by the original dataset authors.

REPRODUCIBILITY STATEMENT

We are committed to making our results reproducible. All model components used to build HIPPOSPATIAL (SAM3, Depth Anything 3, VGGT, Pi3, CLIP, and the open-weight VLM backbones) are publicly released, and the three benchmarks we evaluate on, MINDCUBE, MMSI-BENCH, and SPAR-BENCH, are all publicly accessible. Section 4 and Appendix B document the full experimental setup, including hyperparameters, dataset subsampling procedure, and compute resources, so that our results can be independently verified and reproduced.

REFERENCES

Anthropic. Claude Haiku 4.5. https://www.anthropic.com/news/claude-haiku-4-5, oct 2025. Model release date: October 15, 2025.

Neil Burgess. Spatial memory: how egocentric and allocentric combine. *Trends in cognitive sciences*, 10(12):551–557, 2006.

Patrick Byrne, Suzanna Becker, and Neil Burgess. Remembering the past and imagining the future: a neural model of spatial memory and imagery. *Psychological review*, 114(2):340, 2007.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14455–14465, 2024.

Siyi Chen, Mikaela Angelina Uy, Chan Hee Song, Faisal Ladhak, Adithyavairavan Murali, Qing Qu, Stan Birchfield, Valts Blukis, and Jonathan Tremblay. Spacetools: Tool-augmented spatial reasoning via double interactive rl. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37109–37120, 2026.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024.

Seokju Cho, Ryo Hachiuma, Abhishek Badki, Hang Su, Byung-Kwan Lee, Chan Hee Song, Sifei Liu, Subhashree Radhakrishnan, Seungryong Kim, Yu-Chiang Frank Wang, et al. Spatialclaw: Rethinking action interface for agentic spatial reasoning. *arXiv preprint arXiv:2606.13673*, 2026.

Howard Eichenbaum, Paul Dudchenko, Emma Wood, Matthew Shapiro, and Heikki Tanila. The hippocampus, memory, and place cells: is it spatial memory or a memory space? *Neuron*, 23(2): 209–226, 1999.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Peihao Wang, Huaizhi Qu, Shijie Zhou, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31054–31065, 2026.

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

Google. Gemini 3.6 Flash. https://deepmind.google/models/model-cards/gemini-3-6-flash/, july 2026. Model release date: July 21, 2026.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962. IEEE, 2023.

Wenbo Hu, Jingli Lin, Yilin Long, Yunlong Ran, Lihan Jiang, Yifan Wang, Chenming Zhu, Runsen Xu, Tai Wang, and Jiangmiao Pang. G$ˆ2$vlm: Geometry grounded vision language model with unified 3d reconstruction and spatial reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9535–9546, June 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, et al. Viewspatial-bench: Evaluating multi-perspective spatial localization in vision-language models. In *European Conference on Computer Vision*, pp. 95–111. Springer, 2026.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yirunib8l8.

Zhanpeng Luo, Ce Zhang, Silong Yong, Cunxi Dai, Qianwei Wang, Haoxi Ran, Guanya Shi, Katia P. Sycara, and Yaqi Xie. pyspatial: Generating 3d visual programs for zero-shot spatial reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yv15C8ql24.

Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso De Melo, and Alan Yuille. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6924–6934. IEEE, 2025.

Damiano Marsili, Rohun Agrawal, Yisong Yue, and Georgia Gkioxari. Visual agentic ai for spatial reasoning with a dynamic api. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19446–19455. IEEE, 2025.

John O'keefe and Lynn Nadel. *The hippocampus as a cognitive map*. Oxford university press, 1978.

OpenRouter. Openrouter: A unified interface for large language models. https://openrouter.ai, 2023. Accessed: 2026-09-20.

Yanyuan Qiao, Haodong Hong, Wenqi Lyu, Dong An, Siqi Zhang, Yutong Xie, Xinyu Wang, and Qi Wu. Navbench: Probing multimodal large language models for embodied navigation. *Advances in Neural Information Processing Systems*, 38:95365–95385, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15768–15780. IEEE, 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11854–11864. IEEE, 2023.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Jiajun Wu, Li Fei-Fei, and Manling Li. Mindcube: Spatial mental modeling from limited views. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=0FhrtdKLtD.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=DTQIjngDta.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643. IEEE, 2025.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. MMSI-bench: A benchmark for multi-image spatial intelligence. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=gHRoX4vXm3.

Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Jilin Mei, Chunhui Chen, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *Advances in Neural Information Processing Systems*, 38, 2026.

APPENDIX

A Dataset Details **13**

A.1 MINDCUBE 13

A.2 MMSI-BENCH 14

A.3 SPAR-BENCH 14

B Implementation Details **14**

B.1 Model Components 14

B.2 Hyperparameters 14

B.3 Compute Resources 15

C Tool Set Details **15**

D Prompt Templates **16**

D.1 Object Discovery Prompt 16

D.2 Graph-Guided Reasoning Prompt 16

E Example Spatial Scene Graph **17**

E.1 Allocentric store (objects and relations). 18

E.2 Egocentric store (View Registry). 18

F Algorithmic Details **18**

G Additional Analysis **19**

G.1 Efficiency Analysis 19

G.2 Cross-View Association Sensitivity 20

H Qualitative Examples **21**

A DATASET DETAILS

We evaluate on three multi-view spatial reasoning benchmarks. Following Cho et al. (2026), we cap evaluation at 1,000 samples for benchmarks whose split exceeds that size; otherwise all samples are used. For splits requiring subsampling, we apply stratified sampling with `seed = 42`, preserving the original task distribution.

A.1 MINDCUBE

MINDCUBE (Wang et al., 2026a) evaluates spatial mental modeling under partial observations. Given four orthogonal views of an indoor scene, a model must reason about object positions, orientations, and hypothetical viewpoint changes from perspectives not directly shown. The full benchmark contains 21,154 four-way multiple-choice QA pairs across 3,268 images, organized into 976 multi-view groups drawn from Arkitscenes, WildRGB-D, DL3DV-10K, and a self-collected indoor set.

Questions are partitioned into three settings. **Rotation** (1,081 pairs; 62 image groups) asks models to reason about the scene from a statically rotated observation point. **Among** (18,204 pairs; 733

image groups) requires deducing the lateral arrangement of objects around a central anchor across views. **Around** (1,869 pairs; 180 image groups) tests distance and occlusion reasoning as the camera moves laterally around the scene. We subsample 1,000 questions proportionally by setting and report per-setting accuracy together with the unweighted mean across the three settings.

A.2 MMSI-BENCH

MMSI-BENCH (Yang et al., 2026) targets real-world multi-image spatial intelligence across eleven task categories spanning positional relationships, metric attributes, and object motion. Each item pairs multiple real-world photographs with a question requiring integration of spatial evidence across images. Human accuracy exceeds model accuracy by more than 56 points, establishing it as one of the most demanding probes of cross-image spatial reasoning. The benchmark contains 1,000 questions in total; we evaluate on all of them and report overall accuracy.

A.3 SPAR-BENCH

SPAR-BENCH (Zhang et al., 2026) covers twenty spatial reasoning tasks organized into three difficulty tiers across 7,207 QA pairs. **Low**-difficulty tasks probe fundamental spatial attributes: depth ordering and pairwise distances between an observer and an object (`Depth-OC`, `Dist-OC`) and between two objects (`Depth-OO`, `Dist-OO`). **Medium** tasks require positional matching, camera motion understanding, and view-change inference (`PosMatch`, `CamMotion`, `ViewChgI`). **High**-difficulty tasks demand metric distance inference, cross-view object-relation reasoning, and spatial imagination (`DistI-OO`, `ObjRel-OC`, `ObjRel-OO`, `SpImag-OC`, `SpImag-OO`). Each task additionally has a multi-view variant (suffix `-MV`) supplying images from multiple viewpoints; the nine `-MV` tasks span all three difficulty tiers and constitute a subset of approximately 3,240 questions. We subsample 1,000 questions uniformly across the nine tasks and report overall average accuracy.

B IMPLEMENTATION DETAILS

B.1 MODEL COMPONENTS

Table 2 lists all open-weight model components used in HIPPOSPATIAL.

Table 2: Open-weight model components used in HIPPOSPATIAL.

Role	Model	Source
Segmentation	SAM3	`https://huggingface.co/facebook/sam3`
3D Reconstruction	DA3-Giant	`https://huggingface.co/depth-anything/DA3-GIANT`
	VGGT	`https://huggingface.co/facebook/VGGT-1B`
	Pi3	`https://huggingface.co/yyfz233/Pi3`
Appearance	CLIP	`https://huggingface.co/openai/clip-vit-base-patch32`
VLM backbone	Qwen3.5-9B	`https://huggingface.co/Qwen/Qwen3.5-9B`
	Gemma-4-31B	`https://huggingface.co/google/gemma-4-31B-it`

B.2 HYPERPARAMETERS

Cross-view association (Equation 1) uses three fixed parameters: geometry–appearance trade-off weight $\alpha = 0.7$, distance decay scale $\sigma = 1.0\,\text{m}$, and merge threshold $\tau = 0.6$. The geometric tool-dispatch loop is capped at $T_{\max} = 3$ calls per question. All values are fixed across benchmarks and backbone combinations without per-dataset tuning. VLM inference uses greedy decoding (temperature 0.0) with a maximum of 8192 new tokens per call.

B.3 Compute Resources

All open-weight VLMs are deployed on two NVIDIA H100 80 GB GPUs using the vLLM inference engine (Kwon et al., 2023). Proprietary models are accessed through OpenRouter (OpenRouter, 2023). Scene reconstruction and spatial graph construction run on the same two-GPU node.

C Tool Set Details

HippoSpatial exposes a deterministic tool set $\mathcal{F}$ of five functions that operate directly on the precomputed SSG $\mathcal{G} = (\mathcal{O}, \mathcal{E}, \mathcal{R})$. Every spatial computation is offloaded to these tools; the VLM performs no floating-point arithmetic. Table 3 lists the full signature, inputs, outputs, and purpose of each tool.

Table 3: Deterministic tool set $\mathcal{F}$ used by HippoSpatial.

Tool Name	**Inputs (Type)**	**Output (Type)**	**Description**
Egocentric Spatial Relations			
`query_relation`	`anchor` (*str, optional*) `target` (*str*) `viewpoint` (*int*) `relation` (*str*)	*str*	Returns whether `target` is to the left/right or front/behind of `anchor` as seen from `viewpoint`, by projecting the precomputed direction vector $\hat{\mathbf{u}}_{kk'}$ onto the view-right axis $\hat{\mathbf{w}}_i$ (lateral) or heading $\hat{\mathbf{h}}_i$ (depth). If `anchor` is omitted, the direction is measured from the camera position $\mathbf{p}_i$ of `viewpoint`.
`query_relation_from_hypothetical`	`anchor` (*str, optional*) `target` (*str*) `base_view` (*int*) `delta_rotation` (*float*) `relation` (*str*)	*str*	Answers a counterfactual spatial query: rotates the camera axes of `base_view` by `delta_rotation` degrees in the ground plane (positive = clockwise), then applies the same dot-product projection as `query_relation`. If `anchor` is omitted, the relation is measured from the camera position of `base_view`.
Metric and Ordinal Distance			
`query_distance`	`object_a` (*str*) `object_b` (*str*)	*float* (m)	Retrieves the precomputed Euclidean distance $d_{kk'} = \|\mathbf{x}_k - \mathbf{x}_{k'}\|$ stored in edge $\mathcal{E}_{kk'}$. Because $\mathcal{G}$ is a complete graph over all detected objects, every ordered pair is available regardless of co-visibility.
`query_depth_order`	`objects` (*List[str]*) `viewpoint` (*int*)	*List[str]*	Ranks a list of objects by their projected depth $z_k^{(i)} = (\mathbf{x}_k - \mathbf{p}_i) \cdot \hat{\mathbf{h}}_i$ relative to viewpoint i, returning object labels sorted from nearest to farthest.
Camera Motion			
`query_camera_motion`	`view_a` (*int*) `view_b` (*int*)	*Dict*	Reads two entries from the View Registry $\mathcal{R}$ and returns the translation vector $\Delta\mathbf{p} = \mathbf{p}_b - \mathbf{p}_a$ (metres) and the signed yaw change $\Delta\theta$ (degrees) between the two camera poses.

D PROMPT TEMPLATES

This appendix provides the prompt templates used by the two VLM roles in HIPPOSPATIAL: the Object Discovery prompt, invoked once per query to build the object vocabulary $\mathcal{L}$, and the graph-guided reasoning prompt, which governs Scene Grounding, the tool-dispatch loop, and Answer Generation within a single continuous session.

D.1 OBJECT DISCOVERY PROMPT

```
You are given {n} images of the same scene captured from different
viewpoints, along with a question and its candidate answers.  Identify
every physical object relevant to answering the question.

Question:  {question}
Candidate answers:  {candidate_answers}
Images:  [View 1] ...  [View {n}]

Rules:
1.  Return a short, consistent label for each relevant object (e.g.
``red ball'', not ``the ball that is red in color'').
2.  Use the same label for the same physical object even if its
appearance differs across views.
3.  Do not list objects that are irrelevant to the question.
4.  Output a JSON list of strings, one label per object, with no
duplicates.
```

D.2 GRAPH-GUIDED REASONING PROMPT

The agent receives the following system prompt once per query, which includes the tool documentation, followed by the serialized SSG and the question as the first user turn. The agent's context persists across Scene Grounding, the tool-dispatch loop, and Answer Generation.

You are a spatial reasoning agent. You are given a Spatial Scene Graph $\mathcal{G}$ describing a scene reconstructed from multiple views, and a question about that scene. $\mathcal{G}$ has two parts:

Objects and relations (allocentric, viewpoint-independent):
{"objects": [{"id": "o1", "label": "red ball", "position": [x, y, z], "confidence": c, "seen_in_views": [1, 2]}, ...],
"relations": [{"anchor": "o1", "target": "o2", "distance": d, "direction": [ux, uy, uz]}, ...]}

View registry (egocentric, one entry per input view):
{"views": [{"id": 1, "camera_position": [x, y, z], "heading": [hx, hy, hz], "right": [wx, wy, wz]}, ...]}

Available tools:
query_relation(anchor, target, viewpoint, relation)
query_relation_from_hypothetical(anchor, target, base_view, delta_rotation, relation)
query_distance(object_a, object_b)
query_depth_order(objects, viewpoint)
query_camera_motion(view_a, view_b)

Procedure:
1. **Ground.** Link every object and viewpoint mentioned in the question to an entry in $\mathcal{G}$, stating each grounding explicitly, then decide immediately whether you already have enough evidence to answer or need a tool.
2. If the question only requires existence or counting judgments, answer directly from $\mathcal{G}$ without calling a tool.
3. Otherwise, call at most {T_max} tools, one per turn, choosing the next tool and its arguments based on your groundings and any results already returned. Stop as soon as you have enough evidence to answer.
4. **Answer.** Once you have sufficient evidence, interpret the results in light of the question and output the final answer in the required format (a candidate letter for multiple-choice, a number, or a short free-form phrase).

E EXAMPLE SPATIAL SCENE GRAPH

We illustrate the graph structure produced by Phase 1 on the scene of Figure 1, with 4 views and 6 objects. The serialized graph below follows the format passed to the Phase 2 agent (values rounded for readability), split into the allocentric store (objects and relations, world-frame) and the egocentric store (View Registry, camera-frame), matching the separation defined in Section 3.

E.1 ALLOCENTRIC STORE (OBJECTS AND RELATIONS).


```
{
"objects": [
{ "id": "o1", "label": "red_ball",
"position": [0.00, 0.00, 0.30],
"confidence": 0.91, "seen_in_views": [1, 2, 3, 4] },
{ "id": "o2", "label": "dark_brown_sofa",
"position": [-0.50, -0.20, 0.40],
"confidence": 0.88, "seen_in_views": [1, 2, 3, 4] },
{ "id": "o3", "label": "light_colored_sofa",
"position": [0.90, -0.40, 0.40],
"confidence": 0.93, "seen_in_views": [2, 3, 4] },
{ "id": "o4", "label": "tv_cabinet",
"position": [-0.30, 1.00, 0.60],
"confidence": 0.85, "seen_in_views": [2, 3] },
{ "id": "o5", "label": "white_red_cabinet",
"position": [0.30, 0.90, 0.40],
"confidence": 0.79, "seen_in_views": [1, 2, 3] },
{ "id": "o6", "label": "school_bag",
"position": [-0.30, 1.00, 0.70],
"confidence": 0.82, "seen_in_views": [2, 3] }
],
"relations": [
{ "anchor": "o1", "target": "o2", "distance": 0.55,
"direction": [-0.91, -0.37, 0.18] },
{ "anchor": "o1", "target": "o3", "distance": 0.99,
"direction": [0.91, -0.40, 0.10] },
{ "anchor": "o1", "target": "o4", "distance": 1.09,
"direction": [-0.28, 0.92, 0.28] },
{ "anchor": "o1", "target": "o5", "distance": 0.95,
"direction": [0.31, 0.94, 0.10] },
{ "anchor": "o2", "target": "o3", "distance": 1.41,
"direction": [0.99, -0.14, 0.00] },
{ "anchor": "o4", "target": "o5", "distance": 0.64,
"direction": [0.94, -0.16, -0.31] },
"..." (15 pairs total, one per object pair)
]
}
```


For brevity, we list one entry per unordered pair of objects; the reverse-direction edge $e_{k'k}$ is the negation $-\hat{\mathbf{u}}_{kk'}$ and is omitted here.

E.2 EGOCENTRIC STORE (VIEW REGISTRY).


```
{ "views": [
{ "id": 1, "camera_position": [1.60, 1.20, 1.20],
"heading": [-0.80, -0.60, 0.00], "right": [-0.60, 0.80, 0.00] },
{ "id": 2, "camera_position": [1.60, -1.20, 1.20],
"heading": [-0.80, 0.60, 0.00], "right": [0.60, 0.80, 0.00] },
{ "id": 3, "camera_position": [-1.60, -1.20, 1.20],
"heading": [0.80, 0.60, 0.00], "right": [0.60, -0.80, 0.00] },
{ "id": 4, "camera_position": [-1.60, 1.20, 1.20],
"heading": [0.80, -0.60, 0.00], "right": [-0.60, -0.80, 0.00] }
] }
```


F ALGORITHMIC DETAILS

Algorithm 1 details the construction of the SSG $G = (O, E, R)$ from a set of input views, and Algorithm 2 details graph-guided spatial reasoning given a question Q and the constructed graph.

Algorithm 1 Spatial Scene Graph Construction (Phase 1)

Require: Views $\{I_1, \ldots, I_n\}$, question(s) Q with candidate answers ▷ Q: single question, or union of questions sharing the same views
Ensure: Spatial Scene Graph $G = (O, E, R)$

Object Discovery: query VLM with Q and all views $\{I_i\}$ to obtain object vocabulary $\mathcal{L}$
Metric 3D Reconstruction: $\{(\text{pose}_i, D_i)\}_{i=1}^{n} \leftarrow \text{DEPTHANYTHING3}(\{I_i\}_{i=1}^{n})$ ▷ joint camera poses + depth maps, shared world frame
$\mathcal{S} \leftarrow \emptyset$ ▷ set of observations
for $i = 1, \ldots, n$ **do**
  for $\ell \in \mathcal{L}$ **do**
    $\{(m_{i,j}, c_{i,j})\}_j \leftarrow \text{SAM3}(I_i, \ell)$ ▷ masks and confidences for label ℓ in view i
    for each mask $m_{i,j}$ **do**
      $x_{i,j} \leftarrow$ mean world-frame position of pixels in $m_{i,j}$ ▷ using D_i, pose$_i$
      $\phi_{i,j} \leftarrow \text{CLIP}(\text{crop}(I_i, m_{i,j}))$
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{(i, j, \ell, x_{i,j}, c_{i,j}, \phi_{i,j})\}$
    end for
  end for
end for
Cross-View Association:
for each pair $(i, j), (i', j') \in \mathcal{S}$ with $i \neq i'$ and $\ell_{i,j} = \ell_{i',j'}$ **do**
  $s \leftarrow \alpha \exp\left(-\dfrac{\|x_{i,j} - x_{i',j'}\|_2^2}{\sigma^2}\right) + (1 - \alpha)\cos(\phi_{i,j}, \phi_{i',j'})$
  if $s > \tau$ **then**
    link $(i, j) \leftrightarrow (i', j')$
  end if
end for
$\{S_k\}_{k=1}^{|O|} \leftarrow$ connected components of the linkage graph over $\mathcal{S}$
Graph Construction:
$O \leftarrow \emptyset$; $E \leftarrow \emptyset$
for each component S_k **do**
  $x_k \leftarrow \sum_{(i,j)\in S_k} c_{i,j} x_{i,j} \big/ \sum_{(i,j)\in S_k} c_{i,j}$ ▷ confidence-weighted canonical position
  $O \leftarrow O \cup \{o_k = (\ell_k, x_k, \bar{c}_k, \{i : (i, \cdot) \in S_k\})\}$
end for
for each ordered pair $(o_k, o_{k'})$, $k \neq k'$ **do**
  $d_{kk'} \leftarrow \|x_{k'} - x_k\|_2; \quad \hat{u}_{kk'} \leftarrow (x_{k'} - x_k)/d_{kk'}$
  $E \leftarrow E \cup \{e_{kk'} = (d_{kk'}, \hat{u}_{kk'})\}$
end for
$R \leftarrow \{(\text{pos}_i, \hat{h}_i, \hat{w}_i)\}_{i=1}^{n}$ ▷ View Registry, from pose$_i$
return $G = (O, E, R)$

G Additional Analysis

G.1 Efficiency Analysis

Latency breakdown. We measure per-query latency by executing the full pipeline independently for every query, so that HippoSpatial pays the same reconstruction cost as baselines that also re-derive geometry per query, with no caching across questions that share the same input views. Figure 5 reports the results on the MindCube using the Gemma-4-31B backbone. HippoSpatial's Phase 1, which repeats object discovery, segmentation, and 3D reconstruction for every query under this protocol, accounts for the majority of its total cost (7.80s of 10.20s), comparable in order of magnitude to the reconstruction cost paid by pySpatial, which also re-derives geometry from scratch per query. Phase 2 remains inexpensive regardless of question difficulty (2.40s), bounded by at most $T_{\max} = 3$ tool-dispatch calls. SpatialClaw is the slowest method overall: its iterative loop permits up to $N_{\max} = 30$ code-generation steps per query, and unlike HippoSpatial's deterministic tool calls, each step re-enters the VLM to plan, write, and interpret code before the next action is decided.

Algorithm 2 Graph-Guided Spatial Reasoning (Phase 2)

Require: Question Q, Spatial Scene Graph $G = (O, E, R)$, tool set F, max tool calls $T_{\max}$
Ensure: Answer A

serialize G as structured text, allocentric (O, E) and egocentric R presented separately
Scene Grounding: VLM reads Q and serialized G; links objects mentioned in Q to nodes in O and viewpoints mentioned in Q to entries in R
if Q requires only existence / counting judgments **then**
    $A \leftarrow$ VLM answers directly from G ▷ no tool call needed
else
    $t \leftarrow 0$; $\mathcal{R} \leftarrow \emptyset$ ▷ accumulated tool results
    while $t < T_{\max}$ **and** VLM judges more evidence is needed **do**
        Geometric Tool Call: VLM selects tool $f \in F$ and arguments (anchor, target, viewpoint, relation, ...) grounded in the previous step and $\mathcal{R}$
        $r \leftarrow f(\cdot)$ ▷ e.g., QUERYRELATION, QUERYDISTANCE, QUERYDEPTHORDER, QUERYCAMERAMOTION — see Table 3
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{r\}$; $t \leftarrow t + 1$
    end while
    Answer Generation: VLM interprets $\mathcal{R}$ in light of Q and formats A (multiple-choice, number, or free-form)
end if
return A

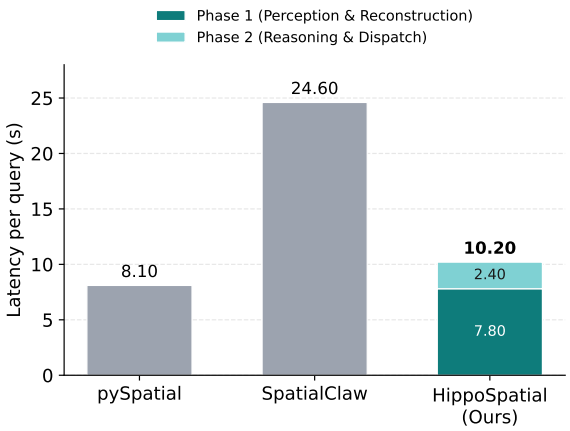


Figure 5: Per-query latency on the MindCube, same backbone and hardware for all methods.

Accuracy versus reasoning cost. Figure 6 relates each method's MindCube accuracy to the average number of VLM calls it issues per query. HIPPOSPATIAL reaches 74.4% accuracy with only 3.5 calls on average, compared to 69.1% for SpatialClaw at 9.7 calls, nearly three times the reasoning cost for a lower score. This gap reflects the two methods' underlying budgets. SpatialClaw's iterative loop permits up to $N_{\max} = 30$ code-generation steps per query (Cho et al., 2026), so harder questions are answered by issuing more calls, whereas HIPPOSPATIAL's tool-dispatch loop is capped at $T_{\max} = 3$, with most of the compositional work already resolved by the precomputed SSG rather than by iterative VLM reasoning. pySpatial issues even fewer calls (2.1) by committing to a complete single-pass program before observing any intermediate result, and this early commitment likely limits its accuracy (66.5%) relative to methods that can act on intermediate evidence. HIPPOSPATIAL is the only method in the upper-left region of the plot, achieving the highest accuracy at a reasoning cost close to the single-pass baseline and far below SpatialClaw's.

G.2 CROSS-VIEW ASSOCIATION SENSITIVITY

We sweep each of the three cross-view association hyperparameters (Section 3, Equation 1) independently, holding the other two at their default value, on the MindCube . Results are in Table 4.

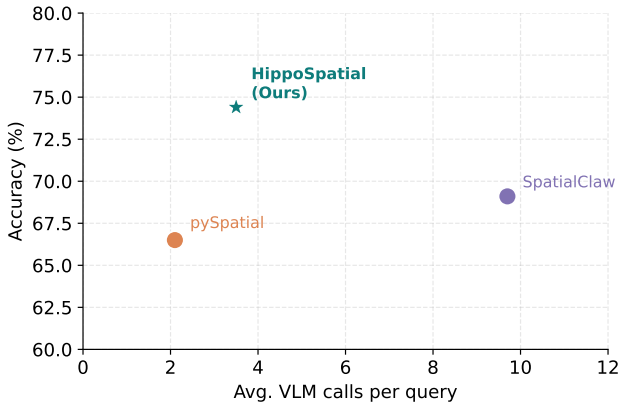


Figure 6: Accuracy vs. average VLM calls per query on MindCube.

Table 4: **Cross-view association sensitivity.** We sweep each hyperparameter independently, holding the other two at their default value (bold), on the MindCube dataset.

Merge threshold τ		Geometry–appearance α		Decay scale σ (m)	
Value	Acc. (%)	Value	Acc. (%)	Value	Acc. (%)
0.3	68.5	0.0	70.1	0.25	70.4
0.4	71.2	0.3	72.0	0.5	72.8
0.5	73.6	0.5	73.5	**1.0**	**74.4**
0.6	**74.4**	**0.7**	**74.4**	2.0	73.2
0.7	73.1	0.9	73.8	4.0	70.9
0.8	69.8	1.0	72.6		

The merge threshold τ is the most sensitive: too low over-merges distinct same-label objects into one node, too high leaves the same object split across several nodes and breaks cross-view relations. $\tau = 0.6$ balances the two.

The geometry-appearance weight α degrades more sharply toward pure appearance ($\alpha = 0$), since CLIP similarity alone cannot separate visually similar instances of the same category; the geometric term resolves this by penalizing spatially distant candidates. $\alpha = 0.7$ keeps geometry dominant while retaining enough appearance signal to tolerate depth noise.

The distance decay scale σ trades off the same two failure modes: too small makes ordinary depth noise push same-object estimates outside the merging range (under-merging), too large flattens the geometric term and shifts merging onto appearance alone (over-merging). $\sigma = 1.0$m matches the typical inter-object spacing in our scenes.

H Qualitative Examples

We trace HippoSpatial end to end on representative MindCube questions.

Example 1: MindCube *Among*, egocentric relation to an object unseen from the queried view

Four images show a black sneaker from the front, left, back, and right, with each camera aligned with the room walls.

Question: From the viewpoint presented in image 2, what is to the **right** of the black sneaker?
Ground truth: C

A. TV **B.** Wooden dining table **C. Light purple sofa** **D.** Brown curtains and windows


View 1 (front)


View 2 (left)


View 3 (back)


View 4 (right)

Phase 1 | Step 1: Object Discovery

Objects: `["black sneaker", "tv", "wooden dining table", "light purple sofa", "brown curtains and windows"]`

Phase 1 | Step 2: Perception and Cross-View Association

The sneaker is segmented in all four views:

Obs.	**View**	$\mathbf{x}_{i,j}$ (m)	$c_{i,j}$
$(1,1)$	1	$[0.04, -0.05, 0.63]$	0.96
$(2,1)$	2	$[0.00, -0.02, 0.61]$	0.95
$(3,1)$	3	$[0.01, -0.01, 0.62]$	0.94
$(4,1)$	4	$[0.03, -0.04, 0.62]$	0.95

Although the sneaker looks different from each side (CLIP similarity 0.86 between views 1 and 2), the geometric term keeps the match: $s = 0.7 \cdot e^{-0.0029} + 0.3 \cdot 0.86 = 0.96 > \tau$. All four observations merge into `o1` at $[0.02, -0.03, 0.62]$. Each background object is detected in a single view and forms its own node.

Phase 1 | Step 3: Spatial Scene Graph

Allocentric store

```
{
"objects": [
{ "id": "o1", "label": "black_sneaker", "position": [0.02,
-0.03, 0.62],
"confidence": 0.95, "seen_in_views": [1, 2, 3, 4] },
{ "id": "o2", "label": "tv", "position": [0.15, 2.60, 1.05],
"confidence": 0.90, "seen_in_views": [1] },
{ "id": "o3", "label": "wooden_dining_table", "position":
[2.45, 0.20, 0.74],
"confidence": 0.84, "seen_in_views": [2] },
{ "id": "o4", "label": "light_purple_sofa", "position": [-0.10,
-2.10, 0.45],
"confidence": 0.88, "seen_in_views": [3] },
{ "id": "o5", "label": "brown_curtains_and_windows", "position":
[-3.05, 0.35, 1.40],
"confidence": 0.81, "seen_in_views": [4] }
],
"relations": [
{ "anchor": "o1", "target": "o2", "distance": 2.67,
"direction": [0.05, 0.99, 0.16] },
{ "anchor": "o1", "target": "o3", "distance": 2.44,
"direction": [0.99, 0.09, 0.05] },
{ "anchor": "o1", "target": "o4", "distance": 2.08,
"direction": [-0.06, -0.99, -0.08] },
{ "anchor": "o1", "target": "o5", "distance": 3.19,
"direction": [-0.96, 0.12, 0.24] },
"..."
]
}
```


Egocentric store (View Registry)


```
{ "views": [
{ "id": 1, "camera_position": [0.00, -0.80, 1.10], "heading":
[0.00, 1.00, 0.00], "right": [1.00, 0.00, 0.00] },
{ "id": 2, "camera_position": [-0.80, 0.00, 1.10], "heading":
[1.00, 0.00, 0.00], "right": [0.00, -1.00, 0.00] },
{ "id": 3, "camera_position": [0.00, 0.80, 1.10], "heading":
[0.00, -1.00, 0.00], "right": [-1.00, 0.00, 0.00] },
{ "id": 4, "camera_position": [0.80, 0.00, 1.10], "heading":
[-1.00, 0.00, 0.00], "right": [0.00, 1.00, 0.00] }
] }
```


Phase 2 | Graph-Guided Spatial Reasoning

Ground. "black sneaker" → `o1` (anchor); "image 2" → view 2; candidates → `o2` (A), `o3` (B), `o4` (C), `o5` (D). Only `o3` appears in view 2, so the relation must be resolved from the allocentric store. The question asks for a viewpoint-dependent direction, so a tool call is required.

Tool call 1 / $T_{\max}$**=3.**
`query_relation(anchor="o1", target="o4", viewpoint=2, relation="left/right")`
→ `"right"` $(\hat{\mathbf{u}}_{o1,o4} \cdot \hat{\mathbf{w}}_2 = 0.99 > 0)$

Stop. Candidate C is confirmed to lie to the right of the anchor from view 2, which is sufficient for a single-answer question.

Answer. **C. Light purple sofa** ✓ **Correct**

Example 2: MINDCUBE *Around*, depth ordering behind an anchor from a queried view

Three images show the same scene from different viewpoints (back, left, and right).

Question: From the viewpoint presented in image 2, what is the **nearest object behind** the dark-green office chair? **Ground truth: A**

A. White round table **B.** Sofa **C.** Black office chair **D.** TV



View 1 (back)



View 2 (left)



View 3 (right)

Phase 1 | Step 1: Object Discovery

Objects: `["dark-green office chair", "white round table", "black office chair", "sofa", "tv"]`

Phase 1 | Step 2: Perception and Cross-View Association

The dark-green chair is segmented in views 1 and 2:

Obs.	View	$\mathbf{x}_{i,j}$ (m)	$c_{i,j}$
$(1, 2)$	1	$[0.64, 0.07, 0.57]$	0.93
$(2, 1)$	2	$[0.60, 0.03, 0.59]$	0.91

View 1 shows the chair from the side and view 2 from the back (CLIP similarity 0.79), yet the two estimates lie 6 cm apart: $s = 0.7 \cdot e^{-0.0036} + 0.3 \cdot 0.79 = 0.93 > \tau$. They merge into `o1` at $[0.62, 0.05, 0.58]$. The black chair is occluded in view 2 and the dark-green chair in view 3, so each chair is localized only from the views in which it is visible.

Phase 1 | Step 3: Spatial Scene Graph

Allocentric store

```
{
"objects": [
{ "id": "o1", "label": "dark_green_office_chair", "position":
[0.62, 0.05, 0.58],
"confidence": 0.92, "seen_in_views": [1, 2] },
{ "id": "o2", "label": "white_round_table", "position": [0.03,
-0.02, 0.62],
"confidence": 0.90, "seen_in_views": [1, 2, 3] },
{ "id": "o3", "label": "black_office_chair", "position":
[-0.58, -0.04, 0.60],
"confidence": 0.93, "seen_in_views": [1, 3] },
{ "id": "o4", "label": "sofa", "position": [-0.45, -2.05,
0.40],
"confidence": 0.87, "seen_in_views": [2, 3] },
{ "id": "o5", "label": "tv", "position": [-3.40, -1.10, 1.60],
"confidence": 0.84, "seen_in_views": [2] }
],
"relations": [
{ "anchor": "o1", "target": "o2", "distance": 0.60,
"direction": [-0.99, -0.12, 0.07] },
{ "anchor": "o1", "target": "o3", "distance": 1.20,
"direction": [-1.00, -0.07, 0.02] },
{ "anchor": "o1", "target": "o4", "distance": 2.36,
"direction": [-0.45, -0.89, -0.08] },
{ "anchor": "o1", "target": "o5", "distance": 4.30,
"direction": [-0.93, -0.27, 0.24] },
"..."
]
}
```


Egocentric store (View Registry)


```
{ "views": [
{ "id": 1, "camera_position": [0.00, -1.80, 1.00], "heading":
[0.00, 1.00, 0.00], "right": [1.00, 0.00, 0.00] },
{ "id": 2, "camera_position": [2.00, 0.00, 1.00], "heading":
[-1.00, 0.00, 0.00], "right": [0.00, 1.00, 0.00] },
{ "id": 3, "camera_position": [-2.00, 0.00, 1.00], "heading":
[1.00, 0.00, 0.00], "right": [0.00, -1.00, 0.00] }
] }
```


Phase 2 | Graph-Guided Spatial Reasoning

Ground. "dark-green office chair" → `o1` (anchor); "image 2" → view 2; candidates → `o2` (A), `o4` (B), `o3` (C), `o5` (D). The black office chair (`o3`) is occluded in view 2 and is resolved from the allocentric store. "Nearest behind" requires ordering objects by depth along the heading of view 2, so a tool call is required.

Tool call 1 / $T_{\max}$**=3.**
`query_depth_order(objects=["o1", "o2", "o3", "o4", "o5"], viewpoint=2)`
→ `["o1", "o2", "o4", "o3", "o5"]` ($z^{(2)}$: 1.38, 1.97, 2.45, 2.58, 5.40 m)

Stop. The anchor `o1` is nearest to the camera; the next object in depth order, `o2`, is the nearest object behind it.

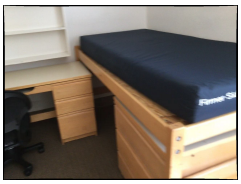
Answer. **A. White round table** **✓ Correct**

Example 3: MINDCUBE *Rotation*, hypothetical turn from a fixed position

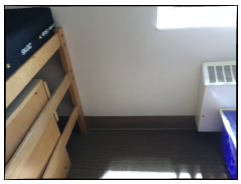
Three images are taken from the same spot: image 2 after turning the camera 90° right from image 1, and image 3 after another 90° right, facing opposite to image 1.

Question: If I am standing at the same spot and facing the same direction as in image 1, then turn **90° to the left**, what is to my **left**? **Ground truth: C**

A. Bed **B.** Window **C. Study table and black chair**



View 1



View 2 (+90°)



View 3 (+180°)

Phase 1 | Step 1: Object Discovery

Objects: `["bed", "window", "study table and black chair", "desk with shelves"]`

Phase 1 | Step 2: Perception and Cross-View Association

The three views share one camera centre and barely overlap; Depth Anything 3 still recovers a common frame with only the heading changing between views. The bed is segmented in views 1 and 2:

Obs.	View	$\mathbf{x}_{i,j}$ (m)	$c_{i,j}$
$(1,1)$	1	$[0.92,\ 1.72,\ 0.86]$	0.94
$(2,1)$	2	$[0.88,\ 1.68,\ 0.84]$	0.90

View 2 shows only the foot of the bed (CLIP similarity 0.72), but the estimates nearly coincide: $s = 0.7 \cdot e^{-0.0036} + 0.3 \cdot 0.72 = 0.91 > \tau$, merging into $\circ 1$ at $[0.90,\ 1.70,\ 0.85]$. The desk in view 1 and the study table in view 3 are two distinct pieces of furniture 3.0 m apart and remain separate nodes.

Phase 1 | Step 3: Spatial Scene Graph

Allocentric store

```
{
"objects": [
{ "id": "o1", "label": "bed", "position": [0.90, 1.70, 0.85],
"confidence": 0.92, "seen_in_views": [1, 2] },
{ "id": "o2", "label": "window", "position": [1.60, -1.45,
1.35],
"confidence": 0.86, "seen_in_views": [2, 3] },
{ "id": "o3", "label": "desk_with_shelves", "position": [-0.95,
1.35, 0.60],
"confidence": 0.89, "seen_in_views": [1] },
{ "id": "o4", "label": "study_table_and_black_chair", "position":
[-0.35, -1.60, 0.60],
"confidence": 0.91, "seen_in_views": [3] }
],
"relations": [
{ "anchor": "o1", "target": "o2", "distance": 3.27,
"direction": [0.21, -0.96, 0.15] },
{ "anchor": "o1", "target": "o4", "distance": 3.54,
"direction": [-0.35, -0.93, -0.07] },
{ "anchor": "o2", "target": "o4", "distance": 2.10,
"direction": [-0.93, -0.07, -0.36] },
{ "anchor": "o3", "target": "o4", "distance": 3.01,
"direction": [0.20, -0.98, 0.00] },
"..."
]
}
```


Egocentric store (View Registry)


```
{ "views": [
{ "id": 1, "camera_position": [0.00, 0.00, 1.20], "heading":
[0.00, 1.00, 0.00], "right": [1.00, 0.00, 0.00] },
{ "id": 2, "camera_position": [0.00, 0.00, 1.20], "heading":
[1.00, 0.00, 0.00], "right": [0.00, -1.00, 0.00] },
{ "id": 3, "camera_position": [0.00, 0.00, 1.20], "heading":
[0.00, -1.00, 0.00], "right": [-1.00, 0.00, 0.00] }
] }
```


Phase 2 | Graph-Guided Spatial Reasoning

Ground. No object anchor: the question is observer-centric, so relations are measured from the camera of base view 1; "turn 90° to the left" → `delta_rotation` $= -90$; candidates → `o1` (A), `o2` (B), `o4` (C). After the turn, the observer's left coincides with the heading of view 3, where both `o2` and `o4` were observed, while `o1` was seen only in views 1 and 2. Both remaining candidates therefore lie on the same side, so the hypothetical pose is checked for each.

Tool call 1 / $T_{\max}$**=3.**

```
query_relation_from_hypothetical(target="o4", base_view=1,
                                 delta_rotation=-90,
relation="left/right")
```

→ `"left"` (lateral component -0.92)

Tool call 2 / $T_{\max}$**=3.**

```
query_relation_from_hypothetical(target="o2", base_view=1,
                                 delta_rotation=-90,
relation="left/right")
```

→ `"left"` (lateral component -0.67)

Tool call 3 / $T_{\max}$**=3.**

```
query_relation_from_hypothetical(target="o2", base_view=1,
```

```
                                    delta_rotation=-90,
relation="front/behind")
→ "behind"
```
(heading component -0.74)

Stop. The window lies mostly behind the rotated observer ($|-0.74| > |-0.67|$), whereas the study table lies almost directly to the left.

Answer. **C. Study table and black chair** **✓ Correct**