SPACEMCP: VISION LANGUAGE MODEL SPATIAL REASONING VIA MODEL CONTEXT PROTOCOL

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite impressive advances in large language models (LLM), vision–language models (VLM) still exhibit limited spatial reasoning capabilities. Recent progress in feed-forward 3D reconstruction has enabled precise, real-time geometric understanding from visual inputs. In this work, we propose SpaceMCP, a framework that leverages these advances to introduce a Model Context Protocol (MCP) style paradigm for spatial reasoning. Our approach augments a VLM with structured function calls, allowing it to query a scene graph for spatial information such as distances, object positioning, and camera properties. The scene graph is constructed in real time using feed-forward 3D reconstruction to recover metric geometry, combined with open-vocabulary segmentation to identify and localize relevant objects within the scene. Paired with VLM's strong reasoning capabilities, the VLM is able to combine this primitive information to solve more complex spatial tasks. Across nine VLMs and without any training, this approach improves scores by an average of 16 points on VSI-Bench, 6 on MMSI-Bench and 14 on VSTI-Bench, and with matched backbones it outperforms the strongest training-free tool-using agent on VSI-Bench. Gains are largest on metric and camera-geometry questions, while questions that hinge on viewpoint conventions, temporal order or fine measurements remain hard. Given the rapid advances in feed-forward 3D computer vision, we propose this paradigm as an augmentation to VLM spatial reasoning rather than a replacement: the vision models contribute precise metric answers and the VLM context and nuance, and improving native VLM spatial reasoning remains necessary.

1 INTRODUCTION

Spatial reasoning is a cornerstone of general intelligence and a prerequisite for deploying autonomous agents in the physical world, from robot navigation and manipulation to AR/VR (Cheng et al., 2025; Gu et al., 2024). Yet a stark asymmetry has emerged in modern foundation models: while large language models now match or exceed human performance on broad knowledge and reasoning benchmarks — frontier models score 94.6% on GPQA Diamond (Rein et al., 2024; Anthropic, 2026) against roughly 70% for PhD-domain experts (OpenAI, 2024) — vision-language models (VLMs) lag dramatically when the same competence is required in 3D space. On MMSI-Bench (Yang et al., 2026b), GPT-5 reaches only 41.9%, and SenseNova-SI (Cai et al., 2026) — an 8B VLM fine-tuned on eight million spatial examples — attains 43.3%, against 97.2% for humans; comparable gaps persist on VSI-Bench (Yang et al., 2025a), ViewSpatial-Bench (Li et al., 2026a), and SPACE (Ramakrishnan et al., 2025). Earlier large-scale efforts paint the same picture: SpatialVLM (Chen et al., 2024) fine-tunes a VLM on millions of synthetic spatial-QA examples and lifts qualitative spatial QA over its base model, but only 37% of its metric estimates fall within a factor of two of the truth. The shortfall is widely attributed to a training distribution dominated by 2D web imagery with limited complex spatial supervision (Ma et al., 2025a;b).

Existing efforts to close this gap broadly fall into three families (Zheng et al., 2025). *Post-training* on spatial-specific data (Chen et al., 2024; Cheng et al., 2024; Daxberger et al., 2025; Batra et al., 2025; Li et al., 2026b; Cai et al., 2026; Lian et al., 2026) teaches a VLM to internalise spatial facts from curated corpora. *Architectural integration* of a 3D expert (Fan et al., 2026; Zhao et al., 2026; Wu et al., 2025a; Yu et al., 2025; Xu et al., 2026; Cheng et al., 2026; Hong et al., 2023) fuses geometric features from a 3D backbone into the VLM at training time. *Test-time scaling and tool-augmented*

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.

agentic methods (Yang et al., 2025b; Marsili et al., 2025; Han et al., 2026; Guo et al., 2026; Cao et al., 2026; Cai et al., 2025; Wan et al., 2026; Li et al., 2025) route inputs through external tools or expanded inference loops at test time.

Test-time scaling is the closest in spirit to our work, but existing instances are typically tied to a single question type. SpatialBot (Cai et al., 2025) exposes a depth-query API, which solves point-distance questions but says nothing about object counts, room layouts, or ego-perspective bearings; SpatialDreamer (Cao et al., 2026) trains a VLM to imagine alternative camera viewpoints with a novel-view synthesis model; MindJourney (Yang et al., 2025b) runs a world model in an iterative reasoning loop optimised for view-change reasoning; TIGeR (Han et al., 2026) wires classical geometric tools into a VLM agent for tasks that reduce to those primitives; MSSR (Guo et al., 2026) curates a per-question minimal sufficient set of 3D perception results. Each bakes a particular tool or template into the agent, which helps on its target distribution but often not on adjacent question types, and several run generative novel-view synthesis in their inner loop, at inference costs far above direct prompting.

Two recent enabling advances make a different, simpler design viable. First, feed-forward 3D reconstructors such as VGGT (Wang et al., 2025a), which builds on DUSt3R (Wang et al., 2024), now produce dense, geometrically consistent point clouds from monocular video in a single forward pass, with some recent variants — notably Depth Anything 3 (Lin et al., 2026) and MapAnything (Keetha et al., 2026) — emitting depth in metric units directly. Second, open-vocabulary segmenters such as the most recent SAM 3 in the SAM lineage (Kirillov et al., 2023; Carion et al., 2026) let us promote those reconstructions to typed scene-graph nodes via short noun-phrase prompts, removing the need for a closed object vocabulary. Together, they convert what was previously a brittle multi-stage pipeline into a few seconds of deterministic preprocessing, and produce an accurate scene graph stable enough to be queried by downstream tools without re-deriving geometry per question.

Building on these advances, we present **SpaceMCP**, a test-time scaling framework that exposes a metric scene graph to the VLM through a generic Model Context Protocol (MCP) interface (Anthropic, 2024). From an indoor video, SpaceMCP reconstructs the scene with Depth Anything 3 (Lin et al., 2026), supplying metric depth and camera extrinsics in a single forward pass; SAM 3 provides language-promptable segmentation, and a 2 cm occupancy grid yields the room layout. The scene graph is populated *statically* on receipt of the video — room footprint, floor plane, and metric point cloud — and then *dynamically*, with object instances added at the VLM's request via two augmentation tools (`add_object`, `add_objects`). On top of this, read-only metric getters expose room size, object dimensions, pairwise distances, relative direction, frame-of-first-appearance, and a full camera-pose API for ego-perspective and inter-frame camera-motion questions. At inference, the VLM emits a single MCP script combining augmentations and the subsequent getter calls in one message; the MCP response is then used by the VLM to provide the answer in the requested format. A final answer takes one plan turn and one answer turn over a scene graph built in seconds.

The crucial design choice is what we expose to the VLM. Rather than introduce a tool engineered around one question type — a novel-view renderer, a depth probe, or a world-model rollout — we expose a small set of *primitive* metric queries (object positions and sizes, pairwise distances, room footprint, relative bearings, camera poses) and let the VLM compose them. Counting, size estimation, absolute distance, room dimensions, ego-perspective "standing-at-X facing-Y" bearings, and inter-frame camera motion all reduce to short call sequences over the same fixed surface, and as feed-forward 3D vision matures, new primitives (e.g. orientation, trajectory) can be added once and benefit every question that decomposes into them.

SpaceMCP is meant to augment a VLM's spatial reasoning, not to replace it. The vision models give exact metric answers — distances, sizes, bearings, camera motion — that a VLM cannot reliably read off pixels, while the VLM brings the context and nuance to interpret the question, decide what to measure and judge whether a returned value is plausible. Accordingly, SpaceMCP helps most on metric and camera-geometry questions and helps little or hurts where perception misfires or the answer turns on conventions, temporal order or multi-step reasoning, such as left/right relations, appearance order and measurement attributes (Sec. 4.2); improving the underlying VLMs remains necessary.

Our contributions are:

- **A general MCP interface for spatial reasoning.** Unlike prior test-time scaling work that engineers a separate tool per question type (novel-view rendering, depth probes, world-model rollouts), we expose a small set of metric primitives over a real-time scene graph and show a single fixed surface covers counting, size, distance, room layout, ego-perspective and inter-frame camera-motion questions without per-task adaptation.
- **A static and dynamic scene-graph generation approach.** We design a two-phase scene-graph generation procedure that distils video-level information statically on receipt (room size, floor plane, metric point cloud) and then expands it dynamically via VLM-driven augmentation calls (object segmentation), allowing the agent to scaffold queries against a uniform tool surface across question types.
- **Consistent gains across nine VLMs.** Without any training, SpaceMCP improves every backbone that can plan on VSI-Bench (Yang et al., 2025a) and VSTI-Bench (Fan et al., 2026) and seven of eight on MMSI-Bench (Yang et al., 2026b), and with matched backbones it outperforms the strongest tool-using method on VSI-Bench. Gains are uneven across question types, and we report where SpaceMCP falls short.

2 Related Work

Spatial reasoning benchmarks for VLMs. VSI-Bench (Yang et al., 2025a), ViewSpatial-Bench (Li et al., 2026a), and SPACE (Ramakrishnan et al., 2025) probe video-based, multi-perspective, and broader spatial cognition; 3DSRBench (Ma et al., 2025a), MMSI-Bench (Yang et al., 2026b), and SpaceVista (Sun et al., 2026) extend the diagnosis to 3D-style queries, multi-image reasoning, and millimetre-to-kilometre scales. The shared finding is that VLMs are weak across a broad range of spatial capabilities — metric estimation, viewpoint reasoning, layout understanding, and ego-perspective grounding.

Post-training for spatial reasoning. A large body of work pushes spatial competence into the model itself. SpatialVLM (Chen et al., 2024), SpatialRGPT (Cheng et al., 2024), and MM-Spatial (Daxberger et al., 2025) curate synthetic spatial-QA datasets from depth, segmentation, and 3D boxes; SpatialThinker (Batra et al., 2025), SpaceR (Ouyang et al., 2025), and Spatial-SSRL (Liu et al., 2026) layer reinforcement learning on top with dense, video-based, and self-supervised rewards respectively; SpatialLadder (Li et al., 2026b) trains progressively from perception to reasoning; SenseNova-SI (Cai et al., 2026) scales the supervised recipe to an 8M-example corpus; and Cambrian-S (Yang et al., 2026a) scales spatial video data and adds a surprise-driven memory for long video. These methods pay the data and compute cost of training, but they improve the VLM itself and are complementary to test-time tools such as ours.

Architectural 3D-expert integration. A third family builds 3D information into the VLM's architecture at training time. VLM-3R (Fan et al., 2026) fuses CUT3R 3D tokens via cross-attention; SpaceMind (Zhao et al., 2026) adds a camera-aware fusion module; Spatial-MLLM (Wu et al., 2025a) aligns a 3D model with the VLM encoder; SpatialLLM (Ma et al., 2025b) bakes in 3D priors through architectural changes and 3D-informed data. All require retraining or fine-tuning.

Test-time scaling and tool-augmented agents. A first family gives the VLM extra views, tools, or compute at inference. AVIC (Yu et al., 2026) gates novel-view synthesis on whether a view change is worth its cost. ViLaSR (Wu et al., 2025b) interleaves reasoning with drawing on the image (boxes, auxiliary lines). TIGeR (Han et al., 2026) integrates classical geometric tools into a VLM agent, and MSSR (Guo et al., 2026) runs a dual-agent perception–reasoning loop that curates a minimal sufficient set of 3D perception results per query. Closest to SpaceMCP are agents that call 3D perception tools directly: pySpatial (Luo et al., 2026) writes programs over a reconstruction and a novel-view renderer, S-Agent (Dai et al., 2026) and GCA (Chen et al., 2026b) drive detectors and metric depth from a VLM planner, SpatialClaw (Cho et al., 2026) runs a code agent with SAM 3 and DA3 for up to 30 turns per question, far from real time, and SpaceTools (Chen et al., 2026a) trains a 3B VLM to call such tools. A further variant passes 3D facts in plain text rather than tool calls: SpatialPIN (Ma et al., 2024) encodes object centres, dimensions, and relations as in-context strings, and Struct2D (Zhu et al., 2025) renders a query-relevant BEV together with a few egocentric keyframes.

Programmatic approaches. A second family casts visual reasoning as program synthesis: the LLM emits a Python program that calls vision sub-modules and combines their outputs. VisProg (Gupta and Kembhavi, 2023) and ViperGPT (Surís et al., 2023) introduced this for compositional 2D image QA; VADAR (Marsili et al., 2025) extends it by having LLM agents grow a shared Pythonic API from a batch of queries. These works are mostly tied to 2D imagery or operate over isolated frames with no global metric reconstruction, are *stateless* across queries, and largely predate recent feed-forward 3D reconstruction (MASt3R (Leroy et al., 2024), VGGT (Wang et al., 2025a), DA3 (Lin et al., 2026)) and open-vocabulary segmentation (SAM 3 (Carion et al., 2026)) that make a stateful metric scene-graph treatment viable.

3 METHOD

We are given an indoor video $V = (I_1, \ldots, I_T)$ and a natural-language question q. SpaceMCP builds the scene graph *statically* from V alone (Sec. 3.1), and then expands it *dynamically* once the VLM sees q (Sec. 3.2); both expose their state through a single MCP interface (Sec. 3.3). Figure 2 summarises the pipeline.

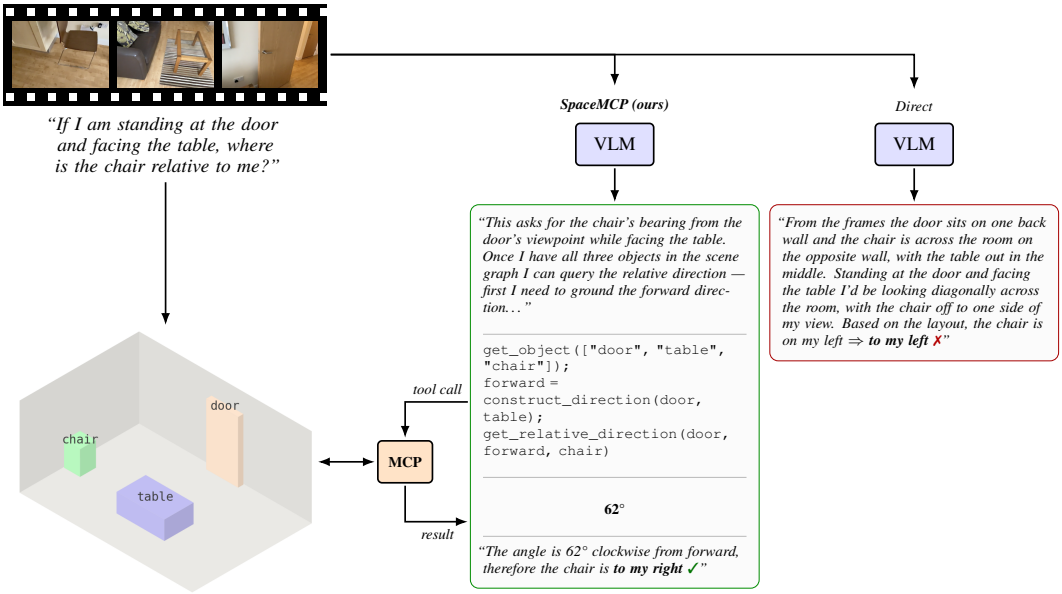


Figure 1: **Inside the VLM's head: SpaceMCP swaps guessing for grounding.** An illustrative ego-perspective question over an indoor video. Direct VLM *(right)* narrates uncertainty and guesses. SpaceMCP *(middle)* routes a short tool-call sequence through MCP, lifts a metric answer (62°) out of the scene graph, and converts it into the cardinal answer.

3.1 STATIC GENERATION

The static generation runs once per video and produces every artefact that is independent of the question: a metric world-frame point cloud, the floor plane and a 2D room frame.

SAM 3's vision tower and text-conditioned grounding head are decoupled. We run the vision tower on each of the frames during the static generation, and keep its activations in memory. Therefore, frames are only encoded once, and per-question `add_object` calls (Sec. 3.2) only invoke the much cheaper grounding head against those activations.

We sample V at 1 fps (reducing sampling for videos over 200s) and feed the frames to Depth Anything 3 (Lin et al., 2026), a feed-forward 3D transformer that returns *metric* per-pixel depth, per-frame camera intrinsics and extrinsics, and a confidence map in a single forward pass. We unproject high-confidence pixels into a fused world-frame point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$ in metres.

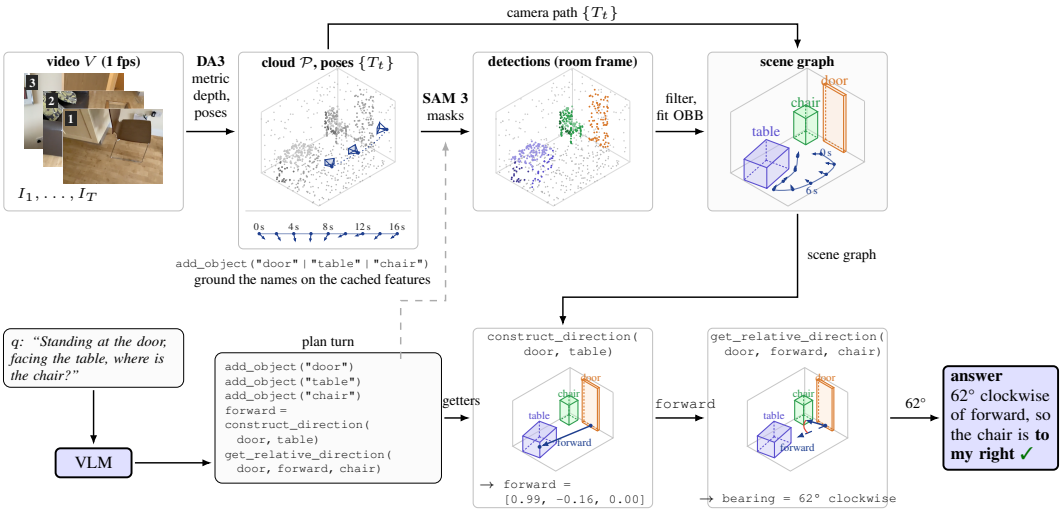


Figure 2: **SpaceMCP overview.** *Top:* DA3 lifts the 1 fps frames to a metric point cloud with per-frame camera poses; SAM 3 grounds the object names in the plan on the cached features, the masked points are filtered and fitted with OBBs, and the boxes and the camera path form the scene graph. *Bottom:* for question q the VLM emits one plan of `add_object` calls and getters; the getters run against the scene graph, here `construct_direction` followed by `get_relative_direction` on its result, and the VLM turns the returned bearing into the answer in a second turn. Clouds and boxes are schematic.

To put $\mathcal{P}$ in a 2D room frame, we run SAM 3 with the prompt "floor" over the cached vision features, project the masked pixels into world coordinates, and fit a robust plane Π via RANSAC (Fischler and Bolles, 1981). Let $\hat{n}$ be the plane normal and c its centroid; we construct an orthonormal basis with $\hat{n}$ as the up axis and rotate $\mathcal{P}$ so that the floor lies in the XY plane:

$$p_{\text{room}} = R_{\text{room}}(p - c), \qquad R_{\text{room}} = \begin{bmatrix} \hat{u}^\top \\ \hat{v}^\top \\ \hat{n}^\top \end{bmatrix}. \tag{1}$$

We then build a 2 cm occupancy grid over XY from the room-frame points, marking cells with at least 10 points as occupied. A weighted-IoU rectangle is fit to the occupied cells, with its rotation swept over $\pm 90^\circ$ around the AABB orientation; the result is the four metric corners $\mathcal{C} \in \mathbb{R}^{4\times 2}$ and the room area A_{room}.

End-to-end, the static generation runs in 8.34 s on an Nvidia B200 for a 60-second indoor clip — about $7.2\times$ faster than real time.

3.2 Dynamic Querying

Once q is known, the VLM names the objects it needs to reason about and the MCP scene-graph executes the corresponding `add_object` / `add_objects` calls (Sec. 3.3). For each call we run SAM 3's text-conditioned grounding head against the cached vision-encoder activations from Sec. 3.1 using the requested noun phrase as prompt, with a confidence cascade ($0.85 \to 0.7 \to 0.5 \to 0.4 \to 0.3$) that falls back to lower thresholds when the higher-confidence pass returns no points. Pixels above the chosen threshold are unprojected via the cached DA3 depth, filtered by DA3 depth confidence (top 85$^{\text{th}}$ percentile per detection), and rotated and scaled into the metric room frame. `add_object` fits a single oriented bounding box (OBB) over the union of all per-frame masks; `add_objects` preserves SAM 3's per-frame per-instance output and resolves cross-frame correspondence with a greedy mutex clustering (Appx. B) that respects the fact that two detections from the same frame must belong to different physical objects, then fits an OBB per resulting cluster. The resulting instances enter the scene graph under the noun phrase as label.

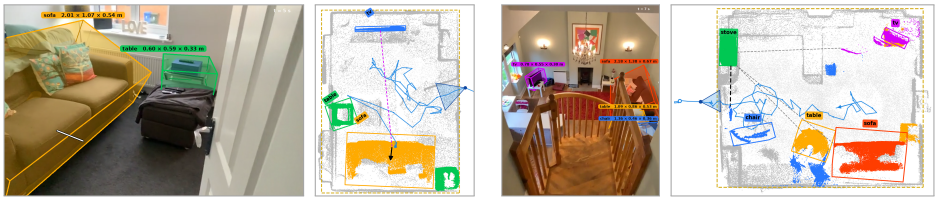


Figure 3: **Scene-graph outputs** for two videos. Each pair shows a frame with the recovered OBBs and their metric extents, then the room-frame map with the point cloud, room rectangle, object footprints, camera path and the geometry returned for the question.

Because only the grounding head runs here, a four-prompt augmentation completes in 6.50 s; subsequent getter calls (Sec. 3.3) read from the resulting graph and never touch the depth or vision backbones again.

3.3 The MCP scene-graph interface

We expose the scene graph to the VLM via a Model Context Protocol (Anthropic, 2024) server. The interface has two augmentation tools and a read-only query API split into three groups: *static room geometry*, *object geometry*, and *camera and pose operations* (Table 3 in Appx. A); Fig. 3 shows the resulting graph for two videos. Augmentation tools trigger the dynamic-querying procedure of Sec. 3.2 and add the resulting instances into the graph; each instance stores both an oriented bounding box and the underlying segmented point cloud. The OBB powers coarse geometric queries (size, dimensions, volume), while the point cloud powers precise calculations such as `get_distance`, which returns the closest-pair distance via a nearest-neighbour search across the two clouds. Pose calls return a JSON dict containing the camera `position_m` and heading. Variables are supported e.g. feeding a camera position from `get_camera_pose` into `get_relative_direction` — so a single plan can compose arbitrary call sequences without an extra round trip.

Given q, the VLM is shown the current scene-graph state (room footprint and labels already in the graph) together with the API documentation, and emits a single JSON plan that combines the augmentation calls naming the targets it needs with the getter calls that return the values used to answer q. The plan is executed against the scene graph and the model is then shown the actual returned values in a second *answer turn* that asks for the answer in the question's expected unit; that answer is the final prediction. The API documentation instructs the agent to use `add_object` for size / dimension / distance / direction / order questions and `add_objects` for counting, with the exact noun phrase from the question. If the plan returns no augmentation calls at all, an LLM-driven *target extractor* pass — a single short prompt focused only on identifying the named objects in q — supplies the missing calls.

Direct-answer fallback. Alongside the plan, the VLM also produces a direct answer to q from V. We treat this as a fallback: when the scene-graph plan fails outright — the model misuses the API, or SAM 3 returns no instances for any of the requested objects — we report the direct answer instead.

4 Experiments

4.1 Setup

Benchmarks. We evaluate on three video / multi-image spatial reasoning benchmarks. *VSI-Bench* (Yang et al., 2025a), on its released debiased split (v1; Brown et al., 2026), has ten categories over indoor videos: appearance order, relative direction (easy, medium, hard), relative distance, route planning, absolute distance, counting, object size and room size. *MMSI-Bench* (Yang et al., 2026b) has eleven categories of multi-image questions over scenes captured from disjoint viewpoints: camera and object motion, six pairwise positional relations between cameras, objects and regions, appearance and measurement attributes, and multi-step reasoning. *VSTI-Bench* (Fan et al., 2026) has nine categories of spatio-temporal questions about a moving camera: camera displacement and movement

Table 1: **SpaceMCP on nine VLM backbones.** Benchmark score (%) of each model alone (Base) and inside SpaceMCP (Ours), and the gain Δ with its 95% paired-bootstrap interval (VSTI-Bench: Δ only); the higher of each pair is in bold. Rows are sorted by the SpaceMCP mean over the three benchmarks.

	VSI-Bench (debiased)			MMSI-Bench			VSTI-Bench			Mean	
Model	Base	Ours	Δ [95% CI]	Base	Ours	Δ [95% CI]	Base	Ours	Δ	Base	Ours
Muse-Spark-1.3	53	**64**	$+11$ $[+6, +16]$	**52**	51	-1 $[-6, +4]$	72	**84**	$+13$	59	**66**
GPT-6-Luna	42	**61**	$+19$ $[+14, +23]$	38	**51**	$+13$ $[+7, +17]$	67	**84**	$+17$	49	**65**
GLM-5.3-Flash	48	**62**	$+14$ $[+10, +19]$	46	**49**	$+3$ $[-2, +8]$	69	**85**	$+16$	54	**65**
Qwen3.5-27B	49	**63**	$+14$ $[+9, +18]$	41	**46**	$+5$ $[-1, +10]$	72	**85**	$+13$	54	**64**
Gemma-4-31B	42	**61**	$+19$ $[+14, +24]$	36	**43**	$+7$ $[+2, +12]$	66	**82**	$+16$	48	**62**
Gemini-3.5-Flash-Lite	53	**63**	$+10$ $[+5, +14]$	35	**40**	$+5$ $[+0, +11]$	71	**83**	$+12$	53	**62**
Gemma-4-26B-A4B	31	**60**	$+28$ $[+24, +33]$	35	**44**	$+10$ $[+5, +14]$	63	**82**	$+19$	43	**62**
DeepSeek-V4.1-Flash	37	**61**	$+24$ $[+19, +28]$	29	**43**	$+13$ $[+8, +18]$	58	**82**	$+24$	41	**62**
Qwen2.5-Omni-7B	30	**33**	$+3$ $[-2, +8]$	**31**	30	-1 $[-6, +3]$	**52**	44	-8	**38**	36

direction, absolute and relative camera–object distance, and object–object position (left/right, near/far, up/down).

Protocol. Every question is answered twice by each model. *Base* is the model alone, given the benchmark media: MMSI-Bench's images; for VSTI-Bench, the M frames the question indexes ("frame N of M"); for VSI-Bench, 16 uniformly spaced frames (native video for Gemini). *SpaceMCP* is the full pipeline of Sec. 3: the model plans calls against the scene graph and answers from the returned values. Both SpaceMCP turns are text-only, and when the tools return nothing usable the pipeline falls back to the base answer. On VSTI-Bench's left/right questions the plan prompt uses a revised version of one in-context example (Appx. C); all other answers use the original prompt.

Metrics. Multiple-choice questions are scored by accuracy and numeric questions by Mean Relative Accuracy,

$$\mathrm{MRA} = \frac{1}{10} \sum_{\theta \in \{0.5, 0.55, \ldots, 0.95\}} \mathbf{1}\left(\frac{|\hat{y} - y|}{y} < 1 - \theta \right),$$

which credits a prediction $\hat{y}$ at each of ten relative-error thresholds. A benchmark score is the unweighted mean over its categories, and an answer with no letter or number scores 0. We report Δ = SpaceMCP $-$ base with a 95% paired-bootstrap interval over questions, resampled within category.

Models. We run nine VLMs, proprietary and open-weight: GPT-6-Luna, Muse-Spark-1.3, GLM-5.3-Flash, Qwen3.5-27B, Gemma-4-26B-A4B, DeepSeek-V4.1-Flash, Gemma-4-31B, Gemini-3.5-Flash-Lite and Qwen2.5-Omni-7B, each at a high reasoning setting where one is offered (Appx. C).

4.2 Gains over the base model

SpaceMCP improves all eight backbones that produce usable plans on VSI-Bench and VSTI-Bench, and seven of them on MMSI-Bench (Table 1); the ninth, Qwen2.5-Omni-7B, is discussed under Limitations. Averaged over the nine models the gain is $+16$ points on VSI-Bench, $+6$ on MMSI-Bench and $+14$ on VSTI-Bench; over the eight planners it is $+17$, $+7$ and $+16$. Scoring the tool answers alone, without the direct-answer fallback, changes no score by more than three points (Table 10), so the gain comes from the scene graph rather than the fallback.

SpaceMCP levels the backbones. The weaker the base model, the larger the gain, and the eight planners converge: from 31–53 to 60–64 on VSI-Bench and from 58–72 to 82–85 on VSTI-Bench, with the standard deviation across backbones shrinking from 7.3 to 1.3 and from 4.6 to 1.2 points. Once geometry comes from the scene graph, the choice of VLM matters less on these two benchmarks but still matters on MMSI-Bench: Gemma-4-26B-A4B, the weakest planner on VSI-Bench without tools (31), reaches 60 / 44 / 82 with them, against 61 / 51 / 84 for GPT-6-Luna. Much of Gemma-4-26B-A4B's low base score comes from refusals: without tools it declines to give a number on 54% of the size and distance questions, usually saying it lacks a scale, and those answers score 0.

Table 2: **Comparison with reported results** (%), with the gain over the same backbone without the method in parentheses where the source reports it. SpaceMCP rows apply each benchmark's official weighting (Table 9); other rows are the numbers reported in each paper. Published VSTI-Bench numbers predate its July 2026 label correction, which changed five of its nine categories; its authors caution that results on old and new labels are not comparable. [a]Re-run by SpatialClaw with each method's official code; gains are over SpatialClaw's no-tool baseline. [b]Evaluated by MSSR; backbone not stated. –: not reported.

Method	Backbone	VSI-Debiased	MMSI	VSTI
Training-free tool use and visual programming				
VADAR[b] (Marsili et al., 2025)	–	–	28.9	–
pySpatial (Luo et al., 2026)	GPT-4o	–	37.3 (+7.0)	–
pySpatial[a]	Gemma-4-31B	44.4 (−3.6)	33.2 (−4.7)	55.0 (+0.3)
SpaceTools (Toolshed)[a] (Chen et al., 2026a)	Gemma-4-31B	33.6 (−14.4)	33.1 (−4.8)	56.0 (+1.3)
S-Agent (Dai et al., 2026)	GPT-5.4	–	46.4 (+4.5)	–
GCA (Chen et al., 2026b)	Qwen3-VL-Thinking	–	47.6 (+15.0)	–
MSSR (Guo et al., 2026)	GPT-4o	–	49.5 (+19.2)	–
SpatialClaw (Cho et al., 2026)	Gemma-4-31B	48.5 (+0.5)	51.3 (+13.4)	67.6 (+12.9)
SpatialClaw (Cho et al., 2026)	Qwen3.6-27B	57.9 (+2.5)	**53.9** (+13.0)	70.3 (+9.8)
SpaceMCP (ours, training-free)				
SpaceMCP	Gemma-4-31B	58 (+15)	44 (+8)	82 (+28)
SpaceMCP	GPT-6-Luna	57 (+14)	51 (+12)	**84** (+29)

Where the gains come from. The largest gains are on metric and camera-geometry categories (Table 4 in Appx. C): camera displacement (+68 points), camera movement direction (+44), relative direction (+22 to +36), absolute distance (+29), room size (+25) and MMSI-Bench's camera–object and camera–camera relations (+22, +18). On these categories the SpaceMCP score barely depends on the backbone — all eight planners score 88 on camera displacement, 69–70 on room size and 59–61 on absolute distance — because the answer is read off the scene graph rather than inferred from pixels. Gains on MMSI-Bench are smaller (−1 to +13), and the 95% interval reaches zero for four of the eight planners: MMSI-Bench supplies a few disjoint images rather than a video, and several of its categories (object motion, multi-step reasoning, region relations) have no direct counterpart in the current API. The clearest losses are VSTI-Bench's left/right relation (−14, no model improves), which the models already answer well alone (89%) and where some plans take the reference object's viewpoint and so mirror left and right (Appx. C); VSI-Bench appearance order (−7), where the tool's first-appearance order often matches none of the options; and MMSI-Bench's measurement attributes (−6). Route planning barely moves, as SpaceMCP almost always keeps the base answer there. The pattern follows the division of labour: the tools win where the answer is a quantity they can measure, and the VLM alone does as well or better where the question turns on context, conventions or reasoning that the geometry does not capture.

4.3 Comparison with other methods

Few tool-using methods report on these benchmarks (Table 2): VADAR is built for single images, SpaceTools for robotics and single images, and among training-free methods only SpatialClaw reports VSI-Bench-Debiased or VSTI-Bench; most pySpatial, SpaceTools and VADAR numbers come from SpatialClaw and MSSR. The SpaceMCP rows use each benchmark's official weighting; counting VSI-Bench's relative direction once rather than three times lowers GPT-6-Luna's VSI-Bench score from 61 to 57.

With GPT-6-Luna, SpaceMCP is level with SpatialClaw's best on VSI-Bench-Debiased (57 vs. 57.9; Muse-Spark-1.3 reaches 61) and scores 51 on MMSI-Bench, above MSSR, GCA and S-Agent but below SpatialClaw's 53.9; its 84 on VSTI-Bench is not comparable with the published, old-label results. Holding the backbone fixed is more telling: SpaceMCP beats SpatialClaw on VSI-Bench-Debiased by 9 points with Gemma-4-31B and 17 with Gemma-4-26B-A4B, but on MMSI-Bench it trails by 7 with Gemma-4-31B and only matches it with Gemma-4-26B-A4B (44 vs. 42.7). On our own MMSI-Bench questions, SpatialClaw driven by GPT-6-Luna also scores 51, the same as SpaceMCP, but it makes a planning call and up to 30 agent turns per question, each a VLM call over the growing trajectory and images with DA3, SAM 3 and further VLM sessions inside the loop, where SpaceMCP uses one plan and one answer turn over a scene graph built once per video.

Limitations. SpaceMCP inherits the failure modes of its perception backbones. DA3 returns confidently wrong depth on reflective, transparent or textureless surfaces, which propagates silently into the geometry. SAM 3 grounds bare noun phrases reliably but not descriptive ones ("chair by the door"), and misses some unusual objects. The cross-frame instance clustering can over-split a single piece of furniture in cluttered scenes, inflating counts. Object orientation is not supported directly: `construct_direction` builds a forward direction from two named points, but the *intrinsic* facing of a single object or person has no such anchor; single-shot orientation estimation (Wang et al., 2025b) could become a per-object getter. The planner adds failure modes of its own: on VSTI-Bench some left/right plans reason from the reference object's viewpoint and so mirror left and right (9% of them), an error of convention that the geometry cannot catch. Harnessing the interface also takes a minimum level of model capability — the VLM must name the right objects, pick the right getters and emit a valid plan. Qwen2.5-Omni-7B, the weakest model we tested, falls below it: it rarely produces an executable plan (Table 10) and loses 8 points on VSTI-Bench, whereas DeepSeek-V4.1-Flash clears the bar even without a reasoning mode. At the other end, SpaceMCP lifts weak and strong models to a similar level, so the headroom shrinks as the base model gets stronger: Muse-Spark-1.3, the strongest MMSI-Bench base model, already sits at that level (52 before, 51 after; 95% CI $[-6, +4]$) and is the only planner that does not improve on a benchmark. More fundamentally, the geometric pipeline has no plausibility prior — a 0.4 m sofa or a 42 m room comes back unflagged — so only the VLM's answer turn, which overrides most gross outliers, catches such failures. These limits are why we propose SpaceMCP as an augmentation: the tools are precise when perception succeeds, but deciding when to trust them, and answering what no getter covers, still rests on the VLM's own spatial understanding.

5 Conclusion

We presented SpaceMCP, a test-time scaling framework that gives a VLM metric spatial reasoning by building a scene graph in real time and exposing it through a generic MCP interface. Without any training, it raises scores by 16 points on average on VSI-Bench, 6 on MMSI-Bench and 14 on VSTI-Bench across nine VLMs.

SpaceMCP is an augmentation, not a substitute for better VLMs: the vision models answer precisely where a question reduces to geometry, the VLM supplies the context and nuance to pose the right queries and judge their results, and the method still falls short on viewpoint conventions, temporal order and fine measurements. Native spatial reasoning, as pursued by large-scale spatial training such as SenseNova-SI (Cai et al., 2026), remains necessary, both for questions no tool covers and for judging when the tools are wrong. As 3D perception matures, we see SpaceMCP as a first step towards standardising the interface between VLMs and 3D perception, as MCP has done for tool use in LLMs, with future versions adding object orientation, motion, part-level queries and dynamic scene updates.

AI use statement

We used generative AI tools to help write the code for the method and its evaluation and to analyse and interpret the experimental results. We also used them to edit the paper, to create or modify figures, and to find related work and format references. We did not use them to design the method or the experiments. The authors reviewed AI-assisted work and take responsibility for the final content of this paper.

Ethics statement

SpaceMCP turns ordinary video into a metric scene graph that records the positions, sizes and trajectories of objects and, where present, people. We evaluate only on public benchmarks, but the same pipeline could extract this information from surveillance or covertly captured footage. Deployments involving people should be gated by consent and retention limits, and cached reconstructions should be treated as personal data wherever the source video is.

REFERENCES

Anthropic. Model Context Protocol. https://modelcontextprotocol.io, 2024.

Anthropic. Project Glasswing: Securing critical software for the AI era. https://www.anthropic.com/glasswing, 2026.

H. Batra, H. Tu, H. Chen, Y. Lin, C. Xie, and R. Clark. SpatialThinker: Reinforcing Scene Graph-Grounded Spatial Reasoning via Dense Rewards. *arXiv preprint arXiv:2511.07403*, 2025.

E. Brown, J. Yang, S. Yang, R. Fergus, and S. Xie. Benchmark Designers Should "Train on the Test Set" to Expose Exploitable Non-Visual Shortcuts. In *COLM*, 2026.

W. Cai, I. Ponomarenko, J. Yuan, X. Li, W. Yang, H. Dong, and B. Zhao. SpatialBot: Precise Spatial Understanding with Vision Language Models. In *ICRA*, 2025.

Z. Cai, R. Wang, C. Gu, F. Pu, J. Xu, Y. Wang, W. Yin, Z. Yang, C. Wei, et al. Scaling Spatial Intelligence with Multimodal Foundation Models. In *CVPR*, 2026.

M. Cao, X. Li, X. Liu, I. Reid, and X. Liang. SpatialDreamer: Incentivizing Spatial Reasoning via Active Mental Imagery. In *CVPR Findings*, 2026.

N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Surís, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, et al. SAM 3: Segment Anything with Concepts. In *ICLR*, 2026.

B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities. In *CVPR*, 2024.

S. Chen, M. A. Uy, C. H. Song, F. Ladhak, A. Murali, Q. Qu, S. Birchfield, V. Blukis, and J. Tremblay. SpaceTools: Tool-Augmented Spatial Reasoning via Double Interactive RL. In *CVPR*, 2026.

Z. Chen, X. Lu, Z. Zheng, P. Li, L. He, Y. Zhou, J. Shao, B. Zhuang, and L. Sheng. Geometrically-Constrained Agent for Spatial Reasoning. In *CVPR*, 2026.

A.-C. Cheng, H. Yin, Y. Fu, Q. Guo, R. Yang, J. Kautz, X. Wang, and S. Liu. SpatialRGPT: Grounded Spatial Reasoning in Vision-Language Models. In *NeurIPS*, 2024.

A.-C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang. NaVILA: Legged Robot Vision-Language-Action Model for Navigation. In *RSS*, 2025.

A.-C. Cheng, Y. Fu, Y. Chen, Z. Liu, X. Li, S. Radhakrishnan, S. Han, Y. Lu, J. Kautz, P. Molchanov, et al. 3D Aware Region Prompted Vision Language Model. In *ICLR*, 2026.

S. Cho, R. Hachiuma, A. Badki, H. Su, B.-K. Lee, C. H. Song, S. Liu, S. Radhakrishnan, S. Kim, Y.-C. F. Wang, and M.-H. Chen. SpatialClaw: Rethinking Action Interface for Agentic Spatial Reasoning. *arXiv preprint arXiv:2606.13673*, 2026.

Y. Dai, H. Li, S. Tian, R. Yao, Y. Dong, F. Hong, Z. Chen, F. Liu, T. Wang, K.-H. Yap, and Z. Liu. S-Agent: Spatial Tool-Use Elicits Reasoning for Spatial Intelligence. *arXiv preprint arXiv:2606.20515*, 2026.

E. Daxberger, N. Wenzel, D. Griffiths, H. Gang, J. Lazarow, G. Kohavi, K. Kang, M. Eichner, Y. Yang, A. Dehghan, and P. Grasch. MM-Spatial: Exploring 3D Spatial Understanding in Multimodal LLMs. In *ICCV*, 2025.

Z. Fan, J. Zhang, R. Li, J. Zhang, R. Chen, H. Hu, K. Wang, P. Wang, H. Qu, S. Zhou, et al. VLM-3R: Vision-Language Models Augmented with Instruction-Aligned 3D Reconstruction. In *CVPR*, 2026.

M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, et al. ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning. In *ICRA*, 2024.

Y. Guo, Y. Hou, W. Ma, M. Tang, and M.-H. Yang. Pursuing Minimal Sufficiency in Spatial Reasoning. In *ICLR*, 2026.

T. Gupta and A. Kembhavi. Visual Programming: Compositional Visual Reasoning Without Training. In *CVPR*, 2023.

Y. Han, E. Zhou, S. Rong, J. An, P. Wang, Z. Wang, C. Chi, L. Sheng, and S. Zhang. TIGeR: Tool-Integrated Geometric Reasoning in Vision-Language Models for Robotics. In *ICRA*, 2026.

Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan. 3D-LLM: Injecting the 3D World into Large Language Models. In *NeurIPS*, 2023.

N. Keetha, N. Müller, J. Schönberger, L. Porzi, Y. Zhang, T. Fischer, A. Knapitsch, D. Zauss, E. Weber, N. Antunes, et al. MapAnything: Universal Feed-Forward Metric 3D Reconstruction. In *3DV*, 2026.

A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment Anything. In *ICCV*, 2023.

V. Leroy, Y. Cabon, and J. Revaud. Grounding Image Matching in 3D with MASt3R. In *ECCV*, 2024.

D. Li, H. Li, Z. Wang, Y. Yan, H. Zhang, S. Chen, G. Hou, S. Jiang, W. Zhang, Y. Shen, W. Lu, and Y. Zhuang. ViewSpatial-Bench: Evaluating Multi-perspective Spatial Localization in Vision-Language Models. In *ECCV*, 2026.

H. Li, D. Li, Z. Wang, Y. Yan, H. Wu, W. Zhang, Y. Shen, W. Lu, J. Xiao, and Y. Zhuang. SpatialLadder: Progressive Training for Spatial Reasoning in Vision-Language Models. In *ICLR*, 2026.

P. Li, P. Song, W. Li, H. Yao, W. Guo, Y. Xu, D. Liu, and H. Xiong. See&Trek: Training-Free Spatial Prompting for Multimodal Large Language Model. In *NeurIPS*, 2025.

S. Lian, C. Wu, L. T. Yang, H. Yuan, B. Yu, L. Zhang, and K. Chen. Euclid's Gift: Enhancing Spatial Perception and Reasoning in Vision-Language Models via Geometric Surrogate Tasks. In *CVPR Findings*, 2026.

H. Lin, S. Chen, J. H. Liew, D. Y. Chen, Z. Li, Y. Zhao, S. Peng, H. Guo, X. Zhou, G. Shi, J. Feng, and B. Kang. Depth Anything 3: Recovering the Visual Space from Any Views. In *ICLR*, 2026.

Y. Liu, B. Zhang, Y. Zang, Y. Cao, L. Xing, X. Dong, H. Duan, D. Lin, and J. Wang. Spatial-SSRL: Enhancing Spatial Understanding via Self-Supervised Reinforcement Learning. In *CVPR*, 2026.

Z. Luo, C. Zhang, S. Yong, C. Dai, Q. Wang, H. Ran, G. Shi, K. Sycara, and Y. Xie. pySpatial: Generating 3D Visual Programs for Zero-Shot Spatial Reasoning. In *ICLR*, 2026.

C. Ma, K. Lu, T.-Y. Cheng, N. Trigoni, and A. Markham. SpatialPIN: Enhancing Spatial Reasoning Capabilities of Vision-Language Models through Prompting and Interacting 3D Priors. In *NeurIPS*, 2024.

W. Ma, H. Chen, G. Zhang, Y.-C. Chou, J. Chen, C. M. de Melo, and A. Yuille. 3DSRBench: A Comprehensive 3D Spatial Reasoning Benchmark. In *ICCV*, 2025.

W. Ma, L. Ye, C. M. de Melo, A. Yuille, and J. Chen. SpatialLLM: A Compound 3D-Informed Design towards Spatially-Intelligent Large Multimodal Models. In *CVPR*, 2025.

D. Marsili, R. Agrawal, Y. Yue, and G. Gkioxari. Visual Agentic AI for Spatial Reasoning with a Dynamic API. In *CVPR*, 2025.

OpenAI. Learning to reason with LLMs. `https://openai.com/index/learning-to-reason-with-llms/`, 2024.

K. Ouyang, Y. Liu, H. Wu, Y. Liu, H. Zhou, J. Zhou, F. Meng, and X. Sun. SpaceR: Reinforcing MLLMs in Video Spatial Reasoning. *arXiv preprint arXiv:2504.01805*, 2025.

S. K. Ramakrishnan, E. Wijmans, P. Krähenbühl, and V. Koltun. Does Spatial Cognition Emerge in Frontier Models? In *ICLR*, 2025.

D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. In *COLM*, 2024.

P. Sun, S. Lang, D. Wu, Y. Ding, K. Feng, H. Liu, Z. Ye, R. Liu, Y.-H. Liu, J. Wang, and X. Yue. SpaceVista: All-Scale Visual Spatial Reasoning from mm to km. In *ICML*, 2026.

D. Surís, S. Menon, and C. Vondrick. ViperGPT: Visual Inference via Python Execution for Reasoning. In *ICCV*, 2023.

J. Wan, X. Wang, M. Xie, H. Zhang, M. Xu, Y. Han, D. Yuan, H. Zhang, and Y. Yang. EagleVision: A Dual-Stage Framework with BEV-grounding-based Chain-of-Thought for Spatial Intelligence. In *CVPR*, 2026.

J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. VGGT: Visual Geometry Grounded Transformer. In *CVPR*, 2025.

S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *CVPR*, 2024.

Z. Wang, Z. Zhang, T. Pang, C. Du, H. Zhao, and Z. Zhao. Orient Anything: Learning Robust Object Orientation Estimation from Rendering 3D Models. In *ICML*, 2025.

D. Wu, F. Liu, Y.-H. Hung, and Y. Duan. Spatial-MLLM: Boosting MLLM Capabilities in Visual-based Spatial Intelligence. In *NeurIPS*, 2025.

J. Wu, J. Guan, K. Feng, Q. Liu, S. Wu, L. Wang, W. Wu, and T. Tan. Reinforcing Spatial Reasoning in Vision-Language Models with Interwoven Thinking and Visual Drawing. In *NeurIPS*, 2025.

B. Xu, S. Zhu, Z. Jin, J. Li, and H. Wang. S^2-MLLM: Boosting Spatial Reasoning Capability of MLLMs for 3D Visual Grounding with Structural Guidance. In *CVPR*, 2026.

J. Yang, S. Yang, A. W. Gupta, R. Han, L. Fei-Fei, and S. Xie. Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces. In *CVPR*, 2025.

S. Yang, J. Yang, P. Huang, E. Brown, Z. Yang, Y. Yu, S. Tong, Z. Zheng, Y. Xu, M. Wang, et al. Cambrian-S: Towards Spatial Supersensing in Video. In *ICLR*, 2026.

S. Yang, R. Xu, Y. Xie, S. Yang, M. Li, J. Lin, C. Zhu, X. Chen, H. Duan, X. Yue, et al. MMSI-Bench: A Benchmark for Multi-Image Spatial Intelligence. In *ICLR*, 2026.

Y. Yang, J. Liu, Z. Zhang, S. Zhou, R. Tan, J. Yang, Y. Du, and C. Gan. MindJourney: Test-Time Scaling with World Models for Spatial Reasoning. In *NeurIPS*, 2025.

H. Yu, W. Li, S. Wang, J. Chen, and J. Zhu. Inst3D-LMM: Instance-Aware 3D Scene Understanding with Multi-modal Instruction Tuning. In *CVPR*, 2025.

S. Yu, Y. Zhang, Z. Wang, J. Yoon, H. Yao, M. Ding, and M. Bansal. When and How Much to Imagine: Adaptive Test-Time Scaling with World Models for Visual Spatial Reasoning. *arXiv preprint arXiv:2602.08236*, 2026.

R. Zhao, Z. Zhang, J. Xu, J. Chang, D. Chen, L. Li, W. Sun, and Z. Wei. SpaceMind: Camera-Guided Modality Fusion for Spatial Reasoning in Vision-Language Models. In *CVPR*, 2026.

X. Zheng, Z. Dongfang, L. Jiang, B. Zheng, Y. Guo, Z. Zhang, G. Albanese, R. Yang, M. Ma, Z. Zhang, et al. Multimodal Spatial Reasoning in the Large Model Era: A Survey and Benchmarks. *arXiv preprint arXiv:2510.25760*, 2025.

F. Zhu, H. Wang, Y. Xie, J. Gu, T. Ding, J. Yang, and H. Jiang. Struct2D: A Perception-Guided Framework for Spatial Reasoning in MLLMs. In *NeurIPS*, 2025.

Table 3: The SpaceMCP MCP interface. Linear sizes are in metres, areas in m^2, volumes in m^3, frames are integers, angles are degrees. Bearings use the $[0, 360)$ compass convention (0°=ahead, 90°=right); pose-pair rotations and pitch use signed $(-180, 180]$ / $[-90, 90]$ to encode direction in the sign. Position arguments accept any of: an object name (resolved to its OBB centre), a coordinate, or a `"%var.path"` reference into the result of an earlier call.

Tool	Description	Tool	Description
Augmentation		*Object geometry (cont.)*	
`add_object(name)`	Segment one instance.	`get_object_volume(n)`	Sum of OBB volumes.
`add_objects(name)`	All instances, clustered.	`get_distance(`n_1, n_2`)`	Closest-pair distance.
Static room geometry		`get_first_appearance(n)`	First visible frame.
`get_room_size()`	Area, long, short side.	`get_relative_direction(`$p_1, \vec{f}, p_2$`)`	Bearing of p_2 from p_1.
`get_room_corners()`	Four $[x, y]$ corners.	*Camera and pose operations*	
`get_room_center()`	3D room centre.	`get_camera_pose(frame)`	Camera extrinsics.
`get_room_axes()`	Long/short/up axes.	`rotate_clockwise(pose, deg)`	Pose rotated CW.
Object geometry		`rotate_anticlockwise(pose, deg)`	Pose rotated CCW.
`get_all_object_labels()`	Labels in graph.	`move_camera_forward(pose, m)`	Pose translated.
`get_object_count(n)`	Number of instances.	`get_distance_to_object(pose, n)`	2D distance.
`get_object_shortest_dim(n)`	Smallest OBB extent.	`get_translation_between(`p_a, p_b`)`	Forward/right delta.
`get_object_middle_dim(n)`	Middle OBB extent.	`get_rotation_between(`p_a, p_b`)`	Signed yaw delta.
`get_object_longest_dim(n)`	Largest OBB extent.	`get_pitch_between(`p_a, p_b`)`	Signed pitch delta.
`get_object_height(n)`	Vertical extent.	`get_camera_displacement(`p_a, p_b`)`	3D L2 between positions.
`get_object_height_above_ground(n)`	Height above floor.		
`get_object_horizontal_extents(n)`	Footprint dims.		
`get_object_center(n)`	3D centroid.		

A MCP Tool-Use Prompts

Example system prompts are included in the supplementary material. The plan prompt enumerates the available getters (Table 3), demands a strict JSON output schema, and includes a small number of worked in-context examples.

B Additional Implementation Notes

Cross-frame instance clustering for `add_objects`. SAM 3 emits per-frame, per-instance masks; we recover physical objects by clustering those detections across frames. Each detection is a 3-D point set in the room frame, obtained by back-projecting the masked pixels via cached DA3 depth and filtering the bottom 15% by depth confidence (and dropping SAM 3 instances with score < 0.5). For all cross-frame pairs (i, j) we compute the min-point-distance (subsampling to 300 points per detection for speed). We walk the pairs in ascending min-distance order and greedily union pair (i, j) whenever *(i)* min_dist$(i, j) \leq 0.5$ m and *(ii)* the resulting union would not place two same-frame detections in the same component. The same-frame mutex is enforced *transitively* via component frame-sets, so a chain of below-threshold cross-frame links cannot collapse two adjacent objects (e.g. neighbouring chairs at a table) into one cluster. An OBB is then fit to each resulting component.

Caching. Per-video DA3 outputs and floor / wall SAM 3 masks are cached on disk and re-used across questions sharing the same video, so the amortised per-question cost is dominated by the per-target SAM 3 forwards triggered by the `add_object` / `add_objects` calls in the plan turn.

C Nine-model run: details and per-category results

Tables 6–8 give every model–category cell as base → SpaceMCP (%). Patterns across models are more informative than single cells. Model abbreviations follow Table 5 (G = Gemma-4).

Caveats. Providers refused eight base (media) requests: the Gemini API returned an error on three image sets for each Gemma model, and DashScope's moderation flagged one frame for each Qwen model; those base answers are empty and scored wrong, while SpaceMCP's text-only turns still ran. GLM-5.3-Flash on Hive fails on requests whose images total about 7.5 megapixels or more; 33 such requests were retried with the images shrunk to 6 megapixels. GLM-5.3-Flash also used up its 32,768-token reasoning budget without answering on 15 MMSI-Bench base questions and one VSI-Bench base question (three attempts each); those base answers are empty and scored wrong. On MMSI-Bench, Gemini-3.5-Flash-Lite wrote code or a new JSON plan instead of an answer 19

Table 4: **Gain by category**: mean Δ (points) over the nine models and the number of models that improve ($\uparrow$, of 9); ^{m} numeric (MRA). Categories are listed in the order of Sec. 4.1; per-model cells are in Tables 6–8.

VSI-Bench	Δ	$\uparrow$	MMSI-Bench	Δ	$\uparrow$	VSTI-Bench	Δ	$\uparrow$
AppOrd	-7	1	Mot-Cam	$+6$	5	CamDisp^m	$+68$	9
RelDir-E	$+22$	8	Mot-Obj	$+1$	4	CamDir	$+44$	9
RelDir-M	$+36$	9	PR Cam–Cam	$+18$	9	CO-Abs^m	$+11$	8
RelDir-H	$+25$	9	PR Cam–Obj	$+22$	8	CO-Rel1	$+7$	8
RelDist	$+5$	7	PR Cam–Reg	$+7$	5	CO-Rel2	-2	3
Route	-1	0	PR Obj–Obj	$+7$	7	CO-Rel3	$+8$	8
AbsDist^m	$+29$	9	PR Obj–Reg	$+1$	4	OO-LR	-14	0
Count^m	$+8$	8	PR Reg–Reg	-4	1	OO-NF	$+6$	7
Size^m	$+16$	9	Attr-Appr	$+16$	9	OO-UD	-5	3
Room^m	$+25$	9	Attr-Meas	-6	2			
			MSR	-1	4			
Score	$+16$	9	**Score**	$+6$	7	**Score**	$+14$	8

Table 5: Models, API providers and reasoning settings of the nine-model run.

Model	API provider	Reasoning setting
GPT-6-Luna	OpenAI	high
Muse-Spark-1.3	Meta Model API, contributor tier	high
GLM-5.3-Flash	Hive	high
Qwen3.5-27B	Alibaba Cloud DashScope	high (thinking capped at 32,768 tokens)
Gemma-4-26B-A4B	Gemini API	high
DeepSeek-V4.1-Flash	Hive	none (not offered by the provider)
Gemma-4-31B	Gemini API	high
Gemini-3.5-Flash-Lite	Gemini API	high
Qwen2.5-Omni-7B	Alibaba Cloud DashScope	none (no reasoning mode)

times, and on VSI-Bench appearance order GPT-6-Luna answered with the order itself instead of a letter on 41% of questions; both count as wrong (the *+Fallback* column shows the effect of falling back instead). Muse-Spark-1.3 ran on Meta's contributor tier, whose terms allow Meta to use the submitted questions and frames.

Left/right viewpoint. For VSTI-Bench's "in frame N, relative to Y, is X to the left or right?" questions, the plan prompt's left/right in-context example ("what is to the left of the kiln from where you stand?") takes the direction from the camera to the object as "ahead" and says not to face the camera; the original prompt, used for all other answers, takes the direction from the object to the camera. On VSTI-Bench, 9% of left/right plans reason from the reference object's viewpoint and so mirror left and right, most of them from Gemini-3.5-Flash-Lite, Gemma-4-26B-A4B and Gemma-4-31B. Plans that look from the camera are right 86% of the time and mirrored ones 10%, so even from the right viewpoint SpaceMCP stays just below the 89% the models reach alone on this category. On MMSI-Bench, plans that take the object's viewpoint score below the model alone on region–region and object–region questions (13% vs. 30%, 30% vs. 43%) but above it on appearance attributes (52% vs. 21%).

Measurement attributes. SpaceMCP loses on MMSI-Bench's measurement-attribute questions for six of the nine models. In the cases we inspected, the tool measured the wrong extent (a door leaf rather than the door opening, the bounding box of hanging clothes) or missed the object, and the answer trusted its numbers anyway, sometimes over a "cannot be determined" option.

Table 6: VSI-Bench (debiased) per category, base → SpaceMCP (%). ^{m} MRA.

Category	Luna	Muse	GLM	Qwen3.5	G-26B	DeepSeek	G-31B	Flash-Lite	Omni-7B	mean Δ
AppOrd	53→35	51→47	55→47	53→47	49→47	39→47	61→53	63→49	37→27	−7
RelDir-E	59→90	65→82	65→88	65→86	47→84	49→80	55→86	80→90	51→47	+22
RelDir-M	31→82	49→82	45→78	53→76	22→82	37→80	33→82	43→78	33→35	+36
RelDir-H	29→63	53→67	41→61	35→67	27→59	29→57	33→63	41→63	29→39	+25
RelDist	47→55	51→55	49→55	47→51	37→55	35→53	43→55	57→55	43→18	+5
Route	49→49	69→69	47→47	65→65	41→39	47→47	41→41	45→45	24→16	−1
AbsDist^m	34→59	36→59	31→61	29→61	10→60	26→60	24→59	38→59	18→24	+29
Count^m	24→47	44→52	42→56	45→46	29→43	32→53	35→49	54→58	26→1	+8
Size^m	48→61	56→58	57→60	53→60	13→57	45→60	46→57	57→60	15→58	+16
Room^m	47→70	57→70	49→70	49→70	38→70	32→70	50→70	53→69	28→70	+25
Score	42→61	53→64	48→62	49→63	31→60	37→61	42→61	53→63	30→33	+16

Table 7: MMSI-Bench per category, base → SpaceMCP (%).

Category	Luna	Muse	GLM	Qwen3.5	G-26B	DeepSeek	G-31B	Flash-Lite	Omni-7B	mean Δ
Mot-Cam	63→63	61→63	55→55	41→49	45→59	27→49	41→29	33→49	27→27	+6
Mot-Obj	31→31	45→39	43→39	33→31	24→27	24→35	39→35	33→35	18→29	+1
PR Cam–Cam	43→69	61→65	51→69	49→57	33→63	31→53	35→63	41→57	29→33	+18
PR Cam–Obj	33→69	51→67	43→57	31→55	20→53	20→53	24→51	24→51	39→27	+22
PR Cam–Reg	39→61	61→67	59→57	51→51	37→45	22→51	49→43	31→37	41→37	+7
PR Obj–Obj	27→49	41→45	39→47	33→45	43→47	33→35	33→43	33→33	27→27	+7
PR Obj–Reg	29→45	45→45	41→47	35→41	27→27	37→33	39→41	43→39	35→20	+1
PR Reg–Reg	43→37	51→35	37→45	49→41	39→37	33→33	41→35	37→35	43→37	−4
Attr-Appr	24→43	43→49	39→45	35→49	22→47	20→41	20→43	16→33	18→31	+16
Attr-Meas	47→45	63→47	53→43	55→45	49→43	41→45	47→53	63→43	33→33	−6
MSR	45→49	47→35	43→33	41→39	41→39	35→43	33→43	29→31	33→29	−1
Score	38→51	52→51	46→49	41→46	35→44	29→43	36→43	35→40	31→30	+6

Table 8: VSTI-Bench per category, base → SpaceMCP (%). ^{m} MRA. OO-LR uses the revised in-context example (Sec. 4.1).

Category	Luna	Muse	GLM	Qwen3.5	G-26B	DeepSeek	G-31B	Flash-Lite	Omni-7B	mean Δ
CamDisp^m	18→88	11→88	18→88	24→88	10→88	20→88	15→88	27→88	2→56	+68
CamDir	49→98	57→94	51→96	51→100	55→100	39→96	43→100	49→100	41→47	+44
CO-Abs^m	45→61	55→61	53→61	50→61	39→59	31→56	48→60	50→62	37→22	+11
CO-Rel1	78→84	82→84	78→84	76→84	59→82	53→84	76→82	78→84	53→27	+7
CO-Rel2	84→80	82→80	78→80	86→82	73→76	69→84	80→80	84→78	59→37	−2
CO-Rel3	69→82	82→86	73→84	78→82	71→82	55→82	76→84	78→86	63→53	+8
OO-LR	90→86	98→86	86→86	96→86	86→67	88→84	90→61	92→69	80→57	−14
OO-NF	78→88	84→88	90→90	90→90	84→90	76→86	76→90	86→90	51→55	+6
OO-UD	90→92	94→92	94→94	94→90	92→96	88→80	90→94	96→94	80→43	−5
Score	67→84	72→84	69→85	72→85	63→82	58→82	66→82	71→83	52→44	+14

Table 9: SpaceMCP under each benchmark's official weighting, base → SpaceMCP (%): VSI-Bench's eight-task mean (relative direction averaged over its three levels), MMSI-Bench's per-question accuracy (using the category sizes of the full benchmark) and VSTI-Bench's mean over its five task groups. Table 2 uses these numbers. They are computed from the rounded per-category scores of Tables 6–8, so a value may be off by one.

	VSI-Bench-Debiased (8 tasks)			MMSI-Bench (per question)			VSTI-Bench (5 groups)		
Model	Base	Ours	Δ	Base	Ours	Δ	Base	Ours	Δ
GPT-6-Luna	43	**57**	+14	39	**51**	+12	55	**84**	+29
Muse-Spark-1.3	52	**61**	+9	**51**	49	−2	59	**83**	+24
GLM-5.3-Flash	48	**59**	+11	45	**47**	+2	58	**84**	+26
Qwen3.5-27B	49	**60**	+11	41	**45**	+4	60	**84**	+24
Gemma-4-26B-A4B	31	**56**	+25	35	**44**	+9	52	**82**	+30
DeepSeek-V4.1-Flash	37	**58**	+21	30	**43**	+13	47	**81**	+34
Gemma-4-31B	43	**58**	+15	36	**44**	+8	54	**82**	+28
Gemini-3.5-Flash-Lite	53	**59**	+6	34	**39**	+5	59	**83**	+24
Qwen2.5-Omni-7B	29	**32**	+3	**32**	30	−2	42	**43**	+1

Table 10: Other ways to count. *Tools only*: the tool-based answer alone, with no fallback to the base answer. *+Fallback*: SpaceMCP, but an answer with no letter or number (for example a new plan instead of an answer) also falls back to the base answer. *Unscorable*: number of answers with no letter or number, which count as wrong. The VSTI-Bench columns use the original prompt on the left/right questions too (Sec. 4.1).

	Tools only			+Fallback			Unscorable, base			Unscorable, SpaceMCP		
Model	VSI	MMSI	VSTI	VSI	MMSI	VSTI	VSI	MMSI	VSTI	VSI	MMSI	VSTI
GPT-6-Luna	60	50	83	64	51	83	0	1	0	25	3	0
Muse-Spark-1.3	62	50	78	64	51	78	6	0	0	2	0	0
GLM-5.3-Flash	62	47	80	62	49	80	1	16	0	1	1	0
Qwen3.5-27B	60	46	82	63	46	82	0	0	1	0	0	0
Gemma-4-26B-A4B	59	44	81	60	44	81	83	0	3	4	0	0
DeepSeek-V4.1-Flash	60	42	80	61	43	82	4	1	1	5	2	8
Gemma-4-31B	60	43	79	61	43	79	14	0	3	1	0	0
Gemini-3.5-Flash-Lite	61	41	80	63	41	80	0	0	0	0	19	0
Qwen2.5-Omni-7B	33	27	41	34	31	47	18	0	1	16	20	26