Denser Input, Worse Agent: Compound Bottlenecks and a Capability Threshold in VLM Spatial Reasoning

Donghyun Lee [1] **Jongwon Jeong** [1]

Abstract

Vision–language models (VLMs) lag text-only LLMs on active spatial reasoning, even though the visual input carries strictly denser information than the matching textual scene description (Zhang et al., 2026a). We trace this persistent gap to two compounding bottlenecks: **(B1)** a VLM-specific *perception bottleneck* concentrated in object orientation, and **(B2)** a universal *belief-instability bottleneck* — both LLM and VLM drift at comparable rates across turns. We attempt to close this gap with a multi-agent pipeline of perception sub-agents and an external cognitive map. Across three frontier models, the pipeline reliably improves *cogmap quality* (the agent's internal spatial belief of the environment), but these representational gains translate to *evaluation accuracy* (downstream task performance) only above a model-capability threshold — yielding *pipeline-helps*, *pipeline-neutral*, and *pipeline-hurts* outcomes. We trace this divergence to a single dimension: *world coverage* (the fraction of world objects placed in the cogmap, driven by the agent's exploration policy).

1. Introduction

Vision–language models (VLMs) are language models extended to accept image input alongside text, allowing them to operate as situated agents in environments observed through visual sensors. A VLM agent perceives its environment through first-person images, acts through text or structured output, and iteratively gathers observations, updates an internal belief state, and selects an action at each step. This architecture has been adopted across embodied tasks in which the observation is fundamentally visual: indoor navigation (Yokoyama et al., 2024; Zhang et al., 2025; Cao et al., 2025) and embodied action planning (Lee et al., 2025), both of which require the agent to construct a spatial understanding of the scene from a stream of partial visual observations.

Despite this growing role, VLM agents struggle on active spatial reasoning. Zhang et al. (2026a) report that across an active spatial-exploration benchmark, the same frontier model reasons more reliably from a textual description of the scene than from the matching first-person image—the denser visual input yields the worse agent. However, *why* VLMs lag their text-input counterparts in this active-exploration setting has not been jointly decomposed: prior work has studied perception extraction (Khezresmaeilzadeh et al., 2026; Cheng et al., 2024) and belief instability (Tang et al., 2026) in isolation, but not jointly in an active-exploration agent setting where the agent must build beliefs from its own multi-turn visual observations and where their relative weight in the LLM-vs-VLM gap can be measured. This motivates our central question:

> *Given that vision provides strictly denser information than text, can VLMs match or outperform their text-input counterparts on spatial reasoning?*

We find that VLMs continue to struggle: across three frontier models, vision-input agents fall behind their text-input baselines. In order to diagnose *why*, we test each candidate failure mode independently — perception fidelity at turn 1 (before any belief has accumulated) and belief stability across the 20-turn exploration window. Both hypotheses are supported, and the two failure modes compound: **(B1)** a VLM-specific *perception bottleneck* concentrated in object orientation, and **(B2)** a universal *belief-instability bottleneck* — both LLM and VLM drift at comparable rates across turns. We diagnose each in detail in Section 4.

The perception side of this gap has been studied in *passive* spatial understanding, where prior work fuses implicit 3D-geometry tokens with the VLM's visual tokens (Zheng et al., 2025; Wu et al., 2025; Fan et al., 2026); however, recent analysis (Zhang et al., 2026b) shows these implicit-fusion approaches often leave geometric cues underutilized. Our setting differs: *active* exploration, in which the agent must build and refine beliefs across turns, exposes belief insta-

[1]University of Wisconsin–Madison. Correspondence to: Donghyun Lee <dlee448@wisc.edu>.

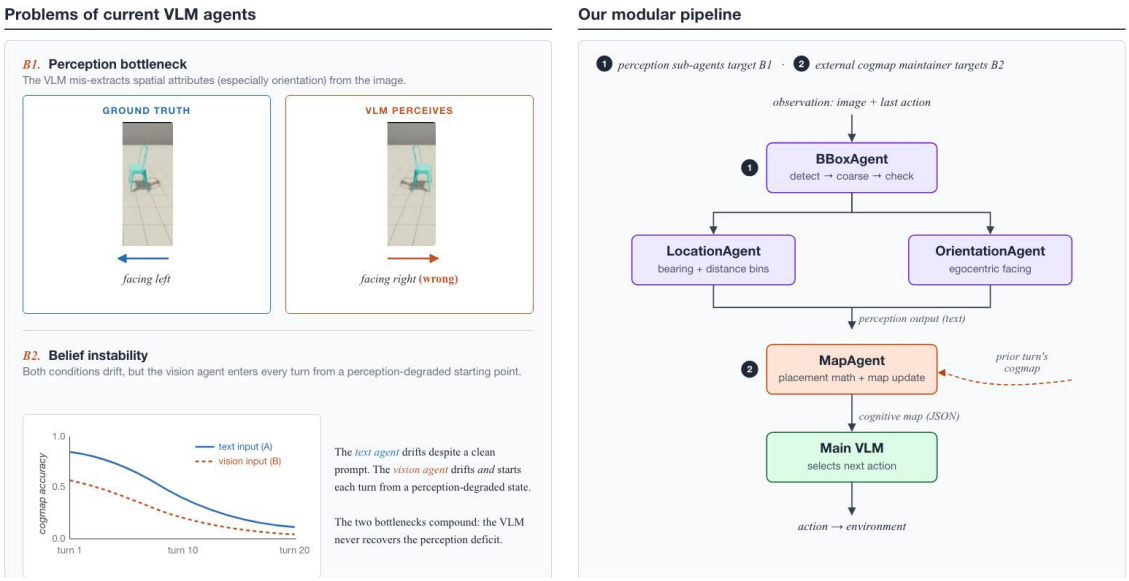

Figure 1. **Denser input, worse agent (left), and our multi-agent pipeline (right). Left:** despite the denser input from first-person images, VLMs perform worse than the same frontier models given a text-only scene description. We trace the gap to two compounding bottlenecks — (B1) a VLM-specific perception deficit and (B2) a universal belief-instability drift. **Right:** per-turn data flow with all sub-agents active (condition E). **purple** = perception sub-agents (BBoxAgent, LocationAgent, OrientationAgent); **orange** = MapAgent (maintains the cognitive map across turns); **green** = main VLM. In condition C, the MapAgent is omitted; in condition D, the three perception sub-agents are omitted and the MapAgent estimates bearing/distance/facing directly from the image.

bility (B2) as a second, orthogonal bottleneck that passive-perception methods do not address. We therefore work at the agent level, leaving the underlying VLM unchanged. Specialized perception sub-agents (bounding-box, location, orientation) supply structured per-turn spatial readings to mitigate B1; an external cogmap maintainer mitigates B2 by holding spatial state across turns rather than relying on context-window memory. We design a multi-agent pipeline (Figure 1) that targets each. Across three frontier VLMs, the pipeline reliably improves cogmap-quality metrics, but these representational gains translate to *evaluation-accuracy* gains only above a model-capability threshold — yielding *pipeline-helps*, *pipeline-neutral*, and *pipeline-hurts* outcomes. We trace this divergence to a single cogmap dimension the pipeline does *not* improve uniformly: *world coverage* — the fraction of world objects placed in the cogmap, driven by the agent's exploration policy. The pipeline stabilizes what the VLM produces but does not equalize what the agent has chosen to explore.

Contributions.

- We **diagnose** the VLM–LLM spatial-reasoning gap as the compound product of two bottlenecks: (B1) a VLM-specific perception bottleneck concentrated in object orientation; and (B2) a universal belief-instability bottleneck that operates on top of the perception deficit and prevents recovery across the 20-turn exploration window (Section 4).
- We introduce a **multi-agent pipeline** combining explicit perception sub-agents (bounding-box, location, orientation) with an external cogmap maintainer that targets B1 and B2 respectively (Section 3).
- We document a **capability-threshold effect**: cogmap-quality gains translate into evaluation-accuracy gains only above a model-capability threshold (Section 5), which corresponds to divergence in a single cogmap dimension across models — *world coverage*.

2. Related Work

Table 1 positions our work against prior multi-modal spatial-reasoning systems along five capability dimensions:

1. **Perception Tools**: dedicated perception sub-modules at inference time — object detection, segmentation, depth estimation, or spatial-attribute extraction — beyond end-to-end VLM perception.
2. **External Memory**: an explicit cogmap maintained across turns.
3. **Active Exploration**: the agent takes actions to gather

information rather than evaluating a static scene.

4. **VLM–LLM Gap**: explicit comparison between vision-input and text-input baselines.
5. **Belief Probing**: revealing the model's internal spatial belief by measuring the cogmap's per-object correctness rather than only black-box task accuracy.

Our paper is the first to combine all five.

Briefly, prior work splits into static-image spatial reasoning (1, 3), navigation systems with cogmap memory (2, 4, 5), VLM-driven agentic exploration (6), and our direct predecessor (7).

1. **VADAR** (Marsili et al., 2025) is a 3D spatial-reasoning system that uses cooperating LLM agents to dynamically extend a Python toolkit with new helper functions per query, applied to static images (perception toolkit only; no memory, no exploration).
2. **VLFM** (Yokoyama et al., 2024) performs zero-shot semantic navigation by combining off-the-shelf detection and segmentation modules with depth-derived occupancy maps and a pretrained vision-language model for frontier scoring.
3. **SpatialVLM** (Chen et al., 2024) trains an end-to-end VLM on automatically-generated spatial-QA data; it has no dedicated perception sub-module, no memory, and no exploration.
4. **MapNav** (Zhang et al., 2025) stores spatial context as a per-episode top-down annotated semantic map updated each turn, in place of replaying historical frames.
5. **CogNav** (Cao et al., 2025) uses an LLM to switch between exploration and identification states during object-goal navigation, guided by an online cognitive map.
6. **IVE** (Lee et al., 2025) uses a VLM-driven imagine-verify-execute loop with scene-graph episodic memory for agentic exploration; the spatial scene representation is constructed end-to-end by the VLM rather than via a dedicated perception sub-agent.
7. **Theory of Space** (Zhang et al., 2026a), the most direct predecessor, introduces the benchmark we adopt and the LLM–VLM gap framing but evaluates a bare VLM without perception tools or external memory.

Our paper extends the framework with the full multi-agent pipeline and asks the natural follow-up: does this pipeline close the LLM–VLM gap documented by Zhang et al. (2026a)?

3. Experimental Setup

Environment and tasks. We adopt the 3-room evaluation environment from Zhang et al. (2026a): three connected rooms, four objects per room (12 total) plus inter-room doors. The agent actively explores, choosing actions (rotate, jump, observe, terminate) over up to 20 turns. We measure two complementary signals: nine *evaluation tasks* (downstream task performance, scored after exploration; see Section A) and a suite of *cogmap metrics* measuring the agent's internal spatial belief of the environment (see Section B). Within the cogmap suite, Correctness is computed end-of-exploration against the full ground truth; the remaining dimensions are per-turn measurements averaged across the trajectory.

3.1. Multi-Agent Pipeline

The pipeline introduces four agents alongside the main VLM. All four use the same backbone model as the main VLM in a given run; only the prompts and call-routing differ. Figure 1 shows the per-turn data flow.

BBoxAgent (3 stages: DETECT → COARSE → CHECK). Per turn, BBoxAgent localizes which room-list objects are visible in the current observation and produces a per-object verification. DETECT asks the model to list every visible object from the room roster (free-form names). COARSE overlays a 3×2 grid on the image and asks for the smallest contiguous rectangle (size 2 or 4 cells) covering each detected object. CHECK crops each picked rectangle and asks the model to either KEEP (object clearly visible in crop) or DROP (not recognizable), and to report whether the object's facing is determinable. The output is a list of verified bounding boxes plus per-object metadata. See Section C for an example grid overlay.

LocationAgent. For each verified object, LocationAgent assigns a discrete *bearing* (one of six bins: front-left, front-slight-left, front, front-slight-right, front-right, beyond-FOV) and *distance* (one of four bins: same / near / mid / slightly-far). A single batched call per turn places all visible objects jointly so the model can use cross-object ordering cues. Output is a JSON map {`name: {bearing, distance}`}.

OrientationAgent. For objects with non-symmetric front/back geometry, OrientationAgent receives a crop of the object plus a 3-tile reference (canonical FORWARD, LEFT, RIGHT views) and predicts which reference tile matches the crop. Output is a per-object egocentric facing label (FORWARD, BACKWARD, LEFT, RIGHT, or PRESENT for symmetric objects). See Section C for the canonical reference tiles.

Method	Perception Tools	External Memory	Active Exploration	VLM–LLM Gap	Belief Probing
VADAR (Marsili et al., 2025)	✓	×	×	×	×
VLFM (Yokoyama et al., 2024)	✓	✓	✓	×	×
SpatialVLM (Chen et al., 2024)	×	×	×	×	×
MapNav (Zhang et al., 2025)	×	✓	✓	×	×
CogNav (Cao et al., 2025)	✓	✓	✓	×	×
IVE (Lee et al., 2025)	×	✓	✓	×	×
Theory of Space (Zhang et al., 2026a)	×	×	✓	✓	✓
Ours (this work)	✓	✓	✓	✓	✓

Table 1. Prior multi-modal spatial-reasoning work along five capability dimensions. ✓ = full support; × = absent. Our pipeline is the first to combine all five — extending the Theory of Space benchmark (columns 3–5: active exploration, VLM–LLM gap, and belief probing) with explicit perception sub-agents and an external cogmap (columns 1–2).

MapAgent (the cogmap maintainer). The MapAgent maintains the cumulative cogmap as a JSON object mapping each observed object to {`position: [x,y], facing: cardinal`}. Each turn it receives: the action just taken (e.g., "rotated clockwise 90°"), the new observation (image), the previous cogmap, and — in condition E only — the perception sub-agents' text output for the current turn, which provides per-object bearing, distance, and (where applicable) orientation from the LocationAgent and OrientationAgent. Its system prompt specifies a per-object placement procedure (pick bearing-bin and distance-bin, compute local-frame coordinates via $\sin / \cos$, rotate into world frame using the agent's current facing, round to integer cells). The MapAgent emits three blocks: a `<position_update>` block asserting the agent's new position+facing, a `<placement_math>` block showing the placement derivation per new object, and a `<map_update>` block containing the full updated cogmap as JSON. We use two prompt variants: FULL (condition E, expects perception input) and MEMORY_ONLY (condition D, instructs the model to estimate bearing/distance/facing from the image directly).

Main VLM. The main VLM receives the current observation, the agent's current pose, the cogmap (in D and E), the perception sub-agent output (in C and E), and the list of available actions. It then chooses the next action.

3.2. Conditions

We run five conditions, defined by which agents are active. Conditions A (text-only LLM) and B (vanilla VLM) are inherited from Zhang et al. (2026a) and establish the LLM–VLM baseline gap. Conditions C, D, E are our additions; each ablates a different combination of the multi-agent pipeline so that gains can be attributed to specific components: C isolates the perception sub-agents (addressing B1), D isolates the MapAgent (addressing B2), and E activates the full pipeline (addressing both).

A **Text-only LLM**. No vision; the prompt describes the room declaratively, providing per-object bearing, distance, and (where applicable) orientation as text in place of the image.

B **Vanilla VLM**. Image input only; no sub-agents, no MapAgent.

C **VLM + perception agents**. BBoxAgent, LocationAgent, OrientationAgent run each turn; their text output is concatenated into the main VLM's prompt as a "Perception Output" block. No MapAgent.

D **VLM + MapAgent**. The MapAgent (MEMORY_ONLY variant) maintains a cogmap from images alone; the cogmap is passed to the main VLM each turn. No perception sub-agents.

E **VLM + perception + MapAgent**. Full pipeline. Perception agents feed both the main VLM and the MapAgent (FULL variant); the MapAgent uses perception text to ground its placement math.

Models. Gemini-3.1-Flash-Lite (Google DeepMind, 2026), GPT-4.1-mini (OpenAI, 2025), and Claude Haiku 4.5 (Anthropic, 2025).

Reporting. We report across n=30 different room configurations.

4. Diagnostic Evidence for B1 and B2

This section diagnoses the LLM–VLM gap by testing each candidate failure mode independently. Section 4.1 isolates *perception fidelity* via turn-1 cogmap correctness, before any belief has accumulated — the cleanest test of how well the agent extracts spatial structure from a single observation. Section 4.2 isolates *belief stability* via cogmap correctness

tracked across the 20-turn exploration window — the cleanest test of whether the agent maintains coherent state over time. The synthesis: both candidates are supported. The gap is driven by both perception failures and memory instability, motivating the multi-agent pipeline.

4.1. Perception gap

At turn 1 — before any cogmap state has accumulated — condition B trails condition A on cogmap correctness (Figure 2). The gap concentrates in object orientation: A's facing correctness is essentially perfect (0.96–1.00 across models) while B's sits at 0.70–0.73 — a 25–30pp shortfall. Position and direction gaps exist but are smaller. This perception-extraction-driven framing is consistent with independent work documenting similar VLM perception bottlenecks across diagnostic and benchmark settings (Khezresmaeilzadeh et al., 2026; Song et al., 2026; Cheng et al., 2024).

4.2. Belief decay across turns is universal, not VLM-specific

Figure 3 plots cogmap correctness across the 20-turn exploration window on the same observed-set metric. Both conditions show substantial drift — cogmap correctness drops by more than half from turn 1 for both A and B. Decay magnitudes are comparable, so the VLM disadvantage does not arise from a VLM-specific memory failure. The persistent A–B offset reflects the perception gap from Section 4.1, not a memory-side gap opening over turns. *Caveat:* our three frontier models are small/cost-tier; Zhang et al. (2026a) report on more capable models that LLM belief is stable while VLM belief decays — universal decay at our scale is plausibly capability-driven.

5. Multi-Agent Pipeline Results

5.1. Headline: a three-tier capability pattern

Figure 4 reports per-condition evaluation accuracy for each model, computed as the mean across the nine evaluation tasks of Zhang et al. (2026a) (see Section A for task definitions and per-condition evaluation accuracy for all five conditions A–E). Three distinct outcomes emerge:

- **Gemini-3.1-Flash-Lite (above-threshold):** pipeline lifts accuracy *above* both vision and text baselines (E = 30.9 vs. B = 23.1 vs. A = 27.5).
- **GPT-4.1-mini (at-threshold):** pipeline produces a small but real lift over vision but does not reach text (E = 22.9 vs. B = 21.4 vs. A = 24.5).
- **Claude Haiku 4.5 (below-threshold):** pipeline regresses *below* the vision baseline (E = 16.8 vs. B = 19.5 vs. A = 28.6).

5.2. Cogmap quality improves broadly — but world coverage diverges across models

Why does evaluation accuracy diverge across the three models? Evaluation accuracy is downstream of cogmap quality, so we examine each cogmap dimension to localize the divergence.

The full pipeline reliably improves cogmap quality across most measured dimensions for all three models. We report a few representative dimensions in Figure 5; the full per-dimension breakdown is in Section B. Position stability across turns rises by $\sim$0.30 uniformly; agent self-pose tracking (orientation) rises by 0.26–0.47; orientation correctness rises by 0.13–0.18. This consistency reflects the MapAgent's model-agnostic coherence-enforcement role: it integrates whatever per-turn measurements the perception sub-agents produce into a stable, self-consistent cogmap.

Yet evaluation performance diverges sharply across models: Gemini $+7.8$pp, GPT $+1.5$pp, Haiku -2.7pp. The cogmap dimension that tracks this divergence is world coverage — the fraction of GT objects placed in the final cogmap (Figure 5, rightmost panel; doors and the agent are counted alongside objects). Gemini covers 0.80 of world objects, GPT 0.63, Haiku 0.55 — matching the eval ordering. Coverage depends on two upstream factors: the agent's exploration policy (visiting different rooms, navigating through doors, encountering objects) and the perception sub-agents producing measurements the MapAgent will accept. Both are bottlenecked by underlying VLM capability.

Coverage, not per-object precision, is the binding constraint. Coverage tells us *how many* objects are placed; *precision-on-placed* tells us *how accurately* each placed object is positioned (Table 2; the world coverage we report is position coverage). We find that Haiku matches or outperforms GPT on per-object precision across all three dimensions — Position, Orientation, and Direction. Yet GPT's overall evaluation accuracy exceeds Haiku's by 6.1pp, matching their coverage ordering (GPT $0.63 >$ Haiku 0.55). The binding constraint is therefore *how many of the world's objects make it into the cogmap*, not how accurately each placed object is positioned.

	Precision-on-placed			Coverage		
Dimension	Gem.	GPT	Haiku	Gem.	GPT	Haiku
Position	0.61	0.47	0.53	0.80	0.63	0.55
Orientation	0.40	0.38	0.42	0.81	0.64	0.50
Direction	0.63	0.43	0.43	0.67	0.44	0.34

Table 2. Per-dimension precision-on-placed and coverage in condition E.

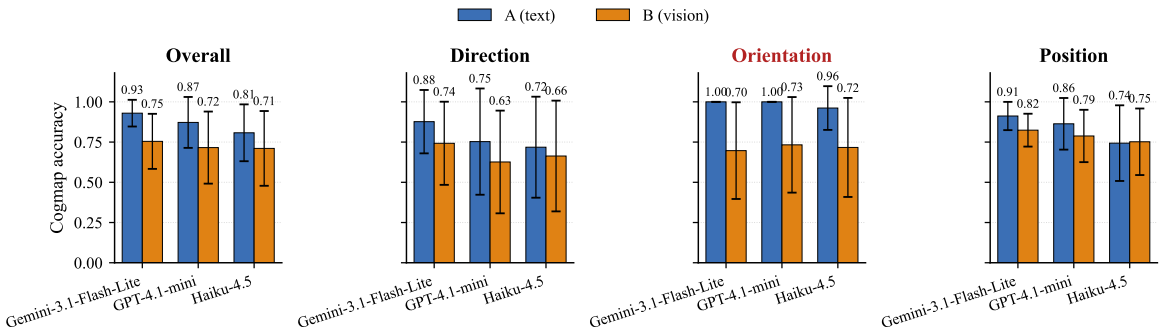

Figure 2. **Turn-1 cogmap correctness for conditions A vs. B across three models.** Bars show mean ± s.d. across n=30 scenes; the predicted cogmap at turn 1 is compared against GT filtered to objects observed so far.

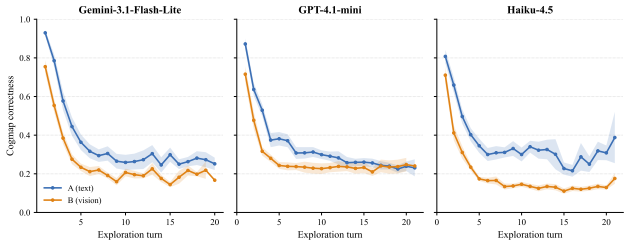

Figure 3. Cogmap correctness over 20 exploration turns (mean ± standard error of the mean across n=30 scenes), per model.

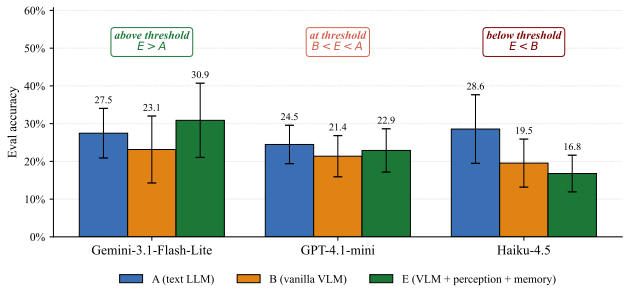

Figure 4. **Three-tier capability pattern.** Eval accuracy (mean ± s.d., n=30) for the bare LLM (A), bare VLM (B), and full multi-agent pipeline (E) per model.

Consequence for evaluation accuracy. The capability threshold operates *upstream* of the cogmap: the multi-agent pipeline reliably stabilizes whatever the VLM produces but does not equalize the VLM's exploration policy across models. Models below the threshold produce stable, accurately-placed cogmaps that simply do not contain enough of the world to support the evaluation suite.

5.3. Each agent targets its intended cogmap dimension

The pipeline assigns each component to a distinct cogmap dimension: perception sub-agents (condition C) target per-turn perception fidelity, and the MapAgent (condition D) targets cross-turn memory. All comparisons below are computed against the vanilla-vision baseline (B). We isolate each by examining *Correctness.overall* — the cogmap's average correctness on object orientation, position, and direction vs. the ground truth (Section B).

Perception tools (C) raise Correctness.overall across all three models (Gemini +1.2pp, GPT +1.9pp, Haiku +3.3pp). In parallel, per-turn perception fidelity rises in step: Perception.ori (orientation correctness on newly-observed objects in the current FOV) lifts by +2.0pp to +4.0pp. Perception sub-agents are doing perception work.

The MapAgent (D) raises Correctness.overall significantly across all three models (Gemini +23.2pp, GPT +12.4pp, Haiku +10.5pp — an order of magnitude larger than the C-induced lift). Driving this lift are large, uniform improvements in memory metrics across the three models: Stability.pos (placement consistency across turns) rises by +27 to +34pp, and SelfTracking.ori (agent's own pose tracked across turns) by +19 to +43pp. The MapAgent is doing memory work.

Each component delivers on its design intent. The combined pipeline (E) inherits both effects, and E achieves the highest evaluation accuracy among the four B/C/D/E conditions for Gemini (30.9%) and GPT (22.9%). Haiku is the exception (E = 16.8%, the lowest among B/C/D/E), where the world-coverage bottleneck identified in Section 5.2 prevents the cogmap-quality gains from translating to evaluation accuracy. The pipeline functions as designed at the cogmap level; whether those representational gains reach evaluation accuracy depends on the model's exploration policy.

6. Discussion

Coverage gates pipeline value. The capability-threshold effect tells us that the multi-agent pipeline's representational improvements (stability, self-tracking, orientation correctness) only become evaluation gains for models whose exploration policy populates enough of the world to support the evaluation queries. The natural next step is to improve the agent's exploration strategy itself — e.g., curiosity-driven

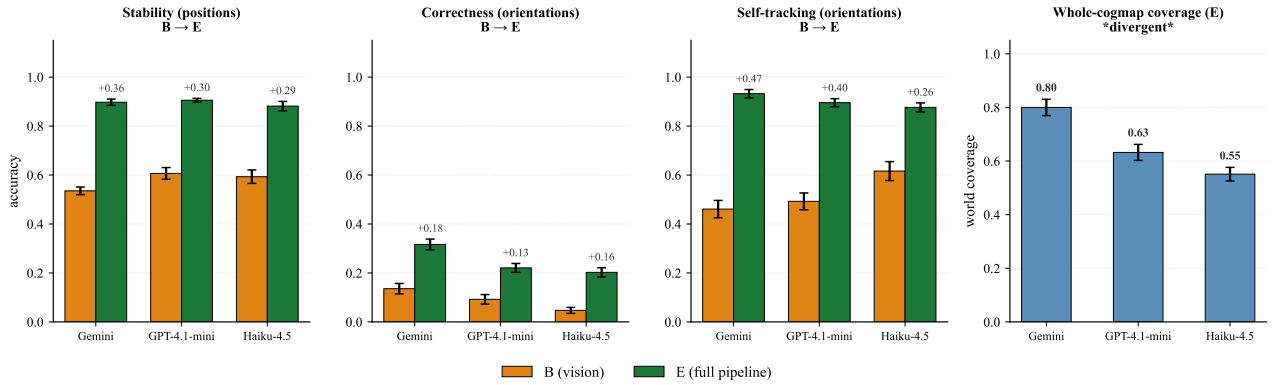

Figure 5. **Left three panels**: B→E pipeline effect on cogmap dimensions per model. **Rightmost panel**: whole-cogmap coverage in condition E. Bars show mean across n=30 scenes; error bars show standard error of the mean (SEM).

exploration or an explicit exploration sub-agent that prioritizes uncovered regions.

External cogmaps help even text-only agents. Both text and vision conditions drift in their cogmap correctness across turns at comparable rates (Figure 3). The maintained-across-turns cogmap we use to mitigate B2 is therefore not a VLM-specific fix; the same component is likely useful for any agent that must reason about a stream of partial observations, regardless of input modality.

Limitations. (i) Single environment (a 3-room layout); larger or different layouts may shift the threshold. (ii) Three frontier models at the small / cost-tier; capability-driven results may not generalize to flagship models. (iii) Our coverage-as-binding-constraint attribution is descriptive: we observe that coverage diverges across models in a way that tracks evaluation performance and that per-object precision does not, but did not run a controlled intervention (e.g., providing one model with another's cogmap) to causally isolate coverage as the binding constraint at evaluation time.

7. Conclusion

We characterize a *capability-threshold effect* in multi-agent VLM spatial-reasoning pipelines: explicit perception tools and external memory reliably improve broad cogmap quality dimensions (stability, self-tracking, orientation correctness), but only translate to evaluation-accuracy gains for sufficiently capable models. Of three frontier VLMs tested, only Gemini-3.1-Flash-Lite clears both the bare-vision and text-only baselines; GPT-4.1-mini lifts above bare vision but does not reach the text baseline; Claude Haiku 4.5 regresses below both. We trace the divergence to a single cogmap dimension that diverges across models even within the pipeline: *world coverage* (the fraction of world objects placed in the cogmap, driven by the agent's exploration policy). Per-object placement precision does not predict evaluation ranking; only coverage does. At the model scales we test, the binding constraint is not the agent's ability to produce and reason about a structured cogmap — but how much of the world it explored to populate that cogmap in the first place.

References

Anthropic. Claude haiku 4.5 system card, 2025. `https://www-cdn.anthropic.com/7aad69bf12627d42234e01ee7c36305dc2f6a970.pdf`.

Cao, Y., Zhang, J., Yu, Z., Liu, S., Qin, Z., Zou, Q., Du, B., and Xu, K. Cognav: Cognitive process modeling for object goal navigation with llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9550–9560, 2025.

Chen, B., Xu, Z., Kirmani, S., Ichter, B., Sadigh, D., Guibas, L., and Xia, F. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14455–14465, 2024.

Cheng, A.-C., Yin, H., Fu, Y., Guo, Q., Yang, R., Kautz, J., Wang, X., and Liu, S. SpatialRGPT: Grounded spatial reasoning in vision-language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Fan, Z., Zhang, J., Li, R., Zhang, J., Chen, R., Hu, H., Wang, K., Wang, P., Qu, H., Zhou, S., Wang, D., Yan, Z., Xu, H., Theiss, J., Chen, T., Li, J., Tu, Z., Wang, Z., and Ranjan, R. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31054–31065, 2026.

Google DeepMind. Gemini 3.1 Flash-Lite model card, 2026. https://deepmind.google/models/model-cards/gemini-3-1-flash-lite/.

Khezresmaeilzadeh, T., Zhong, J., and Psounis, K. Vriq: Benchmarking and analyzing visual-reasoning iq of vlms, 2026. Preprint, arXiv:2602.05382.

Lee, S., Ekpo, D., Liu, H., Huang, F., Shrivastava, A., and Huang, J.-B. Imagine, verify, execute: Memory-guided agentic exploration with vision-language models. In *9th Annual Conference on Robot Learning*, 2025.

Marsili, D., Agrawal, R., Yue, Y., and Gkioxari, G. Visual agentic ai for spatial reasoning with a dynamic api. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19446–19455, 2025.

OpenAI. GPT-4.1, 2025. https://openai.com/index/gpt-4-1/.

Song, Z., Lin, X., Huang, Q., Qin, S., Wang, G., and Lin, L. Siri-bench: Challenging vlms' spatial intelligence through complex reasoning tasks, 2026. Preprint, arXiv:2506.14512.

Tang, Z., Jiao, P., Zhu, B., Qi, H., Chen, J., and Jiang, Y.-G. Spatiotemporal sycophancy: Negation-based gaslighting in video large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14836–14852, 2026.

Wu, D., Liu, F., Hung, Y.-H., and Duan, Y. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. In *Advances in Neural Information Processing Systems*, volume 38, pp. 13569–13597, 2025.

Yokoyama, N., Ha, S., Batra, D., Wang, J., and Bucher, B. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 42–48, 2024.

Zhang, L., Hao, X., Xu, Q., Zhang, Q., Zhang, X., Wang, P., Zhang, J., Wang, Z., Zhang, S., and Xu, R. MapNav: A novel memory representation via annotated semantic maps for VLM-based vision-and-language navigation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13032–13056, 2025.

Zhang, P., Huang, Z., Wang, Y., Zhang, J., Xue, L., Wang, Z., Wang, Q., Chandrasegaran, K., Zhang, R., Choi, Y., Krishna, R., Wu, J., Fei-Fei, L., and Li, M. Theory of space: Can foundation models construct spatial beliefs through active exploration? In *The Fourteenth International Conference on Learning Representations*, 2026a.

Zhang, S., Shen, Q., Wang, S., Pan, T., and Wang, X. Make geometry matter for spatial reasoning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026b.

Zheng, D., Huang, S., Li, Y., and Wang, L. Learning from videos for 3d world: Enhancing MLLMs with 3d vision geometry priors. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

A. Task Suite Reference and Full Per-Task Eval Tables

We adopt the nine-task suite from Theory of Space (Zhang et al., 2026a), organized along two axes: *belief type* (Route, emphasizing egocentric/path reasoning, vs. Survey, emphasizing allocentric mapping) and *dynamic group* (Static — single snapshot reasoning; Forward — given an action / pose, predict the observation; Backward — given an observation, infer the action / pose). Table 3 summarizes the nine tasks; abbreviations match the column headers of Table 4, which reports per-task evaluation accuracy for all five conditions and three models, mean $\pm$ standard deviation across n=30 scenes.

B. Cogmap Metric Reference and Full Tables

Metric definitions. We probe the agent's spatial belief along six dimensions, following the cognitive-map probing protocol of Zhang et al. (2026a). For metrics that decompose into orientation and position components, we report *ori* and *pos* separately; *overall* (Correctness only) is an equally weighted mean of the two components plus a direction-accuracy term.

- **Correctness (final)**: accuracy of the agent's terminal global spatial belief. At the last turn, we score the predicted global cogmap against the ground-truth world. *overall* additionally averages in a pairwise direction-accuracy term, since the global cogmap prioritizes consistent pairwise spatial relations.
- **Perception**: how accurately the agent interprets newly observed local structure. At each turn, we compare the predicted local map to the ground-truth local map for the current field of view (FOV), counting only objects that appear in the FOV for the first time.
- **Local↔Global consistency**: whether new local evidence is incorporated into the global cogmap coherently. Within the same turn, we compare local and global predictions and verify that newly perceived structure is integrated without contradictions.
- **Stability**: whether beliefs about previously observed objects remain non-degrading over time. For each

Dynamic Group	Belief on Route	Belief on Survey
Static	**Pairwise Relation** (*direction*): report allocentric direction and distance from object A to object B.	**Allocentric Mapping** (*alloc.map*): predict global coordinates (and headings) for all objects.
Forward Dynamics	**Perspective Taking** (*persp.take*): output the observation from a specified object's perspective. **Action-to-View** (*act2view*): given a sequence of `Goto`/`Rotate`, predict the final observation (one object in FOV with ego direction/distance bins).	**Mental Rotation** (*ment.rot*): predict the sequence of front-facing objects during a 360° self-rotation. **Location2View** (*loc2view*): given a global pose, predict the observation (one object in FOV with ego bins/distances).
Backward Dynamics	**Perspective Decision** (*perc.dec*): infer which object's perspective the agent is currently adopting. **View-to-Action** (*view2act*): recover an action sequence that produces a target observation.	**View2Location** (*view2loc*): localize the agent (and optionally orientation) from a target observation under the map.

Table 3. **Task suite reference.** Route belief emphasizes egocentric, step-by-step path reasoning; Survey belief emphasizes allocentric mapping and novel-view inference. Definitions follow Zhang et al. (2026a), Table 1.

		Route					**Survey**				
Model	**Cond**	*direction*	*persp.take*	*perc.dec*	*act2view*	*view2act*	*alloc.map*	*ment.rot*	*loc2view*	*view2loc*	**Avg.**
Gemini-3.1-Flash-Lite	A	25.6±16.8	**36.7±24.1**	**35.6±30.2**	**26.7±17.8**	**13.3±20.7**	22.0±10.5	16.7±21.0	30.6±20.6	40.2±16.8	27.5
	B	23.9±19.9	31.1±21.3	25.6±24.3	13.9±13.2	4.4±14.5	20.0±8.3	21.1±27.0	27.2±18.8	41.1±15.7	23.1
	C	22.8±19.3	28.9±16.9	22.2±26.7	22.2±19.2	5.6±12.6	21.0±9.7	14.4±18.9	31.7±15.4	39.1±18.5	23.1
	D	**49.4±21.7**	26.1±15.0	24.4±23.0	15.0±16.0	6.7±16.1	46.5±17.0	**37.8±28.7**	22.8±21.7	41.1±19.4	30.0
	E	46.1±26.9	26.7±19.9	15.6±22.7	26.1±18.4	3.3±10.2	**49.0±17.3**	32.2±33.3	**36.1±25.6**	**42.8±15.8**	**30.9**
GPT-4.1-mini	A	16.1±14.2	**34.4±17.5**	**32.2±29.7**	30.6±18.6	**2.2±8.5**	25.4±8.2	16.7±19.1	27.8±16.6	**34.9±10.3**	**24.5**
	B	15.6±13.8	30.6±20.6	17.8±22.7	**31.7±20.7**	2.2±12.2	27.5±9.6	8.9±15.0	26.1±21.7	32.1±8.8	21.4
	C	13.9±13.2	26.1±22.2	23.3±23.4	23.3±14.9	1.1±6.1	24.8±6.3	13.3±18.8	25.0±19.5	34.2±11.3	20.6
	D	17.8±16.9	25.6±20.9	8.9±21.3	28.3±20.1	0.0±0.0	**35.2±15.5**	15.6±22.7	20.6±19.4	34.3±9.8	20.7
	E	**21.1±16.3**	25.0±18.4	18.9±25.8	29.4±16.2	0.0±0.0	24.1±14.3	**26.7±25.4**	**28.3±21.1**	32.5±12.0	22.9
Claude Haiku 4.5	A	**24.4±20.9**	**37.8±21.9**	**37.8±33.6**	**33.9±17.8**	**6.7±13.6**	**28.6±12.9**	**18.9±27.2**	30.0±22.1	39.1±18.3	**28.6**
	B	13.9±15.2	21.7±18.6	10.0±15.5	23.3±16.7	1.1±6.1	20.4±10.3	15.6±22.7	**30.6±15.2**	39.4±16.7	19.5
	C	12.8±12.1	23.9±16.2	14.4±16.8	28.3±21.1	3.3±13.4	22.3±7.6	14.4±20.9	25.6±20.4	39.5±19.3	20.5
	D	19.4±21.0	17.8±15.7	7.8±16.8	23.9±17.3	0.0±0.0	17.2±15.9	16.7±21.0	23.9±22.2	37.9±19.3	18.3
	E	11.7±15.9	20.0±15.4	10.0±15.5	27.8±16.0	0.0±0.0	12.9±14.4	8.9±19.4	17.8±15.7	**42.0±16.8**	16.8

Table 4. Per-task evaluation accuracy (% mean±SD across n=30 scenes). **Bold** indicates the highest condition for each (model, task) and the highest Avg. per model. Task names follow the Theory of Space convention (Zhang et al., 2026a): Route tasks emphasize egocentric path reasoning, Survey tasks emphasize allocentric mapping. Avg. is the mean across the nine tasks.

		Correctness (%)			**Perception** (%)		**Local↔Global** (%)		**Stability** (%)		**Self-tracking** (%)		**Uncertainty** (%)
Model	**Cond**	*ori*	*pos*	*overall*	*ori*	*pos*	*ori*	*pos*	*ori*	*pos*	*ori*	*pos*	
Gemini-3.1-Flash-Lite	A	23.2±14.4	26.3±9.5	19.6±7.8	**94.8±13.3**	**81.9±4.6**	44.5±19.8	63.1±6.6	**83.8±11.5**	54.6±6.4	48.7±19.6	61.0±12.8	57.5±10.0
	B	13.6±11.7	23.2±11.1	14.9±5.5	28.0±17.6	73.6±6.5	**44.6±19.9**	60.8±9.6	64.0±13.9	53.5±8.5	46.1±19.4	58.9±9.9	35.4±13.3
	C	19.9±14.6	21.0±10.5	16.1±7.3	30.0±18.3	72.7±7.1	34.9±18.5	63.9±6.9	74.0±12.7	52.5±9.1	44.4±16.8	62.8±7.1	58.4±7.8
	D	23.2±9.3	47.3±13.9	38.1±12.2	28.3±18.9	72.5±8.0	37.0±18.2	62.5±8.7	78.0±10.1	87.4±7.9	88.5±17.2	71.5±12.2	58.2±8.2
	E	**31.6±12.0**	**48.4±15.7**	**40.9±12.6**	26.0±16.2	73.3±7.8	36.9±13.5	**66.6±8.7**	81.6±8.0	**89.8±7.1**	**93.2±9.3**	**73.4±12.7**	**62.3±6.5**
GPT-4.1-mini	A	22.0±19.6	31.5±9.6	22.9±9.4	**97.0±5.9**	**73.2±4.4**	42.4±20.1	**64.2±7.0**	81.2±10.5	54.9±6.2	42.8±14.3	52.3±6.7	50.7±7.3
	B	9.2±9.9	26.6±10.3	15.3±7.8	22.0±17.5	58.5±9.4	**69.2±16.1**	57.9±8.8	79.2±12.1	60.7±13.1	49.3±18.8	60.5±15.3	**58.4±6.2**
	C	14.8±11.7	26.0±9.7	17.2±7.7	26.0±17.1	60.5±8.0	53.3±17.2	58.4±6.9	71.1±15.5	54.2±7.9	49.2±16.3	55.4±9.6	47.9±9.2
	D	19.6±13.4	**36.0±13.8**	**27.7±11.1**	29.7±19.9	57.9±10.0	59.7±20.8	61.9±10.4	74.9±18.1	88.8±8.5	**91.8±7.8**	**68.5±11.6**	43.6±10.4
	E	**22.1±9.6**	28.8±10.2	23.0±7.7	27.8±20.1	58.9±9.6	50.2±18.9	62.2±9.5	**90.3±8.4**	**90.6±4.1**	89.6±9.0	61.4±12.2	42.7±8.5
Claude Haiku 4.5	A	**22.9±21.2**	**32.1±10.7**	**23.6±12.3**	**93.9±9.7**	**77.2±7.1**	33.3±19.5	57.5±12.5	82.8±17.9	51.9±14.3	62.4±20.6	66.2±11.6	51.0±10.0
	B	4.7±6.7	21.9±9.5	11.2±5.1	21.3±17.2	52.0±10.2	74.2±14.7	58.9±13.0	78.1±19.2	59.3±15.1	61.6±21.2	62.9±11.0	49.5±10.7
	C	16.8±15.4	20.8±6.7	14.5±7.4	23.7±15.8	54.8±13.3	47.3±17.6	57.4±10.9	63.1±16.2	57.5±12.2	65.6±25.2	65.5±10.2	53.7±11.0
	D	17.1±12.2	31.7±9.5	21.7±8.9	29.6±21.0	54.6±11.8	**75.4±18.1**	65.2±7.4	79.3±20.9	86.8±22.5	80.7±20.4	**74.8±8.6**	54.4±6.6
	E	20.2±10.3	28.2±7.7	20.8±7.1	26.2±18.7	53.1±13.8	51.7±18.2	**65.6±12.6**	**86.9±11.6**	**88.2±10.3**	**87.6±10.1**	74.4±10.2	**54.8±8.1**

Table 5. Cogmap correctness (% mean±SD across n=30 scenes), with orientation (*ori*) and position (*pos*) sub-fields per dimension. Higher is better for all six. *overall* is reported for Correctness only (the equally-weighted mean of *ori*, *pos*, and a pairwise direction-accuracy term). **Bold** indicates the highest condition for each (model, sub-column).

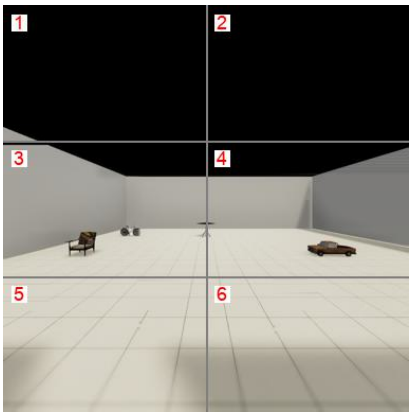

Figure 6. **BBoxAgent example.** The first-person observation is overlaid with a numbered grid; the underlying VLM reports which cells contain each detected object.

previously observed object, at every subsequent turn we check that its predicted state does not worsen; the per-check score is 1 if the prediction is no worse than in the previous turn.

- **Self-tracking**: how well the model estimates its own pose over time. We infer the agent's pose from the predicted global cogmap and compare it against the ground-truth agent state.
- **Uncertainty**: how accurately the agent identifies unexplored regions of the world from its own cogmap. We score the agent's predicted "unexplored" cells against the true unexplored set as a binary F_1 score — the harmonic mean of precision and recall on the unexplored/explored classification, computed over grid cells (fog-probe accuracy).

Table 5 reports the full cogmap correctness numbers across all six dimensions, mean $\pm$ SD across n=30 scenes.

C. Perception Sub-Agent Examples

This appendix illustrates how two of the perception sub-agents operate per turn.

BBoxAgent — coarse localization via grid overlay. Figure 6 shows the COARSE stage of BBoxAgent. Given the agent's first-person observation, the sub-agent overlays a 3×2 grid (numbered 1–6) and asks the underlying VLM to report, for each detected object, the smallest contiguous rectangle of 2 cells (a horizontal or vertical pair) or 4 cells (a 2×2 square) that contains it. Single cells and non-contiguous sets are disallowed. The grid quantizes the localization problem so the model only has to output cell IDs (e.g. "cells 1, 3") instead of free-form pixel coordinates.

OrientationAgent — per-object orientation via reference matching. Figure 7 shows the canonical three-pose reference passed to the OrientationAgent. For each object the BBoxAgent has localized, the OrientationAgent receives two inputs: a crop of the object from the agent's first-person view, and the three-tile reference image showing the same object class facing FORWARD, LEFT, and RIGHT. The sub-agent returns one of four egocentric facing labels: `forward`, `left`, `right` (matching one of the three reference tiles), or `backward` (inferred when the observed crop matches none of the three reference poses — i.e., the agent is looking at the object's back). For symmetric objects without a meaningful front (e.g., vases, lamps, doors), the sub-agent instead receives a 1-tile reference and returns `present`.

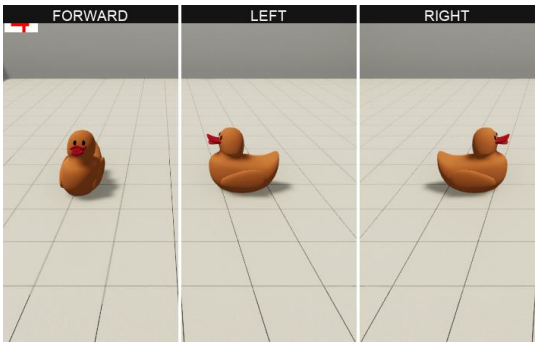

Figure 7. **OrientationAgent reference (*rubberduck*).** The three-pose canonical reference used for orientation matching.