S-SPACE: EXPLORING SPATIAL WORKSPACE IN MULTIMODAL LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

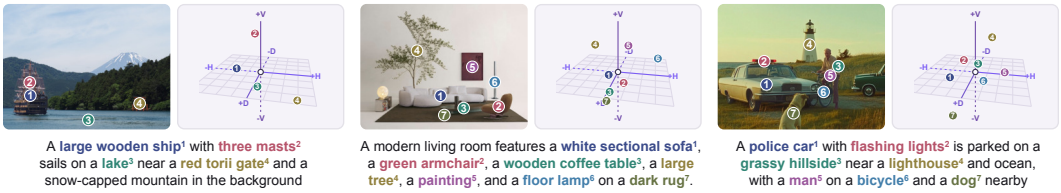
Multimodal language models (MLLMs) exhibit promising spatial reasoning capabilities, yet the internal mechanisms supporting them remain poorly understood. We uncover S-Space, a linear representation subspace in which object-token representations encode continuous coordinates along horizontal, vertical, and distance axes. Although we identify these axes by differentiating spatial-answer scores with respect to object-token activations, S-Space is intrinsic rather than induced by specific data or prompts, remaining readable under non-spatial questions. We show that S-Space functions as an internal spatial workspace of MLLMs. Perception encodes object locations into it, forming spatial representations that can be directly read out to support accurate spatial judgments. Internal reasoning transforms it for changing viewpoints. Verbal responses selectively access it, evidenced by causal interventions that alter spatial judgments while largely preserving non-spatial responses. Case studies further show that S-Space assigns locations to unseen entities and the observer, and connects physical directions with abstract linguistic meanings, suggesting both the capabilities and limitations of large-scale language training. Finally, we use S-Space as a diagnostic tool for spatial reasoning failures. Direction-word readouts reveal systematic coupling between cardinal and image-relative reference frames. More importantly, a perception-computation decomposition exposes a key bottleneck: MLLMs can form capable spatial representations, yet fall short in the spatial transformations required to reason reliably across viewpoint changes. **Code for all experiments is provided in the Supplementary Materials.**

1 INTRODUCTION

Multimodal large language models (MLLMs) (ByteDance Seed, 2026; Kimi Team et al., 2026; Qwen Team, 2026a; Google DeepMind, 2026) have rapidly advanced in spatial intelligence (Yang et al., 2025; Chen et al., 2024), demonstrating utility in reconstruction (Yin et al., 2026; Wang et al., 2026a), planning (Gemini Robotics Team et al., 2025), navigation (Zhang et al., 2024), and manipulation (Brohan et al., 2023; Yuan et al., 2025) in the physical world. Despite this growing range of applications, robust spatial reasoning remains a fundamental challenge for MLLMs, with substantial performance gaps persisting on tasks that humans solve effortlessly (Chen et al., 2026; Wang et al., 2026b; Zhang et al., 2026b). Interpreting these limitations requires looking beyond end-to-end performance: *external* behaviors such as answers and actions alone reveal little about where failures arise, while the *internal* representations and mechanisms remain poorly understood and underdiagnosed.

How is spatial information represented inside an MLLM, and how does it shape the model's responses? Research on human visual cognition offers a useful perspective: there is a *mental workspace* in which internal representations are maintained and manipulated in the human brain (Dehaene et al., 1998; Schlegel et al., 2013). While existing works have explored spatial encodings and representational biases in MLLMs (Kang et al., 2026; Min et al., 2026), a simple and unified explanation of how spatial information is encoded, transformed, and ultimately used to produce responses remains to be established. We ask whether MLLMs similarly possess an internal spatial workspace akin to that of the human brain. Uncovering such a mechanism could help explain spatial reasoning failures and point toward targeted improvements.

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.



(a) Molmo2-ER. (b) Qwen3.5-4B. (c) Qwen3.6-27B.

Figure 1: **Object coordinates read out from S-Space across MLLMs.** For each input image and caption, the intermediate representations of object-name tokens are linearly projected onto the three axes of S-Space to obtain continuous coordinates. Colors and numbers link object names, image regions, and their corresponding coordinates.

In this work, we uncover S-Space, a three-dimensional linear subspace of intermediate representations that functions as an internal spatial workspace in MLLMs. Using a simple gradient-based procedure, we identify fixed spatial orientations that generalize across datasets, tasks, and prompts, defining axes for left-right, above-below, and close-far directions. As illustrated in Figure 1, object locations can then be directly read out as continuous coordinates through linear projection of their representations (Park et al., 2024). Experiments demonstrate that S-Space accurately encodes object locations and provides spatial information that causally shapes the model's judgments, while remaining readable across prompts even when the model is answering questions unrelated to space.

With S-Space, we examine the spatial content represented within MLLMs. Through case studies, we show that the workspace brings visual evidence and linguistic meanings into a shared spatial geometry, extending its content beyond directly visible objects. Our final analysis comes as a diagnosis of spatial reasoning failures, building on S-Space's ability to separate spatial perception from subsequent computation. We study viewpoint-changing problems (Zhang et al., 2026b) by constructing an oracle that transforms S-Space coordinates, revealing that the encoded spatial information supports accurate answers even when the model fails to generate them. Direction-word readouts further reveal systematic coupling between cardinal and image-relative directions, offering insight into reference-frame confusion (Yang et al., 2026). These diagnoses expose a key bottleneck in spatial reasoning: MLLMs can form accurate perceptual representations yet fail to draw reliable spatial conclusions from them.

Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to systematically identify and characterize an internal spatial workspace in MLLMs, revealing how spatial information can be encoded, updated, and causally used within the model's internal representations.
- We propose a simple gradient-based method for identifying S-Space, enabling accurate and generalizable continuous coordinate readouts along fixed spatial directions.
- We establish S-Space as a diagnostic tool for spatial reasoning failures, separating spatial perception from subsequent computation and exposing limitations in how models use available spatial information.

2 METHOD

> **Takeaway:** **S-Space** is a three-dimensional linear subspace of intermediate representations in MLLMs, in which object locations are linearly encoded and can be directly read out.

In this section, we describe how we formulate, identify, and read out S-Space.

2.1 S-SPACE FORMULATION

Large language models are widely hypothesized to represent concepts linearly in their internal representations (Elhage et al., 2022; Bricken et al., 2023). Similarly, we hypothesize that an object's

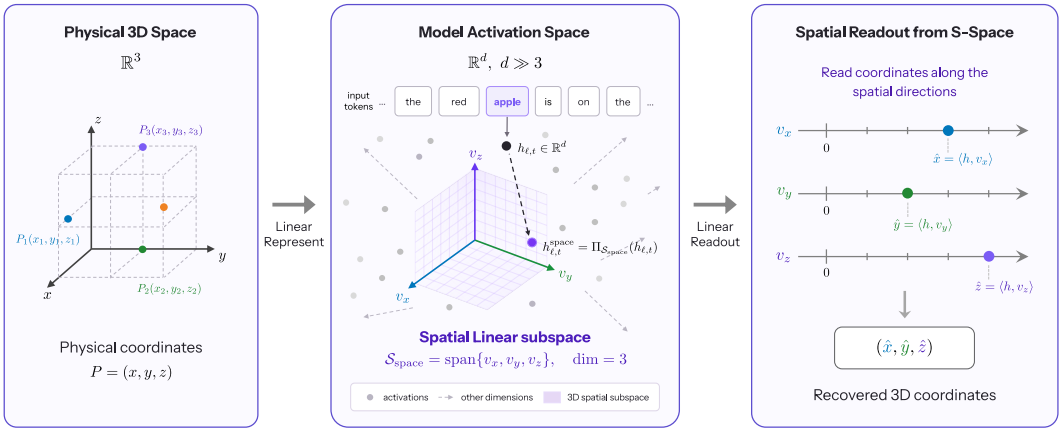


Figure 2: **Linear spatial representation in S-Space.** An object's spatial coordinates *(left)* are linearly embedded in its high-dimensional token representation. Horizontal, vertical, and distance directions span a three-dimensional S-Space *(middle)*. Projections onto them recover viewer-centered coordinates *(right)*.

spatial coordinates, defined in a viewer-centered (or egocentric) reference frame[1], are linearly encoded in the hidden representation of its object-name token (Kang et al., 2026).

Concretely, for a given model and layer ℓ, we define S-Space as a three-dimensional linear subspace of hidden representations $h_{\ell,t} \in \mathbb{R}^{d_{\text{model}}}$. S-Space is specific to the model and layer, but shared across all token positions t within that layer. We search for S-Space in the model's middle layers, as early layers remain relatively anchored to the input representations, and later layers become increasingly aligned with next-token prediction for the production of a verbal response (see layer-wise analysis in Appendix C.1). We fix a layer ℓ and omit its index below for notational simplicity.

2.2 Identifying S-Space

We identify three S-Space directions corresponding to the horizontal, vertical, and distance axes [2] by differentiating the models' responses to relative spatial questions with respect to object-token activations (Gurnee et al., 2026).

Specifically, each relative spatial question example p consists of an image and a prompt comparing a *query* object with a *reference* along one axis $g \in \{x, y, z\}$, e.g., "Is the apple to the left or right of the cup?" Let q and r denote the respective token positions of the query and reference object names[3], we define an answer contrast for each axis as the corresponding logit difference:

$$s_{p,x} = \text{logit(right)} - \text{logit(left)},\ s_{p,y} = \text{logit(above)} - \text{logit(below)},\ s_{p,z} = \text{logit(close)} - \text{logit(far)}. \quad (1)$$

The gradient $\nabla_{h_t} s_{p,g}$ therefore specifies the direction along which perturbing the object representation would most increase the corresponding answer contrast. We then form a role-antisymmetric gradient difference and average it across prompt templates, original/swapped object orders, and examples:

$$a_{p,g} = \frac{\nabla_{h_q} s_{p,g} - \nabla_{h_r} s_{p,g}}{2}, \quad v_g = \text{unit}(\mathbb{E}_p\left[a_{p,g}\right]). \quad (2)$$

Appendix C.1 provides the prompt templates and implementation details.

The resulting unit vectors establish the three S-Space directions: $\mathcal{S} = \text{span}\{v_x, v_y, v_z\}$.

[1] As we will see in Section 5.2 and C.7, this reference frame is not fixed: it can be steered by context and progressively reoriented over the course of chain-of-thought reasoning.

[2] For the horizontal and vertical axes, we use image-plane coordinates rather than world coordinates, without explicitly disentangling depth-dependent perspective effects. These projected coordinates suffice for most spatial reasoning questions considered in this work.

[3] When an object name consists of multiple tokens, we use its last token.

Theoretical sketch. We offer two intuition for Equation (2). (1) Relative spatial judgments treat the query and reference oppositely: a change that makes the query more to the right should favor *right*, while the same change to the reference should favor *left*. Taking the role-antisymmetric difference between gradients therefore reinforces such oppositely acting spatial signals. (2) Other effects arising from object identity, prompt phrasing, token roles, or scene context may not cancel for an individual example, so we further average across prompt templates, original/swapped object orders, and examples. The resulting direction captures the sensitivity that is most consistently associated with the queried spatial relation. See Appendix B for detailed theoretical insights.

2.3 Reading out S-Space

Once we identify the three axes, reading out S-Space requires only a forward pass. Importantly, this readout is not restricted to the spatial QA prompts used for identification: the same fixed directions apply to object-token representations elicited by other prompts and tasks (see Section 3.3). We stack the directions into V and project each object token t to get its coordinate in S-Space:

$$V = [v_x, v_y, v_z] \in \mathbb{R}^{d_{\text{model}} \times 3}, \qquad c_t = V^\top h_t \in \mathbb{R}^3. \tag{3}$$

Relative spatial judgment. Reading out these coordinates supports spatial judgments without requiring the model to verbalize a response. For a relative spatial question, we compare the query and reference coordinates along the queried axis:

$$m_g = v_g^\top (h_q - h_r) = c_{q,g} - c_{r,g}. \tag{4}$$

The sign of margin m_g directly selects the predicted endpoint: a positive value indicates right, above, or close, whereas a negative value indicates left, below, or far.

3 S-Space as Internal Spatial Workspace

> **Takeaway:** Across diverse contexts, S-Space functions as a **persistent** internal spatial workspace in MLLMs, where spatial information is accurately **encoded** into it (§3.1), **transformed** through internal reasoning (§5.2), and causally **used** for verbal responses (§3.2).

In this section, we characterize S-Space as an internal spatial workspace in MLLMs by examining its accuracy, causal role, and persistence across prompting contexts. We identify S-Space for Molmo2-ER (Fang et al., 2026), Qwen3.5-4B (Qwen Team, 2026a), and Qwen3.6-27B (Qwen Team, 2026b) using constructed relative spatial questions from COCO (Lin et al., 2014). To assess the generalizability, we evaluate their readouts on CV-Bench (Tong et al., 2024), EmbSpatial (Du et al., 2024), and SpatialTunnel (Min et al., 2026). All experimental details are provided in Appendix C.

3.1 Accuracy

We first examine how accurately S-Space represents object locations and their spatial relations.

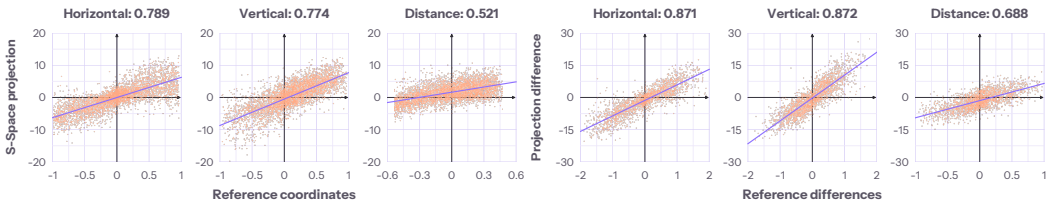
S-Space encodes object coordinates. We project each object-token representation onto S-Space and compare the resulting coordinates with ground-truth object positions. As illustrated in Figure 3a and 3b, both individual object coordinates and pairwise coordinate differences strongly correlate with their ground-truth counterparts along the horizontal, vertical, and distance axes.

S-Space supports spatial QA. We next answer spatial-relation questions using only the coordinates read from S-Space, without relying on the model's verbal response, following Section 2.3. As shown in Figure 3c, this simple coordinate-based readout achieves accuracy comparable to the models' direct verbal answers. Moreover, it generalizes across CV-Bench, EmbSpatial-Bench, and SpatialTunnel, including benchmarks outside the distribution used to identify it.

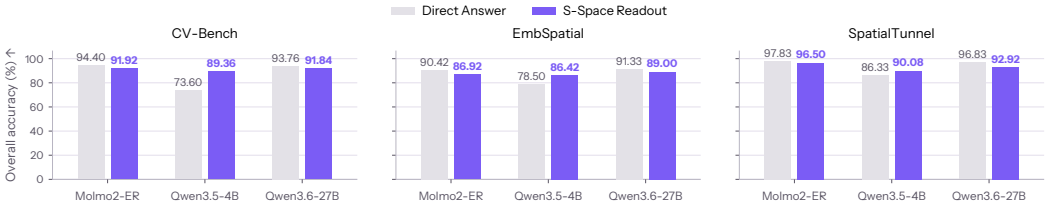
Together, these results show that S-Space **encodes** spatial information with high fidelity.

3.2 Causality

Accurate readout alone does not necessarily imply that the model itself uses S-Space. To test its causal role, we read out object coordinates from S-Space, edit these coordinates, write the modified

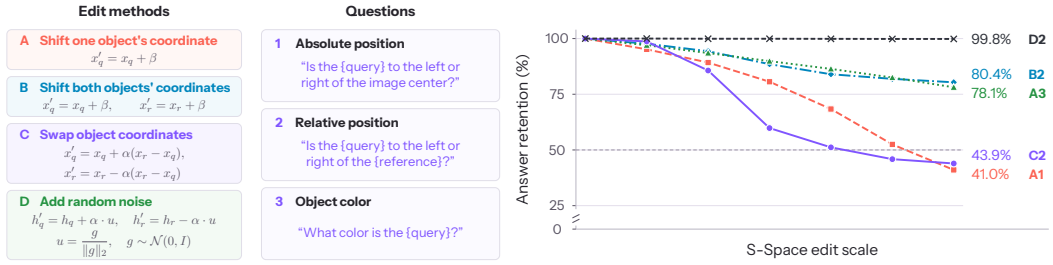


(a) Object coordinates. (b) Pairwise differences.



(c) Spatial QA benchmark accuracy.

Figure 3: **Accuracy of S-Space.** *(a-b)* S-Space coordinates read from layer 21 of Molmo2-ER versus ground-truth object positions, with Spearman correlations and least-squares fits. *(c)* Spatial QA accuracy across benchmarks and models, comparing S-Space readouts with models' direct answers.



(a) Edits and questions. (b) Answer retention.

Figure 4: **Causal intervention of S-Space.** *(a)* Different coordinate-editing and question configurations. *(b)* Answer retention under different interventions and edit scales. Experiments use Molmo2-ER at layer 17 (L17). See Appendix C.3 for details.

values back into the hidden representations, and then resume generation while leaving the image, question, and representation components outside S-Space unchanged.

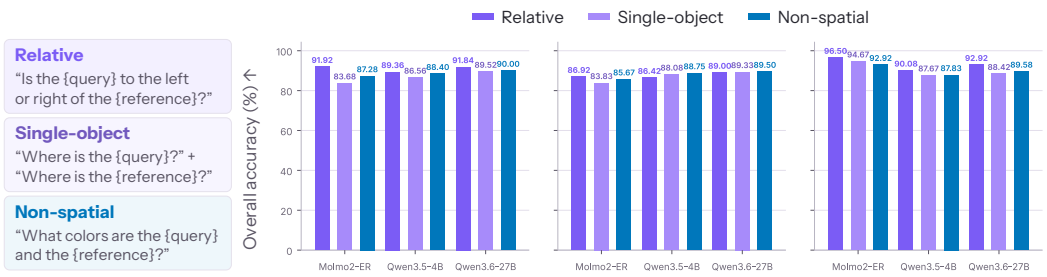
Editing S-Space coordinates changes spatial judgments. As shown in Figure 4, S-Space intervention by coordinate editing substantially alter spatial judgments compared with random editing. The only exception is relative questions under common shifts, where shifting both objects together preserves their relative positions and therefore leaves the judgment merely unchanged.

Non-spatial judgments are largely preserved. The same intervention that strongly alters spatial answers have little effect on unrelated color judgments. This effect suggests that editing S-Space selectively modifies spatial information without broadly disrupting the object representation. Together, these results show that S-Space is causally and selectively **used** when the model produces verbal spatial judgments.

3.3 Persistence

Beyond the dataset generalization shown in Section 3.1, we further replace the relative-comparison prompts with independent single-object queries and even non-spatial questions that ask about colors. Object coordinates are read out in the same way and compared for relative spatial QA.

S-Space persists without explicit spatial questions. As shown in Figure 5, despite substantial change in prompt structure and computational context, the resulting coordinates continue to support accurate spatial judgments. This robustness suggests that S-Space is an **intrinsic and persistent**



(a) Prompt formats. (b) CV-Bench. (c) EmbSpatial. (d) SpatialTunnel.

Figure 5: **S-Space persists across prompts.** The same S-Space readout continues to support accurate spatial judgments on CV-Bench, EmbSpatial, and SpatialTunnel *(b-d)* across pairwise spatial, single-object, and non-spatial prompts *(a)*.

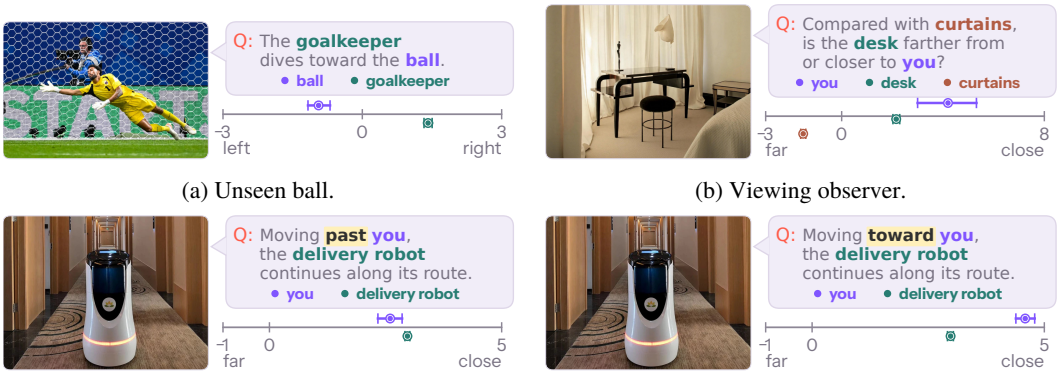
organization of the model's representations, rather than an artifact induced by a particular dataset, prompt template, or pairwise-query format.

4 How does S-Space Emerge? A Case Study

> **Takeaway:** S-Space is hypothesized to emerge from predicting spatial language, which also endows it with broader structure beyond **visible objects** and **physical space**.

We hypothesize that S-Space emerges from the objective of predicting spatial language during large-scale training. To predict tokens such as *left* and *right*, an MLLM must bind visible entities to their relative positions and make this information accessible to language generation. This training pressure may organize internal activations into stable spatial axes without explicit coordinate supervision. Intriguingly, the resulting structure inherited from language may extend beyond visible objects and physical space, as illustrated by the following showcases.

S-Space reflects relations implied by visual evidence. Even when an entity itself is not visible, its position in S-Space can be inferred from visual evidence. As shown in Figure 6a, a goalkeeper diving toward an unseen ball places the token *ball* in the direction of the dive. Beyond unseen objects, even the observer, represented by the abstract token *you*, projects toward the *close* end of the distance axis (Figure 6b). These cases suggest that S-Space does not merely register detected objects, but can complete the surrounding spatial world.



(a) Unseen ball. (b) Viewing observer.

(c) Moving past you. (d) Moving toward you.

Figure 6: **S-Space beyond visible objects.** It localizes off-screen entities *(a)*, places the invisible observer *(b)*, and reflects relations implied by language *(c-d)*. Readout from Molmo2-ER L17. Bars show 95% prompt-template bootstrap intervals.

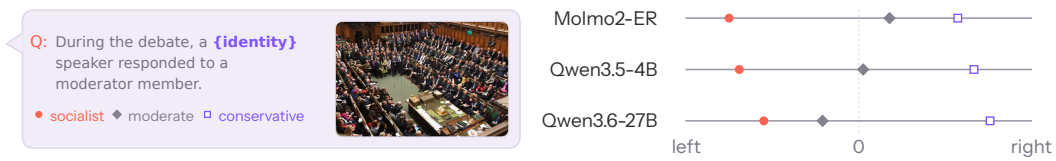


Figure 7: **S-Space beyond physical space**. *Socialist* and *conservative* project toward opposite ends of the horizontal axis of S-Space, mirroring the linguistic metaphor of political left and right.

S-Space reflects relations implied by language. Linguistic context can likewise determine where an entity is represented in S-Space. As shown in Figure 6c and 6d, we keep the delivery-robot image fixed while replacing *moving past you* with *moving toward you*. Correspondingly, the token *you* shifts from a farther to a closer distance coordinate relative to the robot, while the robot's own coordinate remains nearly unchanged.

S-Space blends physical and abstract concepts. Language is a highly abstract representation of the world: spatial terms describe physical geometry, but are also routinely repurposed to express abstract meanings. Models trained on language may therefore inherit associations that do not cleanly separate physical positions from their abstract uses. Strikingly, as illustrated in Figure 7, *socialist*—a concept conventionally associated with the political left—also projects toward the left end of the horizontal axis, and *conservative* toward the right. These cases illustrate both the capabilities and limitations of spatial structure inherited from large-scale language training.

5 Diagnosing Spatial Reasoning Failures with S-Space

> **Takeaway:** MLLMs often possess sufficiently accurate egocentric **spatial representations** in S-Space, yet still fail in downstream spatial reasoning. These failures reveal two key bottlenecks: (1) imperfect **spatial computation** over accurate representations (§5.1), and (2) **reference-frame coupling** when reconciling different coordinate systems (§5.3).

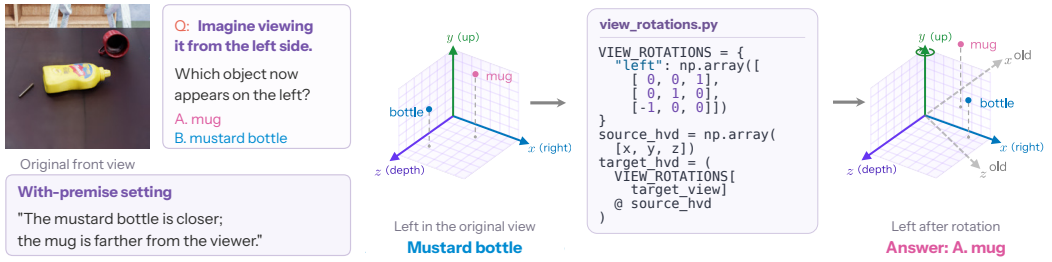
Modern MLLMs still struggle with spatial reasoning, especially when they must build a coherent spatial understanding from partial viewpoints. This requires correctly transforming between viewpoints and reconciling different coordinate frames. S-Space offers a direct view of the model's internal spatial structure, enabling us to diagnose where and how these reasoning failures arise.

5.1 Perception-Computation Decomposition via S-Space

Perspective-taking questions require judging object relations after an imagined change of viewpoint. We use SpinBench (Zhang et al., 2026b) as our evaluation benchmark, which asks which of two objects would appear to the left, or closer to the observer, from a left, right, or back view. Yet frontier VLMs are reported to perform surprisingly poorly on these seemingly simple transformations. This raises a natural question: does the failure arise from spatial perception, or from the computation performed to transform the perceived spatial representation?

S-Space allows us to disentangle these two sources of error by reading the model's internal coordinates independently of its final answer. As shown in Figure 8, we construct a decomposed pipeline that reads the objects' coordinates from S-Space, applies a fixed rotation corresponding to the target viewpoint, and derives the answer with the same deterministic spatial rule as in Section 2.3. We compare this method with the model's native answer generation, both with and without chain-of-thought reasoning (Wei et al., 2022).

Explicit transformation over S-Space substantially improves perspective-taking. As shown in Table 1, direct generation remains limited across all three models. Chain-of-thought improves performance with additional test-time computation, but notable errors persist. In contrast, the prespecified S-Space readout followed by explicit coordinate transformation yields consistently stronger results, which shows that the models encode sufficient spatial information for accurate judgments.



(a) Scene and spatial evidence. (b) S-Space readout and viewpoint rotation.

Figure 8: **Perception-Computation decomposition via S-Space.** *(a)* Perspective-taking question and the optional w/ premise setting. *(b)* We separate spatial perception from spatial computation by reading object coordinates from S-Space, applying the viewpoint transformation explicitly, and deriving the answer with a deterministic spatial rule.

Table 1: **SpinBench perspective-taking accuracy.** Numbers in parentheses show gains over direct answers in the same setting, in percentage points; dashes indicate no reasoning mode.

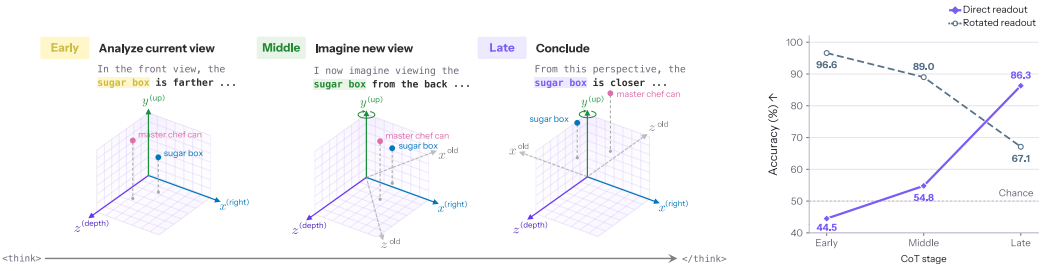
	w/o premise			w/ premise		
Model	Direct	Reasoning	S-Space + Rotation	Direct	Reasoning	S-Space + Rotation
Molmo2-ER	60.96	—	**94.52** (+33.56)	61.81	—	**99.31** (+37.50)
Qwen3.5-4B	60.96	62.33 (+1.37)	**91.10** (+30.14)	60.42	72.22 (+11.80)	**97.92** (+37.50)
Qwen3.6-27B	60.96	83.56 (+22.60)	**95.21** (+34.25)	72.92	91.67 (+18.75)	**99.31** (+26.39)

Reasoning errors persist with explicit spatial premises. Following the premise setting in SpinBench, we further provide the correct source-view relation serves as an oracle for spatial perception, bypassing the model's own perceptual inference (Figure 8a). Even with this relation supplied, models often fail to produce the correct target-view answer. Meanwhile, their S-Space representations become more accurate with language implication, consistent with Section 4, and support near-perfect accuracy. This further highlights a key bottleneck in reliably transforming and operating on correct spatial information during viewpoint reasoning.

5.2 Tracking S-Space Transformation through Reasoning

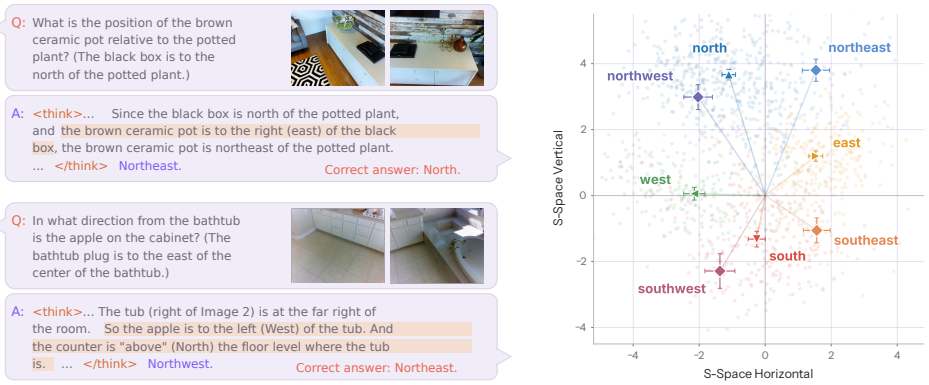
Although still imperfect, chain-of-thought reasoning indeed improves over direct answer generation. We therefore track object coordinates in S-Space throughout the chain of thought to examine how these spatial representations change during the model's own reasoning process. This analysis complements the internal-workspace perspective in Section 3 by revealing internal **transformation** of spatial information in S-Space.

We conduct a controlled analysis on Qwen3.6-27B. Specifically, as shown in Figure 9a, we manually construct a CoT template that explicitly identifies the original spatial relation, imagines the target



(a) S-Space across supplied CoT. (b) Readout accuracy.

Figure 9: **Chain-of-thought progressively transforms S-Space.** *(a)* S-Space coordinates read out from object names at successive stages of a controlled CoT. *(b)* Qwen3.6-27B L43 readout accuracy on SpinBench with and without explicit viewpoint rotation.



(a) Reference-frame errors. (b) Cardinal-direction readouts.

Figure 10: **Direction coupling in S-Space.** *(a)* Representative reasoning failures where image-relative relations are incorrectly reused as cardinal directions. *(b)* S-Space coordinates of cardinal-direction tokens from Qwen3.6-27B chain-of-thought on MMSI-Bench.

viewpoint, and derives the transformed relation. By probing repeated object mentions across these stages, we evaluate how S-Space updates accordingly. At each stage, we read out the S-Space coordinates and use them to answer the corresponding SpinBench perspective-taking question, either directly or after applying the optimal viewpoint rotation.

Reasoning progressively transforms S-Space. As the chain-of-thought unfolds, the objects' relative coordinates become increasingly aligned with the viewpoint requested by the question. This transformation is reflected in a progressive shift from an advantage for the rotated readout at earlier stages to an advantage for the direct readout at later stages, as shown in Figure 9b. This suggests that the model internally simulates part of the required perspective transformation. However, this transformation remains imperfect and does not always produce the correct verbal answer.

5.3 Diagnosing Reference-Frame Coupling in S-Space

S-Space also helps interpret other reasoning errors. In cardinal-direction spatial reasoning, we repeatedly observe reference-frame coupling: models often infer cardinal directions from image-relative relations, mapping *right* to *east* or *above* to *north*, as showcased in Figure 10a. To investigate this phenomenon, using positional relationship tasks of MMSI-Bench (Yang et al., 2026), we collect cardinal-direction tokens (*north*, *south*, etc.) from Qwen3.6-27B's chain-of-thought and read out their coordinates in S-Space.

Cardinal and image-relative directions are coupled in S-Space. As shown in Figure 10b, the resulting geometry makes the coupling explicit: *east* aligns with right in S-Space, *north* with above, and *northeast* reflects both directions. This suggests that the model represents cardinal and image-relative directions through partially shared spatial features rather than fully independent concepts. Inspired by Section 4, we hypothesize that this coupling may partly reflect biases in web-scale pretraining data, where cardinal directions frequently appear in map-like reference frames aligned with horizontal and vertical axes.

6 Discussions

We identified S-Space as an internal spatial workspace whose contents can be measured and operated on to investigate spatial reasoning failures. More broadly, such mechanistic analysis (see related work in Appendix A) helps reveal which key capabilities do and do not emerge through scaling, and for capabilities that remain unreliable, diagnose the missing ingredients. Our results have uncovered biases inherited from pre-training. Another implication is that future systems need not rely on a monolithic MLLM to perform every spatial operation internally. Instead, agentic spatial reasoning (Cho et al., 2026; Luo et al., 2026) can connect dedicated tools for transformation, simulation, and memory, ultimately turning capable perception into reliable reasoning and action.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, drafting and revising portions of the manuscript, language polishing, literature search, reference formatting, and figure preparation.

We have not used generative AI tools to develop theoretical models or conceptual frameworks, design or provide feedback on research methodology or experiments, assist in the writing of proofs, implement experimental methods, generate synthetic datasets, clean or reformat datasets, support qualitative or thematic data analysis, or interpret results. We have not adopted substantive AI-generated suggestions for mathematical assumptions or claims, derivations, critical ingredients of proofs, or the proposal or refinement of research hypotheses.

We have reviewed all AI-assisted work. The authors consulted original references, checked mathematical derivations step by step, tested the experimental code, and verified reported results against the underlying experimental records. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.

ByteDance Seed. Seed2.0 model card: Towards intelligence frontier for real-world complexity. *arXiv preprint arXiv:2607.00248*, 2026.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *CVPR*, 2024.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Y. charles, Yiping Bao, Yuantao Fan, Guopeng Li, Haiyang Shen, Xuanzhong Chen, Wendong Xu, Shuzheng Si, Zefan Cai, Wenhao Chai, Ziqi Huang, Fangfu Liu, Tianyu Liu, Baobao Chang, Ming Wu, Xiaobo Hu, Kaiyuan Chen, Yixin Ren, Yang Liu, Yuan Gong, and Kuan Li. BabyVision: Visual reasoning beyond language. In *ICML*, 2026.

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

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

Kelly Cui, Nikhil Prakash, Shoval Messica, Ayush Raina, David Bau, Antonio Torralba, and Tamar Rott Shaham. The dual mechanisms of spatial variable binding in vision-language models. *arXiv preprint arXiv:2603.22278*, 2026.

Stanislas Dehaene, Michel Kerszberg, and Jean-Pierre Changeux. A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences*, 95(24): 14529–14534, 1998.

Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. EmbSpatial-Bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. In *ACL*, 2024.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable LLM feature circuits. In *NeurIPS*, 2024.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct2: Action reasoning models for real-world deployment. In *CoRL*, 2026.

Shuhao Fu, Esther Goldberg, Ying Nian Wu, and Hongjing Lu. Multimodal function vectors for visual relations. In *ICML*, 2026.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini Robotics: Bringing AI into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Google DeepMind. Gemini 3.5 Flash: Model card. Model card, Google DeepMind, May 2026. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-5-Flash-Model-Card.pdf.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *Transformer Circuits Thread*, 2026. URL https://transformer-circuits.pub/2026/workspace/index.html.

Yining Hong, Jiageng Liu, Han Yin, Manling Li, Leonidas Guibas, Li Fei-Fei, Jiajun Wu, and Yejin Choi. ESI-Bench: Towards embodied spatial intelligence that closes the perception-action loop. *arXiv preprint arXiv:2605.18746*, 2026.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *ICLR*, 2024.

Raphi Kang, Hongqiao Chen, Georgia Gkioxari, and Pietro Perona. Linear mechanisms for spatiotemporal reasoning in vision language models. In *ICLR*, 2026.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Cheng Chen, et al. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Jingli Lin, Runsen Xu, Shaohao Zhu, Sihan Yang, Peizhou Cao, Yunlong Ran, Miao Hu, Chenming Zhu, Yiman Xie, Yilin Long, et al. MMSI-Video-Bench: A holistic benchmark for video-based spatial intelligence. *arXiv preprint arXiv:2512.10863*, 2025.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

Zhanpeng Luo, Ce Zhang, Silong Yong, Cunxi Dai, Qianwei Wang, Haoxi Ran, Guanya Shi, Katia Sycara, and Yaqi Xie. pySpatial: Generating 3D visual programs for zero-shot spatial reasoning. In *ICLR*, 2026.

Cheolhong Min, Jaeyun Jung, Daeun Lee, Hyeonseong Jeon, Yu Su, Jonathan Tremblay, Chan Hee Song, and Jaesik Park. Why far looks up: Probing spatial representation in vision-language models. In *ECCV*, 2026.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *ICML*, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL `https://qwen.ai/blog?id=qwen3.5`.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27b dense model, April 2026b. URL `https://qwen.ai/blog?id=qwen3.6-27b`.

Alexander Schlegel, Peter J. Kohler, Sergey V. Fogelson, Prescott Alexander, Dedeepya Konuthula, and Peter Ulric Tse. Network structure and dynamics of the mental workspace. *Proceedings of the National Academy of Sciences*, 110(40):16277–16282, 2013.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C. Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *NeurIPS*, 2024.

Zifu Wan, Yaqi Xie, Ce Zhang, Zhiqiu Lin, Zihan Wang, Simon Stepputtis, Deva Ramanan, and Katia P. Sycara. InstructPart: Task-oriented part segmentation with instruction reasoning. In *ACL*, 2025.

Hanyang Wang, Yimo Cai, Weiliang Chen, Jiawei Chi, Haowen Sun, Qiyu Dai, Yi-Hsin Hung, Xingzhuo Guo, Jinshan Ren, Runmao Yao, et al. Code as worlds: Agentic discovery of executable world representations for physical reasoning. *arXiv preprint arXiv:2608.27549*, 2026a.

Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Jiajun Wu, Li Fei-Fei, and Manling Li. Spatial mental modeling from limited views. In *ICLR*, 2026b.

Siting Wang, Minnan Pei, Luoyang Sun, Cheng Deng, Yuchen Li, Kun Shao, Zheng Tian, Haifeng Zhang, and Jun Wang. SpatialViz-Bench: A cognitively-grounded benchmark for diagnosing spatial visualization in MLLMs. In *ICLR*, 2026c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *CVPR*, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *NeurIPS*, 2024.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. MMSI-Bench: A benchmark for multi-image spatial intelligence. In *ICLR*, 2026.

Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Chenyang Wang, Xiuyu Li, Michael J. Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning. In *ECCV*, 2026.

Heyang Yu, Yinan Han, Xiangyu Zhang, Baiqiao Yin, Bowen Chang, Xiangyu Han, Xinhao Liu, Jing Zhang, Marco Pavone, Chen Feng, Saining Xie, and Yiming Li. Thinking in 360°: Humanoid visual search in the wild. In *CVPR*, 2026.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. RoboPoint: A vision-language model for spatial affordance prediction in robotics. In *CoRL*, 2025.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. In *RSS*, 2024.

Pingyue Zhang, Zihan Huang, Yue Wang, Jieyu Zhang, Letian Xue, Zihan Wang, Qineng Wang, Keshigeyan Chandrasegaran, Ruohan Zhang, Yejin Choi, et al. Theory of space: Can foundation models construct spatial beliefs through active exploration? In *ICLR*, 2026a.

Yuyou Zhang, Radu Corcodel, Chiori Hori, Anoop Cherian, and Ding Zhao. SpinBench: Perspective and rotation as a lens on spatial reasoning in VLMs. In *ICLR*, 2026b.

A Related Work

Evaluating spatial reasoning. Benchmarks for evaluating the spatial capabilities of MLLMs have become increasingly challenging and comprehensive. Early efforts focused primarily on single-image spatial understanding (Tong et al., 2024; Du et al., 2024), including relative position reasoning and object size estimation. Subsequent studies extended evaluation to multi-view (Yang et al., 2026; Wang et al., 2026b) and video (Yang et al., 2025; Lin et al., 2025; Zhang et al., 2026b) settings, requiring models to integrate observations into coherent spatial representations. These spatial capabilities have been incorporated into embodied tasks involving grounding (Yuan et al., 2025; Gemini Robotics Team et al., 2025) and active exploration (Hong et al., 2026; Zhang et al., 2026a). Nevertheless, spatial reasoning remains a fundamental challenge for MLLMs: visual spatial puzzles (Chen et al., 2026; Wang et al., 2026c) continue to reveal substantial performance gaps relative to humans, motivating deeper diagnosis and mechanistic interpretation of these limitations.

Mechanistic interpretability. Internal representation in LLMs are hypothesized to composed linearly (Park et al., 2024; Elhage et al., 2022). Sparse autoencoders (SAEs) learn dictionaries that recover interpretable features from model activations Huben et al. (2024); Bricken et al. (2023); Templeton et al. (2024). Moving from representation to computation, transcoders approximate MLP computations with sparse features, enabling circuit tracing to identify interactions underlying intermediate reasoning steps (Dunefsky et al., 2024; Ameisen et al., 2025). Most closely related to our work, J-Space (Gurnee et al., 2026) introduces the Jacobian lens, which uses Jacobians to map intermediate activations into the output vocabulary, thereby verbalizing internal representations. Applying this lens across layers reveals internal concepts and intermediate computations. S-Space extends this Jacobian-based perspective to spatial representations in MLLMs, focusing on continuous spatial coordinates rather than discrete verbalizable concepts to investigate how models internally represent and reason about space.

Interpreting spatial representations. MLLMs bind spatial information from visual inputs to the textual representations of object names, enabling both spatial readout and causal intervention (Kang et al., 2026). Prior work have examined spatial binding (Cui et al., 2026), the functional role of attention (Chen et al., 2025; Fu et al., 2026), and bias such as vertical-depth coupling (Min et al., 2026). Much of this work focuses on synthetic settings and two-dimensional pairwise relations. S-Space operates on real-world images and three-dimensional spatial coordinates, providing a simple, unified interface for spatial information readout, causal intervention, and tracking viewpoint transformation.

B Theoretical Insights of S-Space

We now formalize the intuition underlying our procedure for identifying S-Space.

Let $\mathcal{S}^\star$ denote the latent three-dimensional spatial subspace hypothesized in Section 2.1, where u_x, u_y, u_z are unit directions corresponding to the horizontal, vertical, and distance axes. Our procedure in Section 2.2 produces estimated directions v_x, v_y, v_z. Below, we give sufficient conditions under which these estimated directions recover the corresponding latent spatial directions.

Following the linear representation hypothesis (Park et al., 2024; Kang et al., 2026), we assume that distinct types of semantic information are linearly composed within hidden representations. Since object-token representations encode both spatial and non-spatial information, recovering spatial directions requires separating their respective contributions. We introduce the following assumptions to characterize this structure.

Assumption 1 (Representation decomposition)**.** *At an object-token position t, the hidden state can be conceptually decomposed into a spatial component and two non-spatial:*

$$h_t = h_t^{\mathcal{S}^\star} + h_t^{\text{non-spatial}} + h_t^{\text{context}}, \qquad h_t^{\mathcal{S}^\star} \in \mathcal{S}^\star, \qquad h_t^{\text{non-spatial}}, h_t^{\text{context}} \in (\mathcal{S}^\star)^\perp. \quad (5)$$

Here, $h_t^{\mathcal{S}^\star}$ encodes the object's viewer-centered spatial attributes; $h_t^{\text{non-spatial}}$ captures its non-spatial attributes, such as identity, appearance, and function; and h_t^{context} captures information contributed by the question semantics, prompt phrasing, and broader scene context.

Assumption 2 (Gradient decomposition)**.** *Denote $d_{p,t,g} = \nabla_{h_t} s_{p,g}$, the gradient are analogously decomposed into components associated with spatial, non-spatial, and contextual information:*

$$d_{p,t,g} = d^{\mathcal{S}^\star}_{p,t,g} + d^{\text{non-spatial}}_{p,t,g} + d^{\text{context}}_{p,t,g}. \tag{6}$$

Assumption 3 (Gradient behavior)**.** *For a relative spatial question along axis g, the spatial component of the answer-contrast is aligned with the corresponding latent direction u_g and changes sign between the query and reference:*

$$d^{\mathcal{S}^\star}_{p,q,g} = \beta_{p,g} u_g, \qquad d^{\mathcal{S}^\star}_{p,r,g} = -\beta_{p,g} u_g, \tag{7}$$

where $\mathbb{E}_p[\beta_{p,g}] > 0$. Meanwhile:

$$d^{\text{non-spatial}}_{p,q,g} = d^{\text{non-spatial}}_{p,r,g} = 0, \quad \mathbb{E}_p\big[d^{\text{context}}_{p,q,g} - d^{\text{context}}_{p,r,g}\big] = 0. \tag{8}$$

These conditions capture three intuitions. First, the spatial contributions have opposite signs for the query and reference. Second, non-spatial object attributes do not affect the answer contrast. Third, role-asymmetric contextual effects may vary across examples but cancel in expectation. Together, they motivate antisymmetrizing the two object gradients and averaging across prompts and examples.

Property 1. *Under Assumptions 1–3, role antisymmetrization recovers the latent spatial direction u_g in expectation:*

$$\mathbb{E}_p[a_{p,g}] = \mathbb{E}_p[\beta_{p,g}]\, u_g. \tag{9}$$

Consequently, the estimator in Eq. (2) satisfies $v_g = u_g$, up to the choice of orientation, and the recovered subspace $\mathcal{S} = \text{span}\{v_x, v_y, v_z\}$ matches the latent spatial subspace $\mathcal{S}^\star$.

Proof. Substituting the gradient decomposition into Eq. (2) and applying Eq. (7), we obtain

$$\begin{aligned} a_{p,g} &= \frac{1}{2}\left(d^{\mathcal{S}^\star}_{p,q,g} - d^{\mathcal{S}^\star}_{p,r,g}\right) + \frac{1}{2}\left(d^{\text{non-spatial}}_{p,q,g} - d^{\text{non-spatial}}_{p,r,g}\right) + \frac{1}{2}\left(d^{\text{context}}_{p,q,g} - d^{\text{context}}_{p,r,g}\right) \\ &= \beta_{p,g} u_g + \frac{1}{2}\left(d^{\text{context}}_{p,q,g} - d^{\text{context}}_{p,r,g}\right). \end{aligned} \tag{10}$$

The non-spatial terms vanish. Taking expectations eliminates the remaining contextual difference, yielding Eq. (9). Since $\mathbb{E}_p[\beta_{p,g}] > 0$, normalization recovers u_g. □

C Implementation Details and Extended Results

C.1 Identifying S-Space

Models and data. We evaluate Molmo2-ER, MolmoAct2-Pretrain, MolmoAct2 (Fang et al., 2026), Qwen3.5-4B, and Qwen3.6-27B (Qwen Team, 2026a;b). MolmoAct2-Pretrain and MolmoAct2 are VLA models derived from Molmo2-ER after successive stages of action training, allowing us to examine how spatial representations change across these stages. Axis identification and layer selection use disjoint sets of 6,000 and 1,800 COCO-2017 training images, with one object pair per image. For each of the six spatial relations (left, right, above, below, far, and close), we use 1,000 examples to identify the axes and 300 examples to select the layer. The supplementary code provides the image IDs and object pair records for both sets.

We obtain each object's ground truth horizontal and vertical positions from the center of its annotated bounding box, divided by the image width and height, respectively. For depth, we average the inverse depth predicted by Depth Anything V2-Small (Yang et al., 2024) over pixels within each COCO instance mask. Larger values indicate closer objects.

Identification prompts. We use the five templates below for each spatial axis. For each object pair, we fill every template twice, exchanging which object is named in {query} and which is named in {reference}. We compute gradients with respect to the hidden state at the token that ends each object name in the prompt.

Horizontal: left / right

1. Is the {query} to the left or right of the {reference}? Answer with left or right.
2. Where is the {query} relative to the {reference}: left or right? Answer with left or right.
3. In the image, is the {query} to the left or right of the {reference}? Answer with left or right.
4. Which describes the position of the {query} relative to the {reference}: left or right? Answer with left or right.
5. Relative to the {reference}, is the {query} on the left or right? Answer with left or right.

Vertical: above / below

1. Is the {query} above or below the {reference}? Answer with above or below.
2. Where is the {query} relative to the {reference}: above or below? Answer with above or below.
3. In the image, is the {query} above or below the {reference}? Answer with above or below.
4. Which describes the position of the {query} relative to the {reference}: above or below? Answer with above or below.
5. Relative to the {reference}, is the {query} above or below? Answer with above or below.

Distance: far / close

1. Is the {query} farther from or closer to the camera than the {reference}? Answer with far or close.
2. Considering only depth, is the {query} farther from or closer to the camera than the {reference}? Answer far or close.
3. In the image, is the {query} farther from or closer to the camera than the {reference}? Answer with far or close.
4. Choose far or close: is the {query} farther from or closer to the camera than the {reference}? Answer with far or close.
5. Is the {query} farther from the camera than the {reference}, or closer? Answer far or close.

Layer selection. We evaluate validation accuracy for each layer on the 1,800 relative spatial question examples constructed from COCO. For each example, we average S-Space coordinate differences across the five prompt templates for spatial judgments. We select the layer with the highest overall accuracy across the three axes. Table 3 lists the selected layers, indexed from zero after each transformer block. We use these layers unless specified otherwise.

We report S-Space accuracy at every evaluated layer on the COCO selection set in Table 2.

Coordinates and correlations. For the correlations in Figure 3a and 3b, each object's horizontal and vertical ground truth coordinates come from the center of its annotated bounding box. For centering, we linearly map each normalized horizontal and vertical coordinate from $[0, 1]$ to $[-1, 1]$, and center the depth coordinate by subtracting its median. Each axis has 2,116 held-out COCO pairs. Absolute position correlations use the query and reference coordinates from every pair, giving 4,232 readouts. Relative position correlations use the 2,116 differences between query and reference coordinates.

C.2 Reading out S-Space for Spatial QA

We evaluate all five models at their selected layers using the same questions from each benchmark. For CV-Bench (Tong et al., 2024), we use all 1,250 questions in the "Relation" and "Depth" categories. Relation questions compare the horizontal or vertical positions of two objects. Depth questions ask which of two objects is closer to the camera.

For EmbSpatial (Du et al., 2024), we construct a subset of 1,200 questions from the evaluation data released by Min et al. (2026). We use a fixed random seed to select 200 questions for each of six

Table 2: **Layer sweeping accuracy on the COCO validation set.** Layer indices start at zero. Darker purple indicates higher accuracy, and bold marks the selected layer. Qwen3.6-27B is split into two consecutive layer ranges.

Molmo2-ER		MolmoAct2 Pretrain		MolmoAct2		Qwen3.5-4B		Qwen3.6-27B			
Layer	Acc.	Layer	Acc.	Layer	Acc.	Layer	Acc.	Layers 0–31		Layers 32–62	
								Layer	Acc.	Layer	Acc.
0	55.17	0	57.67	0	56.28	0	52.61	0	58.22	32	88.50
1	55.44	1	58.72	1	58.56	1	52.50	1	57.11	33	87.28
2	56.72	2	59.33	2	57.94	2	54.50	2	57.67	34	87.00
3	57.11	3	59.83	3	58.06	3	56.33	3	56.50	35	90.78
4	57.33	4	59.50	4	59.50	4	56.89	4	56.89	36	90.22
5	55.89	5	60.94	5	60.17	5	56.33	5	57.44	37	90.22
6	57.56	6	61.39	6	61.00	6	60.17	6	57.67	38	89.50
7	58.06	7	62.17	7	60.22	7	77.28	7	57.94	39	90.06
8	58.11	8	62.33	8	60.39	8	70.72	8	59.28	40	89.78
9	65.56	9	66.33	9	66.78	9	65.67	9	59.44	41	87.83
10	64.11	10	67.89	10	67.00	10	74.61	10	59.89	42	90.28
11	67.33	11	69.83	11	68.94	11	87.94	11	60.28	43	**91.78**
12	66.72	12	68.39	12	68.83	12	81.94	12	60.11	44	90.72
13	80.44	13	82.28	13	81.83	13	78.39	13	59.28	45	90.22
14	77.78	14	80.44	14	77.83	14	80.39	14	58.61	46	90.28
15	82.83	15	86.22	15	84.00	15	**88.83**	15	64.67	47	88.28
16	87.00	16	87.83	16	88.11	16	85.11	16	64.78	48	88.06
17	89.67	17	88.72	17	88.67	17	82.67	17	68.56	49	85.06
18	92.28	18	**92.22**	18	91.89	18	80.28	18	67.50	50	87.17
19	92.50	19	92.00	19	**92.17**	19	78.28	19	69.50	51	73.39
20	90.67	20	91.61	20	91.94	20	45.61	20	68.78	52	52.94
21	92.28	21	91.94	21	91.39	21	44.56	21	68.39	53	52.50
22	**93.39**	22	90.72	22	89.39	22	45.67	22	67.22	54	48.56
23	77.00	23	67.22	23	64.44	23	50.72	23	69.17	55	47.17
24	68.11	24	61.17	24	56.89	24	51.61	24	69.22	56	47.67
25	70.78	25	65.00	25	61.89	25	50.67	25	68.22	57	46.44
26	32.17	26	35.06	26	33.22	26	49.83	26	74.22	58	47.72
27	40.50	27	41.50	27	39.83	27	51.28	27	86.72	59	44.78
28	46.89	28	52.44	28	51.89	28	49.50	28	84.11	60	45.56
29	48.83	29	53.72	29	54.61	29	49.72	29	81.17	61	46.06
30	47.17	30	51.72	30	53.11	30	50.50	30	79.72	62	44.89
31	48.44	31	52.56	31	53.72			31	89.50		
32	51.17	32	53.22	32	53.56						
33	54.00	33	53.33	33	55.39						
34	54.94	34	51.67	34	50.28						

directions (left, right, above, below, far, and close). For SpatialTunnel (Min et al., 2026), we use all 1,200 questions in the evaluation file, which contains 200 questions for each of the same six directions. Overall accuracy is the percentage of correctly answered questions in each benchmark.

Results We compare direct generation and S-Space readouts for all five models on CV-Bench, EmbSpatial, and SpatialTunnel. Overall accuracy is reported in Table 3, followed by results for each spatial direction in Tables 4, 5, and 6.

Table 3: **Spatial QA accuracy (%).** Direct generates answers, and S-Space compares object coordinates. Each model uses its layer selected on COCO. Bold marks the better method.

		CV-Bench		EmbSpatial		SpatialTunnel	
Model	Layer	Direct	S-Space	Direct	S-Space	Direct	S-Space
Molmo2-ER	22	**94.40**	91.92	**90.42**	86.92	**97.83**	96.50
MolmoAct2-Pretrain	18	**92.32**	91.76	85.33	**90.67**	89.50	**91.25**
MolmoAct2	19	**91.76**	91.44	88.58	**89.92**	89.00	**92.75**
Qwen3.5-4B	15	73.60	**89.36**	78.50	**86.42**	86.33	**90.08**
Qwen3.6-27B	43	**93.76**	91.84	**91.33**	89.00	**96.83**	92.92

Table 4: **CV-Bench accuracy splitted by answer direction.** S-Space uses the selected layer. Bold marks the better method within each model and column, including ties.

Model	Method	Overall	Left	Right	Above	Below	Far	Close
Molmo2-ER	Direct	**94.40**	89.07	**97.00**	**94.44**	**95.93**	**96.33**	**93.33**
	S-Space	91.92	**89.62**	91.50	**94.44**	**95.93**	90.33	92.33
MolmoAct2-Pretrain	Direct	**92.32**	**92.35**	**96.50**	**93.06**	**99.19**	**99.67**	79.00
	S-Space	91.76	88.52	92.00	90.97	94.31	92.67	**92.00**
MolmoAct2	Direct	**91.76**	**94.54**	**95.00**	**95.14**	**96.75**	**100.00**	76.00
	S-Space	91.44	90.71	90.00	90.28	91.06	91.33	**93.67**
Qwen3.5-4B	Direct	73.60	**99.45**	84.50	**97.92**	56.10	**99.67**	20.00
	S-Space	**89.36**	89.62	**89.00**	94.44	**81.30**	85.67	**94.00**
Qwen3.6-27B	Direct	**93.76**	**98.36**	**92.50**	**97.92**	92.68	**93.00**	91.00
	S-Space	91.84	90.16	91.50	95.14	**95.93**	84.33	**97.33**

Table 5: **EmbSpatial accuracy splitted by answer direction.** S-Space uses the selected layer. Bold marks the better method within each model and column, including ties.

Model	Method	Overall	Left	Right	Above	Below	Far	Close
Molmo2-ER	Direct	**90.42**	**92.50**	**95.50**	**97.50**	**97.50**	**88.50**	71.00
	S-Space	86.92	**92.50**	88.50	**97.50**	96.00	73.00	**74.00**
MolmoAct2-Pretrain	Direct	85.33	92.50	**94.00**	98.00	**98.50**	**87.50**	41.50
	S-Space	**90.67**	**95.50**	92.50	**98.50**	97.00	79.50	**81.00**
MolmoAct2	Direct	88.58	**95.50**	**94.00**	97.00	**99.00**	**93.50**	52.50
	S-Space	**89.92**	92.50	90.00	**98.50**	96.50	78.00	**84.00**
Qwen3.5-4B	Direct	78.50	**99.00**	84.50	**99.50**	68.50	**99.00**	20.50
	S-Space	**86.42**	90.50	**94.00**	98.50	**98.00**	47.50	**90.00**
Qwen3.6-27B	Direct	**91.33**	**96.50**	93.00	97.50	96.00	**87.00**	78.00
	S-Space	89.00	93.50	**93.50**	**99.00**	**97.50**	59.50	**91.00**

Table 6: **SpatialTunnel accuracy splitted by answer direction.** S-Space uses the selected layer. Bold marks the better method within each model and column, including ties.

Model	Method	Overall	Left	Right	Above	Below	Far	Close
Molmo2-ER	Direct	**97.83**	99.00	**100.00**	**100.00**	98.50	**100.00**	89.50
	S-Space	96.50	**99.50**	99.00	99.50	**100.00**	85.50	**95.50**
MolmoAct2-Pretrain	Direct	89.50	**100.00**	**100.00**	**100.00**	99.00	**100.00**	38.00
	S-Space	**91.25**	99.50	99.50	97.00	**100.00**	72.00	**79.50**
MolmoAct2	Direct	89.00	**100.00**	**100.00**	**100.00**	99.00	**100.00**	35.00
	S-Space	**92.75**	99.50	99.00	96.50	**100.00**	75.50	**86.00**
Qwen3.5-4B	Direct	86.33	**100.00**	95.50	**100.00**	94.50	**100.00**	28.00
	S-Space	**90.08**	**100.00**	**100.00**	**100.00**	**100.00**	41.00	**99.50**
Qwen3.6-27B	Direct	**96.83**	**100.00**	**100.00**	**100.00**	**100.00**	**94.50**	**86.50**
	S-Space	92.92	**100.00**	99.50	**100.00**	**100.00**	80.00	78.00

C.3 Causal Interventions

Editing hidden states. We edit an object's hidden state to change its S-Space coordinates. Let h denote this hidden state and $V = [v_x, v_y, v_z]$ contain the spatial readout directions. To change the coordinates from $\mathbf{c} = V^\top h$ to $\mathbf{c}'$, we apply

$$B = V(V^\top V)^{-1}, \qquad h' = h + B(\mathbf{c}' - \mathbf{c}). \tag{11}$$

This update gives $V^\top h' = \mathbf{c}'$ when the three axes are linearly independent. It preserves the component of the hidden state orthogonal to S-Space.

Edits. Let c_q and c_r denote the query and reference coordinates on the edited axis, and $h_q, h_r \in \mathbb{R}^d$ their hidden states. We implement four edits:

$$\begin{aligned}
\text{Shift one object:} \quad & c'_q = c_q + \beta, && c'_r = c_r, \\
\text{Shift both objects:} \quad & c'_q = c_q + \beta, && c'_r = c_r + \beta, \\
\text{Swap:} \quad & c'_q = c_q + \alpha(c_r - c_q), && c'_r = c_r - \alpha(c_r - c_q), \\
\text{Random edit:} \quad & h'_q = h_q + \alpha m_q u, && h'_r = h_r - \alpha m_r u.
\end{aligned}$$

Here, β is a signed offset. Shifting both objects preserves their coordinate difference $c_q - c_r$. For swaps, $\alpha = 0$ leaves the coordinates unchanged, $\alpha = 1$ exchanges them, and larger values move them farther apart in the reversed order. We note that amplified intervention is a common practice in related literature of mechanistic analysis (Gurnee et al., 2026; Kang et al., 2026). The shifts and swaps keep the other two coordinates fixed.

For random edits, m_q, m_r are the norms of the query and reference swap updates at $\alpha = 1$. We multiply these norms by α to set the random edit magnitudes:

$$\begin{aligned}
m_t &= \|h_t^{\text{swap}}(\alpha = 1) - h_t\|_2, \quad t \in \{q, r\}, \\
u &= \frac{g}{\|g\|_2}, \quad g \sim \mathcal{N}(0, I).
\end{aligned}$$

Here, $h_t^{\text{swap}}(\alpha = 1)$ denotes the hidden state after a swap with $\alpha = 1$. The query and reference receive opposite directions, each with the same norm as the corresponding swap update at strength α. For each example, we keep u fixed across edit strengths.

Questions. We ask about absolute position, relative position, or object color using the following prompts. The position questions below showcase the horizontal axis.

1. Absolute position

Is the {query} to the left or right of the image center? Answer with only one word.

2. Relative position

Is the {query} to the left or right of the {reference}? Answer with only one word.

3. Object color

What color is the {query}? Answer with only one word.

Main results. We use Molmo2-ER for all interventions. Figure 4 uses 800 EmbSpatial examples at L17. In Figure 4, swaps and random edits use $\alpha = 0, 1, \ldots, 6$, and shifts use $\beta = 0, \pm 5, \ldots, \pm 30$. Table 7 reports the answer retention values plotted in Figure 4.

Table 7: **Answer retention under S-Space intervention** (Figure 4).

Edit method	Question	Editing scale and answer retention (%)						
Shift one object	Absolute position	$\beta = 0$	±5	±10	±15	±20	±25	±30
		100.00	95.15	89.23	80.62	68.37	52.49	41.05
	Color	$\beta = 0$	±5	±10	±15	±20	±25	±30
		100.00	97.00	93.50	89.88	86.38	82.50	78.13
Shift both objects	Relative position	$\beta = 0$	±5	±10	±15	±20	±25	±30
		100.00	97.67	94.32	88.50	83.98	81.91	80.36
Swap	Relative position	$\alpha = 0$	1	2	3	4	5	6
		100.00	98.71	85.66	59.82	51.16	45.87	43.93
Random edit	Relative position	$\alpha = 0$	1	2	3	4	5	6
		100.00	100.00	99.91	99.91	99.87	99.83	99.83

Layer Sweep The interventions at L17 show that editing S-Space changes spatial judgments. We repeat these interventions across layers to examine where this causal effect appears.

We use Molmo2-ER at L13–L20, with 80 examples per task at each layer. For Table 8, we apply the *Swap* and *Shift both objects* edits and ask *Relative position* questions. For Table 9, we apply the *Shift one object* edit along the horizontal axis and ask *Absolute position* and *Object color* questions.

Table 8: **Layer-sweeping results for relative position interventions.** Number refers to the accuracy (%) of relative position questions under corresponding interventions on 80 examples per layer. Unedited accuracy is 97.50%.

Layer	Swap ($\alpha = 5$)	Shift both object ($\beta = -30$)	Shift both object ($\beta = +30$)
13	73.75	85.00	81.25
14	85.00	81.25	83.75
15	50.00	75.00	80.00
16	60.00	92.50	93.75
17	62.50	93.75	93.75
18	43.75	96.25	91.25
19	85.00	96.25	95.00
20	72.50	96.25	92.50

Table 9: **Layer-sweeping results for absolute position and color interventions.** Number refers to the ratio (%) of samples for which the answer changes under either a $+\beta$ or a $-\beta$ intervention at the corresponding layer..

	Absolute position			Color
Layer	$\lvert\beta\rvert = 10$	$\lvert\beta\rvert = 30$	$\lvert\beta\rvert = 50$	$\lvert\beta\rvert = 30$
13	1.25	30.00	76.25	11.25
14	5.00	47.50	83.75	12.50
15	11.25	75.00	93.75	17.50
16	11.25	57.50	73.75	13.75
17	10.00	52.50	55.00	11.25
18	5.00	35.00	43.75	13.75
19	2.50	8.75	6.25	8.75
20	2.50	12.50	11.25	2.50

C.4 Case Studies

Unseen entities and the observer. We present additional examples of S-Space readouts for unseen entities and the observer in Figure 11.

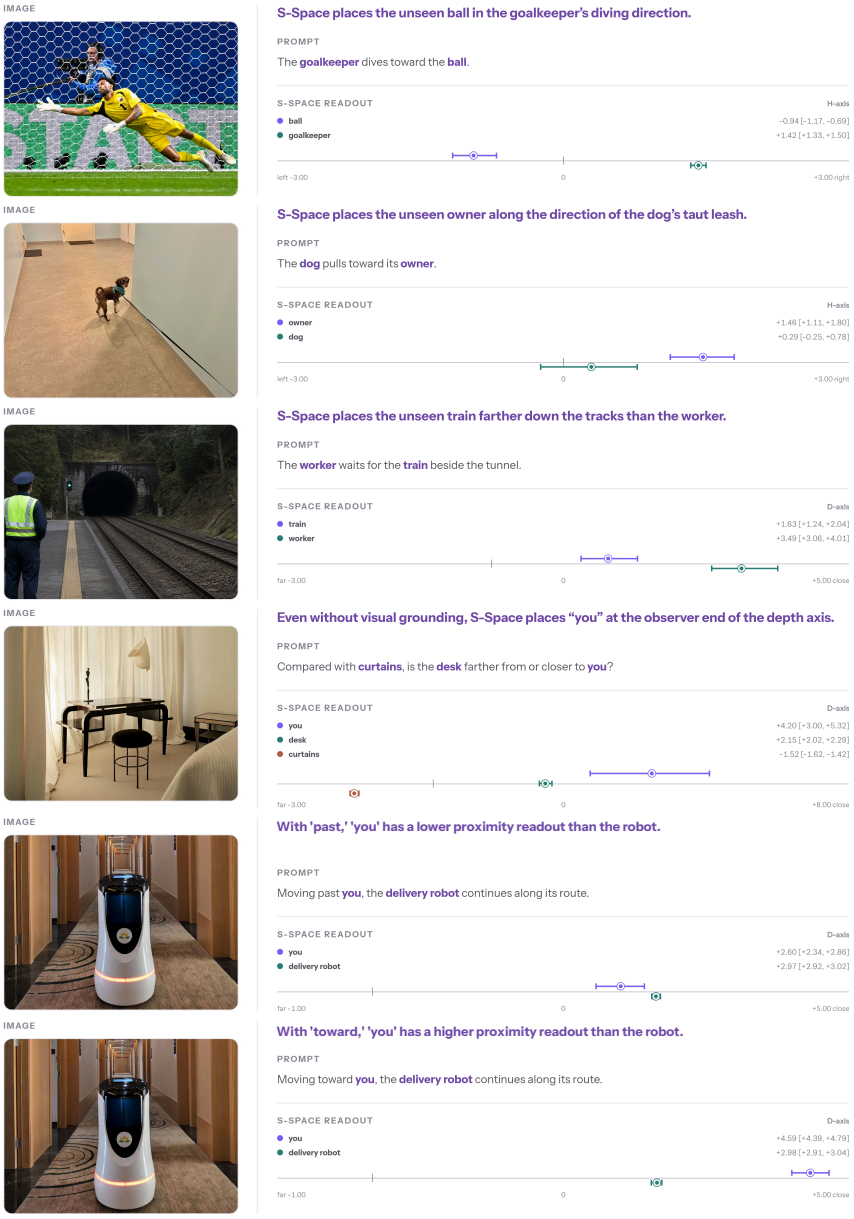


Figure 11: **S-Space beyond visible objects.** Images, prompts, and horizontal or distance readouts for an unseen ball, owner, train, the observer, and a robot moving past or toward. Readouts use Molmo2-ER L17. Means and 95% bootstrap intervals are computed across prompt templates.

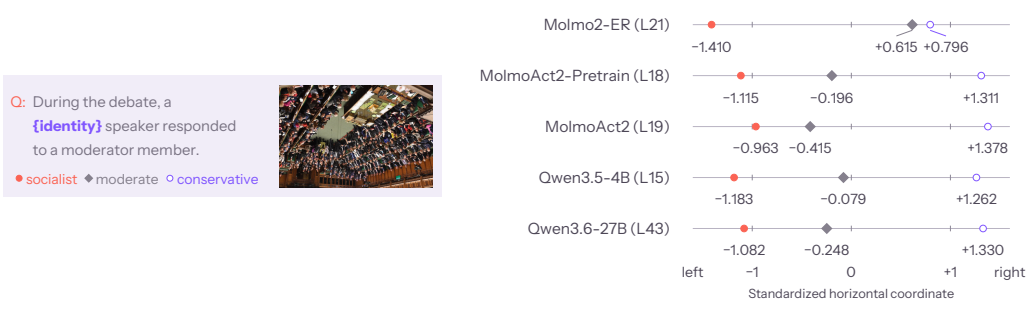


Figure 12: **Political semantics in S-Space.** Standardized horizontal readouts across five models.

Political semantics. The political semantic results in Figure 12 include MolmoAct2-Pretrain and MolmoAct2 alongside the three models in Figure 7, with the readouts standardized. All five models preserve the ordering of *socialist*, *moderate*, and *conservative*.

Cardinal direction readouts. We select 191 of the 1,000 MMSI-Bench questions (Yang et al., 2026) by matching the correct option text to one of eight cardinal directions. The supplementary code includes the selected question IDs. We give Qwen3.6-27B the following instruction.

Cardinal direction readouts

Give a concise step-by-step spatial explanation in at most five steps. End exactly with 'Answer: X', where X is the option letter.

We replay each generated token sequence with the original images and prompt to extract hidden states at L43. For each occurrence of a direction word, we average the hidden states of its subtokens and project the result onto the horizontal and vertical axes identified on COCO. Compound directions written as one word, with a hyphen, or with a space are assigned to the same category and counted as a single occurrence.

Figure 10b presents 1,311 occurrences from both correct and incorrect responses. For each direction, we first average its occurrences within each question, then average the results across questions with equal weight.

C.5 Perspective-Taking Spatial Reasoning

Data and generation. We use the official SpinBench split consisting of 146 questions without a premise and 144 with a premise. Each question provides one image and asks which object would appear on the left or closer to the observer when viewed from the left, right, or back. Direct generation uses greedy decoding with a limit of 128 tokens. The Qwen Reasoning baselines use chain-of-thought generation with a limit of 32,768 tokens. S-Space + Rotation uses the layers selected on COCO in Table 3. All methods use the same examples and object order.

Viewpoint computation. For objects A and B in the benchmark options, we compute the coordinate difference along axis g in the original view as

$$m_g = v_g^\top (h_A - h_B).$$

With x increasing to the right and z increasing toward the camera, we transform the coordinates using

$$R_{\text{left}}(x, z) = (z, -x), \qquad R_{\text{right}}(x, z) = (-z, x), \qquad R_{\text{back}}(x, z) = (-x, -z).$$

After rotation, we select A if $x' < 0$ when the question asks which object appears on the left, or $z' > 0$ when it asks which object is closer to the observer. We select B otherwise. The implementation applies the following matrices.

Viewpoint transformation

```
VIEW_ROTATIONS = {
  "front": np.array([[ 1, 0,  0], [0, 1, 0], [ 0, 0,  1]]),
  "left" : np.array([[ 0, 0,  1], [0, 1, 0], [-1, 0,  0]]),
  "right": np.array([[ 0, 0, -1], [0, 1, 0], [ 1, 0,  0]]),
  "back" : np.array([[-1, 0,  0], [0, 1, 0], [ 0, 0, -1]]),
}

source_hvd = np.array([x, y, z])  # right, below, close
target_hvd = VIEW_ROTATIONS[target_view] @ source_hvd
```

Controlled CoT. For Figure 9, we supply the following reasoning sequence: the relation in the original view, the target viewpoint, and the relation after rotation. We fill in both relations using the benchmark labels.

Controlled CoT

The user wants me to determine {TARGET_QUESTION} if the scene were viewed from the {TARGET_VIEW}.

1. **Analyze the current view:**

* In the front view, the {QUERY} is {FRONT_VIEW_RELATION} the {REFERENCE}.

2. **Imagine the new perspective:**

* I now imagine viewing the {QUERY} and the {REFERENCE} from the {TARGET_VIEW}.

3. **Conclusion:**

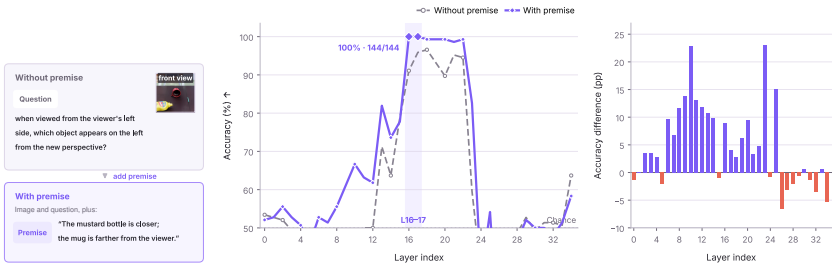
* From this perspective, the {QUERY} is {ROTATED_VIEW_RELATION} the {REFERENCE}.

Main results. SpinBench results for all five models are reported in Table 10. The comparison covers Direct, Reasoning, and S-Space + Rotation, with and without spatial premises.

Table 10: **SpinBench accuracy across all five models.** Direct generates an answer without intermediate reasoning. Reasoning uses chain-of-thought. S-Space + Rotation applies the requested viewpoint rotation to the readout coordinates. A premise supplies the correct spatial relation in the original view. Dashes indicate no reasoning mode. Bold marks the best method.

	Without premise ($n = 146$)			With premise ($n = 144$)		
Model	Direct	Reasoning	S-Space + Rotation	Direct	Reasoning	S-Space + Rotation
Molmo2-ER	60.96	–	**94.52**	61.81	–	**99.31**
MolmoAct2-Pretrain	66.44	–	**96.58**	56.25	–	**99.31**
MolmoAct2	71.23	–	**95.89**	64.58	–	**99.31**
Qwen3.5-4B	60.96	62.33	**91.10**	60.42	72.22	**97.92**
Qwen3.6-27B	60.96	83.56	**95.21**	72.92	91.67	**99.31**

S-Space with additional evidence. For Molmo2-ER, we further compare S-Space accuracy across layers with and without spatial premises. As shown in Figure 13, with the premise included, the coordinates read from S-Space become more accurate, allowing the readout-and-rotation pipeline to answer perspective-taking questions more reliably.



(a) Source-view premise **(b)** Layerwise accuracy **(c)** Accuracy difference

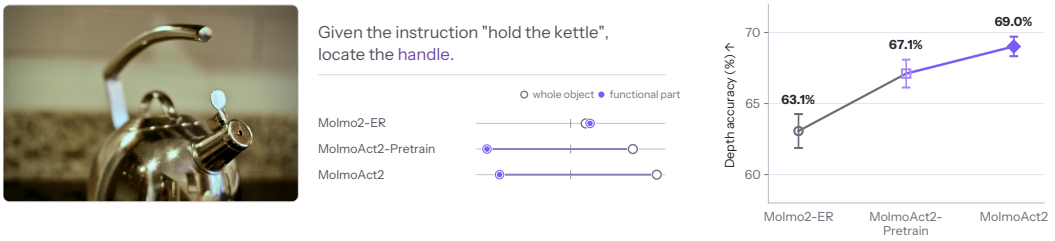
Figure 13: **Molmo2-ER readouts with spatial premises.** **(a)** Source-view premise. **(b)** Layerwise accuracy with ($n = 144$) and without ($n = 146$) premises. **(c)** Accuracy difference.

C.6 S-Space under Action Training

As shown in Section 4, language translates space into words and shapes the emergence of S-Space. Action, by contrast, translates space into movement, providing a more precise interface with the physical world. To reach, grasp, or manipulate an object, a model must resolve both the geometry of the scene and the functional parts through which interaction occurs. We therefore ask whether learning to act sharpens the geometry in S-Space by comparing a VLM with its fine-tuned VLA models.

Models and data. We examine Molmo2-ER, MolmoAct2-Pretrain, and MolmoAct2 on images from the InstructPart dataset (Wan et al., 2025), in which a manipulable object contains a characteristic functional part with estimated depth of the part and the whole object. We construct prompts such as "... Locate the [object]" and assess the S-Space readout on the distance axis by comparison between the part and the whole, as in Section 2.3. Ground-truth depth comparison use the provided object and part masks and Depth Anything V2-Small Yang et al. (2024).

Results. As shown in Figure 14, we find an orderly progression along the model family: from the VLM through the action-model pretrain to the VLA, this local geometry becomes increasingly faithful to the physical relation between them.



(a) Object and part readouts **(b)** Depth-comparison accuracy

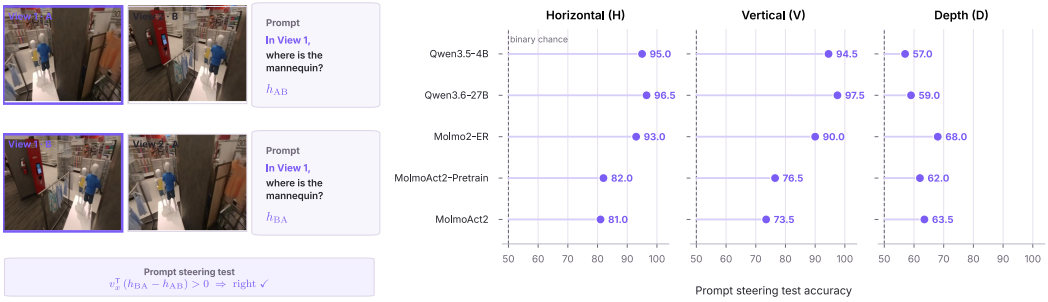
Figure 14: **Action training sharpens part–whole depth geometry in S-Space.** *(a)* Depth readouts for the kettle and its handle. *(b)* Depth comparison accuracy at L18 for all three models. Error bars indicate 95% confidence intervals across prompts.

C.7 S-Space in multi-view contexts

Although S-Space is identified entirely from single-view examples, a workspace should still provide meaningful spatial coordinates when multiple views of the same scene are presented. We examine whether the model can use the prompt to select which view defines the observer-centered frame.

View pairs and prompt steering. We construct 600 view pairs in panoramic scenes from Yu et al. (2026), with 100 per direction. The same object appears in both views. As illustrated in the prompt-steering test (Figure 15a), we keep the prompt fixed on "In View 1" while swapping which image is designated as View 1, and test whether the resulting change in an object's S-Space coordinates follows the corresponding change in viewpoint.

Results. Across models, steering accuracy is consistently well above random choice along the horizontal, vertical, and distance axes (Figure 15b). These results suggest that S-Space is not tied to a single visual input, but can be dynamically anchored to the viewpoint selected through language.



(a) Prompt steering test **(b)** Steer accuracy

Figure 15: **Prompt steering test of S-Space in multi-view contexts.** *(a)* Each example is evaluated twice with the same View-1 query while the two input images are swapped. *(b)* We evaluate the sign agreement between the resulting change in the S-Space coordinates and the object's ground-truth displacement between the two views.