SPACENUM: Revisiting Spatial Numerical Understanding in VLMs

Jianshu Zhang* Northwestern
Yijiang Li* UCSD
Huifeixin Chen USC
Haoran Lu Northwestern
Letian Xue Northwestern
Bingyang Wang GaTech
Han Liu Northwestern

Abstract

Vision-Language Models (VLMs) are increasingly deployed in embodied environments, where they must to produce numerical outputs such as action magnitudes and spatial coordinates. Although these numbers appear meaningful, it remains unclear whether these numerical outputs are genuinely grounded in spatial perception. Therefore, in this work, we revisit spatial numerical understanding through SPACENUM, a unified framework that captures two complementary settings: numbers as dynamic transitions during spatial exploration, and numbers as static layouts in spatial reasoning. We formulate two bidirectional tasks, NUM2SPACE and SPACE2NUM, to evaluate how well VLMs map between vision-side spatial structure and language-side numerical representations. We systematically study whether current VLMs truly understand numerical values in spatial settings. Across dynamic transitions and static layouts, we find that models largely fail to ground numbers in spatial meaning and often perform close to random guess. Through error analysis, reasoning trace analysis, and controlled interventions, we show that current VLMs rely heavily on shallow spatial cues, struggle to build stable coordinate-aware representations, and fail to abstract structured spatial layouts from visual observations. We further show that explicit reasoning provides only marginal gains, while tuning can partially improve spatial numerical understanding and transfer to external spatial reasoning benchmarks.

1 Introduction

Vision-language models (VLMs) have recently progressed from describing what is directly visible in images [6, 16, 22] to actively exploring and understanding complex spatial environments [24, 11, 30, 9, 19]. Two representative spatial task scenarios have emerged: (1) **spatial exploration**, where a VLM-based agent navigates an environment by generating actions conditioned on its observations to actively gather information; and (2) **spatial understanding**, where VLMs infer the global structure of a scene and answer spatially grounded questions by constructing an internal representation of the environment. As illustrated in Figure 1, despite their different objectives, both paradigms share a common requirement: VLMs must produce explicit numerical values whose meanings are grounded in spatial context.

In spatial exploration [31, 27], a VLM-based agent may output an action such as "*rotate_left*(20°)". The value 20 does not describe the current observation, nor does it directly specify the next observation. Instead, it specifies the magnitude of a state change, serving as a transition quantity between consecutive observations, where numbers naturally function as ***dynamic transition magnitudes***.

In contrast, in spatial understanding, prior work has shown that constructing explicit spatial representations [32, 30, 10], often in the form of cognitive maps, improves performance on spatial reasoning

Preprint.

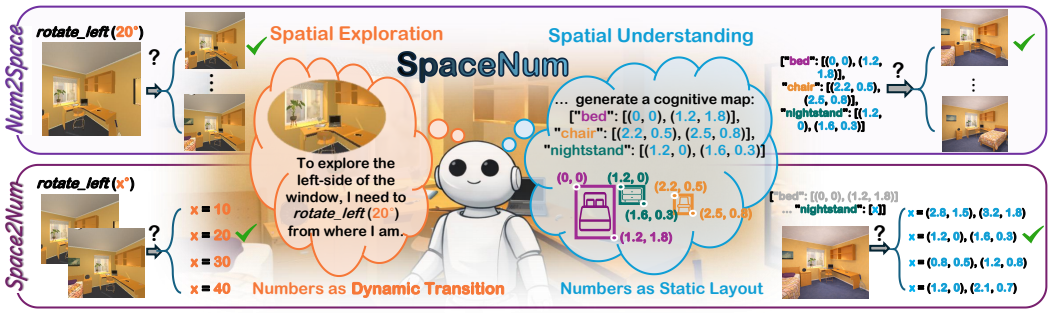


Figure 1: Overview of SPACENUM. We study spatial numerical understanding under two settings: *numbers as dynamic transition* in spatial exploration (left) and *numbers as static layout* in spatial understanding (right). We further investigate the mapping between vision-side space and language-side numbers via two tasks: NUM2SPACE, which maps numbers to visual outcomes (top), and SPACE2NUM, which maps visual inputs to numbers (bottom).

tasks. Here, numbers encode relative spatial relationships and correspond to ***static relative spatial layouts***. A single object's coordinates in isolation carry limited semantic meaning; spatial information becomes interpretable only when multiple objects are considered within a shared coordinate system, where numerical values define their relative positions and overall layout.

This naturally raises a key question: **do VLMs genuinely understand numbers as metric quantities in space and generate them grounded in metric properties of space?** Across both spatial exploration and spatial understanding, **NUM2SPACE** evaluates whether a language-side numerical value can be correctly grounded in its corresponding spatial outcome, while **SPACE2NUM** tests whether an appropriate numerical value can be inferred from a given spatial configuration. Together, these two tasks assess numerical understanding from both directions, enabling a systematic examination of whether VLMs merely generate plausible numbers or genuinely ground them in spatial meaning.

To systematically study spatial numerical understanding, we investigate a series of progressively deeper questions. We first evaluate 18 VLMs across dynamic transitions and static layouts, showing that current models largely fail to ground numerical values in spatial meaning and often perform close to random guess. We then analyze how these failures differ across scenarios and mapping directions, revealing strong asymmetries between vision-to-number and number-to-vision grounding. To further understand the source of these failures, we conduct structured error analysis, reasoning trace analysis, and controlled interventions. Our results show that current VLMs often rely on shallow spatial cues, fail to construct stable coordinate-aware representations, and struggle to abstract structured spatial layouts from visual observations. Surprisingly, enabling explicit reasoning brings only marginal improvements, suggesting that the main limitation is not the absence of reasoning traces, but the lack of spatially calibrated reasoning operations. Finally, we show that spatial numerical understanding can be partially improved through tuning and transfers to external spatial reasoning benchmarks.

2 SpaceNum Data Curation

Data Source and Platform. We setup simulator-based pipelines to enable controllable data generation. For dynamic transition, data is generated in AI2-THOR [13], which supports embodied agents executing parameterized actions across diverse indoor environments. For static layout data, scenes are built in NVIDIA Isaac Sim [21] using assets from BlenderKit [2], allowing controlled layout generation with access to ground-truth spatial annotations for cognitive map construction.

2.1 Number as Dynamic Transition

Data Collection. We construct dataset with careful control over action coverage, transition continuity, visual anchoring, and data validity. (i) **Action coverage:** We define a set of primitive actions that induce spatial transitions, including `Move Forward (F) / Backward (B); Left (L)`

Action	Range	Step
`Move F/B`	0.2–2.4 m	0.2
`Move L/R`	0.2–1.2 m	0.2
`Rotate U/D`	10–70°	10
`Rotate L/R`	10–70°	10

Table 1: Action parameter ranges.

/ Right (R)) and rotations (Rotate Up (U) / Down (D); Left (L) / Right (R)). (ii) **Transition continuity:** The action magnitudes are chosen to ensure sufficient overlap between consecutive observations, as summarized in Table 1, maintaining *visual continuity* while introducing meaningful spatial changes and avoiding abrupt or ambiguous transitions. (iii) **Visual anchoring:** To ensure transitions are visually identifiable, we filter out observations with insufficient anchors by discarding frames containing fewer than *3 object instances*. (iv) **Data validity:** To avoid invalid transitions caused by random initialization or action execution (e.g., identical frames or empty observations), we leverage *occupancy maps* to constrain both the initial agent state and the post-action state to be valid, ensuring all collected samples correspond to informative transitions.

Task Definition. Let o_t denote the initial observation, o_{t+1} the resulting observation, a the action type, and n the numerical parameter representing the transition magnitude.

NUM2SPACE. The model is given (o_t, a, n) and is required to select the correct resulting observation o_{t+1} from a set of candidates. The distractor candidates are constructed by fixing the same initial observation o_t and action type a, while varying the numerical value n, resulting in alternative observations $\tilde{o}_{t+1}$ that correspond to different transition magnitudes.

SPACE2NUM. The model is given (o_t, o_{t+1}, a) and is required to infer the numerical value n that explains the transition. This task requires grounding visual differences between o_t and o_{t+1} to the corresponding transition magnitude.

2.2 Number as Static Layout

Data Collection. We build the layout dataset with controlled generation, covering the reference system, layout construction, scene scale, and representation. **(i) Coordinate system construction.** Each scene uses a clear coordinate system defined by two anchor objects. One anchor sets the origin. The relative position of the two anchors defines a consistent direction. This fixes the coordinate frame (up to scale) and removes ambiguity. The anchors stay fixed across samples in the same scene. **(ii) Layout generation.** Given the coordinate system, we place a third object with different positions and sizes. We enforce simple constraints: objects do not overlap, and distances are within a reasonable range. Under the same reference frame, we create three types of changes: (a) position only, (b) size only, and (c) both position and size. This lets us study each factor in a controlled way. **(iii) Scene scale.** We include both desktop-scale and room-scale scenes. This changes the spatial extent and the distribution of objects, and adds diversity. **(iv) Representation variation.** For each layout, we build multiple coordinate-based representations with different dimensions (1D, 2D, and 3D). These representations describe the same layout in different forms, from simple to more complete ones. This helps us study how models handle spatial information under different representations.

Task Definition. Let $\mathcal{M}$ denote a number-based cognitive map, o the layout observation, and p the numerical coordinates of a target object under a given reference frame.

NUM2SPACE. The model is given a cognitive map $\mathcal{M}$ and is required to select the observation o that is consistent with the specified layout. Distractor candidates are constructed by varying object positions or sizes while preserving the same reference frame.

SPACE2NUM. The model is given an observation o and is required to infer the numerical coordinates p of a target object under the reference coordinate system. This task requires grounding visual spatial structure into numerical representations.

2.3 Statistics

Figure 2 summarizes the benchmark composition that contains 3,800 samples. We further use the same fully automatic pipeline to generate an additional 77,412 training samples for later training-based explorations. The detailed breakdown of this larger training set is also shown in gray in Figure 2.

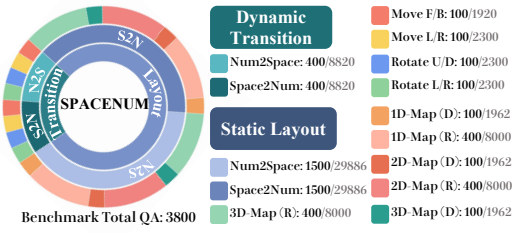


Figure 2: Dataset statistics.

			Dynamic Transition								Static Layout											
			Num2Space				Space2Num				Num2Space						Space2Num					
			Move		Rotate		Move		Rotate		1D-Map		2D-Map		3D-Map		1D-Map		2D-Map		3D-Map	
Methods	Rank	Avg.	F/B	L/R	U/D	L/R	F/B	L/R	U/D	L/R	D	R	D	R	D	R	D	R	D	R	D	R
Random Guess		30.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	50.0	50.0	25.0	25.0	25.0	25.0	50.0	50.0	25.0	25.0	25.0	25.0
Qwen2.5-VL-72B	1	39.8	34.0	**38.0**	34.0	**37.0**	40.0	37.0	44.0	41.0	69.0	64.5	28.0	24.2	36.0	26.8	60.0	51.2	33.0	**33.8**	31.0	32.8
InternVL3.5-38B	2	39.5	**38.0**	27.0	30.0	29.0	42.0	38.0	47.0	42.0	69.0	52.8	**31.0**	24.2	35.0	23.2	53.0	54.5	**43.0**	32.5	40.0	**38.2**
Qwen2.5-VL-32B	3	38.5	32.0	30.0	36.0	22.0	37.0	33.0	41.0	38.0	**71.0**	**67.0**	25.0	23.2	**37.0**	25.2	**63.0**	55.8	38.0	28.5	34.0	33.2
InternVL3.5-14B	4	38.2	36.0	32.0	**37.0**	27.0	40.0	35.0	**53.0**	**48.0**	**71.0**	66.8	20.0	24.0	27.0	25.5	53.0	54.8	30.0	27.5	34.0	23.0
Qwen3-VL-32B	5	35.9	26.0	30.0	36.0	25.0	36.0	**49.0**	44.0	32.0	68.0	50.2	30.0	20.8	32.0	22.8	58.0	57.2	28.0	23.0	29.0	20.8
InternVL3.5-8B	6	34.8	30.0	28.0	35.0	29.0	**45.0**	30.0	38.0	28.0	64.0	64.8	21.0	22.5	31.0	22.0	53.0	52.8	36.0	19.2	25.0	20.8
Ovis2.5-9B	7	34.7	22.0	32.0	31.0	23.0	36.0	44.0	41.0	27.0	70.0	66.2	17.0	25.0	21.0	24.8	53.0	**58.5**	24.0	28.7	26.0	24.5
InternVL3.5-4B	8	34.5	26.0	29.0	25.0	21.0	35.0	29.0	34.0	36.0	70.0	61.0	30.0	23.2	30.0	22.8	56.0	58.2	30.0	18.8	38.0	18.0
Qwen3-VL-8B	9	33.4	26.0	33.0	30.0	25.0	35.0	33.0	43.0	30.0	37.0	43.8	26.0	**30.0**	24.0	26.0	57.0	49.5	39.0	22.0	35.0	22.8
Ovis2.5-2B	10	33.2	26.0	22.0	29.0	31.0	27.0	27.0	23.0	24.0	**71.0**	**67.0**	28.0	22.0	27.0	24.5	51.0	49.5	28.0	26.2	33.0	27.8
Cosmos-Reason2-8B	11	33.1	24.0	37.0	29.0	25.0	31.0	26.0	27.0	33.0	57.0	53.5	20.0	28.0	20.0	27.0	58.0	50.7	34.0	23.8	30.0	27.3
Qwen2.5-VL-7B	12	33.0	37.0	22.0	30.0	32.0	29.0	29.0	27.0	30.0	**71.0**	**67.0**	21.0	26.0	25.0	27.5	46.0	47.5	29.0	23.5	20.0	20.5
Qwen3-VL-4B	13	32.1	22.0	29.0	26.0	26.0	31.0	35.0	29.0	32.0	41.0	55.2	28.0	23.5	23.0	24.5	57.0	56.0	33.0	20.2	31.0	19.5
Qwen2.5-VL-3B	14	31.9	24.0	20.0	23.0	29.0	26.0	16.0	25.0	20.0	**71.0**	**67.0**	19.0	24.8	30.0	**28.0**	55.0	41.5	34.0	25.5	**41.0**	17.8
Cosmos-Reason2-2B	15	31.6	28.0	22.0	23.0	25.0	23.0	26.0	24.0	26.0	**71.0**	**67.0**	13.0	27.0	13.0	23.5	48.0	55.2	25.0	27.0	39.0	27.3
Gemma-3-27B	16	31.2	27.0	25.0	34.0	16.0	24.0	29.0	25.0	27.0	50.0	43.2	25.0	23.8	22.0	22.8	54.0	49.0	32.0	24.5	**41.0**	29.0
Gemma-3-12B	17	30.6	21.0	26.0	35.0	21.0	28.0	29.0	27.0	21.0	67.0	55.8	27.0	22.5	24.0	22.0	48.0	42.2	25.0	19.5	25.0	25.8
Gemma-3-4B	18	28.5	**38.0**	19.0	25.0	21.0	20.0	25.0	24.0	26.0	35.0	34.0	24.0	23.2	22.0	21.2	56.0	45.8	28.0	27.8	30.0	24.5

Table 2: Results on **SpaceNum** benchmark. Accuracy (%) is reported under two major categories: *Dynamic Transition* and *Static Layout*. Each category contains both Num2Space and Space2Num. Avg. denotes the macro-average. **Bold** and underline denote best and second best, and gray values indicate performances that even below random guess.

3 Experiments

Experimental Setup. We evaluate 18 VLMs from 6 model families on SpaceNum, ranging from 2B to 72B [1, 25, 26, 17, 20, 8]. All models are evaluated with the same prompt format, where they are instructed to directly output the option letter without explanations or intermediate reasoning. We run inference in bfloat16 precision with Flash Attention 2 for efficient evaluation, with temperature to 0.7, top-p to 0.9, top-k to 50. All experiments are run on 4 NVIDIA H100 (80GB) GPUs.

3.1 Overall Results

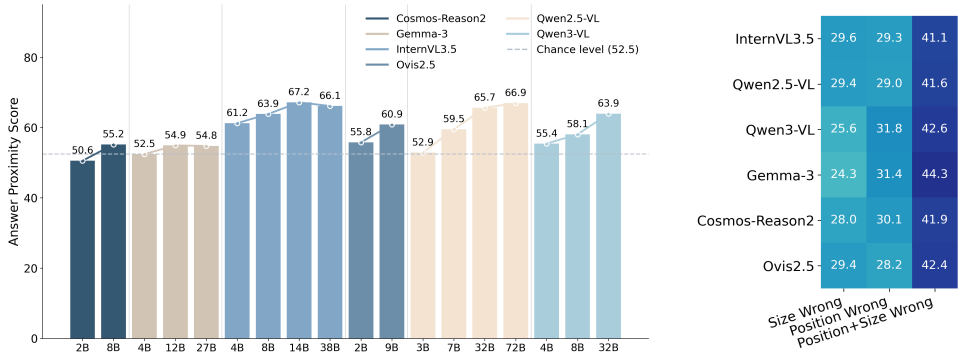
Do VLMs possess spatial numerical understanding? As shown in Table 2, current VLMs struggle to genuinely understand numerical values in spatial settings. Their performance remains close to random guess (30.0%), with the best model reaching only 39.8% on average, and several models even falling below the random baseline. These results suggest that **current models only capture shallow spatial-number correlations instead of truly grounding numerical values in spatial meaning**.

What patterns emerge across different spatial scenarios? Dynamic transitions and static layouts exhibit fundamentally different difficulty structures. In dynamic transitions, performance remains consistently low across all action types, with strong models achieving only around 40.0%, just 10 points above the random baseline (30.0%). Models show little preference or specialization across actions, suggesting **a broad failure to model transition dynamics**. In contrast, static layouts exhibit much clearer structural patterns: models perform relatively well in simpler settings such as 1D layouts and desk-scale scenes, but degrade substantially in higher-dimensional and room-scale settings, often only marginally above the 25.0% random baseline. This suggests that **layout reasoning difficulty grows systematically with spatial complexity and scene scale**.

How does spatial numerical mapping differ across scenarios? The preferred mapping direction differs substantially across scenarios. In dynamic transitions, models consistently perform better in Space2Num than in Num2Space, suggesting that **dynamic transitions are more vision-dependent**: models benefit from observing spatial changes directly, but struggle to predict future visual outcomes from numerical actions alone. In contrast, static layouts show the opposite trend, where Num2Space consistently outperforms Space2Num. This suggests that **static layouts rely more on language-side spatial priors**, where models can project numerical structures into space more easily than recovering structured numerical representations from visual scenes.

3.2 Structured Analysis of Output Patterns

Are larger models making better mistakes? Beyond standard multiple-choice accuracy, SpaceNum enables a more structured analysis of model behavior by leveraging the semantic



(a) Error proximity in dynamic transitions.

(b) Error decomposition in static layouts.

Figure 3: Structured analysis of model errors across spatial scenarios. Left: larger models tend to make numerically closer mistakes in dynamic transitions. Right: static layout failures are dominated by coupled position-and-size errors rather than isolated attribute errors.

relations among answer choices in different spatial scenarios. For dynamic transitions, we analyze not only exact-match accuracy but also the semantic proximity between the selected answer and the ground truth. Specifically, we assign scores of {100, 70, 40, 0} to exact, near, moderate, and far errors according to the numerical distance between the predicted and correct transition magnitudes. Figure 3a shows a clear trend: as model size increases, predictions become progressively closer to the correct answer even when exact-match accuracy changes only slightly. **Larger models make less severe transition errors**, suggesting that scaling improves coarse spatial sensitivity even when precise numerical grounding remains difficult.

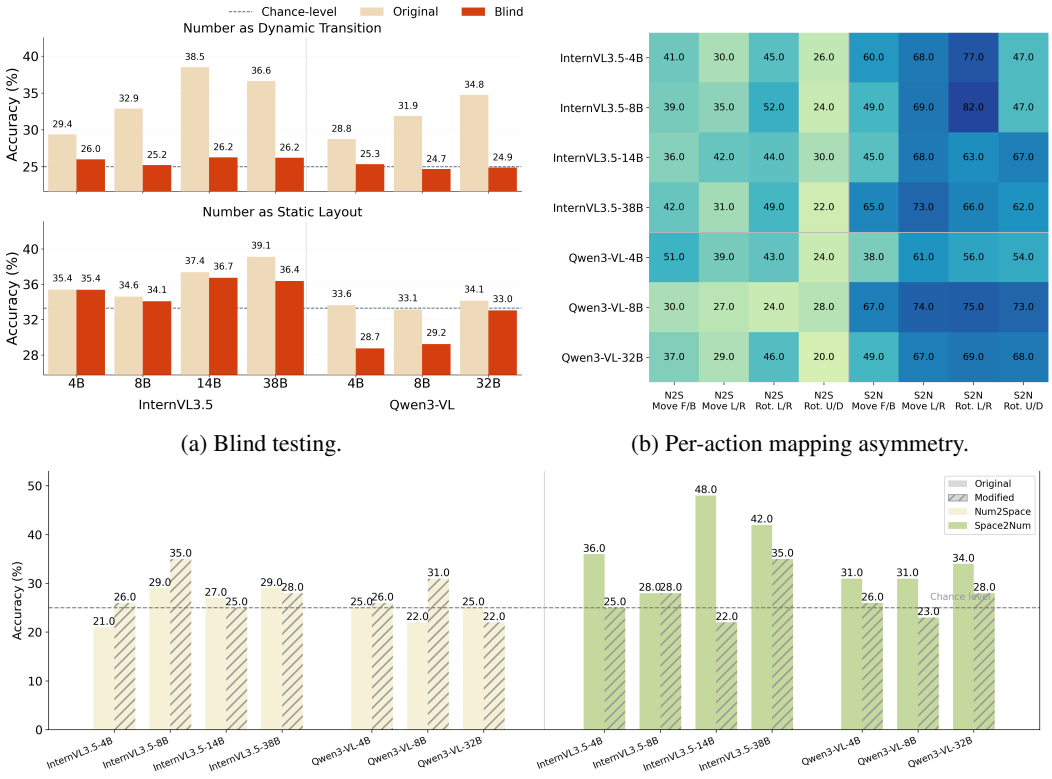
Do spatial errors decompose across attributes? For static layouts, we categorize errors according to whether the predicted layout contains incorrect position, incorrect size, or both. Surprisingly, models consistently favor joint position-and-size errors over single-factor errors across model families, as shown in Figure 3b. **Static layout failures are strongly coupled across spatial attributes**: once models fail to establish a coherent layout, errors tend to propagate jointly across position and scale rather than remain isolated. This suggests that current VLMs rely more on coarse holistic matching than disentangled spatial reasoning.

3.3 Does Reasoning Help Spatial Numerical Understanding?

To answer this question, we compare reasoning-enabled (*think*) and standard (*non-think*) inference across InternVL3.5-4B/8B/14B and Qwen3-VL-4B/8B/32B. Surprisingly, **enabling reasoning produces only marginal changes on SPACENUM, with performance differences typically remaining within 1%.** This suggests that simply generating longer reasoning traces does not substantially improve spatial numerical understanding. We therefore further analyze model traces and identify several recurring failure patterns that explain why reasoning often fails.

Models stop at coarse spatial cues instead of performing fine-grained comparison. A common failure is that models identify a plausible spatial cue and terminate reasoning too early. For example, in dynamic transition tasks, a model may observe that "a new wooden sculpture becomes visible on the left" and immediately select the corresponding candidate. However, the correct solution requires one more step: comparing how far objects shift across candidates to determine the correct transition magnitude. Similarly, in static layout tasks, models often correctly identify cues such as "the sofa is left of the tree," but fail to compare object size across candidates. In both settings, the model performs coarse cue matching but misses the finer comparison needed to disambiguate similar options.

Models fail to reason counterfactually about motion magnitude. Successful SPACE2NUM reasoning often depends on counterfactual magnitude comparison. Correct traces do not only check what changed, but also whether the observed change is large enough to support a candidate magnitude. For example, when estimating a small rotation, correct models explicitly reason that "most objects



(a) Blind testing.

(b) Per-action mapping asymmetry.

(c) Rotational symmetry analysis.

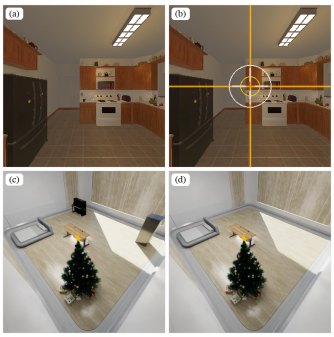
Figure 4: Additional analyses under dynamic transitions. Top left: blind testing by masking visual inputs. Top right: per-action comparison between NUM2SPACE and SPACE2NUM. Bottom: rotational symmetry analysis under equivalent transformations.

remain aligned across the two views," and therefore "a 70° rotation would produce much larger layout changes." In contrast, incorrect traces often map any noticeable visual change directly to a large number, e.g., "the perspective changes noticeably, suggesting a large right rotation." These traces focus only on changed evidence while ignoring stable evidence.

Models reason in image space instead of the defined coordinate system. Another recurring failure is that models rely on generic image-space priors rather than constructing the coordinate system defined by the anchor objects. For instance, some traces directly map "left in the image" to a smaller x value, reasoning that "the piano is positioned on the left side of the image, so it should have a smaller x-coordinate." However, the correct solution requires first establishing the coordinate frame using the provided anchors and then reasoning relative to that frame. Similarly, models may correctly describe an object as "behind" another object but still assign the wrong depth direction because they fail to align the scene with the task-defined coordinate system.

3.4 Modality Asymmetry in Spatial Numerical Understanding

How much do models rely on visual information? To examine whether models truly depend on visual grounding, we conduct a blind testing study by replacing images with fully black inputs while keeping the task format unchanged. As shown in Figure 4a, masking visual inputs causes a substantial performance drop for *number as dynamic transition*, while the effect is much smaller for *number as static layout*. **Dynamic transitions are significantly more vision-dependent**, whereas static layouts can often be partially solved through language-side priors or shortcut patterns without fully grounding the visual scene.

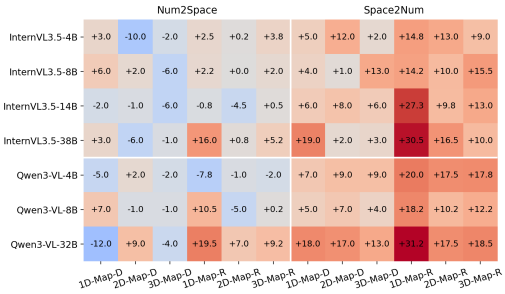


Model	Add Anchor (Transition)	Reduce Objects (Layout)
InternVL3.5-4B	-0.3%	-1.6%
InternVL3.5-8B	-1.3%	-0.0%
InternVL3.5-14B	-2.3%	-0.6%
InternVL3.5-38B	+0.9%	+0.1%
Qwen3-VL-4B	+0.5%	-1.1%
Qwen3-VL-8B	-2.5%	-0.1%
Qwen3-VL-32B	-1.0%	-0.3%

Figure 5: Visual-side interventions. Left: adding anchors for dynamic transitions and reducing objects for static layouts. Right: both interventions lead to only minor and inconsistent performance changes.

Model	NL Δ (Trans.)	Int. Δ (Trans.)	Int. Δ (Layout)
InternVL3.5-4B	-0.3%	+0.7%	-0.9%
InternVL3.5-8B	-1.0%	+0.8%	+1.6%
InternVL3.5-14B	-1.3%	-0.6%	-1.3%
InternVL3.5-38B	-1.8%	+4.5%	-1.8%
Qwen3-VL-4B	-1.2%	-1.7%	+0.2%
Qwen3-VL-8B	+2.0%	-2.0%	+0.5%
Qwen3-VL-32B	-1.2%	+3.3%	+1.3%

(a) Numerical representation changes.



(b) Visual abstraction for layouts.

Figure 6: Representation-side interventions. Left: changing numerical representations in dynamic transitions and layouts. Right: simplifying layouts into structured visual abstractions.

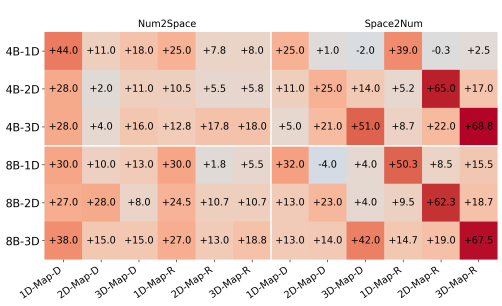
Is spatial numerical mapping balanced across actions? We further compare NUM2SPACE and SPACE2NUM at the level of individual actions. Figure 4b shows that, for almost every action type, *Space2Num* consistently outperforms *Num2Space*. **The asymmetry between the two mapping directions persists even under the same underlying action dynamics**, suggesting that models are systematically better at grounding numbers from observed visual changes than predicting future visual outcomes from numerical actions.

Do models learn geometrically consistent spatial mappings? Finally, we probe *Space2Num* under rotational symmetry transformations. Ideally, equivalent actions such as rotating left by 20° and rotating right by 340° should lead to consistent numerical predictions. However, Figure 4c shows substantial performance drops under these symmetric transformations. **The mapping from vision to numbers lacks geometric consistency and invariance**, suggesting that models fail to build stable numerical representations from visual spatial changes.

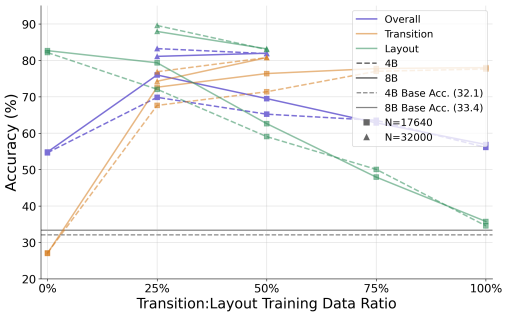
3.5 Disentangling Factors in Spatial Numerical Understanding

Can simple visual interventions improve spatial grounding? We first modify visual inputs in both scenarios. For dynamic transitions, we add explicit visual anchors to help models measure spatial changes. For static layouts, we reduce irrelevant objects to simplify visual grounding. However, Figure 5 shows that both interventions lead to only minor and inconsistent improvements. **The core limitation is not caused by missing visual references or cluttered scenes.**

Does the numerical representation itself matter? We then vary how numerical values are expressed. Converting numbers into natural language yields negligible gains, while integer-scaled representations (e.g., meters to centimeters) provide only limited improvements for larger models in transition tasks. As shown in Figure 6(a), performance in layout reasoning remains largely unchanged. **The bottleneck does not primarily lie in the surface form of numerical representations.**



(a) Cross-dimension tuning transfer.



(b) Training data mixture and scaling.

Figure 7: Tuning analysis for spatial numerical understanding. Left: transfer patterns across different spatial dimensions. Right: effects of data mixture ratios and training scale.

Do models struggle to abstract spatial structure from images? Since neither visual simplification nor numerical reformulation resolves the issue, we further investigate whether models fail to extract structured spatial representations from raw images. We therefore replace layout images with progressively more structured abstractions, including points, 2D boxes, and 3D boxes. Figure 6(b) shows that this substantially improves SPACE2NUM, while providing less effects for NUM2SPACE. **The main bottleneck lies in vision-to-structure abstraction**: current VLMs struggle to transform raw visual observations into structured spatial representations suitable for numerical reasoning.

3.6 Tuning Spatial Numerical Understanding

Can spatial reasoning transfer across dimensions? We fine-tune Qwen3-VL-4B and Qwen3-VL-8B with LoRA using a learning rate of 1×10^{-4}, cosine decay with a 0.1 warmup ratio, bfloat16 precision, a maximum sequence length of 2048, LoRA rank 8 and alpha 16, and an effective batch size of 128 for 3 epochs. Figure 7(a) shows a clear diagonal pattern: tuning on a particular dimension yields the largest improvement on the same dimension, suggesting that different dimensions encode distinct spatial structures. At the same time, tuning on 1D data also improves performance on 2D and 3D settings, especially for larger models and more clearly in NUM2SPACE. **Lower-dimensional spatial reasoning can partially transfer to higher-dimensional settings**, although the transfer remains limited.

What data recipe leads to the best spatial reasoning ability? We next vary the ratio between transition and layout data. As shown in Figure 7(b), the best overall performance consistently emerges when transition data accounts for roughly 25% and layout data accounts for roughly 75%. Increasing the total amount of training data further improves performance under the same ratio. **Both data composition and training scale substantially affect spatial numerical understanding**, with layout-heavy mixtures producing the strongest overall capability.

Does RL help, and does reward design matter? We further study RL tuning on the 4B model using GRPO with LoRA rank 64 and alpha 64, a learning rate of 1×10^{-5}, rollout batch size 128, actor batch size 64, and 5 rollouts per prompt. We compare a strict exact-match reward and a graded reward based on error magnitude. As shown in Table 3, RL brings only limited gains overall, while graded rewards perform slightly better than strict rewards.

Metric	Strict	Graded
Transition	+6.38	**+6.88**
Layout	+6.64	**+7.60**
Num2Space	+8.10	**+9.37**
Space2Num	+5.05	**+5.52**

Table 3: Performance improvement under two different reward designs.

Does the learned ability generalize beyond SPACENUM? Finally, we evaluate tuned models on external spatial reasoning benchmarks. Table 4 shows consistent improvements across all tasks. Gains on OmniSpatial Motion [11] indicate better understanding of camera movement, while improve-

Metric	4B Δ	8B Δ
OS-Motion	+5.5	+4.5
SAT-AC	+8.1	+18.9
SAT-OM	+34.8	+43.5

Table 4: Transfered performance.

ments on SAT Action Consequence and Object Movement [24] demonstrate stronger reasoning about action outcomes and object dynamics. The improvements are particularly large for the 8B model. **The learned capability transfers beyond our benchmark**, suggesting that tuning improves general spatial reasoning ability rather than merely overfitting in our settings.

4 Related Works

Spatial reasoning in dynamic and embodied environments. Recent works study whether VLMs can reason about spatial changes caused by actions, motion, and embodied interactions. SAT evaluates dynamic spatial aptitude through action consequence prediction, object movement, perspective taking, and spatial aiming tasks [24]. OmniSpatial provides a comprehensive benchmark for spatial reasoning over camera motion, object motion, perspective transformation, and interaction-centered scenarios [11]. VSI-Bench evaluates whether MLLMs can see, remember, and recall spatial environments from sequential visual observations [30]. MVoT improves spatial reasoning by encouraging models to imagine intermediate visual states during reasoning [14]. SpaceTools studies tool-augmented spatial reasoning through interactive reinforcement learning with external spatial tools [4]. These works show that current VLMs struggle with dynamic spatial reasoning and spatial transformations. However, they mainly focus on whether models understand spatial changes themselves, rather than whether the numerical values parameterizing these transitions are truly grounded in spatial meaning.

Spatial understanding and structured spatial reasoning. Another line of work studies whether VLMs can infer spatial relations, metric structure, and 3D layouts from visual observations. Early benchmarks evaluate relations such as left/right, above/below, and object-centric configurations, showing that VLMs often struggle with spatial prepositions despite strong object recognition ability [16, 12, 23, 9]. More recent works extend this evaluation to metric reasoning, geometric reasoning, open-space understanding, and domain-specific 3D reasoning [7, 33, 28, 29]. Beyond evaluation, several works inject explicit spatial structures into VLMs through spatial annotations, region-level grounding, coordinates, distances, layouts, and 3D priors [3, 5, 18, 15, 7, 10]. More recently, SpatialReasoner studies explicit and generalizable 3D spatial reasoning through structured spatial representations [19]. Together, these works improve structured spatial understanding and reasoning ability in VLMs, but they mainly treat numbers as auxiliary labels or outputs, rather than directly studying whether numerical values themselves are grounded as meaningful spatial quantities.

In contrast to prior work, SPACENUM directly studies spatial numerical understanding: whether VLMs can ground numerical values as meaningful spatial quantities across both dynamic transitions and static layouts. Beyond benchmark evaluation, we further analyze the asymmetry, failure patterns, reasoning behaviors, and tuning characteristics of spatial numerical grounding in current VLMs.

5 Conclusion

In this work, we study whether current Vision Language Models (VLMs) truly understand numerical values in spatial settings through SPACENUM, a unified benchmark covering both dynamic transitions and static layouts. Our experiments show that current VLMs largely fail to ground numbers in spatial meaning, often relying on shallow spatial cues instead of stable spatial reasoning. Through systematic analyses, we further show that these failures arise from weak spatial abstraction, asymmetric vision-number mappings, and the inability to build structured coordinate-aware representations. Although tuning partially improves performance and transfers to related benchmarks, substantial gaps still remain. We hope SPACENUM can serve as a useful benchmark and diagnostic framework for future research on spatial numerical understanding in VLMs.

Limitations and future work. Our study mainly focuses on controlled spatial settings with discrete candidate-based evaluation and simulated environments. Extending spatial numerical understanding to more open-ended real-world scenes, embodied interactions, and continuous spatial prediction settings remains an important direction for future work. We also mainly analyze failures from the vision and language sides, while how VLMs internally perform spatial reasoning remains largely unexplored. Although we conduct preliminary attention-based analyses, severe attention collapse in current VLMs makes it difficult to obtain clear conclusions. Understanding the internal mechanisms behind spatial numerical reasoning therefore remains an important future direction.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] BlenderKit. Blenderkit: Online asset library for blender. `https://www.blenderkit.com/`, 2023.

[3] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

[4] Siyi Chen, Mikaela Angelina Uy, Chan Hee Song, Faisal Ladhak, Adithyavairavan Murali, Qing Qu, Stan Birchfield, Valts Blukis, and Jonathan Tremblay. Spacetools: Tool-augmented spatial reasoning via double interactive rl. *arXiv preprint arXiv:2512.04069*, 2025.

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

[6] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

[7] Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, et al. Mm-spatial: Exploring 3d spatial understanding in multimodal llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7395–7408, 2025.

[8] Google DeepMind. Gemma 3. `https://deepmind.google/models/gemma/gemma-3/`, 2025. Accessed: 2026-05-01.

[9] Mengfei Du, Binhao Wu, Zejun Li, Xuan-Jing Huang, and Zhongyu Wei. Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 346–355, 2024.

[10] Yibin Huang, Wang Xu, Wanyue Zhang, Helu Zhi, Jingjing Huang, Yangbin Xu, Yangang Sun, Conghui Zhu, and Tiejun Zhao. Video2layout: Recall and reconstruct metric-grounded cognitive map for spatial reasoning. *arXiv preprint arXiv:2511.16160*, 2025.

[11] Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. *arXiv preprint arXiv:2506.03135*, 2025.

[12] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9161–9175, 2023.

[13] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[14] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025.

[15] Yuan-Hong Liao, Rafid Mahmood, Sanja Fidler, and David Acuna. Reasoning paths with reference objects elicit quantitative spatial reasoning in large vision-language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17028–17047, 2024.

[16] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023.

[17] Shiyin Lu, Yang Li, Yu Xia, Yuwei Hu, Shanshan Zhao, Yanqing Ma, Zhichao Wei, Yinglun Li, Lunhao Duan, Jianshan Zhao, et al. Ovis2. 5 technical report. *arXiv preprint arXiv:2508.11737*, 2025.

[18] Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. *Advances in neural information processing systems*, 37:68803–68832, 2024.

[19] Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso de Melo, Jianwen Xie, and Alan Yuille. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. *arXiv preprint arXiv:2504.20024*, 2025.

[20] NVIDIA. Cosmos-reason2: Open reasoning vision-language models for physical ai. `https://huggingface.co/collections/nvidia/cosmos-reason2`, 2026. Accessed: 2026-05-01.

[21] NVIDIA Corporation. Nvidia isaac sim. `https://developer.nvidia.com/isaac-sim`, 2023.

[22] Renjie Pi, Jianshu Zhang, Jipeng Zhang, Rui Pan, Zhekai Chen, and Tong Zhang. Image textualization: An automatic framework for creating accurate and detailed image descriptions. *arXiv preprint arXiv:2406.07502*, 2024.

[23] Navid Rajabi and Jana Kosecka. Gsr-bench: A benchmark for grounded spatial reasoning evaluation via multimodal llms. *arXiv preprint arXiv:2406.13246*, 2024.

[24] Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, et al. Sat: Dynamic spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2024.

[25] Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[26] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[27] Zixuan Wang, Huang Fang, Shaoan Wang, Yuanfei Luo, Heng Dong, Wei Li, and Yiming Gan. Hydra-nav: Object navigation via adaptive dual-process reasoning. *arXiv preprint arXiv:2602.09972*, 2026.

[28] Haoning Wu, Xiao Huang, Yaohui Chen, Ya Zhang, Yanfeng Wang, and Weidi Xie. Spatialscore: Towards unified evaluation for multimodal spatial understanding. *arXiv e-prints*, pages arXiv–2505, 2025.

[29] Zelin Xu, Yupu Zhang, Saugat Adhikari, Saiful Islam, Tingsong Xiao, Zibo Liu, Shigang Chen, Da Yan, and Zhe Jiang. Earthspatialbench: Benchmarking spatial reasoning capabilities of multimodal llms on earth imagery. *arXiv preprint arXiv:2602.15918*, 2026.

[30] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025.

[31] Yuncong Yang, Jiageng Liu, Zheyuan Zhang, Siyuan Zhou, Reuben Tan, Jianwei Yang, Yilun Du, and Chuang Gan. Mindjourney: Test-time scaling with world models for spatial reasoning. *arXiv preprint arXiv:2507.12508*, 2025.

[32] Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*, 2025.

[33] Weichen Zhang, Zile Zhou, Xin Zeng, Xuchen Liu, Jianjie Fang, Chen Gao, Yong Li, Jinqiang Cui, Xinlei Chen, and Xiao-Ping Zhang. Open3d-vqa: A benchmark for comprehensive spatial reasoning with multimodal large language model in open space. *arXiv preprint arXiv:2503.11094*, 2025.