GEOREADER: LEARNING LATENT GEOMETRY FOR SPATIAL REASONING IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

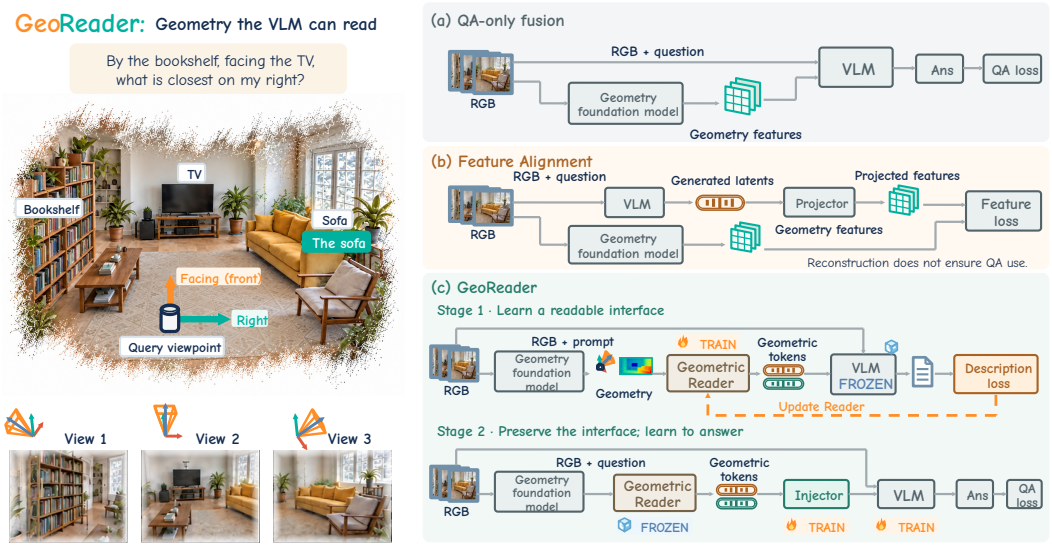


Figure 1: **GeoReader** learns geometry in a form the receiving VLM can read. (a) QA-only fusion learns to use geometry through answer supervision. (b) Feature alignment matches VLM-generated representations to external geometry features. (c) **GeoReader** first trains Geometric Readers through description prediction by a frozen VLM, then freezes the Readers and trains the VLM and geometry injector for QA. All three routes estimate geometry from RGB, which also directly conditions each VLM. Flames and ice cubes mark trainable and frozen modules. The scene, view directions, and answer are illustrative; Fig. 2 details the architecture.

ABSTRACT

Vision-language models (VLMs) struggle with metric and multi-view relationships that remain implicit in RGB observations. Geometry foundation models recover this structure from RGB, but existing methods either fuse the estimated features with visual tokens and learn to use them from question answering alone, or align VLM latents with geometry features. Capturing geometry, however, does not ensure its use: latents aligned with geometry features can be zeroed without lowering accuracy. We introduce **GeoReader**, a framework that learns geometric representations for the VLM that reads them. First, we train lightweight Geometric Readers to produce continuous input tokens, supervising them with geometry descriptions decoded by a frozen VLM, so that the Readers express geometry in a form the target model can already read. We then freeze the Readers to preserve this interface while fine-tuning the VLM on spatial QA, with geometry injection conditioned on the question. Geometry descriptions serve only as training targets: at inference, the VLM answers directly from the continuous geometric tokens without generating an intermediate geometry chain of thought. With Qwen3-VL-4B and geometry predicted from RGB images, **GeoReader** reaches 94.3 on MindCube-Tiny and, with a multi-source QA mixture, an average of 70.6 across six additional spatial benchmarks. These results suggest that the most useful geometric representation is the one the receiving model can use, not necessarily the one that reconstructs geometry most accurately.

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.

1 INTRODUCTION

Vision-language models (VLMs) have demonstrated increasingly strong capabilities in general multimodal understanding, yet continue to struggle with spatial reasoning, a capability essential to embodied intelligence in the 3D world. A central challenge is inferring and reasoning about 3D structure from 2D observations, as required by spatial questions involving camera motion, viewpoint changes, and object distances. Recent advances in geometry foundation models (Wang et al., 2025a; Lin et al., 2026) offer a promising opportunity to address this limitation by providing geometry information. How to effectively harness this information to enhance the spatial reasoning capabilities of VLMs has therefore become an important research question.

Existing approaches to integrating geometry foundation models with VLMs have improved spatial reasoning and can be broadly categorized into two streams. The first introduces geometry as an additional input modality (Fan et al., 2026; Wu et al., 2025a; Zhao et al., 2026), fusing features from geometry foundation models with the VLM's original visual features (Fig. 1a). This provides a direct way to incorporate geometry into existing VLMs, with the learning of using geometry features relying on downstream supervision. However, downstream supervision alone can be insufficient to guarantee proper interpretation of geometry features. The second uses geometry as supervision, typically by aligning VLM latent representations with features from geometry foundation models (Chen et al., 2026) (Fig. 1b). This encourages the latent space to capture geometry information while preserving the VLM's input and output interfaces. Nevertheless, capturing geometry does not ensure its use: zeroing the latents of the released first-stage 3DThinker (Chen et al., 2026) checkpoint leaves MindCube (Wang et al., 2026a) accuracy essentially unchanged, and answer tokens barely attend to them (Tab. 8; Fig. 3). Related analyses of latent visual reasoning methods such as LVR (Li et al., 2026a) and Monet (Wang et al., 2026b) also highlight a disconnect between intermediate representations and answer generation (Viveiros et al., 2026; Guo et al., 2026; Li et al., 2026d; Zhu et al., 2026; Park et al., 2026). Despite their differences, neither approach explicitly bridges the gap between geometry representations and language-grounded spatial relations for VLMs. Consequently, the role of geometry information in the model's reasoning remains uncertain. We bridge this gap by converting geometry into a form that VLMs can interpret and ground in language.

We introduce **GeoReader**, based on the core idea that *geometry should first be converted into a form that VLMs can interpret as spatial relationships in language, before VLMs learn to use it for downstream tasks*. To this end, a two-stage pipeline separates adapting geometry to the VLM from applying it to spatial QA. In Stage 1, we train lightweight Geometric Readers to convert per-view geometric signals into continuous geometric tokens. These tokens and the corresponding images are given to a frozen VLM that predicts geometry descriptions. With the VLM frozen, this supervision trains the Readers to express geometry information in a form that the model can interpret through its existing language capabilities. In Stage 2, we transfer the pretrained Readers to spatial QA, freezing the Readers to preserve their learned mapping while fine-tuning the VLM to use their outputs. Since questions differ in how much they rely on each geometric signal, a geometry injector conditions on the images and question to set each signal's strength. Geometry descriptions are used only for Stage 1 supervision; at inference, the VLM answers directly using the geometric tokens.

We evaluate **GeoReader** across seven spatial reasoning benchmarks and three VLM families. With geometry predicted from RGB images, **GeoReader** built on Qwen3-VL-4B achieves 94.3% accuracy on MindCube-Tiny and, with multi-source QA training, an average accuracy of 70.6% across six additional benchmarks (Sec. 4.2). Comparisons with vanilla supervised fine-tuning (SFT) under matched QA data and training settings show gains concentrated on questions requiring metric 3D understanding. Ablations further demonstrate the benefits of readability pretraining, freezing the Readers during downstream adaptation, and question-conditioned geometry injection (Secs. 4.3 and 4.4). Geometry interventions show that the model uses the supplied geometry: in a controlled MindCube-Tiny experiment with annotation-derived camera geometry, removing geometry content reduces accuracy from 89.2% to 77.2%, while shuffling poses across views reduces it to 63.1%.

Our contributions are summarized as follows:

- We propose explicitly connecting geometry information to spatial relationships expressed in language, separating adapting geometry to the VLM from applying it to spatial reasoning.

- We introduce **GeoReader**, a two-stage framework that trains lightweight Geometric Readers through language supervision from a frozen VLM, then freezes the Readers and adapts the VLM for spatial QA with question-conditioned geometry injection (Sec. 3).
- We demonstrate the effectiveness of **GeoReader** across seven spatial reasoning benchmarks and three VLM families, with ablations and geometry interventions supporting both the framework's design and the model's use of geometry information (Secs. 4.2–4.5).

2 Related Work

Spatial reasoning in vision-language models. SpatialVLM (Chen et al., 2024), SpaRE (Ogezi & Shi, 2025) and SPAR-7M (Zhang et al., 2025) build synthetic or 3D-grounded spatial QA data. SpatialLadder (Li et al., 2026c) and HiSpatial (Liang et al., 2026) organize training from perception toward reasoning, while SenseNova-SI (Cai et al., 2026) studies scaling spatial data. SpatialQA (Cai et al., 2025) and ViewSpatial (Li et al., 2026b) target depth and viewpoint relations; N3D-VLM (Wang et al., 2025c) links 3D object localization to reasoning. These works establish the value of spatial supervision; we study how to present estimated geometry to the answering VLM.

Geometry-augmented vision-language models. DUSt3R (Wang et al., 2024) and MASt3R (Leroy et al., 2024) estimate point maps and correspondences; CUT3R (Wang et al., 2025b) maintains a recurrent scene state. VGGT (Wang et al., 2025a) predicts camera and dense geometry, and Depth Anything V2 (Yang et al., 2024) and Depth Anything 3 (Lin et al., 2026) predict depth and any-view geometry. Geometry-augmented VLMs take such 3D structure as input: 3D-LLM (Hong et al., 2023) and LEO (Huang et al., 2024) ground language in 3D scenes. LLaVA-3D (Zhu et al., 2025), Video-3D LLM (Zheng et al., 2025b) and SR-3D (Cheng et al., 2026) add spatial positions or regions to visual inputs. SpatialRGPT (Cheng et al., 2024), SD-VLM (Chen et al., 2025) and HiSpatial (Liang et al., 2026) use relative depth, depth positional encoding and metric point maps, respectively. VLM-3R (Fan et al., 2026), Spatial-MLLM (Wu et al., 2025a) and VG-LLM (Zheng et al., 2025a) fuse visual and 3D features; SpaceMind (Zhao et al., 2026) uses camera-guided fusion, and SpatialStack (Zhang et al., 2026a) spreads geometry across LLM layers. These studies vary the geometric source and fusion; we train the interface for the receiving VLM.

Learning geometry representations for reasoning. G^2VLM (Hu et al., 2026) combines reconstruction with spatial understanding, and GAP-MLLM (Zhang et al., 2026b) pretrains with sparse point-map and semantic prediction. SpatioLM (Wu et al., 2026) uses pseudo depth and camera supervision but only RGB at inference; GeoSR (Zhang et al., 2026c) finds that fused VGGT tokens are often underused and masks visual tokens during training to force their use, with QA as the only supervision. 3DThinker (Chen et al., 2026) aligns generated latents with VGGT features, and CoVT (Qin et al., 2026) learns continuous visual thoughts; LVR (Li et al., 2026a) and Monet (Wang et al., 2026b) reason with visual latents, and Coconut (Hao et al., 2025) reasons in a continuous latent space. Unlike these methods, **GeoReader** supervises its geometric input through description prediction by the frozen VLM, and takes estimated pose and depth, not descriptions, at inference.

3 Method

We aim to supply a pretrained VLM with geometric input that meets two separate requirements: the geometric tokens must be *readable* by the receiving model, and the model must *use* them for the question at hand. **GeoReader** addresses them with two training stages (Fig. 2). After the preliminaries (Sec. 3.1), we describe how a Geometric Reader encodes each view as a geometric token (Sec. 3.2), how Stage 1 trains it for readability through the frozen VLM (Sec. 3.3), and how Stage 2 injects these tokens according to the question (Sec. 3.4).

3.1 Preliminaries

A VLM with parameters θ encodes the V images $I = \{I_i\}_{i=1}^{V}$ into visual tokens $X_1, \ldots, X_V$, embeds the question q as text tokens Q, and feeds the sequence of d-dimensional embeddings to a language model that predicts the answer a with next-token distribution p_θ. Each view i also

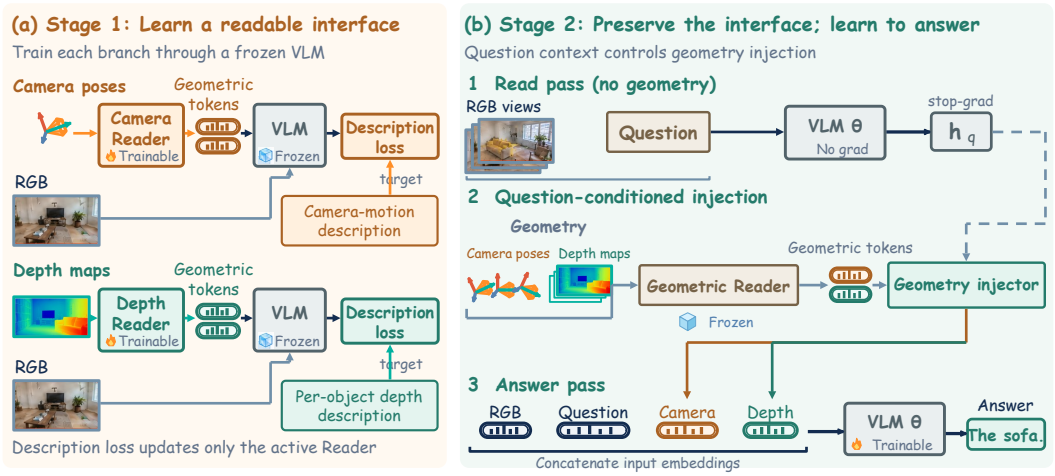


Figure 2: Learn a geometric interface, then preserve it during QA adaptation. (a) The description loss (Eq. 2) trains each Reader separately to support descriptions of inter-view camera motion or per-object depth. Gradients pass through the frozen VLM to the active Reader. (b) With Readers frozen, a read pass extracts detached context h_q from the images and question. This context controls the gates of the geometry injector. The answer pass concatenates gated geometric tokens with visual and question embeddings; both passes share VLM parameters θ. In (b), one Reader and injector depict the per-signal modules, whose parameters remain separate. Slot-embedding residual details are omitted. Inputs are multiview; (a) shows representative views. Scene and QA are illustrative.

has geometric signals G_i^m, $m \in \mathcal{M}$, here camera pose relative to view 1 and a depth map; the formulation applies to any per-view signal. We turn G^m into additional tokens of this input sequence.

3.2 Geometric Token Encoding

To feed geometry through the VLM's existing input channel, a lightweight Geometric Reader E_m maps each view of signal m to one continuous input embedding. For view i, it first computes per-patch descriptors $U_i^m \in \mathbb{R}^{N_i^m \times c_m}$ on the same patch grid as the RGB tokens; each c_m-dimensional descriptor holds the patch position together with pose-related quantities or depth statistics of that patch. A learned query $\mathbf{r}^m$ then reads only that view's descriptors:

$$M_i^m = \mathrm{MLP}_m\big(\mathrm{LN}(U_i^m)\big), \qquad Z_i^m = \mathrm{Dec}_m\big(\mathbf{r}^m, M_i^m\big) \in \mathbb{R}^{1\times d},$$
$$Z^m = E_m(G^m) = \big[Z_1^m; \ldots; Z_V^m\big] \in \mathbb{R}^{V\times d}, \tag{1}$$

where Dec_m applies two Transformer decoder blocks with $\mathbf{r}^m$ cross-attending to M_i^m, then a linear projection. We call Z^m *geometric tokens*: continuous embeddings, one per view in input-view order. The Reader encodes geometry alone; the VLM combines it with RGB through its own attention layers. Each signal has its own Reader, so signals keep their own structure (Sec. A.2.1).

3.3 Frozen-VLM Readability Training

QA supervision alone does not specify what the geometry encodes, and feature alignment measures similarity to an external target rather than what the receiving VLM can read (Sec. 1). We instead supervise each Reader through the frozen target VLM: given the images, the geometric tokens and a fixed instruction, the VLM predicts a geometry description y^m of signal m with teacher forcing. For two ScanNet (Dai et al., 2017) views, the camera description reads "compared with image 1, the observer in image 2 moves 0.4 meters right and 0.1 meters backward, and turns left by about 23 degrees," and the depth description lists objects per view, e.g., "guitar (left) at 1.7 meters." The supervised positions $\mathcal{T}^m$ cover geometric content such as distances, directions and angles; template words, predictable without geometry, are excluded. The description loss is

$$\mathcal{L}_{\mathrm{read}}^m = -\frac{1}{|\mathcal{T}^m|} \sum_{t\in\mathcal{T}^m} \log p_\theta\big(y_t^m \mid I, Z^m, y_{<t}^m\big), \qquad \theta \text{ frozen}, \tag{2}$$

where gradients pass through the frozen VLM to update only E_m, so the Reader cannot reduce this loss by changing the language decoder; *it must act through the VLM's existing computation.* The objective thus trains functional readability; the description is only a training target.

Pair-supervised readability. Descriptions of a signal such as camera pose relate each view to reference view 1. *Group supervision* supplies all V views and their geometry in one example and supervises all target descriptions jointly; the description of view i can then be predicted from the tokens of any view, so the loss does not tell the Reader which token should carry that relation. *Pair supervision* removes this ambiguity by applying Eq. 2 to reference–target pairs: each target view $i \in \{2, \dots, V\}$ forms an independent example with loss $\mathcal{L}^m_{\text{read}}(I_{\{1,i\}}, G^m_{\{1,i\}}; y^m_{1i})$, in which the Reader and the frozen VLM receive only views 1 and i with their geometry, so only these tokens can explain the description of view i. Stage 2 inputs keep all views (Sec. A.4.5).

3.4 Question-Conditioned Geometry Injection

Freezing the Readers. Stage 2 supervises answers, which do not explicitly require the Readers to retain the geometric relationships specified by the Stage 1 descriptions. Updating the Readers under this supervision may weaken their ability to express geometry beyond what the training questions need. We therefore freeze the Readers and adapt the VLM to this fixed interface.

Question-conditioned readout. Questions differ in how much they rely on each geometric signal, so we let the question set each signal's strength. We compute it from the final-layer hidden state of the last valid question token, which attends to all preceding visual and question tokens and thus summarizes the task context. Geometric embeddings, however, enter before layer 1, and one forward pass cannot use its own final-layer state to set earlier inputs. A read pass therefore reads only the images and question to produce this context h_q before any geometry is introduced. Both passes share the VLM parameters; we stop gradients through the read pass to save memory and computation.

Geometry injector. A geometry injector turns the frozen Reader outputs into the geometric tokens of the answer pass. For each signal m, it maps Z^m through an adapter P_m, scales the result by a gate computed from h_q with a routing head R_m, and adds it to the slot embeddings $B^m \in \mathbb{R}^{V \times d}$, the learned input embeddings of the placeholder tokens at the geometric-token positions:

$$\widetilde{Z}^m = B^m + g_m P_m(Z^m), \qquad g_m = \sigma\big(R_m(h_q)\big). \tag{3}$$

Here σ is the sigmoid. The adapter lets Stage 2 adjust how the frozen Reader outputs enter the VLM, and g_m is the question-dependent strength of signal m. When a signal is missing (e.g., inter-view geometry for a single image), $g_m = 0$ and the slot embeddings alone serve as a placeholder.

Training and inference. The read and answer passes encode

$$S_{\text{read}} = \big[X_1, \dots, X_V, Q\big], \qquad S_{\text{ans}} = \big[X_1, \dots, X_V, Q, \{\widetilde{Z}^m\}_{m \in \mathcal{M}}, a\big], \tag{4}$$

and h_q is taken from S_{read}. We fine-tune the language model and the geometry injector with the answer loss $\mathcal{L}_{\text{ans}} = -\sum_t \log p_\theta(a_t \mid I, q, \{\widetilde{Z}^m\}, a_{<t})$, keeping the Readers and the vision encoder frozen. At inference, S_{read} is a prefix of S_{ans}, so the answer pass reuses its key–value cache.

4 Experiments

We test whether **GeoReader** improves spatial QA (Sec. 4.2), how to learn and keep its interface (Sec. 4.3), how questions use it (Sec. 4.4), and how it compares with text (Sec. 4.5).

4.1 Experimental Setup

Data and benchmarks. Stage 1 uses about 65K geometry-description records from 1,550 ScanNet and ScanNet++ training scenes (Sec. A.1.1). For Stage 2, all MindCube experiments use the same 10K QA examples and the multi-source setting uses approximately 135K; held-out test QA is

Table 1: Accuracy (%) on MindCube-Tiny; bold marks the best Overall per backbone and size. **GeoReader** improves over its base model at every backbone size.

Backbone	Size	Method	Rot.	Among	Around	Overall
API models	–	GPT-4.1 (OpenAI, 2025a)	45.5	44.2	47.2	45.1
	230B	Doubao-Seed-1.6 (ByteDance Seed, 2025)	87.0	35.8	38.0	46.1
	–	Gemini-2.5-Pro (Gemini Team, 2025)	84.0	39.7	56.8	52.2
	309B	MiMo-V2.6-Flash (LLM-Core Xiaomi, 2026)	81.0	41.3	64.8	54.5
	–	OpenAI o3 (OpenAI, 2025b)	86.5	42.7	66.0	56.6
	552B	DeepSeek-Flash (thinking) (DeepSeek-AI, 2026)	89.5	58.3	75.6	68.4
Qwen2.5-VL	3B	Qwen2.5-VL (Qwen Team, 2025a)	35.5	36.2	45.6	38.3
		HATCH (Oi et al., 2026)	31.0	50.2	65.6	50.2
		SenseNova-SI (Cai et al., 2026)	–	–	–	52.6
		DR-MV3D (Choi et al., 2026)	43.0	71.3	73.6	66.5
		3DThinker (Chen et al., 2026)	55.5	81.8	75.2	75.2
		GeoReader	40.5	90.3	91.6	**81.1**
	7B	Qwen2.5-VL (Qwen Team, 2025a)	35.0	33.2	38.4	34.8
		SenseNova-SI (Cai et al., 2026)	–	–	–	54.7
		3DThinker (Chen et al., 2026)	55.0	83.0	76.0	76.0
		GeoReader	90.0	94.7	92.4	**93.2**
Qwen3-VL	2B	Qwen3-VL (Qwen Team, 2025b)	29.5	33.3	47.2	35.9
		Spatial-TTT (Liu et al., 2026)	55.5	74.0	89.8	76.2
		3DThinker (Chen et al., 2026)	53.0	83.2	78.4	76.2
		GeoReader	59.5	93.3	91.6	**86.5**
	4B	Qwen3-VL (Qwen Team, 2025b)	25.0	25.3	43.6	29.6
		3DThinker (Chen et al., 2026)	55.0	81.7	78.4	75.8
		GeoReader	94.5	96.2	89.6	**94.3**
	8B	Qwen3-VL (Qwen Team, 2025b)	42.0	32.3	51.2	38.7
		SenseNova-SI (Cai et al., 2026)	79.5	73.2	70.4	73.8
		3DThinker (Chen et al., 2026)	56.5	82.5	78.8	76.7
		GAMSI (Zheng et al., 2026)	–	–	–	**94.1**
		GeoReader	91.5	93.3	95.2	93.4

excluded, and all backbones share one training recipe (Sec. A.1.2). We evaluate on MindCube-Tiny (Wang et al., 2026a), VSI-Bench (Yang et al., 2025), SPBench (Li et al., 2026c), CV-Bench (Tong et al., 2024), SPAR-Bench (Zhang et al., 2025), ViewSpatial (Li et al., 2026b) and MMSI-Bench (Yang et al., 2026). Benchmark comparisons use geometry predicted from RGB by Depth Anything 3 (Lin et al., 2026), whose errors against ground truth are in Tab. 9. Sec. A.3 gives evaluation protocols and result sources. Readers are trained for 10 epochs with the VLM frozen; Stage 2 fine-tunes for 2 epochs on the multi-source collection or 10 epochs on MindCube at learning rate 2×10^{-5}, and our models decode greedily (full settings in Sec. A.2.3).

4.2 Main Results

Comparison with existing models. Tabs. 1 and 2 compare **GeoReader**, using DA3-predicted geometry, with existing models. On MindCube-Tiny, **GeoReader** exceeds 3DThinker at every backbone size. Across six additional benchmarks, **GeoReader** with Qwen2.5-VL-3B averages 63.6 versus 60.4 for 3DThinker-3B; with Qwen3-VL-4B, it reaches 70.6. Compared with vanilla SFT, the same QA fine-tuning without geometry, geometry raises accuracy on questions about metric 3D quantities and leaves the other questions unchanged (Tab. 3); single subtasks gain up to +14.3 on SPBench-SI relative distance and +13.5 on MMSI camera motion (Sec. A.4.1).

Table 2: Accuracy (%) on six spatial benchmarks; Avg. is their mean, bold is the best per group, and **GeoReader** uses DA3-predicted geometry. **GeoReader** raises every base model's average.

Method	VSI	SPBench	CV	SPAR	ViewSpatial	MMSI	Avg. ↑
Qwen2.5-VL family							
Qwen2.5-VL-3B† (Qwen Team, 2025a)	27.3	39.5	70.5	27.9	36.1	25.5	37.8
Spatial-MLLM-4B (Wu et al., 2025a)	47.3	48.4	73.8	35.1	43.6	31.5	46.6
SpatialLadder-3B (Li et al., 2026c)	45.7	**70.6**	73.7	34.4	44.2	29.2	49.6
3DThinker-3B (Chen et al., 2026)	**59.1**	60.2	78.4	58.2	64.7	**41.9**	60.4
GeoReader-3B	56.1	70.0	**83.7**	**59.1**	**71.4**	41.2	**63.6**
Qwen2.5-VL-7B† (Qwen Team, 2025a)	29.9	40.8	77.9	33.2	34.7	26.8	40.5
SpaceR-7B (Ouyang et al., 2025)	44.5	54.0	75.3	37.1	45.5	28.8	47.5
VILASR-7B (Wu et al., 2025b)	45.4	53.9	77.1	37.6	46.1	30.2	48.4
Video-R1 (Feng et al., 2025)	33.4	42.8	69.6	31.5	36.1	29.4	40.5
3DThinker-7B (Chen et al., 2026)	**63.7**	68.3	81.1	**63.3**	68.6	**43.3**	64.7
GeoReader-7B	54.9	**70.4**	**83.7**	61.5	**77.5**	41.9	**65.0**
Qwen3-VL family							
Qwen3-VL-4B† (Qwen Team, 2025b)	55.1	58.2	84.4	33.4	38.6	27.2	49.5
GeoReader-4B	**66.0**	**79.5**	**85.7**	**65.1**	**81.5**	**45.8**	**70.6**
Qwen3-VL-8B† (Qwen Team, 2025b)	55.9	57.8	**85.4**	38.2	40.4	28.9	51.1
SenseNova-SI-8B* (Cai et al., 2026)	**64.8**	65.8	83.4	41.1	51.2	38.1	57.4
GeoReader-8B	60.9	**69.6**	80.4	**60.1**	**76.4**	**41.2**	**64.8**
Qwen3.5 family							
Qwen3.5-4B† (Qwen Team, 2026)	50.0	35.5	**84.9**	34.4	34.5	28.1	44.6
GeoReader-4B	**66.7**	**81.5**	81.8	**68.7**	**77.7**	**45.4**	**70.3**
Qwen3.5-9B† (Qwen Team, 2026)	54.1	50.7	85.9	42.1	38.2	32.0	50.5
GeoReader-9B	**68.9**	**80.4**	**86.3**	**70.4**	**79.4**	**45.4**	**71.8**

Table 3: Accuracy (%) of vanilla SFT and **GeoReader** by question type. Both use Qwen2.5-VL-3B with matched data and settings; the 59 subtasks of the six benchmarks are grouped by whether they ask about a metric 3D quantity (subtask counts in parentheses), weighted by question count. Geometry helps questions about metric 3D quantities without hurting the others.

Question type	Questions	Vanilla	**GeoReader**	Δ
Object and room size (4)	1,862	65.2	70.0	+4.8
Camera motion (4)	967	39.3	43.7	+4.4
Distance and depth (19)	6,613	56.8	59.0	+2.2
All metric 3D questions (27)	9,442	56.7	**59.6**	+2.9
All other questions (32)	13,577	67.3	67.5	+0.3

4.3 Learning VLM-Readable Geometry

MindCube ablations fix Qwen2.5-VL-3B, 10K QA examples, Stage 2 settings and the 1,050-question test set, and use symbolic geometry from annotations so that estimation errors do not affect design comparisons. Tab. 6 and Panel A of Tab. 4 show that both stages matter.

Reader pretraining. Panel A compares the two routes of Sec. 1 with ours. QA-only training of a random Reader reaches 54.7% and readability pretraining 76.4%, both tuning the Reader in QA training; feature alignment, represented by the released 3DThinker checkpoint (Chen et al., 2026) on the same backbone and questions, reaches 62.7%. Readability pretraining thus works best.

Pair versus group supervision. With frozen Readers in QA training, pair supervision, which removes the view ambiguity of Sec. 3.3, reaches 89.2% versus 77.8% for group (Tab. 5; Sec. A.4.5).

Table 4: MindCube-Tiny accuracy (%). A: Reader pretraining; B: complete interfaces. Readability pretraining and **GeoReader** score highest.

Geometric interface	Overall ↑
A. Reader pretraining	
No Stage 1 (QA loss only)	54.7
Feat. align. (released)	62.7
Readability pretraining	76.4
B. Complete interfaces	
VLM-3R-style (VGGT)	86.1
SpaceMind-style (VGGT)	81.1
Per-token additive (pose)	73.8
GeoReader	**89.2**

Table 5: MindCube-Tiny accuracy (%) and inference cost of text and continuous geometry, with precomputed symbolic geometry. Continuous geometry is both more accurate and faster than text geometry.

Recipe	Acc. ↑	Tokens	Latency (s) ↓
A. No geometry			
Vanilla SFT	68.6	16.2	0.525
B. Text geometry			
Joint description + QA SFT	67.0	16.3	0.523
Generated Text-CoT	72.9	119.3	2.544
Descriptions in prompt	81.0	16.3	0.514
C. Continuous geometry			
GeoReader (group)	77.8	16.2	0.504
GeoReader (pair)	**89.2**	16.2	0.504

Table 6: Accuracy (%) per stage. Stage 1 alone is not enough; both are needed.

Configuration	QA ↑
Vanilla SFT	68.6
Stage 1 only	41.4
Stage 1 + Stage 2	**89.2**

Table 7: Accuracy (%) per injection design. Question conditioning works best.

Injection design	QA ↑
Direct (no gate)	78.6
Learned fixed scaling	68.5
Question-conditioned	**89.2**

Table 8: Accuracy (%) per geometry intervention. Only **GeoReader** uses its geometry.

Geometry	**GeoReader**	Feat. align.
Shuffled	63.1	61.5
Zero	77.2	62.9
Correct	**89.2**	62.7

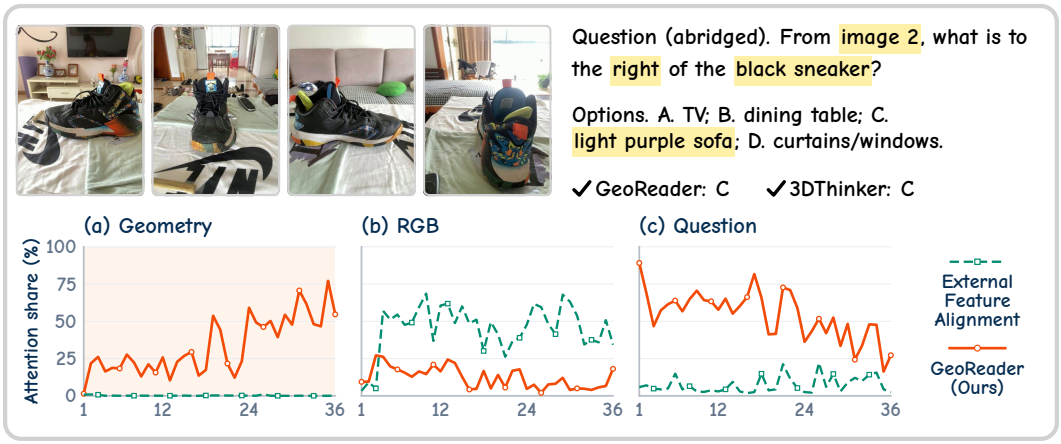


Figure 3: Layer-wise answer-token attention on a MindCube question that both models answer correctly. Answer tokens attend more to geometry in **GeoReader** than to latents in 3DThinker.

Freezing the Readers. Starting from the same pair-supervised Stage 1 Reader, freezing it during QA training raises accuracy from 76.4% to 89.2% (Tab. 4, Panels A/B): QA answers do not require the Readers to keep Stage 1 relationships, so the learned interface is best kept fixed.

Alternative interfaces. Panel B of Tab. 4 compares **GeoReader** with complete direct-injection interfaces under Panel A's protocol. Adapted VLM-3R- and SpaceMind-style fusion with dense VGGT features reach 86.1% and 81.1% (Sec. A.4.6); adding symbolic pose descriptors to image-token embeddings reaches 73.8%, and **GeoReader** reaches 89.2% with the same geometry.

4.4 Using Geometry for the Question

Injection design. Scaling geometry by the question helps: question-conditioned injection reaches 89.2%, versus 78.6% for direct injection and 68.5% for learned fixed scaling (Tab. 7; Sec. A.2.2).

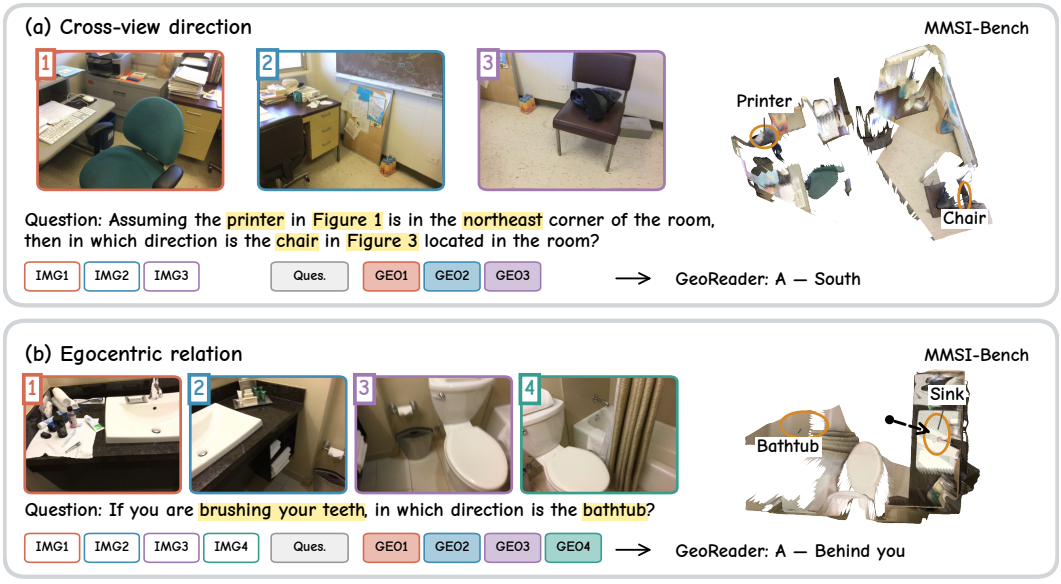


Figure 4: Two MMSI-Bench questions, on (a) a direction across views and (b) a direction from an imagined viewpoint, that **GeoReader** answers correctly with DA3-predicted geometry.

Geometry interventions. The answers depend on both geometric content and its match to the views, which reconstruction alone does not ensure. With the checkpoint, RGB and questions fixed, zeroing the geometric residual, which keeps the slot embeddings, lowers accuracy from 89.2% to 77.2%, and shuffling poses across views lowers it further to 63.1% (Tab. 8; Sec. A.4.7). For feature alignment (Feat. align. in Tab. 8), zeroing the latents of the same released checkpoint or swapping them across questions only moves accuracy from 62.7% to 62.9% and 61.5%, respectively.

Answer-token attention. Answer-token attention shows the same contrast. On a MindCube question that both models answer correctly, **GeoReader**'s answer tokens attend more to geometric tokens than those of 3DThinker attend to its latents (Fig. 3; Sec. A.5.1).

4.5 Interface Trade-offs and Qualitative Analysis

Text versus continuous geometry. Descriptions serve best as Reader supervision. As extra SFT targets, they reach 67.0%, below vanilla SFT (68.6%; Tab. 5). Generating them at inference reaches 72.9% and supplying them in the prompt 81.0%, whereas **GeoReader**, which uses them only to train the Reader, reaches 89.2%. With precomputed geometry, it generates 16.2 tokens in 0.504 s, versus 119.3 and 2.544 s when generating them (Sec. A.4.5).

Qualitative examples. Fig. 4 shows two MMSI-Bench cross-view questions that **GeoReader** answers correctly with predicted geometry; Secs. A.5.2 and A.5.3 give more successes and failures.

5 Conclusion

We presented **GeoReader**, which converts geometry into a form that a frozen VLM can read before the VLM learns to use it. Geometric Readers learn through geometry-description prediction by the frozen VLM, then stay frozen while the VLM and a question-conditioned geometry injector learn spatial QA; descriptions serve only as training targets. Latents aligned with geometry features can be zeroed without lowering accuracy, whereas removing **GeoReader**'s geometry lowers it. With geometry predicted from RGB, **GeoReader** reaches 94.3 on MindCube-Tiny and a six-benchmark average of 70.6 with Qwen3-VL-4B. Capturing geometry does not ensure its use; the representation that helps most is the one the receiving model can read.

AI USE STATEMENT

We used generative AI tools to assist with implementing training code, generating training data and figures, and manuscript organization and editing. The authors manually reviewed the AI-assisted training code and the generated training data. The authors take responsibility for the final content of this work, including its text, claims, code, data, and figures.

REPRODUCIBILITY STATEMENT

Sec. 3 describes the model and training objectives, and Sec. A.2.1 details the Geometric Reader architecture. The training-data composition and split boundaries are described in Sec. A.1.2, and geometry sources in Sec. A.1.3. Training settings for the injection comparisons and geometry-description recipes are provided in Secs. A.2.2 and A.4.5. Sec. A.3 and Sec. A.4.7 document evaluation settings and geometry-content interventions, respectively.

ETHICS STATEMENT

This work uses publicly available datasets for which the authors have obtained permission for the uses in this study. Predicted geometry and answers can be wrong, e.g., under reference-frame confusion. Benchmark performance does not establish reliability for safety-critical applications such as navigation or robotic control; such use requires application-specific validation and safeguards.

REFERENCES

Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-Scale Data for Multiple-View Stereopsis. *International Journal of Computer Vision*, 120 (2):153–168, 2016. doi: 10.1007/s11263-016-0902-9.

Siddhant Bansal, Zhifan Zhu, Shashank Tripathi, Jiahe Zhao, Michael J. Black, and Dima Damen. Towards in-the-wild Egocentric 3D Hand-Object Pose Estimation. In *European Conference on Computer Vision*, 2026. URL https://eccv.ecva.net/virtual/2026/poster/3943.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes: A Diverse Real-World Dataset for 3D Indoor Scene Understanding Using Mobile RGB-D Data. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/66f041e16a60928b05a7e228a89c3799-Paper-round1.pdf.

Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3D: A Large Benchmark and Model for 3D Object Detection in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13154–13164, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/papers/Brazil_Omni3D_A_Large_Benchmark_and_Model_for_3D_Object_Detection_CVPR_2023_paper.pdf.

ByteDance Seed. Introduction to Techniques Used in Seed1.6, 2025. URL https://seed.bytedance.com/en/blog/introduction-to-techniques-used-in-seed1-6.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020. URL http://openaccess.thecvf.com/content_CVPR_2020/papers/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.pdf.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9490–9498. IEEE, 2025. doi: 10.1109/ICRA55743.2025.11128671. URL https://doi.org/10.1109/ICRA55743.2025.11128671.

Zhongang Cai, Ruisi Wang, Chenyang Gu, Fanyi Pu, Junxiang Xu, Yubo Wang, Wanqi Yin, Zhitao Yang, Chen Wei, Qingping Sun, Tongxi Zhou, Jiaqi Li, Hui En Pang, Oscar Qian, Yukun Wei, Zhiqian Lin, Xuanke Shi, Kewang Deng, Xiaoyang Han, Zukai Chen, Xiangyu Fan, Hanming Deng, Lewei Lu, Liang Pan, Bo Li, Ziwei Liu, Quan Wang, Dahua Lin, and Lei Yang. Scaling Spatial Intelligence with Multimodal Foundation Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7879–7890, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Cai_Scaling_Spatial_Intelligence_with_Multimodal_Foundation_Models_CVPR_2026_paper.pdf.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *Proceedings of the International Conference on 3D Vision*, pp. 667–676, 2017. doi: 10.1109/3DV.2017.00081.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14455–14465, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/papers/Chen_SpatialVLM_Endowing_Vision-Language_Models_with_Spatial_Reasoning_Capabilities_CVPR_2024_paper.pdf.

Pingyi Chen, Yujing Lou, Shen Cao, Jinhui Guo, Lubin Fan, Yue Wu, Lin Yang, Lizhuang Ma, and Jieping Ye. SD-VLM: Spatial Measuring and Understanding with Depth-Encoded Vision-Language Models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 33938–33968. Curran Associates, Inc., 2025. doi: 10.52202/085713-1140. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/30bc3a3a44c9d2e3f32e6dd1cd18f552-Paper-Conference.pdf.

Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Xiang An, Yan Feng, Peng Pei, Xunliang Cai, and Ruqi Huang. Think with 3D: Geometric Imagination Grounded Spatial Reasoning from Limited Views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2613–2624, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Chen_Think_with_3D_Geometric_Imagination_Grounded_Spatial_Reasoning_from_Limited_CVPR_2026_paper.pdf.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 135062–135093. Curran Associates, Inc., 2024. doi: 10.52202/079017-4293. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f38cb4cf9a5eaa92b3cfa481832719c6-Paper-Conference.pdf.

An-Chieh Cheng, Yang Fu, Yukang Chen, Zhijian Liu, Xiaolong Li, Subhashree Radhakrishnan, Song Han, Yao Lu, Jan Kautz, Pavlo Molchanov, Hongxu Yin, Xiaolong Wang, and Sifei Liu. 3D Aware Region Prompted Vision Language Model. In *International Conference on Learning Representations*, volume 2026, pp. 123286–123308, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c86d3f43371270c70ed94e024852895d-Paper-Conference.pdf.

Jiho Choi, Seonho Lee, Seojeong Park, and Hyunjung Shim. Dense Reward for Multi-View 3D Reasoning with Global Maps and Local Views. In *European Conference on Computer Vision (ECCV)*, 2026. URL https://eccv.ecva.net/virtual/2026/poster/5231.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017. URL http://openaccess.thecvf.com/content_cvpr_2017/papers/Dai_ScanNet_Richly-Annotated_3D_CVPR_2017_paper.pdf.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the Limits of KV Cache Compression. *arXiv preprint arXiv:2609.19969*, 2026. URL https://arxiv.org/abs/2609.19969.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Peihao Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. VLM-3R: Vision-Language Models Augmented with Instruction-Aligned 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31054–31065, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Fan_VLM-3R_Vision-Language_Models_Augmented_with_Instruction-Aligned_3D_Reconstruction_CVPR_2026_paper.pdf.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing Video Reasoning in MLLMs. In *Advances in Neural Information Processing Systems*, volume 38, pp. 99114–99137. Curran Associates, Inc., 2025. doi: 10.52202/085713-3310. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/8eb3976840e08b80dda9667562574246-Paper-Conference.pdf.

Gemini Team. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261*, 2025. URL https://arxiv.org/abs/2507.06261.

Jiawei Guo, Yu Chen, Xiang Wang, Shuai Li, Xinpei Zhao, Huaxing Liu, Shuai Dong, Feifei Zhai, and Yu Zhou. Beyond Visual Memory: Mechanistic Diagnostics of Latent Visual Reasoning. *arXiv preprint arXiv:2606.01287*, 2026.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training Large Language Models to Reason in a Continuous Latent Space. In *Second Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2412.06769.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3D-LLM: Injecting the 3D World into Large Language Models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 20482–20494. Curran Associates, Inc., 2023. doi: 10.52202/075280-0900. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/413885e70482b95dcbeeddc1daf39177-Paper-Conference.pdf.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning Dexterous Manipulation from Large-Scale Egocentric Video. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/07fcc6e2b89439d3ee5ab60939aaa6a0-Paper-Conference.pdf.

Wenbo Hu, Jingli Lin, Yilin Long, Yunlong Ran, Lihan Jiang, Yifan Wang, Chenming Zhu, Runsen Xu, Tai Wang, and Jiangmiao Pang. G^2VLM: Geometry grounded vision language model with unified 3d reconstruction and spatial reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9535–9546, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Hu_G2VLM_Geometry_Grounded_Vision_Language_Model_with_Unified_3D_Reconstruction_CVPR_2026_paper.pdf.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An Embodied Generalist Agent in 3D World. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20413–20451. PMLR, 21–27 Jul

2024. URL https://raw.githubusercontent.com/mlresearch/v235/main/assets/huang24ae/huang24ae.pdf.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding Image Matching in 3D with MASt3R. In *Computer Vision – ECCV 2024*, pp. 71–91. Springer, 2024. doi: 10.1007/978-3-031-73220-1_5. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/09080.pdf.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent Visual Reasoning. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/efa0939caa8a906fd0a61a9c60e46e8b-Paper-Conference.pdf.

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. ViewSpatial-Bench: Evaluating Multi-perspective Spatial Localization in Vision-Language Models. In *European Conference on Computer Vision*, 2026b. URL https://eccv.ecva.net/virtual/2026/poster/3165.

Hongxing Li, Dingming Li, Zixuan Wang, Yuchen Yan, Hang Wu, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. SpatialLadder: Progressive Training for Spatial Reasoning in Vision-Language Models. In *International Conference on Learning Representations*, volume 2026, pp. 76566–76592, 2026c. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7bc3fe234454107149fa9d44faacaa64-Paper-Conference.pdf.

You Li, Chi Chen, Yanghao Li, Fanhu Zeng, Kaiyu Huang, Jinan Xu, and Maosong Sun. Imagination Helps Visual Reasoning, But Not Yet in Latent Space. In *Forty-third International Conference on Machine Learning*, 2026d. URL https://arxiv.org/abs/2602.22766.

Yunheng Li, Guohong Mu, Hao Li, Shengsheng Qian, Dingwen Zhang, Qibin Hou, and Ming-Ming Cheng. Annotations as Rollouts: Efficient and Scalable Reinforcement Learning for Video MLLMs. *arXiv preprint arXiv:2608.20492*, 2026e. URL https://arxiv.org/abs/2608.20492.

Huizhi Liang, Yichao Shen, Yu Deng, Sicheng Xu, Zhiyuan Feng, Tong Zhang, Yaobo Liang, and Jiaolong Yang. HiSpatial: Taming Hierarchical 3D Spatial Understanding in Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2502–2514, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Liang_HiSpatial_Taming_Hierarchical_3D_Spatial_Understanding_in_Vision-Language_Models_CVPR_2026_paper.pdf.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the Visual Space from Any Views. In *International Conference on Learning Representations*, volume 2026, pp. 141261–141285, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e4cd50120b6d7e8daff1749d6bbaa889-Paper-Conference.pdf.

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 *Computer Vision – ECCV 2014*, pp. 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

Fangfu Liu, Diankun Wu, Jiawei Chi, Yimo Cai, Yi-Hsin Hung, Xumin Yu, Hao Li, Han Hu, Yongming Rao, and Yueqi Duan. Spatial-TTT: Streaming Visual-based Spatial Intelligence with Test-Time Training. In *European Conference on Computer Vision*, 2026. URL https://eccv.ecva.net/virtual/2026/poster/5426.

LLM-Core Xiaomi. MiMo-V2.6: Scaling Reinforcement Learning Towards Self-Improvement. Technical report, LLM-Core Xiaomi, 2026. URL https://huggingface.co/XiaomiMiMo/MiMo-V2.6-Flash-RL/resolve/main/MiMo_V2_6_technical_report.pdf.

Michael Ogezi and Freda Shi. SpaRE: Enhancing Spatial Reasoning in Vision-Language Models with Synthetic Data. *arXiv preprint arXiv:2504.20648*, 2025. URL https://arxiv.org/abs/2504.20648.

Masanari Oi, Koki Maeda, Ryuto Koike, Daisuke Oba, Nakamasa Inoue, and Naoaki Okazaki. From Correspondence to Actions: Human-Like Multi-Image Spatial Reasoning in Multi-modal Large Language Models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://stjohn2007.github.io/HATCH_project/.

OpenAI. Introducing GPT-4.1 in the API, 2025a. URL https://openai.com/index/gpt-4-1/.

OpenAI. OpenAI o3 and o4-mini System Card, 2025b. URL https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf.

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. SpaceR: Reinforcing MLLMs in Video Spatial Reasoning. *arXiv preprint arXiv:2504.01805*, 2025.

Suhyeong Park, Junha Jung, and Jaewoo Kang. Reason Through the Latent! Making Latent Visual Reasoning Necessary. *arXiv preprint arXiv:2609.06746*, 2026.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS Challenge on Video Object Segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Yiming Qin, Bomin Wei, Jiaxin Ge, Konstantinos Kallidromitis, Stephanie Fu, Trevor Darrell, and XuDong Wang. Chain-of-Visual-Thought: Teaching VLMs to See and Think Better with Continuous Visual Tokens. In *European Conference on Computer Vision*, 2026. URL https://eccv.ecva.net/virtual/2026/poster/4447.

Qwen Team. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025a.

Qwen Team. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025b.

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

Fatemeh Shiri, Xiao-Yu Guo, Mona Golestan Far, Xin Yu, Gholamreza Haffari, and Yuan-Fang Li. An Empirical Analysis on Spatial Reasoning Capabilities of Large Multimodal Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21440–21455, 2024. URL https://aclanthology.org/2024.emnlp-main.1195.pdf.

Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 567–576, 2015. URL http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Song_SUN_RGB-D_A_2015_CVPR_paper.pdf.

Team AgiBot-World. AgiBot World Colosseo: A Large-scale Manipulation Platform for Scalable and Intelligent Embodied Systems. *arXiv preprint arXiv:2503.06669*, 2025.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha 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 *Advances in Neural Information Processing Systems*, volume 37, pp. 87310–87356. Curran Associates, Inc., 2024. doi: 10.52202/

079017-2771. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9ee3a664ccfeabc0da16ac6f1f1cfe59-Paper-Conference.pdf.

André G. Viveiros, Nuno Gonçalves, André F. T. Martins, and Matthias Lindemann. What's Holding Back Latent Visual Reasoning? *arXiv preprint arXiv:2605.18445*, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, June 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.pdf.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D Perception Model with Persistent State. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522, June 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Wang_Continuous_3D_Perception_Model_with_Persistent_State_CVPR_2025_paper.pdf.

Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Jiajun Wu, Li Fei-Fei, and Manling Li. MindCube: Spatial Mental Modeling from Limited Views. In *International Conference on Learning Representations*, volume 2026, pp. 82559–82622, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/858de961874816f894285ec02e50c7fe-Paper-Conference.pdf.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in Latent Visual Space Beyond Image and Language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12030–12040, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Wang_Monet_Reasoning_in_Latent_Visual_Space_Beyond_Image_and_Language_CVPR_2026_paper.pdf.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/papers/Wang_DUSt3R_Geometric_3D_Vision_Made_Easy_CVPR_2024_paper.pdf.

Yuxin Wang, Lei Ke, Boqiang Zhang, Tianyuan Qu, Hanxun Yu, Zhenpeng Huang, Meng Yu, Dan Xu, and Dong Yu. N3D-VLM: Native 3D Grounding Enables Accurate Spatial Reasoning in Vision-Language Models. *arXiv preprint arXiv:2512.16561*, 2025c. URL https://arxiv.org/abs/2512.16561.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-MLLM: Boosting MLLM Capabilities in Visual-based Spatial Intelligence. In *Advances in Neural Information Processing Systems*, volume 38, pp. 13569–13597. Curran Associates, Inc., 2025a. doi: 10.52202/085713-0457. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/142b166c8b18b43774c4f0e0c1e9948e-Paper-Conference.pdf.

Jing Wu, Jianhua Wu, Jiayi Guan, Jiahong Chen, Jinghui Lu, Hangjun Ye, Bingzhao Gao, and Long Chen. SpatioLM: Towards general physical spatial intelligence in vision-language models. *arXiv preprint arXiv:2608.01899*, 2026. URL https://arxiv.org/abs/2608.01899.

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing Spatial Reasoning in Vision-Language Models with Interwoven Thinking and Visual Drawing. In *Advances in Neural Information Processing Systems*, volume 38, pp. 143297–143330. Curran Associates, Inc., 2025b. doi: 10.52202/085713-4797. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d2f2ca2c305e1a9c29bf00a676f0a500-Paper-Conference.pdf.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Yang_Thinking_in_Space_How_Multimodal_Large_Language_Models_See_Remember_CVPR_2025_paper.pdf.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *Advances in Neural Information Processing Systems*, volume 37, pp. 21875–21911. Curran Associates, Inc., 2024. doi: 10.52202/079017-0688. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/26cfdcd8fe6fd75cc53e92963a656c58-Paper-Conference.pdf.

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 *International Conference on Learning Representations*, volume 2026, pp. 157051–157088, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/fe5d4bd3e2af823701b5d8f70a3c7602-Paper-Conference.pdf.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12–22, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/papers/Yeshwanth_ScanNet_A_High-Fidelity_Dataset_of_3D_Indoor_Scenes_ICCV_2023_paper.pdf.

Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, and Li Zhang. From Flatland to Space: Teaching Vision-Language Models to Perceive and Reason in 3D. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-5180. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e3a0db7c0a191854c176af1d20cdec80-Paper-Datasets_and_Benchmarks_Track.pdf.

Jiang Zhang, Shijie Zhou, Bangya Liu, Achuta Kadambi, and Zhiwen Fan. SpatialStack: Layered Geometry-Language Fusion for 3D VLM Spatial Reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38678–38688, June 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Zhang_SpatialStack_Layered_Geometry-Language_Fusion_for_3D_VLM_Spatial_Reasoning_CVPR_2026_paper.pdf.

Jiaxin Zhang, Junjun Jiang, Haijie Li, Youyu Chen, Kui Jiang, and Dave Zhenyu Chen. GAP-MLLM: Geometry-aligned pre-training for activating 3d spatial perception in multimodal large language models. In *European Conference on Computer Vision*, 2026b. URL https://eccv.ecva.net/virtual/2026/poster/3455.

Shihua Zhang, Qiuhong Shen, Shizun Wang, Tianbo Pan, and Xinchao Wang. Make Geometry Matter for Spatial Reasoning. In *European Conference on Computer Vision*, 2026c. URL https://eccv.ecva.net/virtual/2026/poster/3231.

Ruosen Zhao, Zhikang Zhang, Jialei Xu, Jiahao Chang, Dong Chen, Lingyun Li, Weijian Sun, and Zizhuang Wei. SpaceMind: Camera-Guided Modality Fusion for Spatial Reasoning in Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16811–16822, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Zhao_SpaceMind_Camera-Guided_Modality_Fusion_for_Spatial_Reasoning_in_Vision-Language_Models_CVPR_2026_paper.pdf.

Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from Videos for 3D World: Enhancing MLLMs with 3D Vision Geometry Priors. In *Advances in Neural Information Processing Systems*, volume 38, pp. 20560–20586. Curran Associates, Inc., 2025a. doi: 10.52202/

085713-0693. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1d8dc55c1f6cf124af840ce1d92d1896-Paper-Conference.pdf.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3D LLM: Learning Position-Aware Video Representation for 3D Scene Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8995–9006, June 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Zheng_Video-3D_LLM_Learning_Position-Aware_Video_Representation_for_3D_Scene_Understanding_CVPR_2025_paper.pdf.

Yufei Zheng, Xuhan Zhu, Zide Liu, Chunpeng Zhou, Chenfeng Wang, Yongchao Xu, Yunnan Wang, Jiawei Liu, Pengfei Yu, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Dual-Pathway Geometry-Aware MLLM for Spatial Intelligence, 2026. URL https://arxiv.org/abs/2605.25334.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. LLaVA-3D: A Simple yet Effective Pathway to Empowering LMMs with 3D Capabilities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4295–4305, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/papers/Zhu_LLaVA-3D_A_Simple_yet_Effective_Pathway_to_Empowering_LMMs_with_ICCV_2025_paper.pdf.

Dongyao Zhu, Zhen Wang, Xi Xiao, Han Jiang, Saeed Vahidian, Wei-Lun Chao, Tanya Berger-Wolf, Yu Su, Raju Vatsavai, and Jianyang Gu. Leveraging Latent Visual Reasoning in Silence. *arXiv preprint arXiv:2605.18641*, 2026.

SUPPLEMENTARY MATERIAL

CONTENTS

A.1 Training Data and Geometry Sources 18
A.1.1 Stage 1: Geometry-Description Data 18
A.1.2 Stage 2: QA Data and Test-Set Exclusion 20
A.1.3 Geometry Sources 20
A.2 Model and Training Details 21
A.2.1 Geometric Reader Architecture 21
A.2.2 Injection Design and Computation 23
A.2.3 Training Settings 23
A.3 Evaluation Protocols and Result Sources 24
A.3.1 Sources of Baseline Results 24
A.3.2 Benchmark Inputs and Metrics 24
A.3.3 Local-Model Evaluation and Scoring 24
A.3.4 API-Model Evaluation 25
A.4 Additional Experiments 25
A.4.1 Geometry versus RGB-Only Training on Six Benchmarks 25
A.4.2 Additional Backbone Results 26
A.4.3 Predicted versus Symbolic Geometry 27
A.4.4 Reader Pretraining 27
A.4.5 Pair versus Group Supervision and Text Geometry 28
A.4.6 Alternative Interfaces 28
A.4.7 Geometry Interventions 28
A.4.8 Inference-Cost Measurement 29
A.5 Qualitative Analysis 29
A.5.1 Answer-Token Attention 29
A.5.2 Additional Qualitative Results 29
A.5.3 Failure Cases 32
A.6 Public Resources 34
A.6.1 Public Datasets 34
A.6.2 Public Implementations and Models 34

A APPENDIX

A.1 TRAINING DATA AND GEOMETRY SOURCES

A.1.1 STAGE 1: GEOMETRY-DESCRIPTION DATA

The window-based Stage 1 corpus is built from training scenes in ScanNet (Dai et al., 2017) and ScanNet++ (Yeshwanth et al., 2023). It contains approximately 29.5K Camera description records and 35.4K Depth description records, drawn from a shared pool of 1,550 scenes. The two branches share the same windows. Camera windows contain 2, 4, 8, 16 or 32 views; Depth windows additionally include single images.

Camera targets are constructed from annotated camera poses and describe planar translation in meters and heading changes relative to the first view. For each window, we adopt the first camera's coordinate frame as the reference world frame and transform all camera poses into it. The first camera therefore has identity pose and lies at the origin; the x and z axes point right and forward relative to that view. Depth targets describe object positions using a 3×3 image grid (upper-left, upper, upper-right, left, center, right, lower-left, lower, and lower-right), together with distances in meters. We retain up to five object entries per view from projected instance geometry and annotations. Single-view windows are omitted from Camera training; windows without any retained object entries are omitted from Depth training. The construction pipeline excludes scenes on its benchmark-scene exclusion list before sampling windows, and its validation report records no overlap with that list. Fig. 5 shows three shared windows, their geometry inputs and description excerpts. The nine-grid regions define the textual position labels; they are distinct from the finer geometry patch grid.

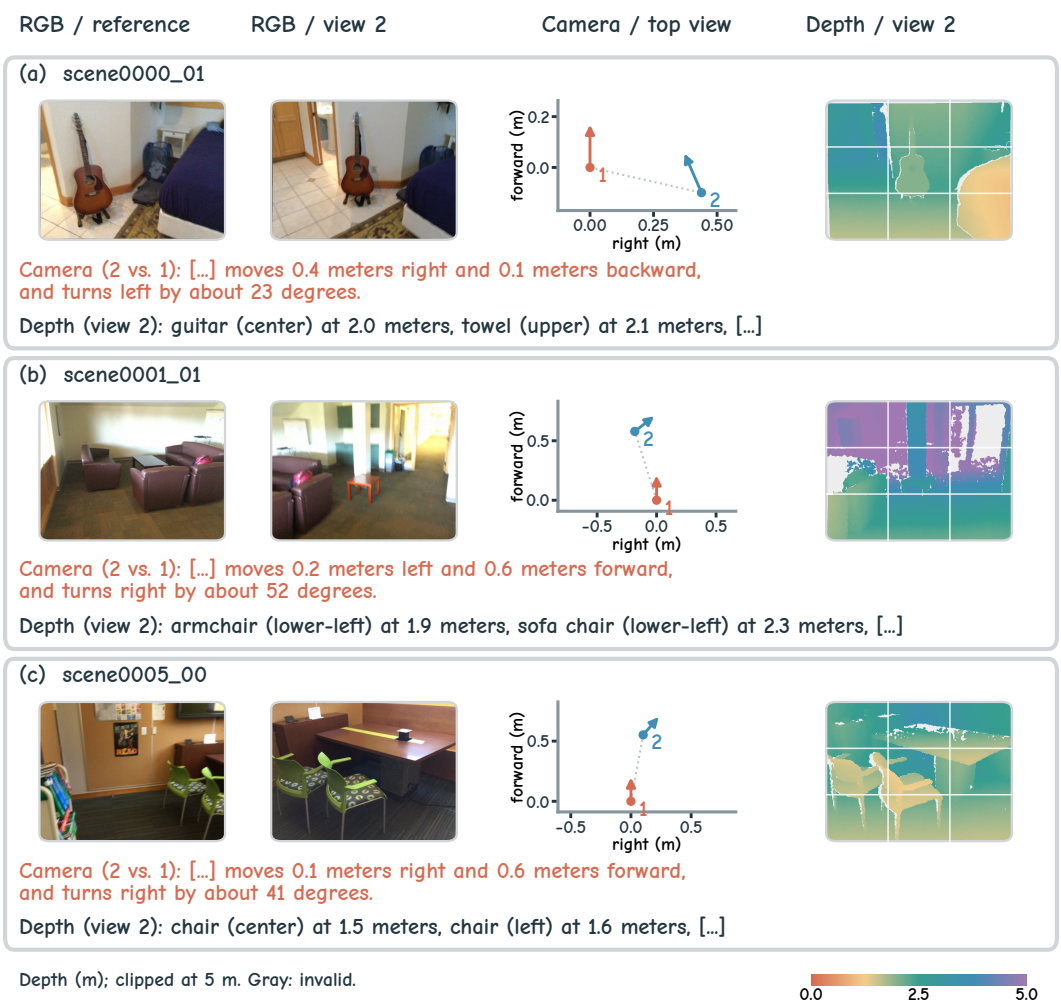


Figure 5: Three Stage 1 training examples from ScanNet (Dai et al., 2017): RGB pairs, camera poses in the first camera's reference frame, and view 2 depth with nine-grid regions. Description excerpts preserve original wording and values; [...] marks omissions. Depth colors are clipped at 5 m; gray is invalid.

The description construction for MindCube (Wang et al., 2026a) is separate from this window corpus. It uses 2,072 training image groups, yielding 3,375 reference-view pairs for pair-based Stage 1 training. Sec. A.4.5 details the group-versus-pair supervision and its counting convention.

Reference–target pair construction. The group–pair construction applies to Camera Reader pretraining (Eq. 2). For example, a four-view group yields three independent pairs $(1, 2)$, $(1, 3)$ and $(1, 4)$, rather than all six possible pairs. Each pair retains only its two images, the corresponding camera geometry and one relative-pose description. The reference coordinate frame and source camera geometry are preserved. Pairs repeated across source groups are retained. Both recipes use the full description for teacher forcing but compute loss only at geometric-content positions, excluding the fixed reference sentence and template text, since template words can be predicted without geometry. Pair construction uses training groups only; QA inputs keep all views.

For example, pairs $(1, 2)$ and $(1, 3)$ supervise each target's relation to view 1. The excluded view and its geometry are absent from that pair's forward pass, rather than merely excluded from the loss. The two pairs share model weights but have no cross-pair attention context.

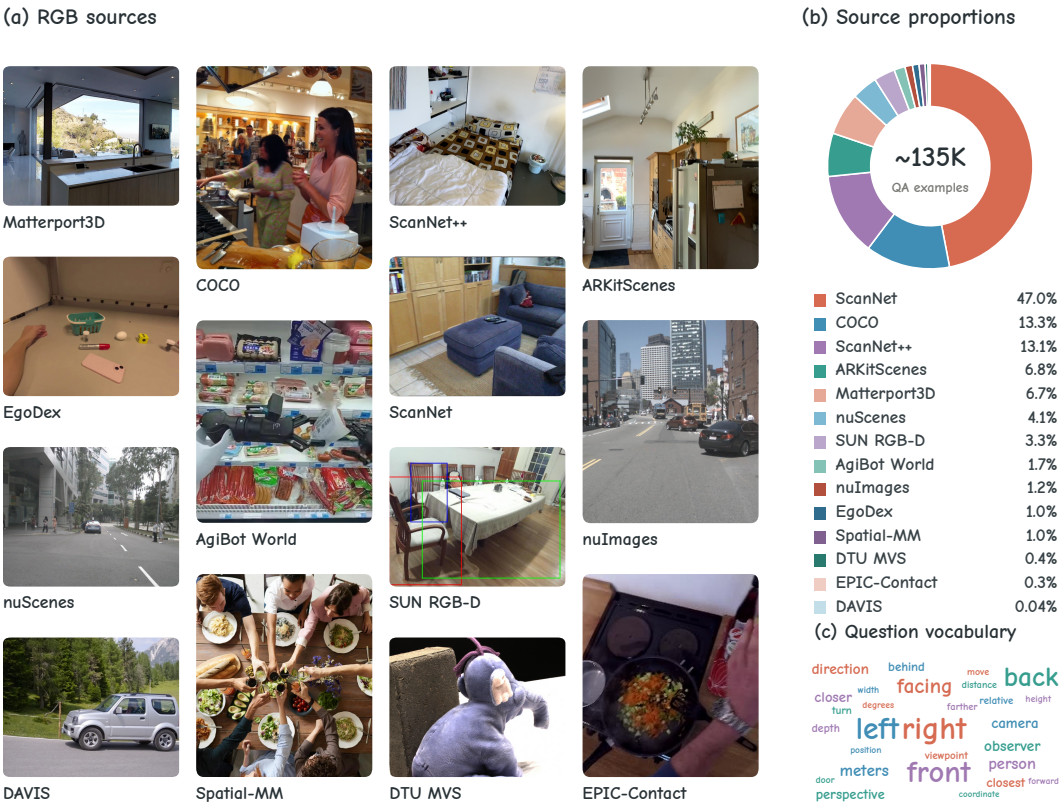


Figure 6: The approximately 135K-example Stage 2 QA collection. Left: cropped RGB examples; image sizes do not encode frequency. Right: QA shares by RGB source and selected spatial terms sized by question-stem frequency. All 14 RGB sources are shown.

A.1.2 Stage 2: QA Data and Test-Set Exclusion

Our larger QA training collection has approximately 135K examples in 36 JSONL files. These files combine training questions from VSI-style tasks (Yang et al., 2025), SpatialLadder (Li et al., 2026c), SPAR (Zhang et al., 2025), ViewSpatial (Li et al., 2026b), Spatial-MM (Shiri et al., 2024), MMSI (Yang et al., 2026), and CV-Bench-style distance tasks (Tong et al., 2024). The RGB images come primarily from ScanNet (Dai et al., 2017), COCO (Lin et al., 2014), ScanNet++ (Yeshwanth et al., 2023), ARKitScenes (Baruch et al., 2021), Matterport3D (Chang et al., 2017), and nuScenes (Caesar et al., 2020). We exclude the official held-out benchmark test questions and answers from Stage 2 training. Sources with published training partitions are drawn from those partitions; separately generated questions use the source training pools. Stage 1 and Stage 2 training scenes do not overlap with the scenes of the evaluation benchmarks. Fig. 6 illustrates the visual range of the inputs and shows QA-example shares for all 14 RGB sources.

The multi-source collection is separate from the 10K MindCube QA set used in the controlled MindCube ablations.

A.1.3 Geometry Sources

For predicted geometry, we use the Depth Anything 3 (Lin et al., 2026) `DA3NESTED-GIANT-LARGE-1.1` checkpoint to estimate camera poses and depth maps from RGB alone. Tab. 9 reports the errors of these predictions on training and evaluation images whose RGB sources provide metric ground truth. Each cell reports the mean over questions, after averaging over valid images or image pairs within a question. Depth and pose were sampled independently. In the evaluation block, the ScanNet pose and ScanNet++ pose values use 175 and 195 valid questions, respectively, after excluding questions with invalid native poses. The ScanNet

Table 9: DA3 geometry errors on training and evaluation images with ground truth. Rows group images by RGB source; A and B each sample up to 200 questions per source and metric from Stage 2 training and evaluation, respectively. Depth errors use valid ground-truth pixels without scale alignment. Pose errors are relative to the earliest image in each question with valid ground-truth pose; only subsequent valid images are scored: rotation error in degrees and camera-position error in meters, without scale alignment. SUN RGB-D (Song et al., 2015) images are single views without pose (–). Matterport3D (Chang et al., 2017), nuScenes (Caesar et al., 2020) and Omni3D (Brazil et al., 2023) images are omitted because they lack pixel-aligned metric ground truth. Errors stay low on indoor scan sources and are highest on SUN RGB-D.

RGB source	Benchmarks	AbsRel (%) ↓	δ_1 (%) ↑	Rot. (°) ↓	Trans. (m) ↓
A. Training images					
ScanNet		3.3	98.7	2.9	0.13
ScanNet++		4.6	97.4	4.3	0.27
ARKitScenes		5.7	96.3	2.7	0.15
SUN RGB-D		26.1	89.3	–	–
B. Evaluation images					
ScanNet	VSI, SPBench, ViewSpatial, SPAR, MMSI	5.9	97.4	2.8	0.11
ScanNet++	VSI, SPAR	8.8	92.3	2.7	0.18
ARKitScenes	VSI	8.0	94.1	3.0	0.18

depth sample is restricted to scenes with locally available sensor data. ARKitScenes training depth is scored only on the pixels covered by official depth after RGB–depth camera registration (16% of the RGB grid on average).

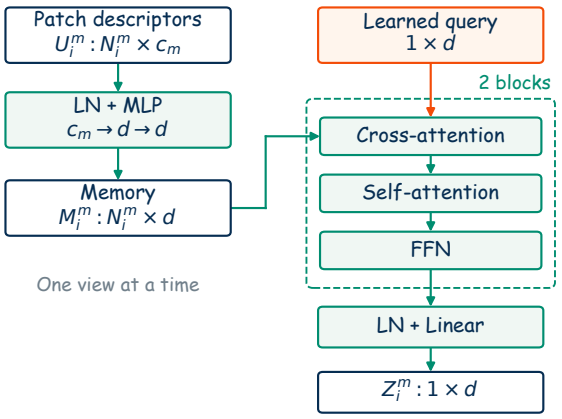
For controlled ablations, we derive symbolic geometry from the grounded cognitive maps provided by MindCube (Wang et al., 2026a). We extract only each view's two-dimensional grid position and discrete facing direction, without using object annotations, reasoning chains, or QA answers to construct the geometric inputs. Views are matched to the ordered input images, and their positions and orientations are expressed relative to the first view. One grid cell defines one input unit; relative height, pitch, and roll are set to zero. Intrinsics follow a fixed normalized convention ($f_x/W = f_y/H = 1$, $c_x/W = c_y/H = 0.5$), rather than camera calibration. These inputs provide annotation-derived symbolic spatial relations. Using them reduces the influence of geometry-estimation errors when comparing interface designs; benchmark comparisons use geometry predicted from RGB images with DA3.

A.2 Model and Training Details

A.2.1 Geometric Reader Architecture

Fig. 7 summarizes the per-view Reader and its Qwen2.5-VL-3B configuration.

Per-view encoding. For view i and geometry branch m, let $U_i^m \in \mathbb{R}^{N_i^m \times c_m}$ contain the patch descriptors derived from the raw signal G^m. The descriptor width is $c_{\text{cam}} = 20$ or $c_{\text{dep}} = 5$. The Reader forms memory $M_i^m = \text{MLP}_m(\text{LN}(U_i^m))$, where MLP_m is the two-layer descriptor projection, its linear maps act independently on the last dimension, and $M_i^m \in \mathbb{R}^{N_i^m \times d}$. A learned $1 \times d$ query, shared across views, is updated by two cross-attention–self-attention–feed-forward blocks. Cross-attention uses normalized queries and memory for its query and key projections, and the unnormalized memory for its value projection. Self-attention and the feed-forward network use pre-LayerNorm; all three sublayers have residual connections. Self-attention operates on the query set within a view, not across views. Final LayerNorm and a linear $d \to d$ projection give Z_i^m. The Qwen2.5-VL-3B configuration uses $d = 2048$, eight attention heads, an 8192-wide GELU feed-forward layer, and zero dropout. The two-layer input MLP has widths $c_m \to 2048 \to 2048$. The geometric branches use the same block structure but independent weights.



Reader configuration

Hidden width d	2048
Blocks	2
Attention heads	8
FFN width	8192
Activation	GELU
Dropout	0
Queries per view	1

N_i^m: patch count in view i.
c_m: descriptor dimension.
LN: LayerNorm.

Figure 7: Architecture of a Geometric Reader. A descriptor MLP builds memory, read by a learned query through two attention blocks. Each branch shares weights across views, without cross-view attention. Outputs feed the Stage 2 geometry injector, so each view contributes one geometric token.

Table 10: Camera descriptor components (20 dimensions per geometry patch). View 1 defines the reference frame; intrinsics are normalized by the processed image dimensions.

Component	Dim.	Meaning / encoding
Image position	2	Normalized patch-center coordinates (u, v).
World ray	3	Unit viewing ray in the source world frame.
Reference ray	3	The same ray expressed in the reference-view frame.
Planar displacement	2	Camera-center displacement $(\Delta x, \Delta z)$ relative to the reference view, in source units.
Viewing direction	3	Camera forward direction in the reference-view frame.
Intrinsics	4	$(f_x/W, f_y/H, c_x/W, c_y/H)$; W, H are the processed image width and height.
View index	1	$(i-1)/31$, using a fixed maximum of 32 views.
Reference flag	1	1 for view 1; 0 for other views.
Temporal position	1	Normalized temporal patch center; 0.5 for a still image.

Table 11: Depth descriptor components (5 dimensions per geometry patch). Depth is measured in meters; the center value is log-transformed, and valid fraction records patch coverage.

Component	Dim.	Meaning / encoding
Image position	2	Normalized patch-center coordinates (u, v).
View index	1	$(i-1)/31$, using a fixed maximum of 32 views.
Center depth	1	$\log(1+z)$ for valid center depth z; 0 otherwise.
Valid fraction	1	Fraction of pixels with depth greater than 0.01 m within the patch.

Geometry descriptors. Tabs. 10 and 11 list the 20-dimensional pose-derived and five-dimensional depth-derived inputs, respectively. View $i = 1$ is the reference view. Displacements retain the units of their geometry source; symbolic grid units are not interpreted as meters. Depth is resized by nearest-neighbor interpolation to the processed image grid; values above 0.01 meters are valid, and invalid center depth is encoded as zero. For this backbone, a geometry patch covers a 2×2 group of 14-pixel visual patches. Camera and depth descriptors share the spatial grid and ordering of the merged RGB tokens: for view i, $N_i^{\text{cam}} = N_i^{\text{dep}} = N_{\text{rgb},i}$. Each descriptor is a vector of geometric measurements associated with one image region, not an RGB embedding. Thus the input is patch-grid aligned, whereas the Reader output is aggregated per view. These descriptors are projected into memory directly, and Reader memory contains only geometric descriptors.

View isolation and output interface. Variable patch counts are padded to the largest count within an example, with padded keys excluded from cross-attention. The implementation treats views as

separate batch items inside the Reader, yielding $[V, 1, d]$ before flattening to $[1, V, d]$ in view order. Reference-relative descriptors can encode cross-view relations even though the learned aggregation itself never attends across views. For four views, the Reader outputs are explicitly

$$Z^m = [Z_1^m]\,[Z_2^m]\,[Z_3^m]\,[Z_4^m], \qquad Z_i^m \in \mathbb{R}^{1\times 2048}.$$

Thus each active branch supplies four embeddings in this example and V embeddings in general, independently of the number of image patches. Stage 1 trains the Reader through the frozen VLM; Stage 2 freezes it. The raw output above is taken before the Stage 1 scalar output gate; the Stage 2 injection designs use their own scaling or adapter and gate, as specified in Sec. A.2.2.

A.2.2 Injection Design and Computation

Tab. 7 compares three injection designs using symbolic geometry and the same frozen, pair-supervised Geometric Reader. For the geometry branch evaluated here, let $Z^m \in \mathbb{R}^{V\times d}$ denote its raw output, with one token per view and $d = 2048$. Direct injection inserts these outputs unchanged. Fixed scaling multiplies them by a single trainable scalar γ_m, shared across all examples and views, initialized to 1 and unconstrained by a sigmoid or normalization. Both replace the slot embeddings without an adapter, an added slot embedding, or a geometry-free context pass. The Reader's internal Stage 1 output gate is bypassed in all three configurations. The question-conditioned method follows Sec. 3.

Sequence and computation. For a four-view example, write X_i for the RGB token block of view i, Q_j for question token j, and $\widetilde{Z}_i^m$ for row i of $\widetilde{Z}^m$. Omitting chat-template delimiters, the question-conditioned design uses

$$\begin{aligned} \text{Read:} \quad & [X_1][X_2][X_3][X_4][Q_1]\cdots[Q_{K_q}], \\ \text{Answer:} \quad & [X_1][X_2][X_3][X_4][Q_1]\cdots[Q_{K_q}][\widetilde{Z}_1^m][\widetilde{Z}_2^m][\widetilde{Z}_3^m][\widetilde{Z}_4^m]. \end{aligned}$$

For variable V, these sequences contain $\sum_i N_{\mathrm{rgb},i} + K_q$ and $\sum_i N_{\mathrm{rgb},i} + K_q + V$ embeddings, respectively, excluding template delimiters; each embedding has width 2048. Both training passes use all 36 language-model layers with causal attention. The read pass contains no geometry or answer tokens and extracts the detached layer-36 hidden state at the last valid question token as h_q. That state can see the preceding RGB and question tokens. The answer pass inserts the geometric embeddings before layer 1; geometric positions can attend to the preceding RGB, question and geometric positions, while question positions cannot attend to the subsequent geometry. Answer tokens follow the prompt and can attend to all preceding inputs. Teacher forcing appends the answer sequence and applies QA loss at answer prediction positions; at inference, each next token is predicted from the final-layer hidden state at the last available position. No additional geometry branches or retrieval tokens are used. Direct injection and fixed scaling use only the answer sequence, with their respective embeddings, in one pass. At inference, the question-conditioned design reuses the read-prefix KV cache as described in Sec. A.4.8.

A.2.3 Training Settings

Six-benchmark models. For the models in Tab. 2, each Geometric Reader is trained with the VLM frozen for 10 epochs on four GPUs, with per-GPU batch size 1, gradient accumulation 1 and learning rate 10^{-4}. Stage 2 then trains on the approximately 135K-example QA collection for 2 epochs on four GPUs, with per-GPU batch size 1 and gradient accumulation 2 (effective batch size 8), learning rate 2×10^{-5}, 100 warm-up steps, weight decay 0.01, and image-area bounds of 3,136–150,528 pixels.

MindCube injection comparisons. All three runs start from Qwen2.5-VL-3B (Qwen Team, 2025a) and use the same 10K MindCube (Wang et al., 2026a) QA examples for 10 epochs (100,000 optimizer steps). Each uses one GPU, batch size 1 and gradient accumulation 1 (effective batch size 1), learning rate 2×10^{-5}, 100 warm-up steps, weight decay 0.01, and image-area bounds of 3,136–150,528 pixels. The Geometric Reader and visual encoder are frozen; the language model is fine-tuned. Fixed scaling additionally trains γ_m; the full design trains its adapter, routing head and slot embeddings, with adapter and routing-head learning rates of 2×10^{-5}. The adapter is a tokenwise 2048×2048 affine map with zero-initialized weights and bias. The routing head applies

LayerNorm, a $2048 \to 256$ linear map, GELU, and a $256 \to 1$ linear map before the sigmoid. Its final map is zero-initialized, giving an initial gate of 0.5 for available geometry. Evaluation covers all 1,050 MindCube-Tiny questions: direct injection, fixed scaling and question conditioning yield 825, 719 and 937 correct answers, respectively.

A.3 Evaluation Protocols and Result Sources

A.3.1 Sources of Baseline Results

The benchmark definitions are MindCube (Wang et al., 2026a), VSI-Bench (Yang et al., 2025), SPBench (Li et al., 2026c), CV-Bench (Tong et al., 2024), SPAR-Bench (Zhang et al., 2025), ViewSpatial (Li et al., 2026b), and MMSI-Bench (Yang et al., 2026). For Tab. 1, we evaluated the Qwen base models (Qwen Team, 2025a;b; 2026), DeepSeek-Flash (DeepSeek-AI, 2026) in thinking mode, MiMo-V2.6-Flash (LLM-Core Xiaomi, 2026), and all **GeoReader** rows. The other rows use the cited literature results. In Tab. 2, Qwen base-model results marked † and all **GeoReader** results are evaluated by us. For Tab. 2, the VSI-Bench, CV-Bench, SPAR-Bench, ViewSpatial and MMSI-Bench scores for SpatialLadder (Li et al., 2026c) come from Tabs. 1–2 of its ICLR 2026 paper; its combined SPBench score is taken from Tab. 2 of 3DThinker (Chen et al., 2026), since the original paper reports SPBench-SI and SPBench-MV separately. For ViLaSR (Wu et al., 2025b), the VSI-Bench, SPAR-Bench and MMSI-Bench scores come from Tab. 1 of its NeurIPS 2025 paper; its other three scores are taken from 3DThinker Tab. 2. The complete Spatial-MLLM (Wu et al., 2025a), SpaceR (Ouyang et al., 2025) and Video-R1 (Feng et al., 2025) rows, and the 3DThinker rows themselves, are taken from the same 3DThinker table. The SenseNova-SI-8B (Cai et al., 2026) row combines VSI-Bench, MMSI-Bench and ViewSpatial from its official Qwen3-VL-8B version 1.1 model card with CV-Bench, SPAR-Bench and SPBench-SI/MV from GAMSI (Zheng et al., 2026) Tab. 1. We compute SPBench as the equal-weight mean of SI/MV ($60.4/71.1$), and recompute the six-benchmark average before rounding. GAMSI does not specify the SenseNova release version. Method citations identify the original methods, whereas these attributions identify the sources of the reported scores. Literature results follow their source evaluation settings.

For Tab. 1, the SenseNova-SI results for Qwen2.5-VL-3B and 7B come from the respective official 3B and 7B model cards (version 1.1). The 3DThinker results with Qwen3-VL-2B/4B/8B use the S1+S2 rows in Tab. 1 of its CVPR 2026 supplementary material (Chen et al., 2026).

A.3.2 Benchmark Inputs and Metrics

MindCube-Tiny. We follow the MindCube-Tiny benchmark (Wang et al., 2026a). The rotation, among, and around subsets contain 200, 600, and 250 questions, respectively. For our evaluations, subtask scores and overall accuracy are computed from the same 1,050 predictions for each model; overall accuracy is weighted by the subset sizes. In Tab. 1, boldface marks the highest Overall accuracy within each Qwen backbone-and-size block. API models are grouped by evaluation interface. DeepSeek-Flash refers to DeepSeek-V4.1-Flash; Size denotes total parameters when reported by the provider, and – denotes unreported values. **GeoReader** uses DA3-predicted geometry; Sec. A.4.3 compares it with annotation-derived symbolic geometry across backbones.

Six-benchmark coverage. The Qwen3-VL-2B/4B/8B evaluations cover 5,130 VSI-Bench, 1,328 SPBench, 2,638 CV-Bench, 7,211 SPAR-Bench, 5,712 ViewSpatial and 1,000 MMSI-Bench questions. All locally evaluated Qwen families use the full VSI-Bench test set, with 32 uniformly sampled frames per video supplied as ordered images.

A.3.3 Local-Model Evaluation and Scoring

Shared settings and model differences. Qwen2.5-VL-3B/7B, Qwen3-VL-2B/4B/8B and Qwen3.5-0.8B/2B/4B/9B use native preprocessing and greedy generation, at most 1,024 new tokens, and image-area bounds of 3,136–150,528 pixels. Qwen3.5 uses non-thinking mode. On MindCube, Qwen2.5-VL uses the official raw-QA instruction requesting an option in an answer tag; Qwen3-VL and Qwen3.5 use question-only prompts.

Answer scoring. All local MindCube base models share an extractor accepting explicit option letters and standalone answers, including answers introduced within an explanation. A later explicit

answer supersedes an earlier one; option lists and ambiguous or unextractable answers are scored as incorrect. For Qwen3-VL-4B, 98 of 1,050 responses reach the token limit, including 68 without an extractable option; three further text-only choices remain unparsed. VSI scoring preserves line breaks and prioritizes explicit numerical answers over numbers within explanations.

A.3.4 API-MODEL EVALUATION

Both API models are evaluated unadapted in September 2026 on all 1,050 MindCube questions, with original images in dataset order, the official raw-QA instruction, and the extractor in Sec. A.3.3. Their 65,536-token limits include reasoning and final output. API models use their own image processing.

DeepSeek-Flash. DeepSeek-V4.1-Flash is requested as `deepseek-flash`, with high reasoning effort and `original` image detail. Temperature is ignored in thinking mode. Four responses reach the output limit; five unextractable answers are counted as incorrect. The reported 552B size denotes total MoE parameters.

MiMo-Flash. MiMo-V2.6-Flash is requested as `mimo-v2.6-flash` with deep thinking enabled; it does not expose the same image-detail control. No response reaches the limit; 26 unextractable answers are counted as incorrect. Xiaomi reports 309B total and 15B active MoE parameters (LLM-Core Xiaomi, 2026).

Reported model sizes. The 230B size for Doubao-Seed-1.6 is the total MoE parameter count reported by ByteDance Seed (ByteDance Seed, 2025); 23B parameters are active.

A.4 ADDITIONAL EXPERIMENTS

These experiments follow the order of Secs. 4.2–4.5.

A.4.1 GEOMETRY VERSUS RGB-ONLY TRAINING ON SIX BENCHMARKS

We compare **GeoReader**-3B from Tab. 2 with vanilla SFT, the same QA fine-tuning without geometric inputs. Both models use Qwen2.5-VL-3B, the same QA data and training settings, and the same evaluation protocol; **GeoReader** additionally reads geometric inputs through the frozen Reader.

Each benchmark mixes questions that depend on 3D geometry with questions that do not, so we analyze the 59 native subtasks in Tab. 12. On the 27 subtasks about metric 3D quantities, geometric inputs raise accuracy from 56.7 to 59.6, and every group improves: object and room size (+4.8), camera motion (+4.4) and distance and depth (+2.2). Individual subtasks gain much more, including +14.3 on SPBench-SI relative distance, +13.5 on MMSI camera motion, +7.8 on SPAR camera-motion inference and +7.4 on VSI-Bench object size. On the other 32 subtasks, accuracy is unchanged (67.3 versus 67.5), and neither group decreases. Geometric inputs thus improve accuracy on geometry-dependent questions without lowering it on the others.

Groups follow the quantity each question asks about. Camera motion covers questions about the camera's own motion or viewpoint change (MMSI camera motion and camera–camera position, SPAR camera-motion and view-change inference); size and distance-and-depth groups cover subtasks named by those quantities; 2D, counting and order covers 2D relations, counting and appearance order; the remaining subtasks form direction, relation and other. Group scores pool subtasks across benchmarks and weight each subtask by its number of questions, whereas benchmark totals follow each benchmark's own averaging (e.g., SPBench averages its SI and MV splits equally); the two are therefore not directly comparable.

As the image counts in Tab. 12 show, the selected subtasks cover inputs of 1, 2, 3, 8 and 32 images or sampled frames.

Table 12: Subtask-level accuracy (%) of vanilla SFT and **GeoReader**. Both models use Qwen2.5-VL-3B with the same QA data and training settings. A and B split all 59 native subtasks of the six benchmarks by whether the question asks about a metric 3D quantity (number of subtasks in parentheses), with accuracies weighted by question count. C lists every subtask in A whose gain is at least 2 points. Images: input RGB images or sampled video frames (ranges indicate mixed input sizes); Questions: number of questions. Δ is computed before rounding. Gains concentrate on metric 3D subtasks, and the other subtasks are unchanged.

Question family		Images	Questions	Vanilla	GeoReader	Δ
A. Questions about metric 3D quantities						
Object and room size (4)		1–32	1,862	65.2	70.0	+4.8
Camera motion (4)		2	967	39.3	43.7	+4.4
Distance and depth (19)		1–32	6,613	56.8	59.0	+2.2
All metric 3D questions (27)		1–32	9,442	56.7	59.6	**+2.9**
B. Other questions						
Direction, relation and other (27)		1–32	10,886	65.5	65.6	+0.1
2D, counting and order (5)		1–32	2,691	74.3	75.1	+0.8
All other questions (32)		1–32	13,577	67.3	67.5	**+0.3**
C. Metric 3D subtasks with gains of at least 2 points						
Benchmark	Subtask	Images	Questions	Vanilla	GeoReader	Δ
VSI-Bench	Object size	32	953	64.5	71.9	+7.4
	Room size	32	288	55.4	62.0	+6.7
	Absolute distance	32	834	38.4	41.8	+3.4
	Relative distance	32	710	47.9	50.6	+2.7
SPBench	Relative distance	1	91	69.2	83.5	+14.3
	Object size	8	158	69.3	72.9	+3.6
	Absolute distance	8	30	38.0	40.3	+2.3
MMSI	Camera motion	2	74	43.2	56.8	+13.5
SPAR	Object–object distance	1	363	48.1	54.2	+6.0
	Distance inference	1	340	73.2	77.6	+4.4
	Object–camera distance	1	393	64.6	68.2	+3.7
	Object–camera depth	1	360	69.6	71.9	+2.3
	Camera-motion inference	2	400	49.8	57.5	+7.8
	Distance inference	3	336	63.4	69.3	+6.0
	Object–object distance	3	297	38.3	41.0	+2.7
CV-Bench	Depth	1	600	86.5	88.8	+2.3

Table 13: Additional MindCube-Tiny results (%). Base models are evaluated by us, Video-ORA uses its cited result, bold marks the best Overall, and **GeoReader** uses DA3 geometry. **GeoReader** improves over its Qwen3.5-4B base model.

Model	Rot.	Among	Around	Overall ↑
Qwen3.5-0.8B (Qwen Team, 2026)	25.0	31.8	44.8	33.6
Qwen3.5-2B (Qwen Team, 2026)	40.5	31.8	51.2	38.1
Qwen3.5-4B (Qwen Team, 2026)	47.5	43.2	59.2	47.8
Video-ORA-4B (Li et al., 2026e)	37.0	49.8	54.8	48.6
GeoReader (Qwen3.5-4B)	35.5	93.5	92.4	**82.2**

A.4.2 Additional Backbone Results

We test whether the recipe transfers to smaller backbones. Tab. 13 reports additional MindCube-Tiny results, including the Qwen3.5-4B comparison. Base-model protocols are given in Sec. A.3; Video-ORA (Li et al., 2026e) uses its cited result.

Table 14: MindCube accuracy (%) with DA3-predicted geometry across recipes; Tab. 5 gives the symbolic-geometry results. Pair-supervised **GeoReader** remains the best recipe.

Training and inference recipe	DA3
Joint description + QA SFT	64.7
Generated Text-CoT	66.7
Descriptions in prompt	79.2
GeoReader (group)	68.9
GeoReader (pair)	**81.1**

Table 15: MindCube accuracy (%) of pair-supervised **GeoReader** with symbolic or DA3-predicted geometry. Bold marks the better source, which differs by backbone.

Backbone	Symbolic	DA3
Qwen2.5-VL-3B	**89.2**	81.1
Qwen2.5-VL-7B	85.6	**93.2**
Qwen3-VL-2B	77.0	**86.5**
Qwen3-VL-4B	92.3	**94.3**
Qwen3-VL-8B	92.5	**93.4**
Qwen3.5-4B	**83.4**	82.2

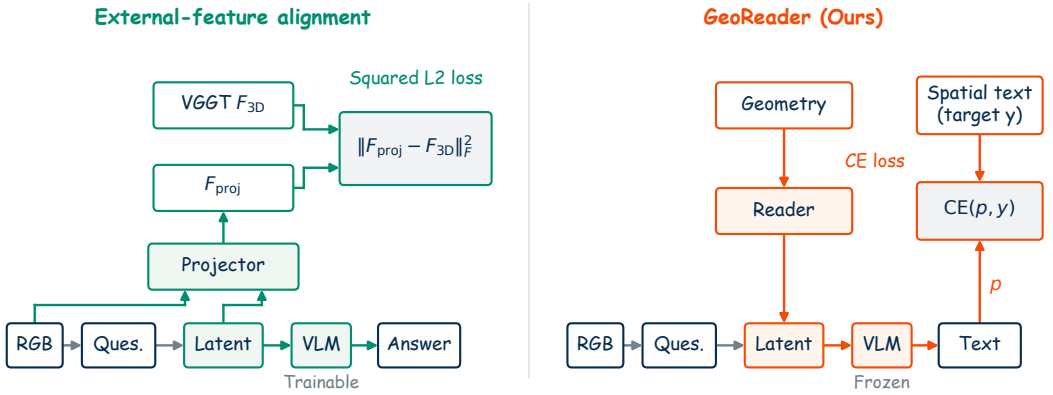


Figure 8: Geometry supervision in feature alignment and in **GeoReader**. 3DThinker (Chen et al., 2026) reconstructs VGGT features with a trainable VLM; **GeoReader** trains the Reader through a frozen VLM using geometry-description cross-entropy. Only geometry-supervision paths are shown; 3DThinker's additional text loss is omitted. Ques. denotes the text prompt supplied to the VLM.

A.4.3 Predicted versus Symbolic Geometry

This comparison measures how geometry-estimation errors affect the full recipe. Using the sources defined in Sec. A.1.3, each configuration applies the same source to description targets, Reader training, QA training and evaluation. With Qwen2.5-VL-3B and pair supervision, DA3-predicted geometry reaches 81.1%, versus 89.2% with symbolic geometry and 68.6% for vanilla SFT (Tab. 14). Sec. A.4.8 describes how inference cost is measured.

Across the six backbones in Tab. 15, DA3 exceeds symbolic geometry on four (by 0.9–9.5 points) and trails it on two (by 1.2–8.1 points). The effect of geometry source therefore varies by backbone under the shared interface-training recipe in Sec. 4.2.

A.4.4 Reader Pretraining

Conditions fixed in the MindCube comparisons. For Tab. 6, the Stage 1-only reference uses a frozen VLM and injection scale 0.25. Vanilla SFT uses the same 10K QA examples; the full recipe uses pair-supervised symbolic geometry.

For Tab. 4, Panel A, the feature-alignment row is the released first-stage 3DThinker checkpoint (Qwen2.5-VL-3B), trained by its authors. Its 62.7% is computed from the released per-question answers on the same 1,050 questions; Sec. A.4.7 uses the same checkpoint for the latent interventions.

Comparison with feature alignment. Fig. 8 contrasts the geometry-supervision interfaces of 3DThinker (Chen et al., 2026) and **GeoReader**. In 3DThinker, RGB features and VLM-generated latents jointly enter a projector supervised by VGGT (Wang et al., 2025a) features; VGGT features

are training targets, not additional inference inputs. **GeoReader** instead learns geometric inputs using geometry-description supervision through the frozen target VLM, then freezes the Reader for QA. The figure contrasts these learning targets and input pathways.

A.4.5 Pair versus Group Supervision and Text Geometry

Tab. 5 uses the same Qwen2.5-VL-3B/10K-QA MindCube setting as Sec. A.4.6, with symbolic geometry; Tab. 14 compares symbolic and DA3-estimated geometry. Joint SFT adds 2,072 whole-group description examples per epoch. Group and pair supervision use 2,072 and 3,375 Stage 1 examples per epoch, respectively. Reference–target pairs are constructed as described in Sec. A.1.1. Generated CoT receives images and the original question at inference; prompt and continuous-input methods additionally receive the corresponding geometry. Direct-answer methods use a generation limit of 128 tokens, while generated CoT uses 512 tokens for the intermediate text and answer.

A.4.6 Alternative Interfaces

Shared setting. During Stage 2, the Panel B alternatives use Qwen2.5-VL-3B, the same 10K MindCube QA examples and answer-only supervision, one GPU with batch size 1 and accumulation 1, ten epochs, learning rate 2×10^{-5}, and the same image-area bounds. We evaluate all rows on the same 1,050 held-out questions. The two adapted fusion rows use pre-extracted final-layer VGGT (Wang et al., 2025a) features from the input RGB images: one camera token and 1,369 patch tokens per view, each 2,048-dimensional; the four register tokens are excluded. The VGGT and Qwen visual encoders are frozen, while the full language model and fusion module are trained with QA answer loss. Both fusion modules use per-frame cross-attention with a 512-dimensional attention space and eight heads. They modify the Qwen image embeddings before one LLM pass, without adding geometric tokens or a Reader.

VLM-3R-style adaptation. Following the spatial–visual fusion idea of VLM-3R (Fan et al., 2026), each frame's Qwen image embeddings query that frame's VGGT camera and patch features. A learned output projection maps the attended features back to the Qwen hidden dimension, and its zero-initialized residual is added to the corresponding image embeddings. We adopt only the fusion mechanism; the original VLM-3R system uses CUT3R (Wang et al., 2025b) features.

SpaceMind-style adaptation. Inspired by the camera-guided fusion of SpaceMind (Zhao et al., 2026), we retain the same per-frame visual queries and VGGT features but use the camera feature to adjust patch keys and values, weight patch values by learned importance, and apply a camera-conditioned channel-wise gate to the fused residual before adding it to the Qwen image embeddings. The residual output is zero-initialized. We use a simplified CGMF in Qwen2.5-VL.

Per-token additive fusion. This row uses the same symbolic pose input as **GeoReader**: the 20-dimensional descriptor of each image token passes through a zero-initialized linear map to the Qwen hidden dimension and is added to that image embedding. It adds no geometric tokens and uses one LLM pass, with the same QA data and schedule as the other rows.

GeoReader instead encodes the same descriptors into per-view geometric tokens with its Reader.

A.4.7 Geometry Interventions

Tab. 8 uses the pair-supervised MindCube checkpoint from Tab. 5, with identical weights, questions, images, and decoding settings across conditions. Correct geometry yields $937/1,050$ correct answers. Zero geometry restores each geometric token to its slot embedding after the Reader, adapter, and gate have been evaluated; it retains the two-pass computation and yields $811/1,050$ correct answers. Frame shuffling applies a within-example permutation with no fixed points (seed 42) to the raw extrinsics and intrinsics, leaving the RGB order unchanged. The original feature builder then regenerates frame identifiers and the reference frame from the newly ordered poses. This condition yields $663/1,050$ correct answers. Thus the intervention tests pose-to-view mismatch under the model's normal feature-construction procedure, including its first-view reference convention. All conditions contain 1,050 unique questions with no answer-parsing failures. For 3DThinker, we use the released MindCube checkpoint described in Sec. A.5.1. The original condition uses the released per-question answers on all 1,050 questions (658 correct, 62.7%). In the zeroed condition, we replace the input embedding at every latent-generation step with a zero vector while keeping the

released visible prefix, and then decode greedily (660 correct, 62.9%). In the shuffled condition, we replace its latents with those generated for another question, using a fixed one-to-one assignment in which the two questions come from different scenes and have different correct options (646 correct, 61.5%). Questions without a valid answer count as incorrect in both conditions. This checkpoint is the released model after the first training stage; the 75.2% in Tab. 1 is the published result after both stages, whose checkpoint is not released.

A.4.8 Inference-Cost Measurement

We measure all six configurations on the same NVIDIA H200 with batch size 1. After five excluded warm-up examples per model, we average generated-token counts and latency over all 1,050 test questions. Generated-token counts include output formatting, control and end-of-sequence tokens, but exclude input tokens. Timing includes image loading, input processing, all VLM passes, generation and decoding. Model loading and offline geometry construction are excluded.

For the continuous-input configurations, the read pass evaluates only the image–question prefix and retains its key–value (KV) cache. The answer pass reuses this cache to process the geometry-conditioned suffix and then decode the answer. Because geometry follows the question in the causal sequence, it does not change the preceding prefix states. Thus, inference does not recompute the full prompt twice; both stages of this cached computation are included in the reported latency. The SFT baselines use the standard generation routine, while the continuous-input configurations use a custom greedy decoding loop.

A.5 Qualitative Analysis

A.5.1 Answer-Token Attention

We show one example from each MindCube category: Among, Around, and Rotation (Fig. 9). For each category, we use the first image-matched entry in our pre-existing evaluation list, selected before extracting attention and without filtering by correctness or attention magnitude. The Among example is also used in Fig. 3.

We use the pair-supervised Camera-only GeoReader checkpoint and the released `jankin123/3DThinker-Mindcube` checkpoint of 3DThinker (Chen et al., 2026) (training step 74,000; source revision `c9469e01b719`), the model released after its first training stage. The latter serves as our representative of feature alignment, and we retain its recurrent latent-generation procedure. Both backbones have 36 language-model layers, 16 query heads, and 2,048-dimensional hidden states. GeoReader first reads the RGB and question prefix, then injects one geometric token per input view after the question before answer generation. 3DThinker processes RGB and question tokens followed by generated latents, intermediate text, and the answer. We retain each model's native preprocessing and generation procedure.

We record eager causal attention at the positions predicting the final answer body, excluding answer tags, and average uniformly over query heads and these prediction positions within each layer. Key tokens are partitioned into Geometry, RGB, Question, CoT, and Other. Other contains instruction and chat templates, boundaries, and preceding answer tokens. Within each layer, attention shares are normalized over Geometry, RGB, Question, and CoT, excluding Other. CoT remains in the denominator for 3DThinker, although its panel is omitted; GeoReader has no CoT body. All 36 layers are retained without smoothing. GeoReader's geometry mass refers to complete geometric tokens, including their slot embeddings and geometric residuals.

Both models answer all three examples correctly. In each example, GeoReader's answer positions place a higher layer-averaged attention share on geometry than those of 3DThinker.

A.5.2 Additional Qualitative Results

MindCube examples in Figs. 10 and 11 use the DA3-pair configuration from Sec. A.4.3. The two MMSI-Bench (Yang et al., 2026) examples in Fig. 4 use the Camera+Depth checkpoint at epoch 1, step 6,335. Questions and predictions come from saved evaluation records.

For Fig. 4, manually marked visible object regions are backprojected using predicted depth and intrinsics. Surface connections crossing depth discontinuities are omitted. Orange outlines mark



Figure 9: Answer-token attention across layers on three MindCube categories. Both models answer every question correctly; Sec. A.5.1 defines the statistics. In all three cases, answer tokens attend more to geometry in GeoReader than to latents under feature alignment.

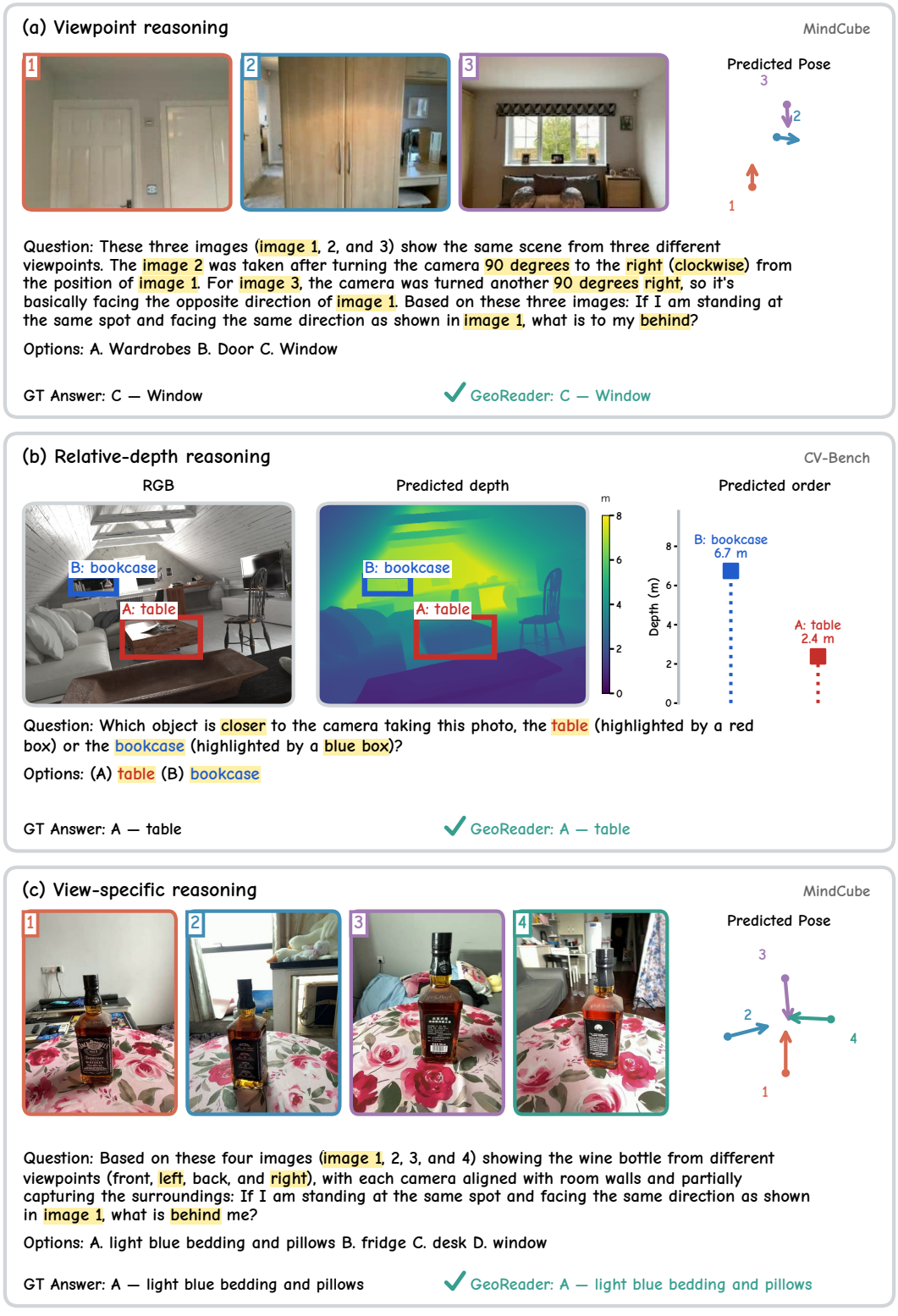


Figure 10: Selected correct responses with DA3-predicted geometry. Camera arrows show poses; depth plots show box-region medians at schematic horizontal positions. Each case includes the question, options, reference answer, and GeoReader response (Sec. A.5.2).

these regions; the bathroom arrow shows the facing direction imagined in the question. IMG, Ques. and GEO denote image, question and per-view geometry groups. GEO blocks are schematic; the checkpoint uses four pose and four depth tokens.

A.5.3 FAILURE CASES

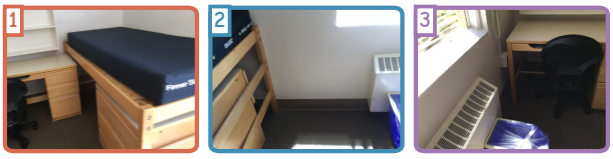
Fig. 11 illustrates three failure patterns.

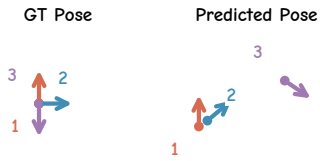
(1) Camera-pose estimation errors. Panel (a) pairs an incorrect answer with a predicted view-3 yaw of $124°$, versus the annotated $180°$. The model receives the predicted pose; GT is shown for reference.

(2) Depth estimation errors. In panel (b), median predicted depths over the central 40% of each box's width and height reverse the benchmark order: car 21.1 m, trailer 25.1 m. The model chooses the car. GT specifies ordinal depth, and horizontal plot positions are schematic.

(3) Reference-frame confusion. In panel (c), the answer matches view 1's reference answer rather than the requested view 2 (cf. Fig. 10c). The predicted relative yaw of view 2 is $76°$, so the input geometry does distinguish the two views.

(a) Camera-pose error — MindCube





View 3 yaw: 180° (GT) vs. 124° (predicted).

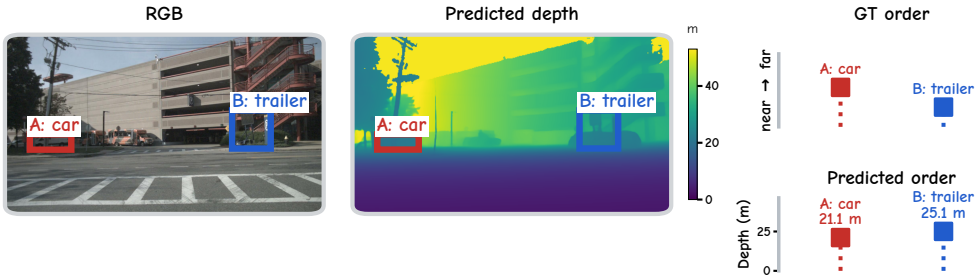
Question: These three images (image 1, 2, and 3) show the same scene from three different viewpoints. The image 2 was taken after turning the camera 90 degrees to the right (clockwise) from the position of image 1. For image 3, the camera was turned another 90 degrees right, so it's basically facing the opposite direction of image 1. Based on these three images: If I am standing at the same spot and facing the same direction as shown in image 1, what is to my right?

Options: A. Study table and black chair B. Bed C. Window

GT Answer: C — Window

✗ GeoReader: A — Study table and black chair

(b) Depth-order error — CV-Bench



Question: Which object is closer to the camera taking this photo, the car (highlighted by a red box) or the trailer (highlighted by a blue box)?

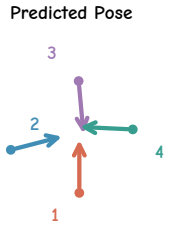
Options: (A) car (B) trailer

GT Answer: B — trailer

✗ GeoReader: A — car

(c) Reference-frame confusion — MindCube





GT from view 2: fridge. Wrong answer matches view 1: bedding and pillows.

Question: Based on these four images (image 1, 2, 3, and 4) showing the wine bottle from different viewpoints (front, left, back, and right), with each camera aligned with room walls and partially capturing the surroundings: If I am standing at the same spot and facing the same direction as shown in image 2, what is behind me?

Options: A. light blue bedding and pillows B. fridge C. desk D. window

GT Answer: B — fridge

✗ GeoReader: A — light blue bedding and pillows

Figure 11: Selected failure cases: (a) an incorrect response alongside a pose error; (b) reversed predicted depth order for the sampled car/trailer regions; and (c) a response matching an alternative reference view. GT pose is annotation-derived; GT order is ordinal. Horizontal depth-plot positions are schematic.

A.6 Public Resources

We acknowledge the public data, models, and implementations used in this work. The dataset list covers all 14 RGB sources in our approximately 135K-example training collection, together with MindCube. QA construction and the remaining evaluation benchmarks are described in Secs. A.1.2 and A.3.2.

A.6.1 Public Datasets

- ScanNet (Dai et al., 2017)[1] ..... ScanNet Terms of Use
- ScanNet++ (Yeshwanth et al., 2023)[2] ..... ScanNet++ Terms of Use
- COCO images (Lin et al., 2014)[3] ..... Image-specific licenses
- ARKitScenes (Baruch et al., 2021)[4] ..... Apple dataset license
- Matterport3D (Chang et al., 2017)[5] ..... Matterport3D Terms of Use
- nuScenes (Caesar et al., 2020)[6] ..... nuScenes Terms of Use
- Spatial-MM (Shiri et al., 2024)[7] ..... Not specified
- AgiBot World (Team AgiBot-World, 2025)[8] ..... CC BY-NC-SA 4.0
- nuImages (Caesar et al., 2020)[9] ..... nuImages Terms of Use
- EgoDex (Hoque et al., 2026)[10] ..... CC BY-NC-ND
- EPIC-Contact (Bansal et al., 2026)[11] ..... CC BY-NC 4.0
- DTU MVS (Aanæs et al., 2016)[12] ..... Not specified
- DAVIS images (Pont-Tuset et al., 2017)[13] ..... Source-video terms
- SUN RGB-D (Song et al., 2015)[14] ..... Constituent-dataset terms
- MindCube (Wang et al., 2026a)[15] ..... MIT License

A.6.2 Public Implementations and Models

Our code builds on 3DThinker. VLM-3R and SpaceMind inform our adapted fusion controls (Sec. A.4.6). Code and checkpoint licenses are listed separately where they differ.

- 3DThinker (Chen et al., 2026)[16] ..... Apache License 2.0
- VLM-3R (Fan et al., 2026)[17] ..... Apache License 2.0
- SpaceMind (Zhao et al., 2026)[18] ..... Apache License 2.0
- MindCube evaluation code (Wang et al., 2026a)[19] ..... MIT License
- Depth Anything 3 (Lin et al., 2026)[20] ..... Apache License 2.0

[1] https://github.com/ScanNet/ScanNet
[2] https://scannetpp.mlsg.cit.tum.de/scannetpp/
[3] https://cocodataset.org/#termsofuse
[4] https://github.com/apple/ARKitScenes
[5] https://github.com/niessner/Matterport
[6] https://www.nuscenes.org/terms-of-use
[7] https://huggingface.co/datasets/advaitgupta/spatial-mm-obj
[8] https://github.com/OpenDriveLab/AgiBot-World
[9] https://www.nuscenes.org/terms-of-use
[10] https://github.com/apple-aiml-research/ml-egodex
[11] https://huggingface.co/datasets/Sid2697/epic-contact
[12] https://roboimagedata.compute.dtu.dk/?page_id=36
[13] https://davischallenge.org/
[14] https://rgbd.cs.princeton.edu/
[15] https://huggingface.co/datasets/MLL-Lab/MindCube
[16] https://github.com/zhangquanchen/3DThinker
[17] https://github.com/VITA-Group/VLM-3R
[18] https://github.com/RealMikeDuke/SpaceMind
[19] https://github.com/mll-lab-nu/MindCube
[20] https://github.com/ByteDance-Seed/Depth-Anything-3

- Depth Anything 3 Nested (Lin et al., 2026)[21] . . . . . CC BY-NC 4.0
- VGGT (Wang et al., 2025a)[22] . . . . . VGGT License
- Qwen2.5-VL-3B-Instruct (Qwen Team, 2025a)[23] . . . . . Qwen Research License
- Qwen3-VL-4B-Instruct (Qwen Team, 2025b)[24] . . . . . Apache License 2.0
- Qwen3.5-4B (Qwen Team, 2026)[25] . . . . . Apache License 2.0

[21] https://huggingface.co/depth-anything/DA3NESTED-GIANT-LARGE-1.1
[22] https://github.com/facebookresearch/vggt
[23] https://huggingface.co/Qwen/Qwen2.5-VL-3B-Instruct
[24] https://huggingface.co/Qwen/Qwen3-VL-4B-Instruct
[25] https://huggingface.co/Qwen/Qwen3.5-4B