SCOPE: LEARNING SPHERICAL COORDINATE GROUNDING FOR SPATIAL UNDERSTANDING IN MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatial understanding is pivotal for machines to advance toward physical world intelligence, yet current MLLMs still struggle to estimate object distances, judge relative directions, and reason about 3D layout. One line of work adds 3D encoders, which changes the architecture and weakens general multimodal ability. The other fine-tunes on spatial reasoning data, where supervision targets final answers rather than the 3D locations of relevant objects. Models may then rely on language priors without learning where objects are in 3D space. We therefore seek to train MLLMs to localize objects in 3D, which raises a question: which location representation best supports MLLMs' spatial reasoning? Under a protocol that varies only this choice across three MLLMs, we compare four representations common in 3D vision and find spherical coordinates the most effective across all three models. In spherical coordinates, distance and direction occupy separate axes: an observer's yaw rotation changes only azimuth, while moving directly toward an object changes only distance. To train MLLMs with spherical coordinate grounding, we develop an automatic pipeline that generates grounding samples from annotated 3D scans, yielding SPHEREQA, a large dataset of 300K samples in egocentric and allocentric reference frames. We then introduce SCOPE, an MLLM trained on SPHEREQA to predict spherical coordinates through per-axis classification with a spherical coordinate tokenizer instead of raw-text generation. Its two variants improve over their respective base MLLMs by 10.2/11.1 points on average across four spatial benchmarks, surpass all evaluated open-source and specialist baselines on average, and preserve general multimodal ability.

1 INTRODUCTION

Spatial understanding is essential for machines to perceive and interact with the physical world, with applications in embodied AI (Zhou et al., 2025; Liu et al., 2025), autonomous driving (Cui et al., 2024), and wearable devices (Chen et al., 2025a). Despite substantial progress in visual recognition, captioning, and visual question answering (Song et al., 2025b; Bai et al., 2025a), multimodal large language models (MLLMs) still struggle with spatial reasoning, such as estimating object distances, judging relative directions, and reasoning about 3D scene layout (Yang et al., 2025b; 2026c).

Previous efforts to improve MLLM spatial awareness follow two main directions. The first augments the model with extra spatial encoders (Wu et al., 2025a; Hu et al., 2026) or depth and BEV inputs (Cheng et al., 2024; Cai et al., 2025; Zhu et al., 2025), which adds architectural complexity and weakens general multimodal ability. The second keeps the architecture unchanged and supervises spatial competence through large-scale fine-tuning (Chen et al., 2024; Ouyang et al., 2025; Wang et al., 2026) or reinforcement learning (Wu et al., 2025b; Liu et al., 2026; Zhan et al., 2025) on spatial reasoning data. However, this supervision targets final spatial answers rather than the precise 3D location of each object on which they rest. This may lead models to rely on language priors instead of genuine visual evidence (Asadi et al., 2026). Recent work begins to fill this gap by supervising per-object localization with 2D boxes (Li et al., 2026b) or 3D boxes (Yang et al., 2026a). These choices expose different geometric variables to MLLMs, yet prior work rarely isolates the effect of the representation itself, leaving open *which representation best supports MLLMs' spatial reasoning*.

To address this question, we conduct a study (§3) that holds the image and query fixed while varying only the grounding representation provided to the MLLM, comparing depth, 3D boxes, Cartesian

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.

coordinates, and spherical coordinates from the same 3D estimate. Across three MLLMs, spherical coordinates consistently yield the largest gains. Although the spherical and Cartesian anchors describe the same centroid up to an invertible transform, spherical coordinates state the queried distance and direction as separate coordinates, r and (θ, φ), whereas from (x, y, z) both must be computed through $\sqrt{x^2 + y^2 + z^2}$ and $\mathrm{atan2}(x, z)$. Moreover, rotating the observer by α gives $\theta' = \theta - \alpha$ while leaving r and φ unchanged. Spatial reasoning thereby becomes separable for this case, with each question and each such reorientation acting on one axis at a time, which also underlies the use of spherical coordinates in point-cloud detection (Lai et al., 2023) and segmentation (Shin et al., 2024).

Motivated by this observation, we seek to improve MLLM spatial reasoning by training models to ground objects in spherical coordinates. This poses two challenges: existing datasets lack training data for object-level spherical coordinate grounding, and standard LLM tokenizers fragment numerical coordinates into digit tokens, so supervision applies to digits rather than to the location they denote (Zausinger et al., 2025). We address them as follows. **First**, we develop an automatic data curation pipeline that generates spherical coordinate grounding data from existing 3D scan datasets. The pipeline selects visual observations based on object visibility and transforms 3D object locations into spherical coordinates in egocentric and allocentric reference frames, yielding SPHEREQA, a dataset of 300K grounding QA pairs spanning single-image, multi-image, and video inputs. **Second**, we introduce SCOPE (**S**pherical **C**oordinate gr**O**unding for s**P**atial und**E**rstanding), an MLLM trained with SPHEREQA and equipped with a spherical coordinate tokenizer. The tokenizer divides each coordinate axis into bins and assigns a token to each bin, allowing SCOPE to predict coordinates through per-axis classification. We initialize the new token embeddings from textual descriptions of their axes and bin centers while leaving the backbone architecture unchanged.

Contributions. *First*, we provide the first controlled comparison of four geometric representations for object grounding in MLLMs and show that spherical coordinates consistently yield the largest gains. *Second*, we develop an automated pipeline to construct SPHEREQA, a large-scale dataset of 300K spherical coordinate grounding samples spanning egocentric and allocentric reference frames and multiple visual input formats. *Third*, we propose SCOPE, which uses a spherical coordinate tokenizer to cast object-level coordinate prediction as per-axis classification to improve spatial reasoning. SCOPE improves its base models by 10.2/11.1 points on average across four spatial benchmarks and achieves the highest average score among all evaluated open-source MLLMs, including spatial specialists, while preserving general multimodal capabilities.

2 RELATED WORK

Spatial Understanding in MLLMs. Prior efforts to improve spatial understanding in MLLMs follow two routes. The first route enhances *3D awareness* with auxiliary geometric inputs, either at training time through RGB-D inputs (Cheng et al., 2024; Cai et al., 2025), depth-encoded features (Chen et al., 2025b; 2026), 3D-aware encoders (Wu et al., 2025a; Hu et al., 2026), and camera-parameter conditioning (Zhang et al., 2026a;c), or at inference time through verbalized geometric prompts (Ma et al., 2024; Li et al., 2025; Cheng et al., 2026), BEV images (Zhu et al., 2025), cognitive maps (Yang et al., 2025b; Huang et al., 2025), and synthesized novel views (Yang et al., 2025c). While effective, this route either modifies the MLLM architecture in ways that can weaken general multimodal ability, or relies on extra 3D inputs unavailable in typical deployments.

The second route keeps the standard MLLM architecture unchanged and supervises spatial competence through targeted instruction tuning, with supervision targets covering relative distance and direction (Chen et al., 2024; Ma et al., 2025b), spatial layout reasoning (Zhang et al., 2026b), region-grounded spatial QA (Cheng et al., 2024; Yuan et al., 2025; Daxberger et al., 2025), embodied spatial reasoning (Song et al., 2025a; Zhou et al., 2025; Gholami et al., 2026), and multi-view (Wang et al., 2026) or video-based spatial relations (Yang et al., 2026b; Ouyang et al., 2025), recently strengthened by reinforcement learning rewards (Liu et al., 2026; Zhan et al., 2025). Complementarily, Cambrian-1 studies vision-centric model and instruction-data design (Tong et al., 2024), while Cambrian-S scales spatially focused video supervision (Yang et al., 2026b). However, most methods supervise only final spatial answers, leaving the per-object 3D grounding underlying such reasoning weakly supervised. The few methods that explicitly supervise this grounding, including SpatialLadder (Li et al., 2026b) and VST (Yang et al., 2026a), adopt 2D and 3D boxes, respectively, as fixed grounding representations. This leaves the effect of representation choice on downstream spatial reasoning underexplored.

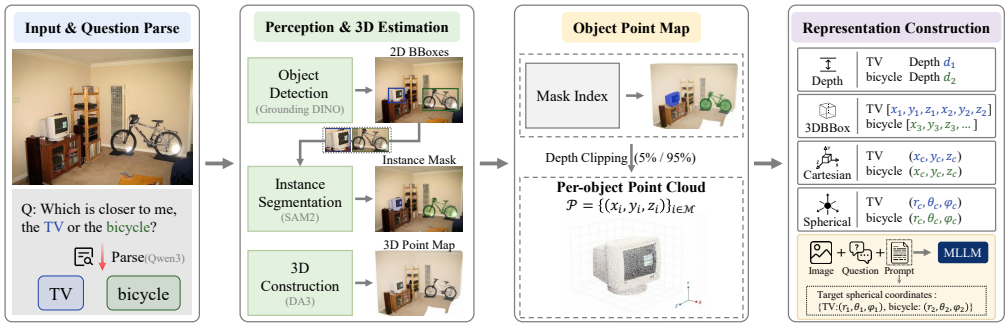


Figure 1: **Grounding representation generation.** For each entity in the question, we refine its detected box into an instance mask and use the mask to index a dense 3D point map, yielding the per-object point cloud $\mathcal{P}$. We derive four representations from $\mathcal{P}$ and use each to build a prompt for the MLLM. See Sec. 3 for details.

We therefore compare four common geometric representations used in 3D vision (Guo et al., 2021) and find that spherical coordinates yield the largest gains in MLLM spatial reasoning. Based on this finding, we train MLLMs with spherical coordinate grounding as auxiliary supervision.

Spherical Coordinate Representations. Spherical coordinates represent a 3D position by radial distance, azimuth, and elevation, separating range from horizontal and vertical direction. This viewpoint-centered parameterization appears across spatial understanding tasks, including depth estimation (Shen et al., 2022; Li et al., 2022; Zhang et al., 2025a), layout prediction (Sun et al., 2021), and panoramic scene understanding (Cohen et al., 2018; Coors et al., 2018) in the image domain, as well as 3D object detection (Fan et al., 2021), scene segmentation (Lai et al., 2023), and instance segmentation (Shin et al., 2024) in the point-cloud domain. The representation depends only on a chosen origin and forward axis, allowing the same parameterization to express different reference frames. The recent SoPE (Yip et al., 2026) exploits this structure in MLLMs by injecting spherical positional embeddings, but relies on point-cloud input. We instead study spherical coordinates as object-level supervision for spatial reasoning, without requiring 3D sensor input at inference.

3 Which Representation Best Supports Spatial Reasoning?

Study Design. We design a controlled prompting study to examine how object-level geometric representations affect MLLM spatial reasoning. For each sample, we hold the visual input and question fixed and vary only the prompted grounding representation. We compare depth, 3D boxes, Cartesian coordinates, and spherical coordinates, all derived from the same visual input using a shared estimation pipeline (Fig. 1). This design isolates the effect of representation choice.

Representation Generation. We first use Qwen3-4B (Yang et al., 2025a) to parse the target entities from the question. For each entity, Grounding DINO (Liu et al., 2024) predicts a 2D bounding box, which SAM2 (Ravi et al., 2025) refines into an instance mask $\mathcal{M}$. In parallel, DA3 (Lin et al., 2026) predicts a per-pixel 3D point map in the camera frame. We obtain the object point cloud $\mathcal{P} = \{(x_i, y_i, z_i) \mid i \in \mathcal{M}\}$ by indexing the point map with $\mathcal{M}$, then apply depth clipping to discard points in the lowest and highest 5% of z values. From $\mathcal{P}$, we construct four representations: *depth* (z-depth), given by the median camera-space z coordinate along the optical axis; the *3D box*, defined as the axis-aligned bounding box $(x_{\min}, y_{\min}, z_{\min}, x_{\max}, y_{\max}, z_{\max})$; the *Cartesian* anchor (x_c, y_c, z_c), defined as the point-cloud centroid; and the *spherical* anchor, obtained by transforming the centroid as

$$r = \sqrt{x_c^2 + y_c^2 + z_c^2}, \quad \theta = \operatorname{atan2}(x_c, z_c), \quad \varphi = \operatorname{atan2}\left(-y_c, \sqrt{x_c^2 + z_c^2}\right), \tag{1}$$

under the OpenCV camera convention ($+x$ right, $+y$ down, $+z$ along the optical axis), where r is distance, θ the azimuth (positive rightward), and φ the elevation (positive upward).

Evaluation Protocol. We evaluate on MindCube-Tiny (Wang et al., 2026), which tests spatial reasoning from partial visual observations. This setting allows us to compare how different entity-level geometric representations support MLLMs' spatial reasoning. For each sample, we provide the MLLM with the question, the images, and a key-value prompt that maps each target entity to one of the four representations. In multi-image queries, we attach a separate prompt to each image. The

representation is the only factor varied across conditions. We also include an *image-only* baseline with no representation prompt to quantify each representation's gain. To avoid conclusions based on a single backbone, we evaluate Qwen3-VL-32B-Instruct (Bai et al., 2025a), GPT-5.4-mini (Singh et al., 2025), and InternVL3.5-14B (Wang et al., 2025), with temperature set to zero for all models.

Findings. Fig. 2 shows that all four representations improve over the image-only baseline, yet the gain does not grow with how complete the 3D description is: a single depth value is as effective as a 3D box. Cartesian and spherical coordinates isolate expression from content: both name the same centroid in the same camera frame and are related by an invertible transform, yet spherical coordinates reach the highest accuracy on every model. We attribute the gap to two properties of the spherical parameterization. First, distance and direction appear as separate coordinates r and (θ, φ), whereas recovering them from Cartesian coordinates (x, y, z) requires $\sqrt{x^2+y^2+z^2}$ and $\mathrm{atan2}(x, z)$. Second, a yaw of the observer by α about a fixed origin is a shift of one coordinate, $\theta' = \theta - \alpha$, leaving r and φ unchanged, while two of the three Cartesian components change even though the object has not moved. Spherical coordinates are thus the representation that best supports MLLM spatial reasoning among the four.

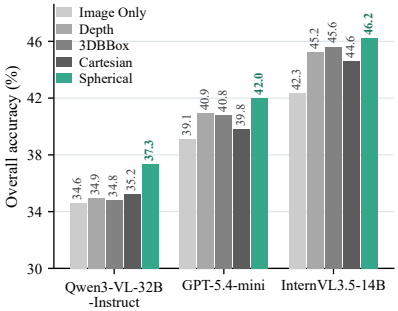


Figure 2: **Comparing grounding representations.** Spherical coordinates yield the highest accuracy on all three MLLMs.

4 Learning Spherical Coordinate Grounding

To internalize this grounding in MLLMs through training and thereby improve spatial reasoning, we first construct SphereQA with an automatic pipeline that derives spherical-coordinate grounding QA pairs from annotated 3D scans (§4.1). We then present Scope, an MLLM trained with this data and equipped with a spherical coordinate tokenizer for per-axis classification (§4.2).

4.1 SphereQA Dataset Construction

The pipeline proceeds in four stages: keyframe sampling, visibility filtering, spherical coordinate labeling, and instruction synthesis, producing both egocentric and allocentric supervision (Fig. 3).

Pose-based Keyframe Sampling. Consecutive frames of a scan often observe the scene from nearly the same pose, so uniform sampling yields many near-duplicate views. To reduce this redundancy, we retain a frame only when the camera has moved enough since the last retained frame. For a candidate with camera-to-world pose $(\mathbf{R}_2, \mathbf{t}_2)$ and the last retained pose $(\mathbf{R}_1, \mathbf{t}_1)$,

$$\Delta_{\text{trans}} = \|\mathbf{t}_2 - \mathbf{t}_1\|_2, \qquad \Delta_{\text{rot}} = \arccos\left(\frac{\mathrm{tr}\left(\mathbf{R}_2\mathbf{R}_1^\top\right) - 1}{2}\right), \tag{2}$$

where tr is the matrix trace. We retain the candidate when $\Delta_{\text{trans}} > 0.5\,\text{m}$ or $\Delta_{\text{rot}} > 30^\circ$. Both quantities are measured from the last retained frame, so they grow with camera motion and even slow motion eventually crosses a threshold. The retained frames form the set $\mathcal{I}$, each frame $i \in \mathcal{I}$ carrying a camera-to-world pose $(\mathbf{R}_i, \mathbf{t}_i)$ and intrinsics $(f_{x,i}, f_{y,i}, c_{x,i}, c_{y,i})$.

Visibility Filtering. We next determine which objects each frame in $\mathcal{I}$ actually shows, since an occluded or off-frame object would otherwise be turned into an unanswerable QA pair. An object j carries a semantic label, an annotated 3D box center $\mathbf{c}_j$, and the set $\mathcal{P}_j$ of annotated mesh vertices of its instance, read from the segmentation annotation. For each frame $i \in \mathcal{I}$ we rasterize the full scene mesh into a depth map D_i storing the camera-space z-depth of the nearest surface at each pixel, so that occluders belonging to other objects and to the room shell are both represented. A vertex $\mathbf{p} \in \mathcal{P}_j$ has camera-frame coordinates $\mathbf{p}_i = (x_i, y_i, z_i)$ and pixel position $\pi_i(\mathbf{p}_i)$ in frame i, given by

$$\mathbf{p}_i = \mathbf{R}_i^\top(\mathbf{p} - \mathbf{t}_i), \qquad \pi_i(\mathbf{p}_i) = \left(f_{x,i}\frac{x_i}{z_i} + c_{x,i},\;\; f_{y,i}\frac{y_i}{z_i} + c_{y,i}\right). \tag{3}$$

A vertex is visible when three conditions hold: it lies in front of the camera, $z_i > 0$; its rounded projection $(u, v) = \pi_i(\mathbf{p}_i)$ falls inside the $W_i \times H_i$ image, $(u, v) \in [0, W_i) \times [0, H_i)$; and it is not

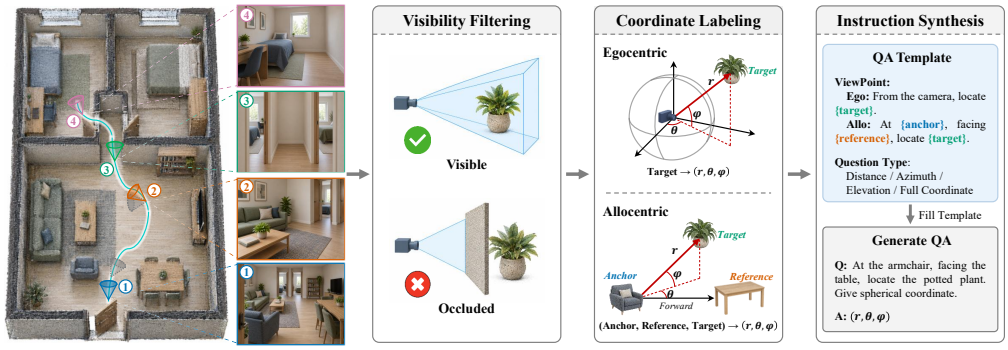


Figure 3: **SPHEREQA construction pipeline.** We first sample keyframes by camera pose along a scan trajectory, then apply *Visibility Filtering* to keep an object only where it is unoccluded. *Coordinate Labeling* expresses each target center as (r, θ, φ) from the camera (egocentric) or an anchor facing a reference (allocentric), and *Instruction Synthesis* fills a QA template under one viewpoint and one of four question types.

occluded, $z_i \leq D_i(u, v) + \epsilon$, where $\epsilon = 0.05$ m tolerates rasterization and registration error. Let $\mathcal{P}_{ij}^{\text{vis}} \subseteq \mathcal{P}_j$ denote the visible vertices of object j in frame i. The eligible frames of that object are

$$V_j = \left\{ i \in \mathcal{I} \;:\; |\mathcal{P}_{ij}^{\text{vis}}| \,/\, |\mathcal{P}_j| \geq \rho_{\min} \right\}, \tag{4}$$

where $\rho_{\min} = 0.3$ keeps enough of an object to identify it.

Spherical Coordinate Labeling. From the eligible frames we construct the labels that supervise grounding, in an egocentric and an allocentric reference frame. A label is a spherical coordinate triplet $\mathbf{s} = (r, \theta, \varphi)$, a distance in meters and two angles in degrees, stating where a target object lies for one observer. The two frames share a single construction and differ only in the observer, which we describe by its world-space position $\mathbf{o}$ and an orientation $\mathbf{R}$ whose columns are its right, down, and forward axes, following the camera convention of Eq. 1. Transforming an object center $\mathbf{c}$ into that observer's frame and applying the mapping $\mathcal{S} : (x, y, z) \mapsto (r, \theta, \varphi)$ of Eq. 1 gives the label

$$\mathbf{s} = \mathcal{S}\Big(\mathbf{R}^\top (\mathbf{c} - \mathbf{o})\Big). \tag{5}$$

For the egocentric label $\mathbf{s}_{\text{ego}}$, we set $\mathbf{c} = \mathbf{c}_{\text{tgt}}$ and use the recording camera in an eligible frame $i \in V_{\text{tgt}}$ (Eq. 4) as the observer, so $(\mathbf{R}, \mathbf{o}) = (\mathbf{R}_i, \mathbf{t}_i)$. The allocentric observer is instead imagined, following the allocentric question setup in VSI-Bench (Yang et al., 2025b): it stands at an anchor object, faces a reference object, and the query asks where the target lies. We place it at the anchor center $\mathbf{c}_{\text{anc}}$ and orient it toward the reference center $\mathbf{c}_{\text{ref}}$, building its right, down, and forward axes $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ from that 3D direction and the world up vector $\mathbf{u}$:

$$\hat{\mathbf{z}} = \frac{\mathbf{c}_{\text{ref}} - \mathbf{c}_{\text{anc}}}{\|\mathbf{c}_{\text{ref}} - \mathbf{c}_{\text{anc}}\|}, \qquad \hat{\mathbf{x}} = \frac{\hat{\mathbf{z}} \times \mathbf{u}}{\|\hat{\mathbf{z}} \times \mathbf{u}\|}, \qquad \hat{\mathbf{y}} = \hat{\mathbf{z}} \times \hat{\mathbf{x}}, \tag{6}$$

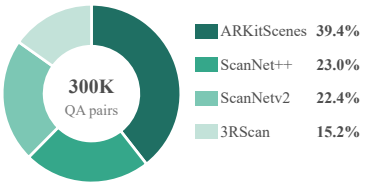
Substituting this observer pose and target center $\mathbf{c}_{\text{tgt}}$ into Eq. 5 gives the allocentric label $\mathbf{s}_{\text{allo}}$.

Instruction Synthesis. A QA pair consists of a visual input, a question, and the label as its answer. The visual input must show every object the question names. An egocentric QA pair uses the frame on which $\mathbf{s}_{\text{ego}}$ was computed. An allocentric QA pair names the anchor, the reference, and the target, and takes one of three visual forms. It is single-image when V_{anc}, V_{ref}, and V_{tgt} share a frame. Otherwise we take one frame per object and order them along the camera trajectory, requiring consecutive frames to share at least two visible objects so that the views chain into one layout. The third form replaces them with the scene video, which we also use when no chain exists. Objects are named by semantic labels, unique within the shown frame for egocentric pairs and within the scene otherwise.

Each question is assembled from two templates in App. A.2. A viewpoint stem names the observer and the target, the camera for an egocentric label and the anchor and the reference for an allocentric one, with the object slots filled by semantic labels. A quantity request asks for the distance, the azimuth, the elevation, or all three, which gives four questions per observer and target. We sample among paraphrases of both templates to vary the wording. The answer is the label, $\mathbf{s}_{\text{ego}}$ or $\mathbf{s}_{\text{allo}}$, in meters and degrees, and Fig. 4(c) gives an allocentric example. We then validate the corpus with automatic checks and a human inspection of samples, with protocol and results in App. A.1.

Statistic	Value
Scenes	6,892
Objects	23.1K
Images	49.9K
Videos	1,798
Ego/Allo QA	115K/185K

(a) Corpus statistics



(b) Dataset composition



Q: Stand at the *table*, face the *bookshelf*, and locate the *whiteboard*.
A: $(2.97\,\text{m}, 26.8°, 7.8°)$

(c) A grounding example

Figure 4: **Overview of SPHEREQA.** (a) The table summarises the scale and composition of SPHEREQA. (b) QA pairs by scan source, with percentages giving each source's corpus share. (c) An allocentric example: the observer stands at the anchor facing the reference and the answer is the target's (r, θ, φ) in that frame.

Corpus Statistics. Applied to four annotated 3D scan collections, the pipeline yields 300K QA pairs, about 75K for each of the four coordinate question types. Of the 300K QA pairs, 50% use a single image as visual input, 25% use multiple images, and 25% use a video. Fig. 4 summarizes the corpus scale and scan-source composition and shows an allocentric example.

4.2 THE SCOPE MODEL

Fine-tuning an MLLM on SPHEREQA generates each coordinate as a raw-text string. Because a subword tokenizer splits that string, next-token cross-entropy matches tokens rather than comparing values (Zausinger et al., 2025), so numerical proximity to the target is not explicitly reflected in the loss.

To address this, we therefore propose SCOPE, which recasts coordinate prediction as per-axis classification. As shown in Fig. 5, we leave the architecture unchanged and extend only the vocabulary with a spherical coordinate tokenizer that discretizes distance, azimuth, and elevation into bins, so that predicting an axis becomes a classification over its bins. Each token is initialized from a phrase naming its axis and bin center, placing it in the pretrained embedding space by the quantity it denotes.

Spherical Coordinate Tokenizer. Inspired by Pix2Seq (Chen et al., 2022), which discretizes 2D bounding-box coordinates into vocabulary tokens for autoregressive prediction, we extend this idea from homogeneous image-plane coordinates to the three heterogeneous axes of spherical space. We partition each axis into half-open bins, assign one token per bin, and define the tokenizer as

$$\mathcal{T} : (r, \theta, \varphi) \;\mapsto\; (t_r,\, t_\theta,\, t_\varphi), \tag{7}$$

where each output token denotes the bin containing the corresponding input coordinate. Azimuth and elevation are partitioned uniformly, one token per $5°$, so <az_0_5> stands for $\theta \in [0°, 5°)$. Radial distance is instead partitioned with growing width, $0.2\,\text{m}$ below $2\,\text{m}$, then $0.5\,\text{m}$ up to $5\,\text{m}$, $1\,\text{m}$ up to $10\,\text{m}$, and a single bin beyond, since the same metric error matters more at short range.

We train SCOPE with the standard next-token cross-entropy over the extended vocabulary, so no regression head or auxiliary loss is needed. A coordinate answer is supervised with one bin token per requested axis rather than with a numeric string. Supervision thereby acts on a region of space instead of on digits: the $3.1\,\text{m}$ prediction shares the token of its $3.0\,\text{m}$ label, while $2.0\,\text{m}$ does not. A special token <SPHERE_MODE> prefixes such queries and triggers this coordinate mode, leaving other queries answered in natural language.

Semantically Initialized Embeddings. The newly added spherical coordinate tokens require initial embeddings that reflect the axis and magnitude each bin denotes. Random or mean initialization is ill-suited: random vectors ignore the existing embedding topology, and a shared mean erases distinctions among spatial bins (Dobler & De Melo, 2023). We instead initialize each token b from the averaged subword embeddings of a phrase naming its axis and bin center,

$$\mathbf{e}_b \;=\; \frac{1}{|\mathcal{W}_b|} \sum_{s \in \mathcal{W}_b} \mathbf{e}_s, \tag{8}$$

where $\mathcal{W}_b$ is the set of subword tokens in that phrase and $\mathbf{e}_s$ is the pretrained embedding of subword s, so the bins of one axis begin grouped together, with adjacent bins closer than distant ones. Gradient masking then restricts updates to these new embeddings, preserving the model's general ability.

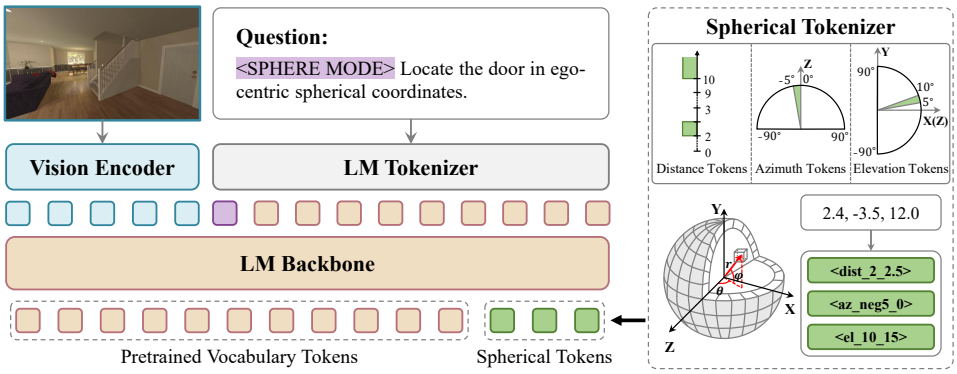


Figure 5: **Overview of SCOPE.** We extend the LM vocabulary with spherical tokens, and a <SPHERE_MODE> prefix triggers decoding in this vocabulary (left). The spherical tokenizer bins azimuth and elevation every $5°$ and radial distance with growing width, assigning one token per bin (right). A coordinate is then mapped to the three tokens whose bins contain its values, one per axis.

5 EXPERIMENT

5.1 EXPERIMENTAL SETUP

Benchmarks and Metrics. We evaluate on four recent benchmarks covering distinct axes of spatial reasoning: MindCube-Tiny (Wang et al., 2026) (mental modeling from limited views), MMSI-Bench (Yang et al., 2026c) (multi-view reasoning), ViewSpatial-Bench (Li et al., 2026a) (egocentric and allocentric reasoning), and ReVSI (Zhang et al., 2026d) (video-based reasoning). We follow the official protocol of each, which scores multiple-choice questions by exact-match accuracy and the numerical questions of ReVSI by mean relative accuracy. All scores are reported in percent.

Model Training. We adopt Qwen2.5-VL-7B-Instruct (Bai et al., 2025b) and Qwen3-VL-8B-Instruct (Bai et al., 2025a) as our base models. We update only the LoRA adapters (r=64, α=128) on the LLM decoder and the new spherical coordinate token embeddings, under the gradient masking of §4.2. Following VST (Yang et al., 2026a) and SenseNova (Cai et al., 2026), we mix SPHEREQA with an *auxiliary mixture* of public spatial QA and general VQA, added to preserve general instruction following. The resulting 600K training set comprises 300K SPHEREQA and a 300K auxiliary mixture of 200K SenseNova (Cai et al., 2026), 50K GQA (Hudson & Manning, 2019), 40K VSI-590K (Yang et al., 2026b), and 10K SAT-v2 (Ray et al., 2025). SPHEREQA is synthesized with numeric answers, which we convert into the formats of §4.2 in a 45:45:10 token/numeric/joint ratio, prepending <SPHERE_MODE> to the token and joint queries. We train with batch size 128 and learning rate 5×10^{-5} for 1 epoch on 8 Ascend 910B NPUs.

Inference Setup. We evaluate the four spatial-QA benchmarks with lmms-eval (Zhang et al., 2025b). We include SenseNova-SI-1.1-Qwen2.5-VL-7B (Cai et al., 2026) (SenseNova-SI-1.1-7B in tables). For fair comparison, every model built on Qwen2.5/3-VL, including SCOPE, is given a visual token budget of 512 per image. The two remaining baselines use a different visual encoder and cannot be reduced to this budget, so we grant them a larger one: 768 tokens per image for InternVL3.5 and 1,512 for Cambrian-S. For ReVSI benchmark, we uniformly sample 32 frames per video.

5.2 MAIN RESULT

Spatial Reasoning Capability. In Table 1, SCOPE improves the average score of Qwen2.5-VL-7B by 10.2 points and that of Qwen3-VL-8B by 11.1, with gains on all four benchmarks for both backbones. The two largest gains, $+21.4$ on MindCube-Tiny for the 8B model and $+18.9$ on ReVSI for the 7B one, fall on the benchmarks that ask a model to assemble one layout from observations each showing only part of a scene, which is what object-level grounding supervises. MMSI-Bench is also multi-view, but poses many of its questions between two scene entities instead of between the observer and an entity, and gains less ($+5.8$/ $+7.9$). This ordering is not specific to one model: both backbones improve by a comparable margin, 10.2 and 11.1 points, and even our 7B variant exceeds every specialist in the table on average. App. A.3 reports the per-subtask results, and App. A.4 directly evaluates object-center localization.

Table 1: **Evaluation on four spatial benchmarks.** Avg. is computed over the benchmarks for each model. Values in gray denote the gain of SCOPE over its corresponding base model. Bold denotes the best result among open-source general and specialist models. † Results from the original paper because model weights are unavailable. ‡ Results from the ReVSI paper, evaluated with GPT-5.2 at 64 frames per video.

Model	Pub	MindCube-Tiny	MMSI-Bench	ViewSpatial	ReVSI	Avg.
• ***Proprietary Models***						
GPT-4o (Hurst et al., 2024)	–	38.8	30.3	35.0	–	–
GPT-5 (Singh et al., 2025)	–	56.3	41.8	45.6	50.9‡	48.7
Gemini-2.5-Pro (Comanici et al., 2025)	–	57.6	38.0	46.1	–	–
• ***Open-Source General Models***						
Qwen2.5-VL-7B (Bai et al., 2025b)	arXiv 25.02	33.0	27.8	37.9	26.6	31.3
Qwen3-VL-8B (Bai et al., 2025a)	arXiv 25.11	27.9	28.6	40.8	44.4	35.4
InternVL3.5-8B (Wang et al., 2025)	arXiv 25.08	41.6	28.2	38.9	45.3	38.5
• ***Spatial Specialist Models***						
SpaceR-SFT-7B (Ouyang et al., 2025)	arXiv 25.04	36.6	26.7	39.2	28.7	32.8
ViLaSR-7B (Wu et al., 2025b)	NeurIPS 25	33.6	28.8	35.2	38.1	33.9
SpatialReasoner-7B (Ma et al., 2025a)	NeurIPS 25	32.9	23.5	39.4	33.1	32.2
SpatialLadder-3B (Li et al., 2026b)	ICLR 26	44.5	27.3	40.9	36.5	37.3
Cambrian-S-7B (Yang et al., 2026b)	ICLR 26	38.7	24.1	37.8	46.3	36.7
Spatial-SSRL-7B (Liu et al., 2026)	CVPR 26	33.1	26.2	38.6	29.1	31.8
SpaceVista-SFT-7B† (Sun et al., 2026)	ICML 26	–	29.1	–	–	–
VST-SFT-7B (Yang et al., 2026a)	ECCV 26	38.9	30.7	**50.5**	38.4	39.6
SenseNova-SI-1.1-7B (Cai et al., 2026)	CVPR 26	**50.4**	29.7	43.3	40.3	40.9
• ***Ours***						
SCOPE Qwen2.5-VL-7B	–	47.2(+14.2)	33.6(+5.8)	39.6(+1.7)	45.5(+18.9)	41.5(+10.2)
SCOPE Qwen3-VL-8B	–	49.3(+21.4)	**36.5**(+7.9)	49.8(+9.0)	**50.4**(+6.0)	**46.5**(+11.1)

Among the evaluated open-source general and spatial specialist models, SCOPE-8B achieves the highest average score (46.5%), surpassing SenseNova-SI-1.1-7B and VST-SFT-7B by 5.6 and 6.9 points, respectively. It ranks first on MMSI-Bench and ReVSI, and second on MindCube-Tiny and ViewSpatial, trailing the respective leaders by only 1.1 and 0.7 points. This broad performance suggests that object-level geometric grounding supports spatial reasoning across both multi-view images and video. Notably, it surpasses Cambrian-S-7B on ReVSI by 4.1 points despite Cambrian-S's dedicated video spatial reasoning training. On ViewSpatial, it also exceeds all proprietary baselines.

General Multimodal Capability. To assess whether SCOPE preserves capabilities beyond spatial reasoning, we compare it against each backbone on three general benchmarks spanning image, video, and document understanding: SEED-Bench (Li et al., 2024a), MVBench (Li et al., 2024b), and DocVQA (Mathew et al., 2021). As shown in Table 2, results are essentially unchanged: two of the six comparisons improve, by up to +0.8 on SEED for Qwen3-VL, and none of the remaining four decreases by more than 0.5 points. Thus SCOPE enhances spatial reasoning without sacrificing general capability.

Table 2: **Performance on general multimodal benchmarks.** SCOPE preserves general capability.

Benchmark	Qwen2.5-VL		Qwen3-VL	
	Base	SCOPE	Base	SCOPE
SEED	79.0	78.8(−0.2)	80.4	81.2(+0.8)
MVBench	67.8	67.3(−0.5)	68.6	68.5(−0.1)
DocVQA	93.5	93.4(−0.1)	94.1	94.2(+0.1)

5.3 DIAGNOSTIC EXPERIMENTS

We diagnose SCOPE along two factors: the coordinate representation, spanning both the coordinate system and its tokenization, to check whether either still matters once trained rather than only prompted, and the training corpus, spanning its composition, scale, and efficiency relative to prior specialists. Every variant is trained on the 8B model, otherwise following the setup of §5.

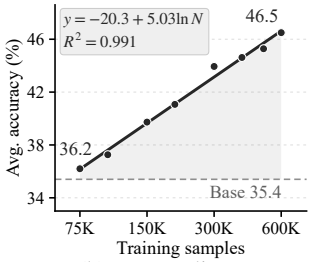
Coordinate System. We found in §3 that coordinate representation affects in-context reasoning; we now test whether it also matters for supervision. The raw-text variants in Table 3 supervise the same object centers as Cartesian (x, y, z) or spherical (r, θ, φ) coordinates, isolating the coordinate system; Base is the untuned backbone. Both improve over Base, by 7.0 and 9.1 points,

Table 3: **Ablation on coordinate representation.** We compare Cartesian and spherical coordinates in raw text against spherical tokens under random or semantic initialization.

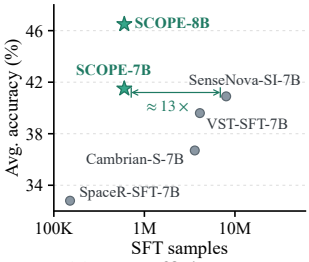
Benchmark	Base	Raw text		Spherical Tokens	
		Cartesian	Spherical	Rand. init	Sem. init
MindCube	27.9	41.7	43.1	47.3	**49.3**
MMSI	28.6	34.3	36.0	35.8	**36.5**
ViewSpatial	40.8	47.2	**50.8**	50.0	49.8
ReVSI	44.4	46.5	48.2	48.9	**50.4**
Avg.	35.4	42.4(+7.0)	44.5(+9.1)	45.5(+10.1)	**46.5**(+11.1)

#Samples	Base	300K			600K	
SPHEREQA		✓	✗	✓	✗	✓
Aux. mixture		✗	✓	✓	✓	✓
MindCube	27.9	42.2	44.8	45.4	47.0	**49.3**
MMSI	28.6	29.5	32.6	34.2	35.6	**36.5**
ViewSpatial	40.8	41.7	46.3	46.9	47.8	**49.8**
ReVSI	44.4	45.4	46.6	49.3	46.9	**50.4**
Avg.	35.4	39.7	42.6	43.9	44.3	**46.5**
△	–	+4.3	+7.2	+8.5	+8.9	**+11.1**

(a) Data composition



(b) Data scaling



(c) Data efficiency

Figure 6: **Analysis of the training corpus.** (a) Comparison of SPHEREQA, auxiliary data, and their equal-part mixtures at matched training budgets. (b) Average score of SCOPE-8B across training set sizes, showing approximately log-linear gains up to 600K samples. (c) Average score versus SFT corpus size for SCOPE (stars) and 7B specialists (circles); upper-left positions indicate higher scores with fewer samples.

and the spherical form leads on all four benchmarks. Its advantage is largest on ViewSpatial, which directly queries egocentric and allocentric directions. Spherical labels expose the requested angular quantities, whereas Cartesian coordinates require combining components to recover them.

Coordinate Tokenization. We next test whether SCOPE's tokenizer yields further gains with the coordinate system fixed to spherical. In Table 3, both token variants outperform raw-text coordinate generation on average, suggesting that per-axis classification makes spherical grounding easier to learn than raw-text coordinate generation. Semantic initialization further improves the average score by 1.0 point over random initialization, suggesting that pretrained representations of axes and numerical values provide a better starting point for learning the new coordinate vocabulary.

Data Composition. We examine SPHEREQA's contribution to spatial reasoning at matched training budgets (Fig. 6(a)). At 300K samples, fine-tuning on SPHEREQA alone improves the average score over the base model by 4.3 points without using dedicated spatial reasoning QA. This suggests that learning to localize objects in 3D develops spatial knowledge that transfers to downstream reasoning. The auxiliary mixture yields a larger gain (+7.2), consistent with its use of dedicated spatial reasoning QA. Combining the two in equal parts achieves the highest average score at 300K, outperforming the auxiliary mixture alone by 1.3 points. This improvement, despite replacing half of the auxiliary samples, suggests that explicit object localization provides useful supervision beyond reasoning answers alone. The same pattern holds at 600K, where the advantage of mixed training over the auxiliary mixture increases from 1.3 to 2.2 points. These results support object-level grounding as an effective training signal for spatial reasoning, with its benefit increasing at the higher training budget.

Data Scaling. Following prior work on neural scaling laws (Kaplan et al., 2020), we evaluate SCOPE-8B at eight training budgets from 75K to 600K, fine-tuning independently with fixed data proportions. The average score rises from 36.2% to 46.5% (Fig. 6(b)). The approximately log-linear trend ($R^2 = 0.991$) corresponds to a fitted gain of 3.5 points per doubling of the training data.

Data Efficiency. Fig. 6(c) shows that both SCOPE variants achieve higher average scores with substantially less SFT data. Using 600K samples, our 7B and 8B models reach 41.5% and 46.5%, respectively, surpassing VST-SFT-7B (39.6%) and SenseNova (40.9%), which use 6.8× and 13.3× as much data. This favorable accuracy–data trade-off demonstrates that strong spatial reasoning on these benchmarks is achievable with a compact SFT corpus.

6 CONCLUSION

In this work, a controlled prompting study across three MLLMs shows that spherical coordinates outperform depth, 3D boxes, and Cartesian coordinates as grounding representations for spatial reasoning. Building on this finding, we introduce SPHEREQA, a dataset of 300K object-level grounding QA pairs derived from 3D scans and covering both egocentric and allocentric reference frames, and propose SCOPE, which recasts spherical coordinate prediction as per-axis classification with semantically initialized tokens. Across four spatial benchmarks, SCOPE improves Qwen2.5-VL-7B and Qwen3-VL-8B by 10.2 and 11.1 points on average, respectively, while limiting degradation on three general multimodal benchmarks to at most 0.5 points. These results support object-level spherical grounding as an effective supervision signal for improving MLLM spatial reasoning.

THE USE OF LARGE LANGUAGE MODELS (LLMS)

We used generative AI tools to edit and review the code for this work, and later to edit this paper to improve readability. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Mohammad Asadi, Jack W O'Sullivan, Fang Cao, Tahoura Nedaee, Kamyar Rajabalifardi, Fei-Fei Li, Ehsan Adeli, and Euan Ashley. Mirage: The illusion of visual understanding. *arXiv preprint arXiv:2603.21687*, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

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

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. In *ICRA*, 2025.

Zhongang Cai, Ruisi Wang, Chenyang Gu, Fanyi Pu, Junxiang Xu, Yubo Wang, Wanqi Yin, Zhitao Yang, Chen Wei, Tongxi Zhou, Qingping Sun, Hui En Pang, Jiaqi Li, Oscar Qian, 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 *CVPR*, 2026.

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

Mingfei Chen, Zijun Cui, Xiulong Liu, Jinlin Xiang, Yang Zheng, Jingyuan Li, and Eli Shlizerman. SAVVY: Spatial awareness via audio-visual LLMs through seeing and hearing. In *NeurIPS*, 2025a.

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 *NeurIPS*, 2025b.

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *ICLR*, 2022.

Zhangquan Chen, Ruihui Zhao, Chuwei Luo, Mingze Sun, Xinlei Yu, Yangyang Kang, and Ruqi Huang. Sifthinker: Spatially-aware image focus for visual reasoning. In *AAAI*, 2026.

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 *NeurIPS*, 2024.

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 *ICLR*, 2026.

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *ICLR*, 2018.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *ECCV*, 2018.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. A survey on multimodal large language models for autonomous driving. In *WACV*, 2024.

Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, and Peter Grasch. Mm-spatial: Exploring 3d spatial understanding in multimodal llms. In *ICCV*, 2025.

Konstantin Dobler and Gerard De Melo. Focus: Effective embedding initialization for monolingual specialization of multilingual models. In *EMNLP*, 2023.

Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *ICCV*, 2021.

Mohsen Gholami, Ahmad Rezaei, Zhou Weimin, Sitong Mao, Shunbo Zhou, Yong Zhang, and Mohammad Akbari. Spatial reasoning with vision-language models in ego-centric multi-view scenes. In *ICLR*, 2026.

Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(12):4338–4364, 2021.

Wenbo Hu, Jingli Lin, Yilin Long, Yunlong Ran, Lihan Jiang, Yifan Wang, Chenming Zhu, Runsen Xu, Tai Wang, and Jiangmiao Pang. G^2 vlm: Geometry grounded vision language model with unified 3d reconstruction and spatial reasoning. In *CVPR*, 2026.

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

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3d recognition. In *CVPR*, 2023.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. Seed-bench: Benchmarking multimodal large language models. In *CVPR*, 2024a.

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, et al. Viewspatial-bench: Evaluating multi-perspective spatial localization in vision-language models. In *ECCV*, 2026a.

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 *ICLR*, 2026b.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024b.

Pengteng Li, Pinhao Song, Wuyang Li, Huizai Yao, Weiyu Guo, Yijie Xu, Dugang Liu, and Hui Xiong. See&trek: Training-free spatial prompting for multimodal large language model. In *NeurIPS*, 2025.

Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Ye Duan, and Liu Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *CVPR*, 2022.

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 *ICLR*, 2026.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Le Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024.

Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Trans. Mechatron.*, 30(6):7253–7274, 2025.

Yuhong Liu, Beichen Zhang, Yuhang Zang, Yuhang Cao, Long Xing, Xiaoyi Dong, Haodong Duan, Dahua Lin, and Jiaqi Wang. Spatial-ssrl: Enhancing spatial understanding via self-supervised reinforcement learning. In *CVPR*, 2026.

Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. In *NeurIPS*, 2024.

Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso M de Melo, Jianwen Xie, and Alan Yuille. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. In *NeurIPS*, 2025a.

Wufei Ma, Luoxin Ye, Celso M de Melo, Alan Yuille, and Jieneng Chen. Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models. In *CVPR*, 2025b.

Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021.

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.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *ICLR*, 2025.

Arijit Ray, Jiafei Duan, Ellis L Brown II, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A. Plummer, Ranjay Krishna, Kuo-Hao Zeng, and Kate Saenko. Sat: Dynamic spatial aptitude training for multimodal language models. In *COLM*, 2025.

Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. Panoformer: panorama transformer for indoor 360° depth estimation. In *ECCV*, 2022.

Sangyun Shin, Kaichen Zhou, Madhu Vankadari, Andrew Markham, and Niki Trigoni. Spherical mask: Coarse-to-fine 3d point cloud instance segmentation with spherical representation. In *CVPR*, 2024.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *CVPR*, 2025a.

Shezheng Song, Xiaopeng Li, Shasha Li, Shan Zhao, Jie Yu, Jun Ma, Xiaoguang Mao, Weimin Zhang, and Meng Wang. How to bridge the gap between modalities: Survey on multimodal large language model. *IEEE Trans. Knowl. Data Eng.*, 37(9):5311–5329, 2025b.

Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *CVPR*, 2021.

Peiwen Sun, Shiqiang Lang, Dongming Wu, Yi Ding, Kaituo Feng, Huadai Liu, Zhen Ye, Rui Liu, Yun-Hui Liu, Jianan Wang, et al. Spacevista: All-scale visual spatial reasoning from mm to km. In *ICML*, 2026.

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 *NeurIPS*, 2024.

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 *ICLR*, 2026.

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

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. In *NeurIPS*, 2025a.

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 *NeurIPS*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

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

Rui Yang, Ziyu Zhu, Yanwei Li, Jingjia Huang, Shen Yan, Siyuan Zhou, Zhe Liu, Xiangtai Li, Shuangye Li, Wenqian Wang, et al. Visual spatial tuning. In *ECCV*, 2026a.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis L. Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, Rob Fergus, Yann LeCun, Li Fei-Fei, and Saining Xie. Cambrian-s: Towards spatial supersensing in video. In *ICLR*, 2026b.

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

Yuncong Yang, Jiageng Liu, Zheyuan Zhang, Siyuan Zhou, Reuben Tan, Jianwei Yang, Yilun Du, and Chuang Gan. Mindjourney: Test-time scaling with world models for spatial reasoning. In *NeurIPS*, 2025c.

Koonting Yip, Qiyan Zhao, Wenhao Yu, Liangyu Yuan, Mingkai Li, Xiaofeng Zhang, Jianmin Ji, Yanyong Zhang, Qing Jiang, and Ka-Veng Yuen. Sope: Spherical coordinate-based positional embedding for enhancing spatial perception of 3d lvlms. In *CVPR*, 2026.

Yuqian Yuan, Hang Zhang, Wentong Li, Zesen Cheng, Boqiang Zhang, Long Li, Xin Li, Deli Zhao, Wenqiao Zhang, Yueting Zhuang, Jianke Zhu, and Lidong Bing. Videorefer suite: Advancing spatial-temporal object understanding with video llm. In *CVPR*, 2025.

Jonas Zausinger, Lars Pennig, Anamarija Kozina, Sean Sdahl, Julian Sikora, Adrian Dendorfer, Timofey Kuznetsov, Mohamad Hagog, Nina Wiedemann, Kacper Chlodny, Vincent Limbach, Anna Ketteler, Thorben Prein, Vishwa Mohan Singh, Michael Danziger, and Jannis Born. Regress, don't guess: A regression-like loss on number tokens for language models. In *ICML*, 2025.

Xiaoyu Zhan, Wenxuan Huang, Hao Sun, Xinyu Fu, Changfeng Ma, Shaosheng Cao, Bohan Jia, Shaohui Lin, Zhenfei Yin, LEI Bai, Wanli Ouyang, Yuanqi Li, Jie Guo, and Yanwen Guo. Actial: Activate spatial reasoning ability of multimodal large language models. In *NeurIPS*, 2025.

Gongjie Zhang, Wenhao Li, Quanhao Qian, Jiuniu Wang, Deli Zhao, Shijian Lu, and Ran Xu. On the generalization capacities of mllms for spatial intelligence. In *ICLR*, 2026a.

Junsong Zhang, Zisong Chen, Chunyu Lin, Zhijie Shen, Lang Nie, Kang Liao, and Yao Zhao. Sgformer: Spherical geometry transformer for 360 depth estimation. *IEEE Trans. Circuits Syst. Video Technol.*, 35(6):5738–5748, 2025a.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of NAACL*, 2025b.

Pingyue Zhang, Zihan Huang, Yue Wang, Jieyu Zhang, Letian Xue, Zihan Wang, Qineng Wang, Keshigeyan Chandrasegaran, Ruohan Zhang, Yejin Choi, Ranjay Krishna, Jiajun Wu, Li Fei-Fei, and Manling Li. Theory of space: Can foundation models construct spatial beliefs through active exploration? In *ICLR*, 2026b.

Xuejun Zhang, Aditi Tiwari, Zhenhailong Wang, and Heng Ji. Predicting camera pose from perspective descriptions for spatial reasoning. *arXiv preprint arXiv:2602.06041*, 2026c.

Yiming Zhang, Jiacheng Chen, Jiaqi Tan, Yongsen Mao, Wenhu Chen, and Angel X Chang. Revsi: Rebuilding visual spatial intelligence evaluation for accurate assessment of vlm 3d reasoning. In *ICML*, 2026d.

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and Shanghang Zhang. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. In *NeurIPS*, 2025.

Fangrui Zhu, Hanhui Wang, Yiming Xie, Jing Gu, Tianye Ding, Jianwei Yang, and Huaizu Jiang. Struct2d: A perception-guided framework for spatial reasoning in mllms. In *NeurIPS*, 2025.

A APPENDIX

A.1 SPHEREQA CONSTRUCTION DETAILS

Scene-level holdout. Public indoor scan collections with camera poses and object-level 3D annotations are limited and widely reused in spatial reasoning benchmarks. To avoid overlap between training and evaluation scenes, we first match source scene identifiers against the evaluation scenes of the four benchmarks and exclude all matches. We construct SPHEREQA from the remaining scenes. We apply the same scene-overlap checks and exclusion rules to the auxiliary training data.

Object geometry and visibility. For ScanNet++, ScanNetv2, and 3RScan, we obtain object vertex sets from the instance annotations on the scene meshes. For ARKitScenes, we use the center, dimensions, and rotation of each annotated oriented 3D box to extract the scene-mesh vertices within it, forming a box-derived vertex set $\mathcal{P}_j$. We express the mesh, boxes, and camera poses in a common coordinate frame. We apply Eq. 4 using depth rendered from the full scene mesh, which accounts for occlusion by other objects and surrounding surfaces. The annotated box center is retained as the ARKitScenes localization target.

Object eligibility. An object is eligible when its semantic label is not a structural or non-referring one (wall, floor, ceiling, door, window, room, object, unknown) and is unique within the shown frame for egocentric pairs or within the scene otherwise.

Accepted label range. With the capture cameras' field of view as a reference, we define a common angular supervision range for egocentric and allocentric queries, retaining labels with $\theta \in [-70°, 70°)$ and $\varphi \in [-60°, 60°)$. Object visibility is checked in the input views using Eq. 4. To avoid unstable directions when the observer and target centers nearly coincide, we also require $r \geq 0.2\,\text{m}$. For allocentric queries, we exclude configurations where the basis in Eq. 6 becomes unstable because the anchor and reference centers nearly coincide or are nearly vertically aligned.

Visual forms and sampling. A multi-image sample chains 2–3 views along the camera trajectory with adjacent pairs sharing at least two visible objects other than the queried ones, and no single view, or single pair when there are three views, may show the anchor, the reference, and the target together, to encourage integration of the scene layout across views. A video sample uses 16 uniformly sampled frames, is allocentric only, and requires each named object to be visible in at least one frame. We draw at most 24, 16, and 24 observer–target combinations per scene for the single-image, multi-image, and video forms, and fill targets of 150K, 75K, and 75K pairs by round-robin over sources and then scenes, seeded by a hash of the scene key.

Quality control. An independent validation pass checks that coordinates lie in the valid ranges, that answers follow the requested format, and that all referenced media are available. We further inspect 500 randomly sampled QA pairs for unambiguous referring labels (489/500, 97.8%), referent visibility (486/500, 97.2%), and angular-sign consistency with the depicted layout (496/500, 99.2%).

A.2 SPHEREQA TEMPLATES

Each question is built in three steps. (1) Select a viewpoint template from Table 4 and fill {`object`} (and, for allocentric templates, {`anchor`} and {`reference`}) with semantic labels. (2) Select a quantity request from part (a) of Table 5: distance, azimuth, elevation, or the full spherical coordinate triplet. (3) Select an answer format from part (b) and append the selected components in this order. SPHEREQA stores numerical answers, and the token and joint formats are produced at training time in the ratio of §5.1, with `<SPHERE_MODE>` prepended to the token and joint queries only. For example: "Stand at the table, face the sofa, and locate the chair. Give the distance r. Answer in meters." Image and video placeholders are omitted for readability.

Token answers contain the requested axis token or the ordered triplet. Numerical answers contain the corresponding scalar or triplet rounded to two decimal places. Joint answers state both forms with explicit units. For example, a joint training target is "Spherical coordinate tokens: `(<dist_3_3.5>, <az_15_20>, <el_neg15_neg10>)`; Numeric values: (3.15 m, 18.69 degrees, -14.65 degrees)."

Table 4: **Templates for defining the spatial viewpoint.** Each question first uses one template to specify the observation format and the viewpoint from which the target object is described.

Setting	Stem
Single-image ego	In this image, locate the {object} relative to the camera From the camera viewpoint in this image, locate the {object} Using this image, report where the {object} is relative to the viewer
Single-image allo	In this image, stand at the {anchor}, face the {reference}, and locate the {object} Using this image, imagine standing at the {anchor} and looking toward the {reference}; locate the {object}
Multi-image allo	Use the ordered images as views of one scene. Stand at the {anchor}, face the {reference}, and locate the {object} Using all the ordered images, imagine standing at the {anchor} and looking toward the {reference}; locate the {object}
Video allo	Use the entire video. Stand at the {anchor}, face the {reference}, and locate the {object} After watching the full video, imagine standing at the {anchor} and looking toward the {reference}; locate the {object}

Table 5: **Templates for the prediction target and answer format.** After selecting a viewpoint template, the generator selects one item from (a) to specify what spatial quantity to predict and one item from (b) to specify how to express the answer. The two selected texts are appended in that order.

Component	Text
(a) Quantity request: what to predict	
Distance	Give the distance r.
Azimuth	Give the azimuth theta; right is positive and left is negative.
Elevation	Give the elevation phi; up is positive and down is negative.
Full coordinate	Give spherical coordinates in the order (r, theta, phi): distance, azimuth, elevation.
(b) Answer format: how to express it	
Token, scalar	Answer with a spherical coordinate token.
Token, full	Answer with spherical coordinate tokens.
Numerical, distance	Answer in meters.
Numerical, angle	Answer in degrees.
Numerical, full	Use meters for r and degrees for both angles.
Joint, scalar	Answer with both the spherical coordinate token and numeric value.
Joint, full	Answer with both spherical coordinate tokens and numeric values.

A.3 Results by Subtask

Table 1 summarizes overall benchmark performance. Here, we report results for each benchmark's official subtasks to examine how the improvements vary across spatial abilities. Several of the largest gains occur in distance estimation, camera-relative positioning, and reasoning across viewpoints. This pattern is consistent with our object-level positional supervision, while the magnitude of improvement varies across subtasks and backbones.

Table 6: **Per-setting results on MindCube-Tiny.** Accuracy (%). Bold marks the best result per column, and dashes mark unavailable scores.

Model	Around	Among	Rotation	Overall
Qwen2.5-VL-7B	34.8	32.2	33.0	33.0
Qwen3-VL-8B	25.6	27.2	33.0	27.9
InternVL3.5-8B	47.6	41.2	35.5	41.6
SpaceR-SFT-7B	40.8	35.7	34.0	36.6
ViLaSR-7B	42.0	29.0	37.0	33.6
SpatialReasoner-7B	42.4	31.7	24.5	32.9
SpatialLadder-3B	54.6	45.5	39.0	44.5
Cambrian-S-7B	38.0	41.3	31.5	38.7
Spatial-SSRL-7B	36.4	30.0	38.5	33.1
VST-SFT-7B	48.4	35.7	36.5	38.9
SenseNova-SI-1.1-7B	**59.2**	51.2	37.0	**50.4**
SCOPE Qwen2.5-VL-7B	52.0	49.5	34.5	47.2
SCOPE Qwen3-VL-8B	46.8	**52.8**	**42.0**	49.3

MindCube-Tiny. Table 6 shows improvements across all three settings for both backbones. Relative to their base models, SCOPE-7B and SCOPE-8B improve by 17.2/21.2 points on *Around* and 17.3/25.6 points on *Among*, respectively. These gains are consistent with stronger integration of spatial observations across viewpoints. SCOPE-8B achieves the highest scores on *Among* (52.8) and *Rotation* (42.0), exceeding SenseNova-SI-1.1-7B by 1.6 and 5.0 points, respectively. Its overall score reaches 49.3, a 21.4-point gain over Qwen3-VL-8B.

Table 7: **Results by subtask on MMSI-Bench.** Accuracy (%). Cam, Obj, and Reg denote camera, object, and region, and Meas., App., Mot., and MSR denote measurement, appearance, motion, and multi-step reasoning. Bold marks the best result per column, and dashes mark unavailable scores.

Model	Positional						Attribute		Motion		MSR	Avg.
	Cam-Cam	Cam-Obj	Cam-Reg	Obj-Obj	Obj-Reg	Reg-Reg	Meas.	App.	Cam Mot.	Obj Mot.		
Qwen2.5-VL-7B	26.9	30.2	33.7	25.5	35.3	23.5	26.6	21.2	21.6	34.2	26.8	27.8
Qwen3-VL-8B	28.0	26.7	32.5	27.7	36.5	**40.7**	32.8	18.2	23.0	25.0	25.8	28.6
InternVL3.5-8B	32.3	31.4	25.3	25.5	30.6	29.6	31.2	24.2	16.2	**38.2**	26.8	28.2
SpaceR-SFT-7B	28.0	27.9	22.9	31.9	28.2	28.4	29.7	31.8	13.5	31.6	23.7	26.7
ViLaSR-7B	28.0	38.4	41.0	31.9	**44.7**	21.0	28.1	19.7	14.9	32.9	21.7	28.8
SpatialReasoner-7B	20.4	14.0	28.9	14.9	32.9	24.7	25.0	27.3	9.5	32.9	26.3	23.5
SpatialLadder-3B	38.7	36.0	22.9	30.9	27.1	32.1	21.9	31.8	21.6	22.4	20.7	27.3
Cambrian-S-7B	19.4	43.0	31.3	27.7	18.8	23.5	29.7	15.2	10.8	25.0	21.7	24.1
Spatial-SSRL-7B	20.4	29.1	28.9	24.5	34.1	23.5	26.6	12.1	23.0	32.9	**28.3**	26.2
VST-SFT-7B	36.6	29.1	28.9	36.2	30.6	32.1	31.3	**48.5**	**32.4**	31.6	19.2	30.7
SenseNova-SI-1.1-7B	43.2	45.7	36.6	25.7	27.4	33.8	32.4	29.8	19.6	33.5	16.2	29.7
SCOPE Qwen2.5-VL-7B	48.4	44.2	54.2	**37.2**	28.2	30.9	29.7	30.3	24.3	31.6	21.7	33.6
SCOPE Qwen3-VL-8B	**50.5**	**57.0**	**62.7**	36.2	37.6	32.1	**40.6**	30.3	25.7	27.6	19.7	**36.5**

MMSI-Bench. For SCOPE-8B, Table 7 shows the largest gains in camera-centered positional reasoning. Relative to Qwen3-VL-8B, SCOPE-8B improves camera–camera, camera–object, and camera–region accuracy by 22.5, 30.3, and 30.2 points, respectively, achieving the highest score in all three columns. It also leads on measurement (40.6) and overall accuracy (36.5). On the same Qwen2.5-VL-7B backbone, SCOPE-7B outperforms SenseNova-SI-1.1-7B by 3.9 points overall (33.6 vs. 29.7). These gains span camera-centered and inter-object positional reasoning.

Table 8: **Results by subtask on ViewSpatial-Bench.** Accuracy (%). Rel. Dir., Obj. Ori., and Sce. Sim. denote relative direction, object orientation, and scene simulation. Bold marks the best result per column, and dashes mark unavailable scores.

Model	Camera (Ego)		Person (Allo)			Overall
	Cam-Rel. Dir.	Cam-Obj. Ori.	Per-Rel. Dir.	Per-Obj. Ori.	Per-Sce. Sim.	
Qwen2.5-VL-7B	47.7	31.9	35.0	39.3	28.5	37.9
Qwen3-VL-8B	53.9	30.1	38.1	44.7	28.1	40.8
InternVL3.5-8B	48.8	20.9	34.3	52.3	30.9	38.9
SpaceR-SFT-7B	46.0	31.2	36.3	38.3	38.4	39.2
ViLaSR-7B	46.8	25.3	32.7	36.0	26.4	35.2
SpatialReasoner-7B	43.6	22.4	46.0	50.0	33.6	39.4
SpatialLadder-3B	42.8	20.6	31.6	**68.3**	38.4	40.9
Cambrian-S-7B	51.4	21.1	39.1	42.5	25.7	37.8
Spatial-SSRL-7B	47.0	33.0	37.4	43.1	26.8	38.6
VST-SFT-7B	50.5	**37.4**	**48.4**	52.3	**62.3**	**50.5**
SenseNova-SI-1.1-7B	45.3	26.2	40.6	53.9	47.7	43.3
SCOPE Qwen2.5-VL-7B	51.3	28.7	35.5	34.5	38.5	39.6
SCOPE Qwen3-VL-8B	**63.5**	28.4	46.6	49.3	49.8	49.8

ViewSpatial-Bench. Table 8 shows substantial gains for the 8B backbone in camera-relative and person-relative direction, improving by 9.6 and 8.5 points over Qwen3-VL-8B, respectively. SCOPE-8B achieves the highest camera-relative direction score (63.5), while scene simulation improves by 21.7 points to 49.8. It exceeds SenseNova-SI-1.1-7B by 6.5 points overall (49.8 vs. 43.3), with higher scores on both direction subtasks and scene simulation. On the shared Qwen2.5-VL-7B backbone, SCOPE-7B scores higher on camera-relative direction and object orientation (51.3/28.7 vs. 45.3/26.2). The direction and scene-simulation gains are consistent with improved reasoning across actual and imagined viewpoints.

Table 9: **Results by subtask on ReVSI.** Scores (%) are mean relative accuracy for the numerical subtasks and exact-match accuracy for the multiple-choice ones. Avg. is the aggregate benchmark score. Bold marks the best result per column.

Model	Numerical Question			Multiple-Choice Question				Avg.
	Obj. Count	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	
Qwen2.5-VL-7B	19.2	6.0	35.8	35.7	29.0	32.1	28.3	26.6
Qwen3-VL-8B	39.3	48.8	66.5	36.0	48.6	36.7	34.5	44.4
InternVL3.5-8B	41.3	45.6	61.4	48.2	40.3	37.2	43.0	45.3
SpaceR-SFT-7B	18.0	13.1	41.7	25.2	31.2	35.9	35.7	28.7
ViLaSR-7B	35.2	44.5	60.5	18.6	32.0	35.3	40.3	38.1
SpatialReasoner-7B	27.5	37.6	46.4	26.2	30.6	25.9	37.6	33.1
SpatialLadder-3B	24.1	43.4	53.6	30.6	29.7	36.4	37.6	36.5
Cambrian-S-7B	45.5	56.7	63.3	45.4	36.0	**44.1**	33.3	46.3
Spatial-SSRL-7B	30.1	9.4	32.9	35.2	33.2	30.7	32.2	29.1
VST-SFT-7B	29.6	36.7	61.9	42.2	35.9	33.1	29.5	38.4
SenseNova-SI-1.1-7B	41.9	35.2	59.9	45.7	33.9	32.8	31.8	40.3
SCOPE Qwen2.5-VL-7B	**45.9**	41.6	60.8	47.2	41.6	36.7	**44.6**	45.5
SCOPE Qwen3-VL-8B	44.7	**57.0**	**66.9**	**52.4**	**54.5**	39.3	37.6	**50.4**

ReVSI. Table 9 shows improvements across all subtasks for both backbones. SCOPE-8B achieves the highest scores on absolute distance, object size, room size, and relative distance, reaching 50.4 overall. Relative to Qwen3-VL-8B, absolute distance and room size improve by 8.2 and 16.4 points, respectively. SCOPE-7B leads on object counting (45.9) and route planning (44.6). With the same Qwen2.5-VL-7B backbone, it surpasses SenseNova-SI-1.1-7B on every subtask and by 5.2 points overall (45.5 vs. 40.3), including a 12.8-point margin on route planning. These improvements span both metric estimation and relational reasoning in video.

A.4 Direct Evaluation of 3D Grounding

Evaluation protocol. We evaluate SCOPE and its original backbones on 4,309 RGB image–object queries from the public test split of SUN RGB-D (Song et al., 2015). Each query names a category that occurs once in the image's 3D annotations; the corresponding box's geometric center is the ground-truth location. We apply the inverse of the supplied tilt rotation and convert to camera coordinates (x right, y down, and z forward), then obtain spherical coordinates using Eq. 1.

Each model receives only the RGB image and target category and returns numerical (r, θ, φ): distance in meters and angles in degrees. All four models use the same image–object pairs and prompt, with greedy decoding, a 64-token output limit, and a 262,144-pixel input limit. The prompt is: "In this image, locate the {`target_label`} relative to the camera. Give spherical coordinates in the order (r, theta, phi): distance in meters, azimuth in degrees (right positive), and elevation in degrees (up positive). Return only three numbers in parentheses."

Metrics. We invert Eq. 1 and measure the 3D Euclidean center error $e_i = \|\hat{\mathbf{c}}_i - \mathbf{c}_i\|_2$. Mean and median errors use predictions containing three finite coordinates, without clipping. Accuracy at $e_i < 0.5$ m and $e_i < 1$ m is computed over all 4,309 queries, counting unparseable outputs as failures.

Table 10: **Object-center localization results.** Mean and median errors are in meters; threshold accuracies are percentages. Bold marks the better result within each backbone pair.

Model	Mean ↓	Median ↓	< 0.5 m ↑	< 1 m ↑
Qwen2.5-VL-7B	10.14	2.06	5.80	20.10
SCOPE-7B	**0.74**	**0.55**	**44.44**	**78.49**
Qwen3-VL-8B	1.32	1.19	3.34	32.33
SCOPE-8B	**0.51**	**0.36**	**65.86**	**89.58**

Results. SCOPE improves all four localization metrics for both backbones (Table 10). Accuracy within 1 m increases by 58.39 and 57.25 percentage points for 7B and 8B, respectively. The corresponding 95% confidence intervals from 2,000 paired bootstrap resamples are $[56.72, 59.97]$ and $[55.70, 58.69]$. The selected examples in Fig. 7 illustrate corrections to both image projections and radial distance.

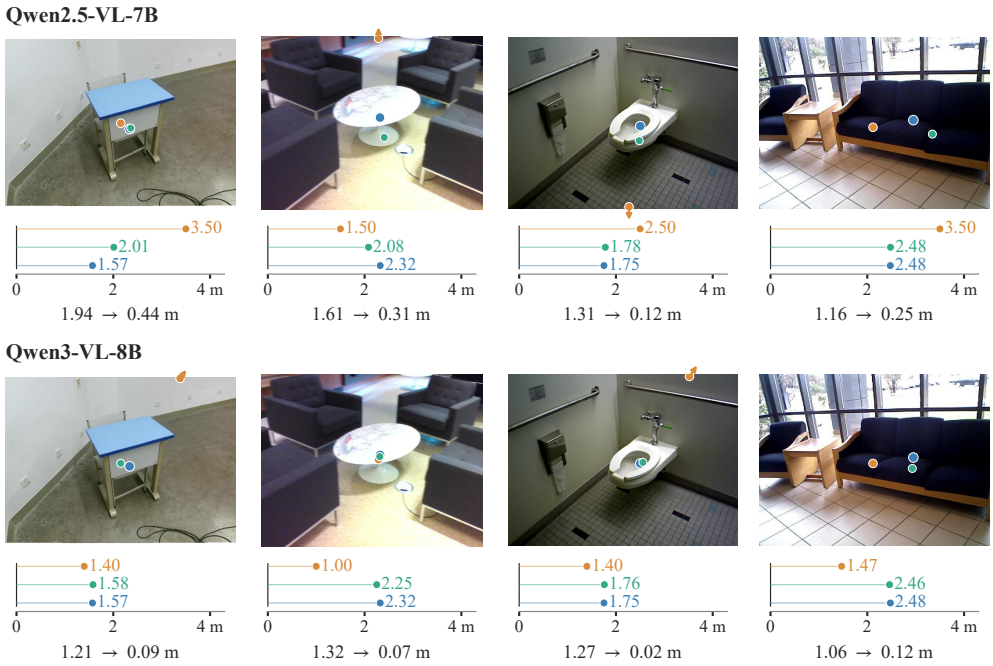


Figure 7: **3D localization examples.** Image dots mark Base Model (●), SCOPE (●), and Ground Truth (●). Distance strips show radial distance in the same order from top to bottom; numbers below compare 3D center errors (Base → SCOPE), all in meters. Orange boundary dots and outward arrows indicate off-image baseline projections; the dots are boundary indicators, not predicted locations.

A.5 QUALITATIVE EXAMPLES OF SPHEREQA

This section shows four SPHEREQA samples, one per requested quantity: distance, azimuth, elevation, and the full coordinate triplet. They cover all three input modalities, with the single image carrying both an egocentric and an allocentric question. In the allocentric questions, the **anchor** the observer stands at, the **reference** it faces, and the **target** to be located are colour-coded.

Single image: egocentric and allocentric



Q: In this image, locate the **refrigerator** relative to the camera. Give the distance r. Answer in meters.
A: `3.23`

Q: In this image, stand at the **oven**, face the **table**, and locate the **refrigerator**. Give the azimuth theta; right is positive and left is negative. Answer in degrees.
A: `−55.28`

Multiple images: allocentric




Q: Using all the ordered images, imagine standing at the **bathtub** and looking toward the **mirror**; locate the **toilet**. Give the elevation phi; up is positive and down is negative. Answer in degrees.
A: `−16.72`

Video: allocentric



time ▶

Q: Use the entire video. Stand at the **oven**, face the **sink**, and locate the **washing machine**. Give spherical coordinates in the order (r, theta, phi): distance, azimuth, elevation. Use meters for r and degrees for both angles.
A: `(1.74, -19.85, -10.57)`

A.6 Prompts of the Representation Study

This section gives the prompt templates of the study in §3. All five conditions share the template in the first box below, where angle brackets mark slots filled at run time, and differ only in the grounding block prepended to the question: the image-only baseline leaves it empty, while the four grounded conditions use the blocks that follow. A block is generated per image, carrying a header `Image` i`:` for multi-image questions, and lists one line per referenced entity whose grounding succeeded, with `<name>` the entity's referring expression and the remaining placeholders its estimated values, rounded to two decimals. The four blocks follow one schema, stating what is estimated, in which frame, in which units and sign conventions, and in which layout, and say nothing about how the values should be interpreted or compared, so that the representation is the only factor varying across conditions. All runs use greedy decoding (temperature 0), and answers are parsed from the backtick-delimited letter.

For the depth condition, d denotes the median camera-space z coordinate of the object point cloud, measured along the optical axis (z-depth), as defined in §3.

Prompt template shared across all conditions

You are a helpful assistant.

`<Image 1>, <Image 2>, ...`
`<Grounding Block>`
`<Question>`
Answer with the single correct option's letter wrapped in backticks, e.g. `` `A` ``.

Condition: depth

Estimated z-depth of referenced objects in the camera frame (meters; depth along the optical axis):
- `<name>: <d>`
- `...`

Condition: 3D box

Estimated egocentric 3D axis-aligned bounding boxes of referenced objects in the camera frame (x, y, z in meters; x right positive, y down positive, z forward positive), as [x_min, y_min, z_min, x_max, y_max, z_max]:
- `<name>: [<x_min>, <y_min>, <z_min>, <x_max>, <y_max>, <z_max>]`
- `...`

Condition: Cartesian coordinates

Estimated egocentric 3D positions of referenced objects in the camera frame (x, y, z in meters; x right positive, y down positive, z forward positive), as (x, y, z):
- `<name>: (<x>, <y>, <z>)`
- `...`

Condition: spherical coordinates

Estimated egocentric 3D positions of referenced objects in the camera frame (r in meters, theta and phi in degrees; theta azimuth right positive, phi elevation up positive), as (r, theta, phi):
- `<name>: (<r>, <theta>, <phi>)`
- `...`