GEOMFUNNEL: COORDINATE-ANCHORED SPARSE VOXEL FUNNELING FOR METRIC-AWARE 3D-VLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern 3D vision-language models increasingly compress rich 3D scene evidence into compact representations for efficient reasoning, inevitably losing some information in the process. Our design addresses the risk that semantic scene understanding and relative spatial relationships can remain accessible after compression while the fine-grained metric geometry required for precise physical measurement becomes harder to recover. To address this gap, we introduce GeomFunnel, which constructs a coordinate-grounded sparse physical field that anchors multi-view scene evidence in a shared metric 3D space and preserves it as an explicit physical substrate. Specifically, its representation-decoupled reasoning uses compact tokens for semantic and relative spatial reasoning while retaining direct access to physical geometry for queries that require precise metric quantities. Within the metric pathway, language-guided physical grounding connects the referred scene elements to their relevant physical supports, allowing the requested quantities to be computed from explicit geometry rather than recovered from compressed tokens. GeomFunnel achieves a new state-of-the-art average score on VSI-Bench while remaining competitive on general 3D vision-language benchmarks.

1 INTRODUCTION

The rapid development of vision-language models (VLMs) and embodied AI has made spatial understanding increasingly important (Hong et al., 2023; Chen et al., 2024a; Huang et al., 2024; Fu et al., 2025). An embodied agent must understand not only what objects are, but also where they are and how they are arranged in physical space (Man et al., 2024; Chen et al., 2026).

Early 3D vision-language methods addressed this need by constructing or directly operating on explicit 3D scene representations (Peng et al., 2023; Zhu et al., 2023). Although these representations preserve rich geometric structure, processing detailed 3D evidence directly can be costly for language-oriented reasoning.

Recent methods improve efficiency by projecting, tokenizing, selecting, or compressing scene evidence (Zhu et al., 2025; Zheng et al., 2025b; Guo et al., 2025; Baek & Kong, 2026; Huang et al., 2025a). Parallel efforts strengthen spatial inference through grounding, structured reasoning, chain-of-thought supervision, and auxiliary geometry (Chen et al., 2024c; Cheng et al., 2024; Chen et al., 2026; Li et al., 2026b; Gao et al., 2026; Zhang et al., 2026b). As illustrated by the existing-model branch in Fig. 1, compact tokens support both qualitative scene reasoning (QSR), which includes semantic understanding, relative spatial relations, and scene layout, and metric geometric reasoning (MGR), which requires precise physical quantities such as distances and object dimensions. However, compressing scene evidence into these tokens inevitably loses some information in the process.

The retention diagnostic in Fig. 2 illustrates the distinction between changes in QSR, MGR, and metric-structure recoverability under compression. The plotted model values show larger relative degradation for the metric quantities; Appendix J details the plot construction and identifies the protocol and uncertainty information still required for empirical interpretation. This distinction motivates a design that preserves explicit physical geometry alongside a compact interface for QSR, so precise metric computation need not depend on the recoverability of geometry from compressed tokens.

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.

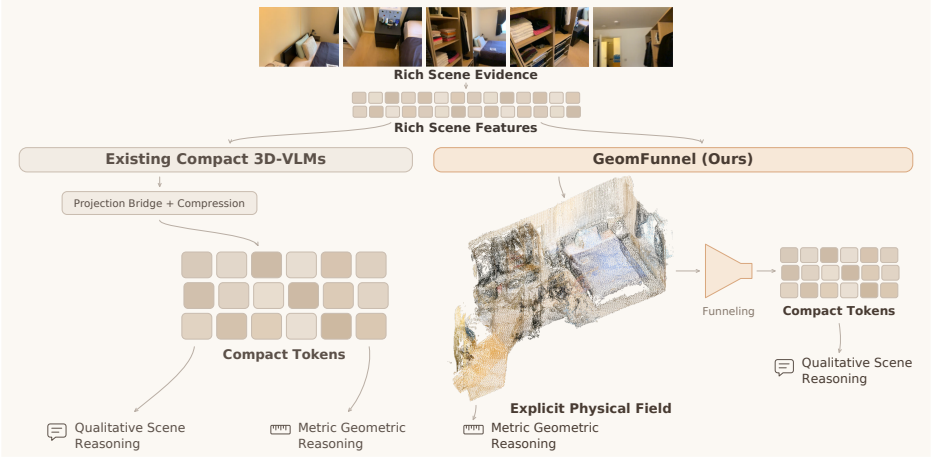

Figure 1: Existing compact 3D-VLMs compress rich scene evidence into compact tokens for both QSR and MGR. GeomFunnel instead retains an explicit physical field for MGR while funneling scene evidence into compact tokens for QSR, preserving direct access to metric geometry alongside efficient compact reasoning.

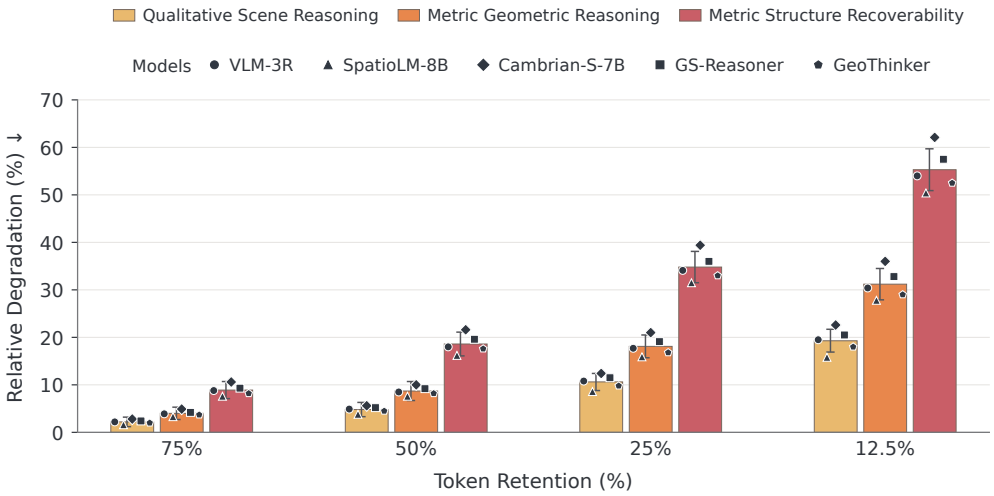

Figure 2: Representation-retention diagnostic for five existing models. Bars are five-model means and markers denote individual model values. Lower relative degradation indicates better preservation. Whiskers reproduce the supplied predicted intervals, not verified empirical confidence intervals; see Appendix J.

To address this gap, we introduce GeomFunnel, which preserves explicit physical geometry alongside a compact reasoning interface. Its coordinate-grounded sparse physical field anchors multi-view evidence in a shared 3D space and aligns the reconstructed geometry to metric scale. Representation-decoupled reasoning then assigns compact tokens to QSR and direct physical-field access to MGR. Within the Metric Geometry Track, language-guided physical grounding identifies the supports of referred objects or regions for deterministic metric computation. This design retains efficient compact reasoning without recovering precise physical quantities from compressed tokens.

Our main contributions are summarized as follows.

- We propose GeomFunnel, a coordinate-grounded 3D vision-language framework that preserves explicit physical geometry alongside a compact reasoning interface. Its Coordinate-Grounded Sparse Physical Field preserves metric scene structure before representation compression. This provides direct access to physical geometry for precise measurement while retaining efficient compact reasoning.

- We introduce Representation-Decoupled Reasoning, which separates the reasoning pathways for Qualitative Scene Reasoning and Metric Geometric Reasoning. It assigns the two reasoning types to distinct information pathways. This allows each to access the scene information best suited to its requirements.
- We present Language-Guided Physical Grounding, which connects language understanding to query-relevant physical evidence. It links query semantics to the corresponding physical supports in the scene. This enables metric quantities to be computed from the relevant physical geometry.
- GeomFunnel achieves state-of-the-art performance on VSI-Bench while remaining competitive on general 3D vision-language benchmarks.

2 Related Work

2.1 3D Vision-Language Representations

3D vision-language models encode scene evidence through explicit 3D structures, object- or voxel-level features, and increasingly compact language-oriented representations (Hong et al., 2023; Zhu et al., 2023; Chen et al., 2024b; Fu et al., 2025; Deng et al., 2025; Yu et al., 2025; Zhu et al., 2024). Recent methods further improve efficiency through feature projection, tokenization, selection, or compression while retaining spatial cues for downstream reasoning (Man et al., 2024; Zhu et al., 2025; Zheng et al., 2025b; Guo et al., 2025; Baek & Kong, 2026; Huang et al., 2025a; Li et al., 2026c; Zheng et al., 2025a). However, as scene evidence becomes more compact, fine-grained metric structure becomes increasingly implicit, making precise physical quantities difficult to recover from tokens alone. To address this limitation, GeomFunnel preserves an explicit metric physical field alongside a compact reasoning interface and decouples the information pathways for QSR and MGR. This design retains efficient token-based reasoning for QSR while giving MGR direct access to physical geometry for deterministic metric computation.

2.2 Geometry-Grounded Spatial Reasoning

A complementary line of work strengthens spatial reasoning through referential grounding, structured reasoning, chain-of-thought supervision, or explicit geometric structures (Chen et al., 2024c; Cheng et al., 2024; Chen et al., 2026; Li et al., 2026b; Gao et al., 2026; Zhang et al., 2026b; Li et al., 2026a). These methods improve how language connects to scene structure, but do not by themselves ensure that the physical geometry required for precise metric computation remains accessible after compression. To address this limitation, GeomFunnel introduces language-guided physical grounding, which maps query semantics to the relevant physical supports in the preserved field. This separates semantic identification from deterministic metric computation, allowing MGR to derive physical quantities directly from grounded geometric evidence.

3 Method

3.1 GeomFunnel Framework

Figure 3 summarizes GeomFunnel. Given sampled RGB frames $\mathcal{V} = \{I_t\}_{t=1}^{T}$ and a query q, we construct a coordinate-grounded sparse physical field $\mathcal{F} = \{(\mathbf{p}_i, \mathbf{f}_i)\}_{i=1}^{M}$, where $\mathbf{p}_i$ is a retained metric 3D coordinate and $\mathbf{f}_i$ is its fused visual evidence. Representation-decoupled reasoning routes QSR through compact tokens and MGR through direct field access. For MGR, language-guided physical grounding identifies query-relevant supports for deterministic geometric computation.

3.2 Coordinate-Grounded Sparse Physical Field

GeomFunnel first converts multi-view observations into an explicit physical field with consistent metric coordinates. We use a frozen VGGT-Ω (Wang et al., 2025b; 2026) to obtain self-consistent multi-view depth D_t^{Ω}, camera intrinsics $\mathbf{K}_t$, camera poses $(\mathbf{R}_t^{\Omega}, \mathbf{t}_t^{\Omega})$, and visual feature maps $\mathbf{X}_t$. While this reconstruction preserves coherent cross-view geometry, its global scene scale is not guaranteed to be metric. We therefore use UniDepthV2 (Piccinelli et al., 2026) only on a sparse set of

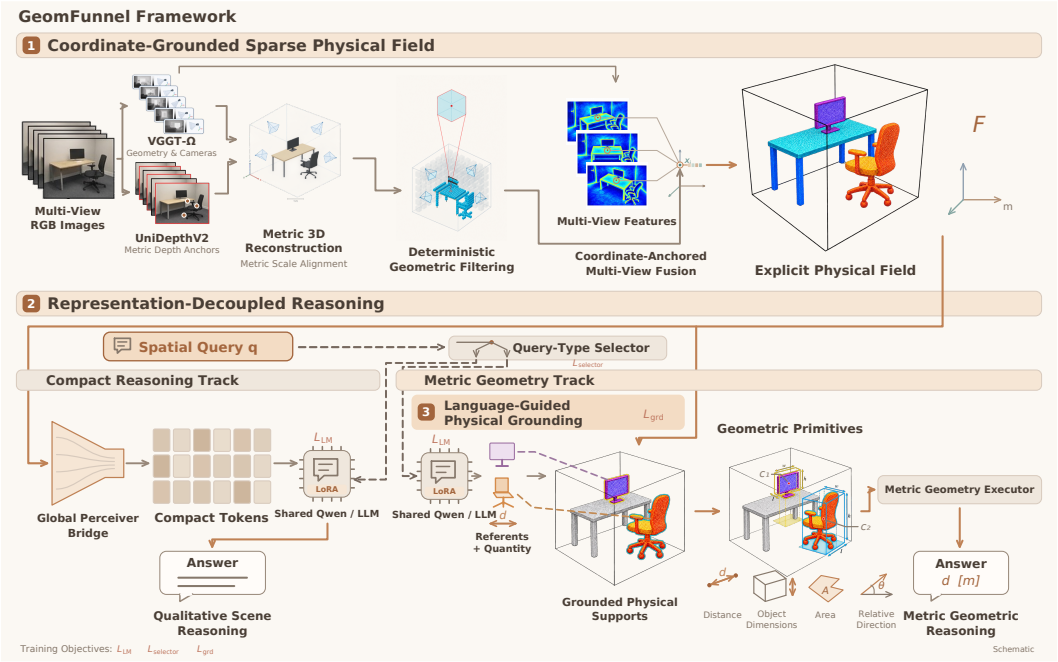

Figure 3: Multi-view RGB frames are fused into a Coordinate-Grounded Sparse Physical Field. Representation-Decoupled Reasoning routes queries to either the Compact Reasoning Track for Qualitative Scene Reasoning or the Metric Geometry Track for Metric Geometric Reasoning. For metric queries, language identifies what to measure, Language-Guided Physical Grounding locates the corresponding physical supports, and the Metric Geometry Executor computes the answer from explicit geometry.

anchor frames $\mathcal{A}$ to provide metric depth references. Directly replacing the reconstructed depth with independently predicted metric depth would break the original multi-view consistency. Instead, we estimate a single sequence-level scale from the two depth predictions using a robust median of their log-depth ratios,

$$\hat{s} = \exp\left(\text{median}_{t\in\mathcal{A}}\, \text{median}_{\mathbf{u}\in\Omega_t}\left[\log D_t^U(\mathbf{u}) - \log D_t^\Omega(\mathbf{u})\right]\right), \tag{1}$$

where Ω_t denotes valid high-confidence pixels shared by the two depth estimates. The resulting scale $\hat{s}$ is applied to the reconstructed depth and camera translations, while the camera rotations and intrinsics remain unchanged. This restores physical scale while preserving the original multi-view geometry. Details of anchor-frame selection and valid-pixel construction are provided in Appendix B.

After metric alignment, we retain only geometrically supported 3D locations rather than learning a new occupancy or geometry predictor. Starting from candidate locations in the reconstructed scene, we deterministically filter them according to reconstruction confidence, visibility, depth consistency, and multi-view geometric agreement. The resulting sparse coordinate set excludes unsupported regions while keeping every retained location tied to reliable reconstructed geometry. The voxel spacing is defined in meters rather than by normalizing each scene into a fixed unit volume, so distances and extents retain a common physical interpretation across scenes. Details of the filtering criteria and implementation are provided in Appendix B.

We then fuse multi-view visual evidence directly at these retained metric coordinates. For each $\mathbf{p}_i \in \mathcal{P}$, we project the coordinate onto the image planes of its valid views, sample the corresponding visual feature $\mathbf{x}_{it}$ from $\mathbf{X}_t$, and aggregate the observations at the shared 3D location. Let $\mathcal{T}_i$ denote the set of valid views for $\mathbf{p}_i$, and let w_{it} reflect reconstruction confidence and geometric consistency. The fused feature is

$$\mathbf{f}_i = \frac{\sum_{t\in\mathcal{T}_i} w_{it}\mathbf{x}_{it}}{\sum_{t\in\mathcal{T}_i} w_{it}}. \tag{2}$$

The resulting Coordinate-Grounded Sparse Physical Field $\mathcal{F} = \{(\mathbf{p}_i, \mathbf{f}_i)\}_{i=1}^{M}$ keeps each fused visual feature explicitly anchored to a metric 3D coordinate. It therefore preserves cross-view visual evidence together with directly accessible physical geometry for subsequent reasoning.

3.3 Representation-Decoupled Reasoning

QSR and MGR require different forms of scene access. GeomFunnel therefore introduces Representation-Decoupled Reasoning rather than forcing both reasoning types through the same compressed interface. A lightweight Query-Type Selector, implemented as a small MLP over the query representation $\mathbf{h}_q$, routes each query to either the Compact Reasoning Track or the Metric Geometry Track. The decision depends on the required scene access rather than the answer format: counting can produce a number without requiring a metric length or area. The selector receives the query, not a benchmark task identifier, at inference time; its architecture and training supervision are detailed in Appendix E.

For the Compact Reasoning Track, a Global Perceiver Bridge converts the variable-length physical field $\mathcal{F}$ into a fixed-length set of compact tokens. Writing $\mathcal{B}_\theta$ for the bridge, the compact representation is $\mathbf{Z} = \mathcal{B}_\theta(\mathcal{F}) = \{\mathbf{z}_\ell\}_{\ell=1}^{L}$. These tokens provide an efficient interface for the language model to perform QSR while keeping the scene-token budget independent of the number of retained physical locations. The field remains available separately, constructing $\mathbf{Z}$ does not replace or discard $\mathcal{F}$. Architectural details of the bridge and compact-token configuration are provided in Appendix A.

The Metric Geometry Track follows a different information pathway. It does not first compress the physical field into the compact token interface, but instead retains direct access to $\mathcal{F}$ and its explicit metric coordinates for MGR. The mechanism that connects a metric query to the relevant physical evidence is described next.

3.4 Language-Guided Physical Grounding

Once a query enters the Metric Geometry Track, GeomFunnel first uses the language model to determine what should be measured and which scene elements are involved. The LoRA-adapted Qwen model produces a lightweight structured interpretation that identifies the referring phrases and the geometric quantity requested by the query. This interpretation determines which grounded physical supports and geometric primitives are required for subsequent metric computation. The structured interpretation is supervised through the language-modeling objective described in Sec. 3.5.

For each referring phrase e_r, we reuse its language representation $\mathbf{h}_r$ to identify the corresponding physical support in the Coordinate-Grounded Sparse Physical Field $\mathcal{F}$. A shared grounding decoder relates the language representation to the voxel features through cross-attention and predicts a soft support score for every retained physical location,

$$\tilde{\mathbf{h}}_r = \mathrm{CrossAttn}\left(\mathbf{h}_r, \{\mathbf{f}_i\}_{i=1}^{M}\right), \qquad g_i^{(r)} = \sigma\left((W_m \tilde{\mathbf{h}}_r)^\top (W_k \mathbf{f}_i)\right). \tag{3}$$

The scores $\{g_i^{(r)}\}_{i=1}^{M}$ define a soft physical support for the referred object or region while retaining the original metric coordinates $\mathbf{p}_i$. A lightweight deterministic refinement removes spatially inconsistent responses without modifying the underlying scene geometry. The refinement procedure is detailed in Appendix D.

The grounded supports are then converted into reusable geometric primitives that describe the relevant physical structure. These include object centroids, oriented bounding boxes and their extents, and floor or room footprints. For example, a refined support $\mathcal{S}_q^{(r)} = \{(\mathbf{p}_i, \omega_i^{(r)})\}_{i \in C_r^*}$ gives the centroid $\mathbf{c}_r = \sum_{i \in C_r^*} \omega_i^{(r)} \mathbf{p}_i$, where the retained grounding weights sum to one. Weighted principal axes provide object extents, and floor-aligned support provides a room footprint. The primitive must match the requested quantity: the VSI-Bench absolute-distance operator measures the closest separation between object supports, rather than centroid-to-centroid distance. Relative-direction relations are an additional geometric capability, while the evaluated VSI-Bench relative-direction category uses the Compact Reasoning Track (Appendix E).

For object supports C_a^* and C_b^*, voxel side length δ, and robust object extents $e_{r,k}$, the basic measurements are

$$d(a,b) = \min_{i \in C_a^*,\, j \in C_b^*} \left\| \max(|\mathbf{p}_i - \mathbf{p}_j| - \delta, 0) \right\|_2 , \tag{4}$$

$$s_r = \max_{k \in \{1,2,3\}} e_{r,k}, \qquad A_{\text{room}} = N_f r_f^2. \tag{5}$$

The absolute value and maximum in Eq. 4 act element-wise, accounting for the finite extent of the axis-aligned voxels when measuring their closest separation. Here N_f is the number of retained floor cells at metric grid spacing r_f. Object-level measurements use language-grounded supports, whereas room area uses the scene-level floor geometry and does not require an object mask. The same distance and extent primitives can be reused across multiple operations without reconstructing the scene or predicting new coordinates for each query.

The computed value is not passed back to the language model for numerical estimation. Because the primitives are reusable, the executor can compose measurements through comparison, differences, ratios, and selection followed by another measurement; it is not a separate learned regressor for each task. Thus, language determines what is being asked, grounding locates the relevant physical supports, and explicit geometry provides the final value. Details of primitive construction and deterministic geometric computation are provided in Appendix D.

3.5 Training Objectives

GeomFunnel is trained in two stages. Stage 1 optimizes the visual projection and Global Perceiver Bridge with the language-modeling objective $\mathcal{L}_{\text{LM}}$, keeping the base language model frozen. Stage 2 enables LoRA adaptation of Qwen and trains the Query-Type Selector and Language-Guided Physical Grounding. Language modeling supervises generation and structured query interpretation; $\mathcal{L}_{\text{selector}}$ and $\mathcal{L}_{\text{grd}}$ supervise routing and grounding, respectively. The objective is

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_{\text{selector}} \mathcal{L}_{\text{selector}} + \lambda_{\text{grd}} \mathcal{L}_{\text{grd}}. \tag{6}$$

The grounding term combines class-balanced binary cross-entropy and Dice overlap on the valid support labels:

$$\mathcal{L}_{\text{grd}}^{(r)} = w_{d(r)} \left(\mathcal{L}_{\text{bal-BCE}}^{(r)} + \lambda_{\text{dice}} \mathcal{L}_{\text{Dice}}^{(r)} \right), \tag{7}$$

where $\mathcal{L}_{\text{grd}}^{(r)}$ is the contribution of entity r, and $w_{d(r)}$ down-weights weak box-derived targets; the training objective aggregates these contributions over supervised entities. Ambiguous support labels are excluded from both terms. Geometry reconstruction, metric scale alignment, physical-field filtering, geometric primitive construction, and final metric computation are either fixed or deterministic throughout training. No metric-answer regression loss is used. Optimization settings, grounding supervision, and routing details are provided in Appendices A, C, and E, respectively.

4 Experiments

We evaluate MGR, broader 3D understanding, inference efficiency, and the contributions of the core design components.

4.1 Datasets and Benchmarks

We train GeomFunnel on ScanNet (Dai et al., 2017), ScanNet++ (Yeshwanth et al., 2023), ARKitScenes (Dehghan et al., 2021), and Sr3D (Achlioptas et al., 2020). ScanNet and ScanNet++ provide instance-level 3D annotations for supervising physical supports, while ARKitScenes provides oriented 3D boxes that are converted into weaker support-supervision signals. Sr3D provides referring expressions linked to 3D object instances, strengthening Language-Guided Physical Grounding and qualitative spatial understanding. Ground-truth geometry is used only to construct training supervision and is never provided at inference time. Details of the supervision construction are provided in Appendix C.

VSI-Bench (Yang et al., 2025) is our primary benchmark for qualitative and metric spatial reasoning. We additionally evaluate general 3D question answering on ScanQA (Azuma et al., 2022), situated spatial understanding on SQA3D (Ma et al., 2023), and object-centric captioning on Scan2Cap (Chen et al., 2021). All benchmarks follow their official evaluation protocols and standard inference inputs.

4.2 Implementation Details

Frozen VGGT-Ω (Wang et al., 2026) and UniDepthV2 (Piccinelli et al., 2026) construct the metric-aligned physical field using a fixed voxel size and 1024-dimensional fused features. The Global Perceiver Bridge produces 256 compact tokens, which are projected into the embedding space of Qwen3-VL-8B-Instruct (Bai et al., 2025); its native vision encoder is unused. We follow the two-stage training in Sec. 3.5, enabling LoRA (Hu et al., 2022), routing, and grounding in Stage 2. Appendix A details architecture, optimization, and hyperparameters.

4.3 Main Results

As shown in Table 1, GeomFunnel shows its clearest advantage on tasks that depend on precise physical scene geometry, while remaining competitive on broader spatial reasoning tasks. Compared with methods that primarily strengthen compact scene representations or enhance downstream reasoning through grounding and structured geometry (Yang et al., 2026; Fan et al., 2026; Cai et al., 2026; Zhao et al., 2026; Wu et al., 2026; Guan et al., 2026; Liu et al., 2026a; Chen et al., 2026; Li et al., 2026a), GeomFunnel preserves a directly accessible physical field throughout reasoning. The resulting performance pattern suggests that the benefit does not come from uniformly stronger reasoning, but from providing MGR with explicit physical evidence while retaining the compact pathway for QSR.

Table 1: **Quantitative comparison on VSI-Bench.** Bold/dark pink, underlined/medium pink, and light pink indicate the best, second-best, and third-best results, respectively. Higher is better.

Method	Numerical				Multiple Choice				Avg. ↑
	Obj. Count ↑	Abs. Dist. ↑	Obj. Size ↑	Room Size ↑	Rel. Dist. ↑	Rel. Dir. ↑	Route Plan ↑	Appear. Order ↑	
GS-Reasoner (ICLR'26)	69.1	61.9	70.0	65.7	65.4	88.9	44.3	52.3	64.7
Cambrian-S-7B (ICLR'26)	73.2	50.5	74.9	72.2	71.1	76.2	41.8	80.1	67.5
VLM-3R-7B (CVPR'26)	70.2	49.4	69.2	67.1	65.4	80.5	45.4	40.1	60.9
SenseNova-SI-8B (CVPR'26)	75.7	53.2	72.8	67.5	77.0	72.7	48.5	82.6	68.7
SpaceMind (CVPR'26)	73.3	61.4	77.3	74.2	67.2	88.4	44.3	70.6	69.6
GeoThinker (ICML'26)	74.6	62.8	76.9	74.9	75.9	90.2	48.4	77.5	72.6
SpatioLM-8B (ICML'26)	**81.9**	55.7	75.2	71.5	78.2	74.4	47.9	**87.8**	71.6
VST-SFT-7B (ECCV'26)	72.0	44.4	74.3	68.3	59.7	55.8	44.9	65.2	60.6
Spatial-TTT-2B (ECCV'26)	70.8	47.8	71.7	65.9	61.8	73.0	47.4	77.0	64.4
GeomFunnel (Ours)	78.3	**69.4**	**78.4**	**76.1**	**78.8**	**90.7**	**49.1**	81.4	**75.3**

Table 2 reports object-centric captioning results, followed by the ScanQA and SQA3D results in Table 3. ScanQA tests object semantics, attributes, and spatial relations, while SQA3D adds agent-centric context. The results indicate that the Compact Reasoning Track retains broader scene understanding alongside the dedicated metric pathway.

Figure 4 compares absolute-distance reasoning. GS-Reasoner and GeoThinker capture the relevant scene elements but infer the final quantity from intermediate representations. GeomFunnel uses language-guided physical grounding to associate the referred-to objects with their supports in the explicit physical field. In this example, direct access to metric geometry yields a distance estimate closer to the ground truth, illustrating the benefit of preserving physical evidence for MGR. Additional compositional MGR examples are provided in Appendix I (Fig. 5).

Table 2 evaluates object-centric captioning through the Compact Reasoning Track. GeomFunnel remains strong compared with recent video-based, reconstructive, region-prompted, and geometry-aware methods (Zheng et al., 2025b; Wang et al., 2025a; Huang et al., 2025b; Qi et al., 2026; Cheng et al., 2026; Zhang et al., 2026a; Liu et al., 2026b). These results indicate that preserving a dedicated physical pathway for MGR is compatible with effective language-centric scene understanding through compact tokens.

Table 2: **Scan2Cap results.**

Method	B-4 ↑	ROUGE ↑	CIDEr ↑	METEOR ↑
Video-3D LLM (CVPR'25)	42.4	62.3	83.8	28.9
ROSS3D (ICCV'25)	43.4	66.9	81.3	30.3
3DRS (NeurIPS'25)	41.6	62.3	86.1	29.0
GPT4Scene-HDM (ICLR'26)	40.6	59.3	86.3	28.2
SR-3D-8B (ICLR'26)	44.7	67.3	97.9	31.5
GS-Reasoner (ICLR'26)	47.6	69.2	101.0	**32.1**
3D-IDE (CVPR'26)	40.8	62.1	79.0	28.8
PoseAlign-T (CVPR'26)	37.1	57.6	76.1	27.1
GeomFunnel (Ours)	**48.4**	**71.8**	**106.2**	32.0

Table 3: **Quantitative comparison on ScanQA and SQA3D.**

Method	ScanQA				SQA3D
	B-4 ↑	ROUGE ↑	CIDEr ↑	METEOR ↑	EM ↑
ROSS3D (ICCV'25)	17.9	50.7	107.0	20.9	**63.0**
3DRS (NeurIPS'25)	17.2	49.8	104.8	20.5	60.6
GPT4Scene-HDM (ICLR'26)	15.5	46.5	96.3	18.9	59.4
SR-3D-8B (ICLR'26)	18.1	51.2	109.3	21.2	62.2
GS-Reasoner (ICLR'26)	16.2	49.2	102.6	19.8	59.9
3D-IDE (CVPR'26)	17.4	48.8	102.1	20.8	59.2
SpatioLM-8B (ICML'26)	16.3	48.9	101.8	19.8	56.5
VST-SFT-7B (ECCV'26)	0.5	41.9	82.1	9.6	45.7
GeomFunnel (Ours)	**18.4**	**52.8**	**113.2**	**21.5**	62.6

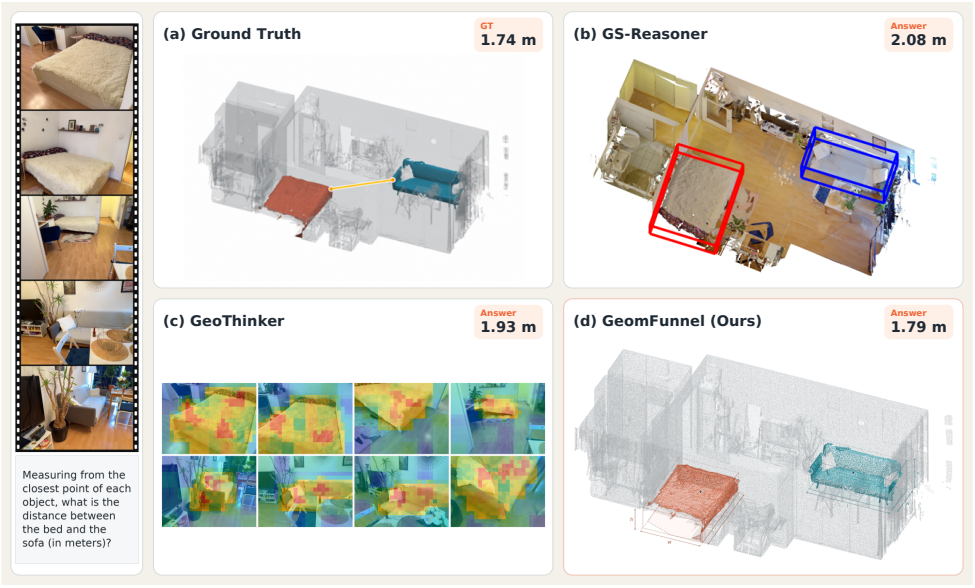

Figure 4: **Qualitative comparison of absolute-distance reasoning.** For this example, the ground-truth distance is 1.74 m; GS-Reasoner, GeoThinker, and GeomFunnel predict 2.08 m, 1.93 m, and 1.79 m, respectively. GeomFunnel grounds the referred-to objects in explicit physical geometry before measurement.

4.4 Efficiency Analysis

We evaluate all methods on a single NVIDIA H200 GPU with a batch size of 1 and 32 input frames. GeomFunnel's latency includes geometry reconstruction, scale alignment, physical-field construction, and reasoning. Despite its additional frozen geometry components, Table 4 shows that its memory use and latency are close to those of the most efficient methods compared. Sparse field access and compact tokens limit downstream computation, while deterministic operators produce

metric answers. Appendix H provides a complexity analysis and an accounting of the geometry components.

Table 4: **Efficiency comparison on VSI-Bench.** All methods are evaluated with a batch size of 1 under the same 32-frame input setting. Inference time denotes end-to-end latency per sample. ↓ indicates lower is better.

Method	Total Params (B) ↓	Peak Memory (GB) ↓	Time (s) ↓
GS-Reasoner (ICLR'26)	**8.0**	24.8	5.6
SpaceMind (CVPR'26)	9.3	26.1	4.8
GeoThinker (ICML'26)	9.0	25.3	4.5
SpatioLM-8B (ICML'26)	8.2	**19.6**	**2.9**
GeomFunnel (Ours)	9.7	20.8	3.2

4.5 Ablation Study

We examine three ablations corresponding to the core design components. Removing the coordinate-grounded physical field eliminates direct metric coordinate access. Removing representation-decoupled reasoning forces MGR through compact tokens. Removing language-guided physical grounding replaces the learned language-to-physical correspondence used to obtain query-relevant supports.

Table 5 shows the largest degradation without the physical field, followed by the reasoning-pathway ablation. Grounding provides a further consistent benefit. These results support preserving metric geometry, providing a dedicated access pathway, and grounding the query in physical supports. Appendix G reports sensitivity analyses for the anchor count, voxel resolution, latent capacity, and routing.

Table 5: **Ablation of the core GeomFunnel designs on MGR.** The variants remove the named high-level design components. Metric Avg. is the average of absolute distance, object size, and room size. ↑ indicates higher is better.

Variant	Abs. Dist. ↑	Obj. Size ↑	Room Size ↑	Metric Avg. ↑
w/o Coordinate-Grounded Sparse Physical Field	49.8	59.6	43.9	51.1
w/o Representation-Decoupled Reasoning	64.8	74.7	72.3	70.6
w/o Language-Guided Physical Grounding	66.8	75.6	75.8	72.7
GeomFunnel	**69.4**	**78.4**	**76.1**	**74.6**

5 Conclusion

We presented GeomFunnel, a 3D vision-language framework that preserves explicit physical geometry alongside a compact language representation. Its Representation-Decoupled Reasoning uses the Compact Reasoning Track for QSR and the Metric Geometry Track for deterministic computation over explicit physical supports. Experiments show clear gains on metric reasoning while retaining strong performance on general 3D vision-language and spatial-understanding tasks. These results support separating language-oriented representation from direct physical access when precise metric reasoning is required. Limitations and scope are discussed in Appendix F.

AI Use Statement

In this work, we used generative AI tools for language polishing and improving clarity and readability. We did not use generative AI tools to originate the research idea, develop the core scientific contributions, or formulate original arguments. We also used generative AI tools to provide feedback on presentation and aspects of the methodology. We have carefully reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. ReferIt3D: Neural listeners for fine-grained 3D object identification in real-world scenes. In *European Conference on Computer Vision*, pp. 422–440, 2020.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. ScanQA: 3D question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19129–19139, 2022.

Changwoo Baek and Kyeongbo Kong. 3DZip: Spatial-aware feature diversity-guided token compression for 3D question answering. In *European Conference on Computer Vision*, 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Zhongang Cai, Ruisi Wang, Chenyang Gu, Fanyi Pu, Junxiang Xu, Yubo Wang, Wanqi Yin, Zhitao Yang, Chen Wei, et al. Scaling spatial intelligence with multimodal foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7879–7890, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024a.

Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. Scan2Cap: Context-aware dense captioning in RGB-D scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3193–3203, 2021.

Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. LL3DA: Visual interactive instruction tuning for omni-3D understanding, reasoning, and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26428–26438, 2024b.

Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Runsen Xu, Ruiyuan Lyu, Dahua Lin, and Jiangmiao Pang. Grounded 3D-LLM with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024c.

Yiming Chen, Zekun Qi, Wenyao Zhang, Xin Jin, Li Zhang, and Peidong Liu. Reasoning in space via grounding in the world. In *International Conference on Learning Representations*, 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 *Advances in Neural Information Processing Systems*, volume 37, pp. 135062–135093, 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 *International Conference on Learning Representations*, 2026.

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.

Afshin Dehghan, Gilad Baruch, Zhuoyuan Chen, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, 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.

Jiajun Deng, Tianyu He, Li Jiang, Tianyu Wang, Feras Dayoub, and Ian Reid. 3D-LLaVA: Towards generalist 3D LMMs with omni superpoint transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3772–3782, 2025.

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*, pp. 31054–31065, 2026.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-LLM: Extending language model for 3D visual reasoning. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 2195–2206, 2025.

Xiangjun Gao, Zhensong Zhang, Dave Zhenyu Chen, Songcen Xu, Long Quan, Eduardo Pérez-Pellitero, and Youngkyoon Jang. Map2Thought: Explicit 3D spatial reasoning via metric cognitive maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 7154–7164, 2026.

Yiran Guan, Liang Yin, Dingkang Liang, Jianzhong Ju, Zhenbo Luo, Jian Luan, Yuliang Liu, and Xiang Bai. Video streaming thinking: VideoLLMs can watch and think simultaneously. In *European Conference on Computer Vision*, pp. 80–99, 2026.

Wenxuan Guo, Xiuwei Xu, Ziwei Wang, Jianjiang Feng, Jie Zhou, and Jiwen Lu. Text-guided sparse voxel pruning for efficient 3D visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3666–3675, 2025.

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, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

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 *International Conference on Machine Learning*, volume 235, pp. 20413–20451, 2024.

Jiangyong Huang, Xiaojian Ma, Xiongkun Linghu, Yue Fan, Junchao He, Wenxin Tan, Qing Li, Song-Chun Zhu, Yixin Chen, Baoxiong Jia, and Siyuan Huang. LEO-VL: Efficient scene representation for scalable 3d vision-language learning. *arXiv preprint arXiv:2506.09935*, 2025a.

Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. 3DRS: MLLMs need 3D-aware representation supervision for scene understanding. In *Advances in Neural Information Processing Systems*, volume 38, pp. 75725–75752, 2025b.

Haoyuan Li, Qihang Cao, Tao Tang, Kun Xiang, Zihan Guo, Jianhua Han, Jiawang Bian, Hang Xu, and Xiaodan Liang. Thinking with geometry: Active geometry integration for spatial reasoning. In *International Conference on Machine Learning*, 2026a.

Jinpeng Li, Haiping Wang, Jiabin Chen, Yuan Liu, Zhen Dong, and Bisheng Yang. SCoT: Teaching 3D-LLMs to think spatially with million-scale CoT annotations. In *International Conference on Learning Representations*, 2026b.

Wenhao Li, Xueying Jiang, Quanhao Qian, Deli Zhao, Ran Xu, Shijian Lu, and Gongjie Zhang. 3D-Aware VLMs with implicit and explicit geometries. In *European Conference on Computer Vision*, pp. 168–185, 2026c.

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*, 2026a.

Quan Liu, Weihao Xuan, Junjue Wang, Naoto Yokoya, Ling Shao, and Shijian Lu. Direction-aware 3D large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9668–9678, 2026b.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. SQA3D: Situated question answering in 3D scenes. In *International Conference on Learning Representations*, 2023.

Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. Situational awareness matters in 3D vision language reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13678–13688, 2024.

Songyou Peng, Kyle Genova, Chiyu Max Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 815–824, 2023.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):2354–2367, 2026.

Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. GPT4Scene: Understand 3D scenes from videos with vision-language models. In *International Conference on Learning Representations*, 2026.

Haochen Wang, Yucheng Zhao, Tiancai Wang, Haoqiang Fan, Xiangyu Zhang, and Zhaoxiang Zhang. Ross3D: Reconstructive visual instruction tuning with 3D-awareness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9275–9286, 2025a.

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*, pp. 5294–5306, 2025b.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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. In *International Conference on Machine Learning*, 2026.

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*, pp. 10632–10643, 2025.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, et al. Cambrian-S: Towards spatial supersensing in video. In *International Conference on Learning Representations*, 2026.

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.

Hanxun Yu, Wentong Li, Song Wang, Junbo Chen, and Jianke Zhu. Inst3D-LMM: Instance-aware 3D scene understanding with multi-modal instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14147–14157, 2025.

Chushan Zhang, Ruihan Lu, Jinguang Tong, Yikai Wang, and Hongdong Li. 3D-IDE: 3D implicit depth emergent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23837–23847, 2026a.

Shihua Zhang, Qiuhong Shen, Shizun Wang, Tianbo Pan, and Xinchao Wang. Make geometry matter for spatial reasoning. In *European Conference on Computer Vision*, pp. 231–249, 2026b.

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*, pp. 16811–16822, 2026.

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. 23356–23382, 2025a.

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*, pp. 8995–9006, 2025b.

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*, pp. 4295–4305, 2025.

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3D-VisTA: Pre-trained transformer for 3D vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2911–2921, 2023.

Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. Unifying 3D vision-language understanding via promptable queries. In *European Conference on Computer Vision*, pp. 188–206, 2024.

A Detailed Implementation and Training Configuration

GeomFunnel takes 32 uniformly sampled RGB frames as input for each scene. We use a frozen VGGT-Ω model for multi-view geometry recovery and UniDepthV2-ViT-L/14 for metric scale estimation. UniDepthV2 is applied to eight uniformly distributed anchor frames in each sequence. The resulting metric scene is discretized using a fixed voxel size of $\delta = 0.05$ m. Candidate voxels are restricted to the valid reconstructed scene extent before Geometry-Guided Active Voxel Filtering and Coordinate-Anchored Multi-View Fusion. The fused voxel feature dimension is 1024.

The Global Perceiver Bridge contains four layers and maintains 256 learned latent queries with a hidden dimension of 1024. Each layer uses eight attention heads. The resulting latent representation $\mathbf{Z} = \{\mathbf{z}_l\}_{l=1}^{256}$ is projected to the embedding space of Qwen3-VL-8B-Instruct. We do not use the native vision encoder of Qwen3-VL. The Language-Guided Physical Grounding decoder contains two cross-attention layers with a hidden dimension of 512 and eight attention heads. Both entity representations and voxel features are projected into this shared grounding space. The Query-Type Selector is implemented as a two-layer MLP with a hidden dimension of 512.

Training follows the two-stage procedure described in Sec. 3.5. Stage 1 trains the visual projection and Global Perceiver Bridge for one epoch using the language-modeling objective. Training continues for two additional epochs in Stage 2. It introduces the grounding decoder and Query-Type Selector, enables LoRA adaptation of Qwen3-VL, and continues to optimize the visual projection and Global Perceiver Bridge with a reduced learning rate. LoRA is applied to the query, key, value, and output projections of the language-model attention layers with a rank of 16, a scaling factor of 32, and a dropout rate of 0.05. The original Qwen3-VL parameters remain frozen.

We use AdamW for optimization. Newly introduced modules are optimized with a learning rate of 1×10^{-4}. During Stage 2, the visual projection and Global Perceiver Bridge use a reduced learning rate of 5×10^{-5}, while the Qwen3-VL LoRA parameters use 1×10^{-5}. We use a weight decay of 0.01, cosine learning-rate decay, a warmup ratio of 0.03, and gradient clipping at 1.0. Training is performed on four NVIDIA H200 GPUs with a per-GPU batch size of 1 and gradient accumulation over four steps, giving an effective batch size of 16.

For Stage 2, we set $\lambda_{\text{grd}} = 1.0$, $\lambda_{\text{selector}} = 0.1$, and $\lambda_{\text{dice}} = 1.0$. Grounding supervision from ScanNet and ScanNet++ is assigned full weight. The weaker box-derived supervision from ARKitScenes is weighted by $w_d = 0.5$. All geometry recovery modules, metric scale alignment, active voxel filtering, support refinement, and deterministic metric operators remain fixed throughout training. We further study the effects of the number of metric anchor frames, voxel resolution, and the number of Perceiver latent tokens in Appendix G.

B Metric Geometry Recovery and Physical Field Construction

GeomFunnel constructs the physical field directly from the sampled RGB sequence. We use a frozen VGGT-Ω (Wang et al., 2026) to recover per-view visual features $\mathbf{X}_t$, relative depth D_t^{Ω}, camera intrinsics $\mathbf{K}_t$, camera rotations $\mathbf{R}_t^{\Omega}$, camera translations $\mathbf{t}_t^{\Omega}$, and reconstruction confidence. VGGT-Ω provides a coherent multi-view reconstruction, but its global scale is not guaranteed to correspond to physical units. UniDepthV2 (Piccinelli et al., 2026) is therefore used only to recover the missing metric scale.

For each sequence, UniDepthV2 is applied to eight uniformly distributed anchor frames $\mathcal{A}$. For each anchor frame t, we construct the valid pixel set Ω_t from pixels with positive depth and reliable predictions from both VGGT-Ω and UniDepthV2. Confidence filtering is performed independently within each frame, and the top 70% confidence observations are retained. The sequence-level scale $\hat{s}$ is then estimated using the robust median formulation in Eq. 1. We estimate only one scale for the entire sequence. Independent per-frame scaling is avoided because it would distort the multi-view consistency established by VGGT-Ω.

After estimating $\hat{s}$, we apply the same scale to both the recovered depth and the camera translations:

$$D_t = \hat{s} D_t^{\Omega}, \qquad \mathbf{t}_t = \hat{s} \mathbf{t}_t^{\Omega}, \qquad \mathbf{R}_t = \mathbf{R}_t^{\Omega}. \tag{8}$$

The camera rotations and intrinsics remain unchanged. UniDepthV2 therefore does not reconstruct or modify the scene geometry. It only restores the physical scale of the coherent VGGT-Ω reconstruction.

We next construct the candidate physical-support set. The valid metric reconstruction defines the spatial extent of the scene, which is discretized using a fixed voxel size of $\delta = 0.05$ m. This produces the candidate voxel centers $\mathcal{P}_0$. We do not normalize different scenes into a fixed canonical grid. Each candidate coordinate therefore retains its physical meaning in metric 3D space.

Geometry-Guided Active Voxel Filtering then removes unsupported candidate locations. For a candidate voxel center $\mathbf{p}_i$, we first project it onto the image plane of view t using the recovered metric camera geometry:

$$\mathbf{u}_{it} = \pi\left(\mathbf{K}_t, \mathbf{R}_t, \mathbf{t}_t, \mathbf{p}_i\right), \tag{9}$$

where $\mathbf{u}_{it}$ denotes the projected image coordinate and z_{it} denotes the corresponding camera-space depth. A view is considered valid only when the projection lies inside the image, $z_{it} > 0$, and the observation passes the per-frame reconstruction-confidence filtering.

We further require consistency between the projected voxel depth and the recovered metric depth:

$$\frac{|z_{it} - D_t(\mathbf{u}_{it})|}{D_t(\mathbf{u}_{it})} < \tau_d, \qquad \tau_d = 0.05. \tag{10}$$

We denote whether view t provides valid support for $\mathbf{p}_i$ by $v_{it} \in \{0, 1\}$. A candidate voxel is retained only when it receives geometrically consistent support from at least two views:

$$G(\mathbf{p}_i) = \mathbb{I}\left[\sum_{t=1}^{T} v_{it} \geq N_{\min}\right], \qquad N_{\min} = 2. \tag{11}$$

The final active coordinate set consists of the candidate centers satisfying $G(\mathbf{p}_i) = 1$ in Eq. 11. This procedure is fully deterministic and does not introduce a learned occupancy predictor.

Finally, Coordinate-Anchored Multi-View Fusion associates visual evidence with each retained physical coordinate. For an active voxel $\mathbf{p}_i$, we define its valid-view set as $\mathcal{T}_i = \{t \mid v_{it} = 1\}$. The coordinate is projected onto the image plane of each view in $\mathcal{T}_i$, and the corresponding feature $\mathbf{x}_{it}$ is sampled from $\mathbf{X}_t$. Each observation is assigned a deterministic weight w_{it} according to reconstruction confidence and geometric consistency. The valid observations are then aggregated using the confidence-aware fusion defined in Eq. 2.

The resulting feature $\mathbf{f}_i$ remains attached to its metric coordinate $\mathbf{p}_i$. Repeating this process over all retained voxels produces the Coordinate-Grounded Sparse Physical Field $\mathcal{F} = \{(\mathbf{p}_i, \mathbf{f}_i)\}_{i=1}^{M}$ used by both reasoning tracks. The physical field therefore preserves metric geometry and multi-view semantic evidence without replacing the recovered scene structure with a learned geometric representation.

C Grounding Supervision and Data Preprocessing

Language-Guided Physical Grounding is supervised to identify where a referred-to entity is physically supported in the reconstructed scene. The language-modeling objective separately supervises structured query interpretation, as described in Sec. 3.5. Final metric values are not used as regression targets. Instead, instance-level annotations from different training datasets are converted into a common voxel-level physical-support target. For a referred-to entity e_r, we denote its annotated physical geometry by $\mathcal{G}_r^*$. Each active voxel center $\mathbf{p}_i$ is assigned a binary grounding target according to its distance to the annotated geometry:

$$m_i^{(r)*} = \mathbb{I}\left[d\left(\mathbf{p}_i, \mathcal{G}_r^*\right) < \eta\right], \tag{12}$$

where $m_i^{(r)*} \in \{0, 1\}$ and η accounts for small misalignments between the reconstructed physical field and the annotated scene geometry. We set $\eta = 1.5\delta = 0.075$ m for the voxel size $\delta = 0.05$ m used in our implementation. This tolerance allows nearby reconstructed surface voxels to receive positive supervision without labeling the surrounding empty volume as part of the object.

For ScanNet (Dai et al., 2017), we use the annotated scene mesh together with semantic and instance annotations to recover the physical surface associated with each object instance. The annotated object geometry is aligned with the reconstructed scene coordinate system and voxelized at the same physical resolution as GeomFunnel. The resulting object surface provides $\mathcal{G}_r^*$ for Eq. 12. Referring expressions are associated with their corresponding object instances during training, allowing the grounding decoder to learn a mapping from language entities to their physical supports.

ScanNet++ (Yeshwanth et al., 2023) is processed using the same target representation. Its semantic and instance annotations are mapped to the reconstructed scene geometry, and each annotated instance is converted into a physical surface-support representation at the GeomFunnel voxel resolution. Although the original annotation formats of ScanNet and ScanNet++ differ, both datasets are therefore converted into the same voxel-level target $m_i^{(r)*}$. This allows the grounding decoder to share a common supervision interface across datasets.

ARKitScenes (Dehghan et al., 2021) provides annotated 3D oriented bounding boxes rather than the dense object-surface annotations available in ScanNet and ScanNet++. We therefore treat its grounding targets as weak supervision. We first identify reconstructed or observed surface points that fall inside the annotated 3D bounding box and use only these observed surfaces as the positive object support. We do not label the entire bounding-box volume as positive, since this would incorrectly supervise empty space as part of the object. Voxels near uncertain box boundaries or overlapping multiple annotated boxes are ignored when computing the grounding loss. Because these targets are weaker than mesh-derived instance support, ARKitScenes grounding supervision is weighted by $w_d = 0.5$, while ScanNet and ScanNet++ use $w_d = 1.0$.

Sr3D (Achlioptas et al., 2020) provides referring expressions paired with 3D object instances. We use these expressions to increase the diversity of language-conditioned grounding queries. The referred-to object instance supplies the physical target, while the expression is encoded as the entity query e_r. This supervision strengthens the connection between language descriptions and object-level physical support, especially when multiple objects of the same semantic category are present in the scene.

For each referred-to entity, the grounding decoder predicts independent soft support scores $\{g_i^{(r)}\}_{i=1}^{M}$. Because positive object-support voxels are usually far fewer than background voxels, we use a class-balanced binary cross-entropy loss:

$$\mathcal{L}_{\text{bal-BCE}}^{(r)} = -\frac{1}{2}\left[\frac{1}{M_+^{(r)}}\sum_{\substack{i\in\mathcal{J}_r\\ m_i^{(r)*}=1}}\log g_i^{(r)} + \frac{1}{M_-^{(r)}}\sum_{\substack{i\in\mathcal{J}_r\\ m_i^{(r)*}=0}}\log\left(1-g_i^{(r)}\right)\right], \tag{13}$$

where $M_+^{(r)}$ and $M_-^{(r)}$ denote the numbers of valid positive and negative voxels for entity e_r, respectively. Let $\mathcal{J}_r$ denote the valid voxel indices for entity e_r. Voxels marked as ambiguous are excluded from $\mathcal{J}_r$ and from both class counts.

We additionally use a Dice loss to encourage overlap between the predicted support and the physical target:

$$\mathcal{L}_{\text{Dice}}^{(r)} = 1 - \frac{2\sum_{i\in\mathcal{J}_r} g_i^{(r)} m_i^{(r)*} + \epsilon}{\sum_{i\in\mathcal{J}_r} g_i^{(r)} + \sum_{i\in\mathcal{J}_r} m_i^{(r)*} + \epsilon}, \tag{14}$$

where ϵ is a small numerical constant and the sums exclude ambiguous voxels. The grounding objective combines these two losses with $\lambda_{\text{dice}} = 1.0$, using the dataset weights described above.

This supervision teaches Language-Guided Physical Grounding where each referred-to entity is physically supported in the scene. Metric quantities such as distance, size, and room area are not used as regression targets. They are instead derived from the grounded physical geometry through the deterministic operators described in Appendix D.

D Metric Geometry Track and Deterministic Metric Operators

The Metric Geometry Track accesses the Coordinate-Grounded Sparse Physical Field directly, bypassing the compact scene-token interface. As described in Sec. 3.4, the LoRA-adapted Qwen model

first interprets the query to identify the referring phrases and requested geometric quantity. For each required entity, Language-Guided Physical Grounding then predicts soft physical-support scores using Eq. 3. We convert these scores into a spatially coherent physical support before extracting geometric primitives for the Metric Geometry Executor.

For a referred-to entity e_r, we first retain voxels whose grounding confidence meets a fixed threshold $\gamma = 0.5$:

$$\mathcal{I}_r = \left\{ i \mid g_i^{(r)} \geq \gamma \right\}, \qquad \gamma = 0.5. \tag{15}$$

We construct connected components over the retained voxel centers using 26-neighbor connectivity on the 3D voxel grid. To suppress isolated false responses, we retain the component with the largest total grounding mass:

$$C_r^* = \arg\max_{C} \sum_{i \in C} g_i^{(r)}. \tag{16}$$

Instead of assigning equal weight to all retained voxels, we preserve the original grounding confidence after support refinement. The normalized physical-support weight is

$$\omega_i^{(r)} = \frac{g_i^{(r)}}{\sum_{j \in C_r^*} g_j^{(r)}}, \qquad i \in C_r^*. \tag{17}$$

The final grounded support is therefore represented as $\mathcal{S}_q^{(r)} = \{(\mathbf{p}_i, \omega_i^{(r)})\}_{i \in C_r^*}$. This procedure keeps the learned grounding confidence while ensuring that the retained support is spatially coherent.

For object-level reasoning, we derive geometric primitives directly from the grounded physical support. The weighted object centroid is

$$\mathbf{c}_r = \sum_{i \in C_r^*} \omega_i^{(r)} \mathbf{p}_i. \tag{18}$$

We estimate object orientation and extent using weighted principal component analysis. The weighted covariance matrix is

$$\mathbf{\Sigma}_r = \sum_{i \in C_r^*} \omega_i^{(r)} \left(\mathbf{p}_i - \mathbf{c}_r\right) \left(\mathbf{p}_i - \mathbf{c}_r\right)^\top. \tag{19}$$

The eigenvectors of $\mathbf{\Sigma}_r$ define the principal object axes. We project the grounded coordinates onto these axes and compute the extent along each axis using the weighted 2nd and 98th percentiles. This is more robust to isolated support errors than using the raw minimum and maximum. To account approximately for the finite voxel size, we add a fixed padding of δ to each resulting span. We denote the three physical extents by $\{e_{r,k}\}_{k=1}^3$.

For object-size queries, we use the longest physical extent:

$$s_r = \max_{k \in \{1,2,3\}} e_{r,k}. \tag{20}$$

The resulting value is converted to the unit required by the evaluation protocol.

Absolute-distance queries require the closest physical distance between two object surfaces rather than the distance between their centroids. We therefore compute the minimum distance between their grounded voxel supports. Since each voxel represents a cube of side length δ, we correct the center-to-center displacement by the voxel extent:

$$d(a, b) = \min_{i \in C_a^*, j \in C_b^*} \left\| \max\left(|\mathbf{p}_i - \mathbf{p}_j| - \delta, 0\right) \right\|_2, \tag{21}$$

where the absolute value and maximum are applied element-wise. This operator measures the closest distance between the two grounded physical supports and avoids the bias introduced by centroid-based distance.

Room-size queries are handled at the scene level and do not require object grounding. We first estimate candidate planes from the reconstructed physical scene using RANSAC. A robust scene-up direction is obtained from the recovered camera orientations. Among the candidate planes, we select the dominant plane whose normal is aligned with the scene-up direction and whose location

is consistent with the recovered camera trajectory. The selected floor plane is then refined using its inlier points. We use an inlier tolerance of $\epsilon_{\text{plane}} = 1.5\delta = 0.075$ m.

After estimating the floor plane, we retain physical support within $\epsilon_f = 2\delta = 0.10$ m of the floor and project it onto a 2D metric floor coordinate system. The projected support is rasterized using a grid resolution of $r_f = \delta = 0.05$ m. We apply a small 3×3 morphological closing operation followed by hole filling to repair local gaps caused by furniture and partial observations. We retain the union of valid floor regions rather than only the largest connected component, allowing the footprint to represent multiple visible parts of the same room.

If N_f floor cells remain after refinement, the room area is computed as

$$A_{\text{room}} = N_f r_f^2. \tag{22}$$

The resulting object- and scene-level primitives form the query-specific geometry set $\mathcal{P}_q$. The Metric Geometry Executor applies the requested deterministic operator to these primitives and returns the metric prediction. No learned distance, size, or area regressor is used. The learned components interpret the query and locate the query-relevant physical supports; the final metric quantity is computed directly from the reconstructed scene geometry.

E Query Routing and Representation-Decoupled Reasoning

GeomFunnel uses a learned Query-Type Selector to choose between the Compact Reasoning Track and the Metric Geometry Track. The selector is trained with manually defined routing supervision, but the routing decision itself is predicted from the input query at inference time. In particular, we do not provide the benchmark task identity, ground-truth query category, or a manually specified reasoning track during inference.

We define the routing supervision according to whether answering a query requires direct access to an explicit physical quantity. Queries that require metric length, object extent, or physical area are assigned the metric label. Queries that can be answered through semantic understanding, relational spatial reasoning, counting, or planning are assigned the compact label. Importantly, this criterion is not based on whether the final answer is numerical. For example, object counting may produce a numerical answer but does not require explicit metric geometry. Relative-distance reasoning similarly compares spatial relations without requiring a metric distance in physical units.

Let $\mathbf{h}_q$ denote the language representation of query q. The Query-Type Selector is implemented as a two-layer MLP with a hidden dimension of 512. With ϕ denoting the MLP activation, it predicts the probabilities of the two reasoning tracks as

$$\mathbf{p}_q = \text{softmax}\left(W_2 \phi\left(W_1 \mathbf{h}_q\right)\right), \qquad \rho(q) = \underset{c \in \{\text{direct}, \text{geo}\}}{\arg\max} \mathbf{p}_q[c]. \tag{23}$$

Here, direct and geo are symbolic labels for the Compact Reasoning Track and Metric Geometry Track, respectively. The selector is trained using the supervised selector objective $\mathcal{L}_{\text{selector}}$ in Eq. 6. The supervision label is constructed from the query requirement described above and is used only during training.

Table 6 summarizes the routing supervision used in our experiments. These labels define the training targets for the Query-Type Selector rather than hard-coded inference rules.

At inference time, only the natural-language query q is given to the selector. The predicted route $\rho(q)$ determines which representation interface is used. No benchmark-specific task identifier is provided. This allows queries with different surface forms to be routed according to their underlying reasoning requirement rather than their dataset category.

When $\rho(q) = \text{direct}$, the Coordinate-Grounded Sparse Physical Field is compressed by the Global Perceiver Bridge into the latent representation $\mathbf{Z}$. The language model then produces the final answer from these compact tokens. This track handles semantic understanding, object attributes, counting, relative spatial relations, situated reasoning, and other queries that do not require an explicit physical measurement.

When $\rho(q) = \text{geo}$, the compact scene-token interface is bypassed. Qwen first produces the structured query interpretation specifying the relevant referring phrases and geometric operation. Object-level queries then access the physical field through Language-Guided Physical Grounding, while

Table 6: **Routing supervision used to train the Query-Type Selector.** The labels are used only during training. At inference time, the reasoning track is predicted directly from the input query.

Query Type	Training Routing Label
Absolute distance	Metric
Object size	Metric
Room size	Metric
Object counting	Compact
Relative distance	Compact
Relative direction	Compact
Route planning	Compact
Appearance order	Compact
General 3D question answering	Compact
Object-centric captioning	Compact

scene-level metric queries can directly access the reconstructed physical geometry without an object-level support mask. The Metric Geometry Executor applies the deterministic operators described in Appendix D to the required geometric primitives and returns the final metric prediction.

For queries routed to the Metric Geometry Track, the resulting metric quantity is treated as the final numerical prediction. It is converted only to the output unit and format required by the evaluation protocol and is not passed back to the language model for additional numerical estimation. This prevents the explicit physical measurement from being replaced by language-based numerical reasoning.

Representation-Decoupled Reasoning therefore separates two decisions. The routing supervision defines which type of representation is appropriate for each reasoning requirement, while the learned Query-Type Selector predicts that requirement directly from the query at inference time. This allows GeomFunnel to preserve a compact language-oriented interface for general spatial understanding while directly accessing the physical scene representation when explicit metric geometry is required.

F Discussion and Scope

GeomFunnel relies on the quality of the recovered physical scene. Errors in VGGT-Ω reconstruction, UniDepthV2-based metric scale alignment, or Language-Guided Physical Grounding can propagate to the final metric prediction. Although the Metric Geometry Track directly operates on explicit physical geometry, its accuracy is still bounded by the reliability of these upstream components.

Our current evaluation mainly focuses on indoor 3D scene benchmarks. The behavior of GeomFunnel under stronger domain shifts, in outdoor environments, and in very large-scale scenes has not been fully studied. Evaluating the robustness and scalability of the physical representation in these settings remains an important direction for future work.

G Additional Ablation Studies

We study the effect of the number of metric anchor frames used by UniDepthV2 for sequence-level scale estimation. We vary the number of uniformly distributed anchor frames over 1, 4, 8, and 16 while keeping all other model components and training settings unchanged. UniDepthV2 is applied only to the selected anchor frames, and the resulting sequence-level scale is used to calibrate the full VGGT-Ω reconstruction. We report performance on Absolute Distance, Object Size, and Room Size, which directly depend on metric geometry.

As shown in Table 7, performance improves consistently as more anchor frames are introduced and largely saturates at eight anchors. Increasing the number of anchors from 8 to 16 improves the Metric Avg. from 74.6 to only 74.8, while doubling the number of UniDepthV2 evaluations. We therefore use eight anchor frames as the default setting. We speculate that uniformly distributed anchors already provide sufficient coverage for robust sequence-level scale estimation, while additional anchors contribute mostly redundant scale evidence.

We study the effect of physical voxel resolution by varying the voxel size δ over 0.025 m, 0.05 m, and 0.10 m while keeping all other model components and training settings unchanged. The voxel size determines the spatial resolution of the Coordinate-Grounded Sparse Physical Field and directly

Table 7: **Effect of the number of metric anchor frames.** UniDepthV2 is applied only to the selected anchor frames for sequence-level metric scale estimation. The default setting is marked with †.

Anchors	Abs. Dist.	Obj. Size	Room Size	Metric Avg.
1	64.8	75.1	69.6	69.8
4	68.2	77.6	74.7	73.5
8†	69.4	78.4	76.1	74.6
16	**69.6**	**78.5**	**76.2**	**74.8**

affects both geometric precision and the number of active voxels. We therefore evaluate the three metric tasks together with peak GPU memory and end-to-end inference time.

As shown in Table 8, reducing the voxel size from 0.10 m to 0.05 m substantially improves metric performance, while further reducing it to 0.025 m provides only a small additional gain. In contrast, the finer 0.025 m representation increases peak memory from 20.8 GB to 25.4 GB and inference time from 3.2 s to 4.0 s. We therefore use $\delta = 0.05$ m as the default setting, which retains nearly all of the metric accuracy of the finer representation at a noticeably lower computational cost. This suggests that, beyond this resolution, reconstruction and grounding uncertainty may become more influential than voxel discretization, limiting the benefit of further increasing spatial resolution.

Table 8: **Effect of physical voxel resolution.** Smaller voxels preserve finer geometric detail but increase the size of the sparse physical field. The default setting is marked with †.

Voxel Size (m)	Abs. Dist.	Obj. Size	Room Size	Metric Avg.	Peak Memory (GB)	Time (s)
0.025	**69.8**	**78.8**	**76.2**	**74.9**	25.4	4.0
0.050†	69.4	78.4	76.1	74.6	20.8	3.2
0.100	67.5	76.3	73.4	72.4	**18.6**	**2.8**

We study the effect of the number of Perceiver latent tokens by varying L over 128, 256, 512, and 1024 while keeping the rest of the architecture and the training settings unchanged. The number of latent tokens controls the capacity of the compact representation used by the Compact Reasoning Track, but does not change the physical field accessed by the Metric Geometry Track. We therefore report performance on VSI-Bench categories assigned to the Compact Reasoning Track, ScanQA, SQA3D, and Scan2Cap, together with the Metric Avg. and inference cost.

As shown in Table 9, increasing the number of latent tokens consistently improves performance on the Compact Reasoning Track, but the gains become small beyond 256 tokens. Increasing L from 256 to 1024 improves the VSI Compact Avg. from 75.7 to 76.0, while peak memory increases from 20.8 GB to 26.7 GB and inference time from 3.2 s to 4.2 s. We therefore use $L = 256$ as the default setting, which provides a favorable balance between representation capacity and computational cost. In contrast, the Metric Avg. remains within a narrow range across different latent capacities. This suggests that metric reasoning is largely insensitive to the degree of language-oriented compression, consistent with the Metric Geometry Track directly accessing the preserved physical representation.

Table 9: **Effect of the number of Perceiver latent tokens.** VSI Compact Avg. is the average over the five categories assigned to the Compact Reasoning Track: object counting, relative distance, relative direction, route planning, and appearance order. The default setting is marked with †.

Latents	VSI Compact Avg.	Metric Avg.	ScanQA CIDEr	SQA3D EM	Scan2Cap CIDEr	Peak Memory (GB)	Time (s)
128	74.7	74.5	110.6	61.8	104.7	**19.8**	**3.0**
256†	75.7	**74.6**	113.2	62.6	106.2	20.8	3.2
512	75.9	74.3	113.8	62.8	106.7	22.4	3.5
1024	**76.0**	74.5	**114.0**	**62.9**	**106.9**	26.7	4.2

We evaluate the Query-Type Selector by comparing three routing strategies while keeping the underlying Compact Reasoning Track and Metric Geometry Track unchanged. Always Compact forces every query through the language-oriented representation and therefore removes access to explicit metric execution. Learned Selector uses the predicted routing decision from the input query at inference time, without access to the benchmark task identity or ground-truth routing label. Oracle Routing instead uses the ground-truth routing label and serves as an upper-bound reference. We report routing accuracy and class-wise recall together with compact-track, metric-track, and overall VSI-Bench performance.

As shown in Table 10, the learned selector achieves 97.1% routing accuracy and closely approaches Oracle Routing in downstream performance. Its VSI Avg. reaches 75.3 compared with 75.4 under Oracle Routing, indicating that routing errors introduce only a small performance gap. In contrast, forcing all queries through the Compact Reasoning Track reduces the Metric Avg. from 74.6 to 70.2 while leaving performance on the Compact Reasoning Track largely unchanged. We therefore use the learned selector at inference time. These results indicate that the selector can reliably identify queries that require explicit physical measurement without relying on manually specified test-time routing.

Table 10: **Analysis of query routing.** Always Compact forces every query through the language-oriented representation. Oracle Routing uses the ground-truth routing label and provides an upper bound. GeomFunnel uses the learned Query-Type Selector at inference time.

Routing Strategy	Route Acc. (%)	Metric Recall (%)	Compact Recall (%)	Compact Avg.	Metric Avg.	VSI Avg.
Always Compact	–	–	100.0	75.7	70.2	73.6
Learned Selector†	97.1	96.4	97.5	75.7	74.6	75.3
Oracle Routing	100.0	100.0	100.0	**75.8**	**74.8**	**75.4**

H Computational Complexity and Inference Efficiency

Although GeomFunnel preserves an explicit physical scene representation, downstream reasoning is performed over a sparse set of active voxels rather than a dense 3D volume. Let M denote the number of active voxels after Geometry-Guided Active Voxel Filtering, L the number of Perceiver latent queries, and d the hidden dimension of the Global Perceiver Bridge. In our implementation, $L = 256$ is fixed for all scenes.

For the Compact Reasoning Track, each Perceiver layer first performs cross-attention from the L latent queries to the M sparse voxel features. The cost of its attention score and value aggregation is

$$\mathcal{O}(MLd). \tag{24}$$

The subsequent latent self-attention operates only over the fixed set of latent queries and has complexity

$$\mathcal{O}(L^2 d). \tag{25}$$

Therefore, scene-dependent computation before the language model grows with the number of retained active voxels rather than with the volume of a dense 3D grid. After the Global Perceiver Bridge, the language model always receives a fixed number of 256 scene tokens. Its language-side context length is therefore independent of the original number of reconstructed 3D locations.

For the Metric Geometry Track, scene access bypasses the compact scene-token interface after language-based query interpretation. Let R denote the number of query entities and d_g the grounding hidden dimension. Entity-to-voxel cross-attention has complexity

$$\mathcal{O}(RMd_g). \tag{26}$$

In practice, R is small because metric queries usually refer to only one or a few entities. After grounding, geometric primitives such as centroids, covariance matrices, and object extents are computed only over the corresponding sparse physical supports. The closest-support operator used for absolute distance has a worst-case pairwise cost of $\mathcal{O}(|C_a^*||C_b^*|)$, but it is applied only to the two grounded object supports rather than the complete scene. Scene-level operations such as floor estimation and footprint rasterization similarly operate directly on the sparse reconstructed geometry and do not require autoregressive numerical reasoning. These geometric complexity expressions exclude the separate cost of language-based query interpretation.

The storage required for the downstream scene representation is approximately linear in the number of active voxels:

$$\mathcal{O}(Md + Ld). \tag{27}$$

The first term stores the sparse physical field, while the second corresponds to the fixed latent representation used by the Compact Reasoning Track. This expression describes representation storage

rather than total peak GPU memory; it excludes model parameters and intermediate attention activations. This differs from a dense voxel representation whose storage grows with the number of spatial cells inside the scene volume, including unsupported empty space.

The total parameter count reported in Table 4 also includes the frozen VGGT-Ω geometry backbone and UniDepthV2 metric-scale module. These components increase the nominal model size but are not optimized during training. More importantly, their outputs are reduced to the sparse physical field before downstream reasoning. The Compact Reasoning Track then compresses this field into 256 latent tokens, while the Metric Geometry Track accesses only the physical support required by the metric query. These properties limit the cost of downstream scene access. The measured end-to-end memory consumption and latency in Table 4, which also include geometry recovery and language-based query processing, are close to those of the compact 3D vision-language baselines compared.

I Additional Qualitative Illustrations

Figure 5 illustrates compositional Metric Geometric Reasoning beyond the three primary VSI-Bench measurement categories. For each scene, the same sparse physical field and reusable geometric primitives are shared across multiple queries, while only the deterministic geometric program changes with the requested operation. The examples progress from primitive selection (**L1**) to arithmetic composition over recovered quantities (**L2**) and finally to chained multi-step execution involving object selection, ranking, and subsequent metric operations (**L3**). These examples illustrate how the Metric Geometry Executor can extend beyond isolated distance, object-size, or room-area predictors; instead, these quantities serve as reusable intermediate primitives that can be composed into more complex metric queries.

The examples use held-out scenes from ARKitScenes, ScanNet, and ScanNet++, covering different room layouts, object sets, and geometric scales. In particular, the Metric Geometry Executor can reuse the same grounded scene geometry for cross-object comparisons, numerical differences and ratios, multi-object ranking, and chained operations that combine multiple primitive types. These qualitative illustrations demonstrate the intended separation between *representation* and *execution*: the physical field provides a scene-level metric substrate once, whereas query-specific answers are obtained by composing deterministic operations over this shared representation rather than relearning geometry for each task. Scene D is included only to illustrate the execution structure because its ground-truth values are incomplete; it does not provide numerical evidence of metric accuracy.

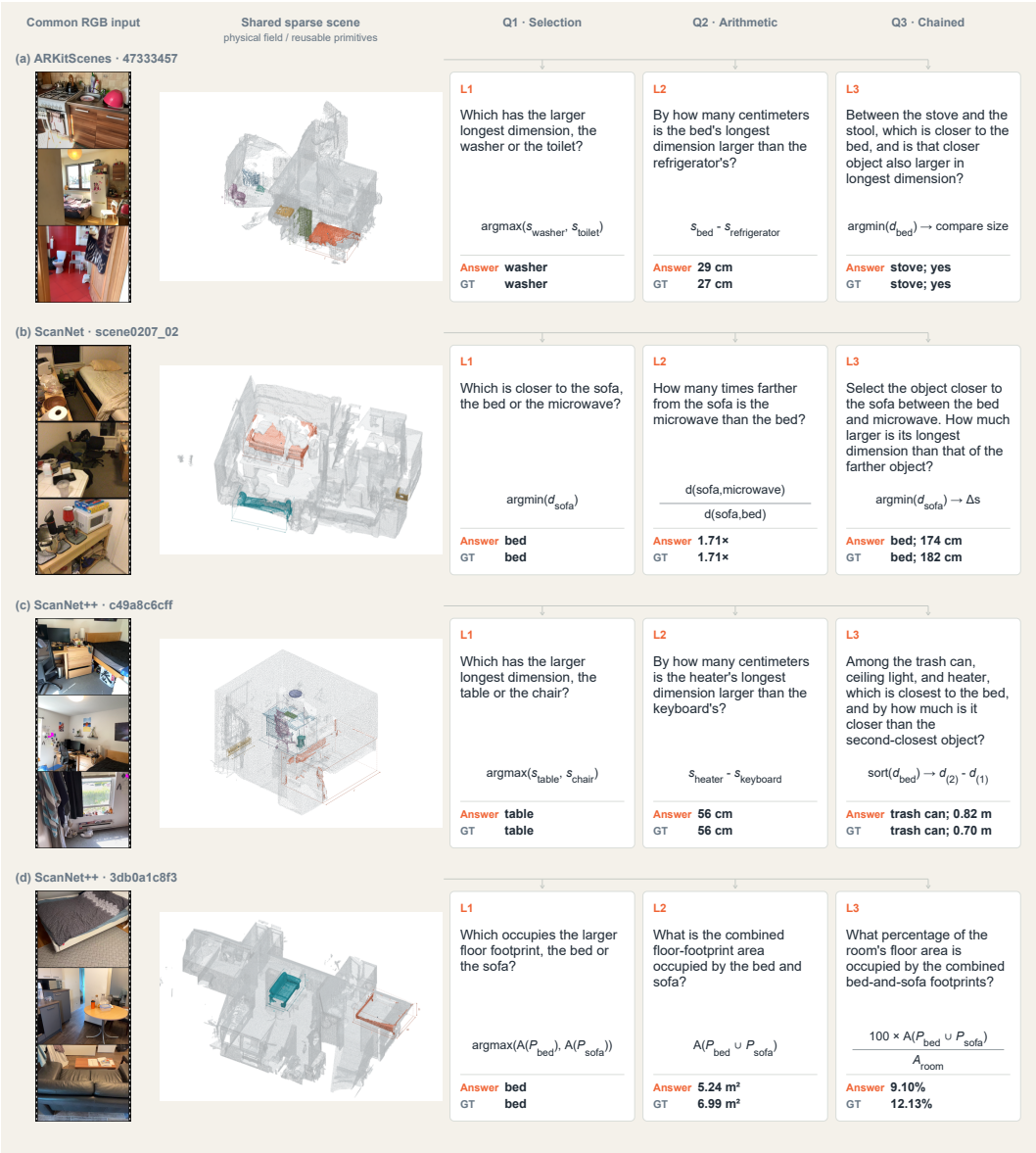

Figure 5: We show four held-out scenes spanning ARKitScenes, ScanNet, and ScanNet++. For each scene, a shared sparse physical field and its geometric primitives are reused across three query types of increasing compositional complexity: primitive selection (**L1**), arithmetic composition (**L2**), and chained multi-step execution (**L3**). Each query is resolved by composing deterministic operations over the same underlying scene geometry, illustrating that the Metric Geometry Executor is not limited to isolated distance, object-size, or room-area prediction. Scene D illustrates execution structure only; its ground-truth values are incomplete.

J TOKEN COMPRESSION DIAGNOSTIC IN FIGURE 2

J.1 SCOPE AND STATUS OF THE DIAGNOSTIC

Figure 2 compares the relative degradation of Qualitative Scene Reasoning (QSR), Metric Geometric Reasoning (MGR), and Metric Structure Recoverability (MSR) as the retained visual-token budget decreases. It contains five model series: VLM-3R, SpatioLM-8B, Cambrian-S-7B, GS-Reasoner, and GeoThinker. The four retention ratios are $r \in \{0.75, 0.50, 0.25, 0.125\}$. Retention is the fraction kept, so smaller values indicate stronger compression. The diagnostic concerns changes relative to each model's own full-budget reference; it is not an absolute-accuracy ranking of these models.

Provenance of the supplied plot. The figure is constructed from the supplied table of 60 model-specific degradation values and 12 interval pairs, reproduced in Table 11. The accompanying plotting records explicitly identify the intervals as *predicted* intervals, originally labeled "Pred. 95% CI." The available records do not include full-budget scores, per-example predictions, an identified resampling unit, or a bootstrap computation. Thus, the renderer verifies the marker-to-bar arithmetic but does not reconstruct the raw-score-to-marker calculation, and the displayed intervals do not have verified empirical 95% coverage. They must not be interpreted as experimentally validated confidence intervals. The controlled evaluation requirements below state what must be documented to establish that interpretation; they are not a claim that missing experimental steps have been performed.

J.2 EXPERIMENTAL CONDITIONS REQUIRED FOR A CONTROLLED INTERPRETATION

A controlled compression study compares each retained-budget condition with the corresponding full-budget condition while holding the evaluation data, model checkpoint, frame selection, image resolution, prompt, and decoding settings fixed. The exact evaluation split, scene count, question count, and assignment of questions to QSR and MGR must be recorded. The five non-metric query categories in GeomFunnel's routing supervision do not, by themselves, establish the composition of the QSR subset used in this diagnostic.

For model m and example j, let N_{mj} denote the number of visual scene tokens at the specified compression interface before token retention, excluding text and any special tokens that are held fixed. A reproducible protocol must state where this interface lies in each model, how the retained count is rounded, and whether tokens are removed, masked, merged, or selected by a learned operator. It must also specify the selection rule, preservation of token positions and order, allocation across frames, handling of global tokens, and any random seeds. Equal retention ratios do not imply equal absolute token counts or equal computational cost across different architectures. Any adaptation of a model or readout at a reduced budget must be reported separately from an evaluation with fixed weights.

QSR measures semantic and relative spatial reasoning that does not require precise physical quantities. MGR measures answers requiring metric quantities, such as lengths, object extents, or physical areas. MSR is a separate representation diagnostic: it concerns the metric information accessible through a specified readout of the retained representation. An empirical MSR definition requires the geometric target, units, ground-truth source, error or score, readout architecture, training and evaluation split, and any alignment procedure. These details are not supplied by the plot alone. In particular, an unreported scale alignment or access to external geometry during the readout could change what the diagnostic measures. An MGR answer score cannot substitute for an independently defined MSR measurement.

J.3 FROM RAW MEASUREMENTS TO RELATIVE DEGRADATION

For clarity, the following equations specify how relative degradation can be computed once the original measurements and score direction are available. They are not a reconstruction of absent evaluation logs. Let $S_{mk}(r)$ be a higher-is-better score for model m, quantity k, and retention ratio r. For $S_{mk}(1) > 0$, the relative degradation is

$$D_{mk}(r) = 100 \frac{S_{mk}(1) - S_{mk}(r)}{S_{mk}(1)}. \tag{28}$$

If a quantity is instead measured using a lower-is-better error $E_{mk}(r)$, with $E_{mk}(1) > 0$, its corresponding relative increase is

$$D_{mk}(r) = 100\frac{E_{mk}(r) - E_{mk}(1)}{E_{mk}(1)}. \tag{29}$$

The appropriate definition must be fixed for each quantity and used at every retention ratio. These expressions are relative percentage changes, not percentage-point changes in the original score. A zero full-budget baseline requires a separately specified convention. Improvements yield negative degradation and should not be silently clipped. The full-budget condition has zero degradation by construction and is omitted from the figure.

When a quantity aggregates several question categories, its within-model aggregation must be stated before applying the normalization. Each model is normalized against its own full-budget reference before averaging over models. Normalizing an average raw score across models is generally different and is not the marker-to-bar operation used in this figure.

J.4 Markers, bars, and the plotted values

The marker shapes identify models: circles denote VLM-3R, triangles denote SpatioLM-8B, diamonds denote Cambrian-S-7B, squares denote GS-Reasoner, and pentagons denote GeoThinker. Their horizontal offsets within a bar are only for visibility and do not encode another variable. Yellow, orange, and red bars indicate QSR, MGR, and MSR, respectively. Each marker is the supplied degradation value $D_{mk}(r)$ for one model, quantity, and retention ratio. With $N_{\text{model}} = 5$ models, the corresponding bar is the equal-weight arithmetic mean

$$\overline{D}_k(r) = \frac{1}{N_{\text{model}}} \sum_{m=1}^{N_{\text{model}}} D_{mk}(r), \qquad N_{\text{model}} = 5. \tag{30}$$

This calculation is verified for all 12 bars. For example, the 75% QSR bar is $(2.2 + 1.6 + 2.8 + 2.4 + 2.0)/5 = 2.2$. The exact QSR means at 25% and 12.5% retention are 10.62 and 19.28, respectively, which round to 10.6 and 19.3 when displayed to one decimal place.

Table 11: **Values used to construct Figure 2.** All values are relative degradation in percent. Model columns give marker heights; Mean gives the exact arithmetic mean used for each bar. The final column reproduces the supplied predicted interval endpoints and does not represent a statistically verified confidence interval.

Retention	Quantity	VLM-3R	SpatioLM	Cambrian-S	GS-Reasoner	GeoThinker	Mean	Supplied interval
75%	QSR	2.2	1.6	2.8	2.4	2.0	2.20	[1.2, 3.2]
	MGR	3.9	3.3	4.9	4.2	3.7	4.00	[2.7, 5.3]
	MSR	8.8	7.6	10.6	9.3	8.2	8.90	[7.1, 10.7]
50%	QSR	4.9	3.8	5.6	5.2	4.5	4.80	[3.3, 6.3]
	MGR	8.5	7.6	10.0	9.2	8.2	8.70	[6.7, 10.7]
	MSR	18.0	16.2	21.6	19.6	17.6	18.60	[16.1, 21.1]
25%	QSR	10.8	8.6	12.4	11.5	9.8	10.62	[8.8, 12.4]
	MGR	17.7	15.9	21.0	19.1	16.8	18.10	[15.7, 20.5]
	MSR	34.1	31.5	39.4	36.0	33.0	34.80	[31.5, 38.1]
12.5%	QSR	19.5	15.8	22.6	20.5	18.0	19.28	[16.9, 21.7]
	MGR	30.4	27.8	36.0	32.8	29.0	31.20	[27.9, 34.5]
	MSR	54.0	50.4	62.1	57.5	52.5	55.30	[50.9, 59.7]

J.5 Interval rendering and uncertainty estimation

Let $[L_k(r), U_k(r)]$ denote one supplied interval. Its lower and upper distances from the exact bar mean are

$$e_k^-(r) = \overline{D}_k(r) - L_k(r), \qquad e_k^+(r) = U_k(r) - \overline{D}_k(r). \tag{31}$$

This converts supplied endpoints into drawable whiskers; it does not estimate uncertainty. For 25% QSR, these distances are 1.82 and 1.78; for 12.5% QSR, they are 2.38 and 2.42. The small asymmetry follows from retaining the exact marker mean while the supplied interval is centered on its

one-decimal summary. The interval pairs should therefore be preserved as endpoints rather than silently re-centered by rounding.

The existing whiskers cannot be derived from the five markers by a consistent standard-deviation, standard-error, or Student-t confidence-interval rule. For example, the sample standard deviation of the 75% QSR markers is approximately 0.447, giving a standard error of 0.200 and a conventional Student-t 95% half-width of approximately 0.555; the supplied half-width is 1.0. This difference does not rule out a separately computed data-sampling interval, but no such computation is documented in the supplied plotting record. The five selected architectures also should not be treated automatically as independent random samples from a population of models.

To validate empirical uncertainty in a future reconstruction of this diagnostic, the original scene identifiers and paired full-budget and retained-budget predictions are required. A suitable candidate procedure is a paired scene bootstrap: sample scenes with replacement, retain all associated questions and both budget conditions together, recompute each model's scores and relative degradation in each draw, and then apply Eq. 30. Pointwise percentile intervals can be obtained from the 2.5th and 97.5th percentiles of the resulting means. The number of draws, random seed, resampling unit, aggregation rule, and interval method must be reported. This is a proposed validation procedure, not the established source of the currently displayed whiskers.

J.6 Interpretation and limits

Within the supplied values, MSR degrades more than MGR, and MGR more than QSR, at every retention ratio for every model. This pattern is consistent with the motivation for retaining explicit metric evidence alongside compact tokens. However, an MSR result is conditional on the target, alignment, and readout used; reduced recoverability through one readout does not prove that all metric information has disappeared. The supplied values and predicted intervals alone cannot establish statistical significance, universal behavior across compact representations, or a causal explanation for each model's degradation. Those empirical conclusions require the documented controlled evaluation and raw-result verification described above.