ANCHOR3D: DO 3D LLMS USE 3D POSITIONS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent 3D large language models (3D LLMs) inject explicit geometry into a language backbone through 3D positional embeddings, voxel-aligned features, or specialized rotary encodings, and report steady gains on 3D question answering and grounding benchmarks. We ask whether these models actually use the geometry they are given. Across five representative 3D LLMs, including a model with a dedicated 3D rotary encoding, we find that randomly permuting the 3D positions assigned to visual tokens leaves performance essentially unchanged across the benchmarks we evaluate, and so does setting them all to zero. Simple MLP probes show that, the LLM's hidden states can still carry accurate relative 3D offsets between tokens, so the geometry is encoded but not consulted: current benchmarks can be solved from appearance and language cues, and a purely autoregressive decoder is never forced to bind the entities named in a question to their 3D locations. Motivated by this, we propose to make question-conditioned localization an explicit bottleneck. Our model first detects and segments the objects named in the question, then selects one mask per named object with a question-conditioned matcher, and passes only the patch tokens covered by these masks, together with their 3D positions, to the LLM for metric reasoning. Trained on ScanQA, SQA3D, and R3D-Bench-style metric questions built on ScanNet, the resulting model, Anchor3D, becomes sensitive to positional perturbation, matches GPT-5.5 and Gemini 3.1 Pro on R3D-Bench with 4B parameters, and improves ScanQA and SQA3D over the same pipeline.

1 INTRODUCTION

Large multimodal models have become strong at understanding 2D images and videos, and a growing line of work extends them to **3D scene understanding**. Progress is measured on a small set of benchmarks: ScanQA for scene-level question answering, SQA3D for situated and egocentric reasoning, and ScanRefer and its relatives for language-guided 3D grounding (Azuma et al., 2022; Ma et al., 2022; Chen et al., 2020). To move these numbers, 3D LLMs have adopted increasingly explicit ways of injecting geometry into the language backbone, from point-cloud and object-centric tokens to voxel-aligned features, 3D positional embeddings, and dedicated rotary encodings such as QuatRoPE (Hong et al., 2023; Huang et al., 2023; 2024; Zhu et al., 2024a; Zhou et al., 2026). These designs are validated on the same downstream benchmarks, which raises a basic question: *do the benchmarks measure, and do the models use, the 3D geometry these designs are meant to provide?*

Real-3DQA (Ma et al., 2026) already showed that a text-only model fine-tuned on the task can match full 3D LLMs, i.e., much of the benchmark is answerable from language priors. This does not mean the scene is useless: when we re-evaluate the released checkpoints (Table 1), giving the model the scene still improves over text-only inference by 7.9–12.8 EM points on SQA3D. The open question is *what* in the scene is being used.

We answer it by perturbing the geometry given to five representative 3D LLMs (LLaVA-3D, Video-3D LLM, Spatial-MLLM, Proxy3D, and QuatRoPE) at inference time, without retraining (Section 3). Two findings emerge. **(i) 3D positions are not consulted.** Randomly permuting the 3D positional embeddings across visual tokens, or setting them all to zero, changes ScanQA and SQA3D EM by at most 2.2 points, a small fraction of the gap to text-only inference. The same holds for grounding: ScanRefer Acc@0.25 of QuatRoPE, a method whose contribution is a 3D rotary encod-

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.

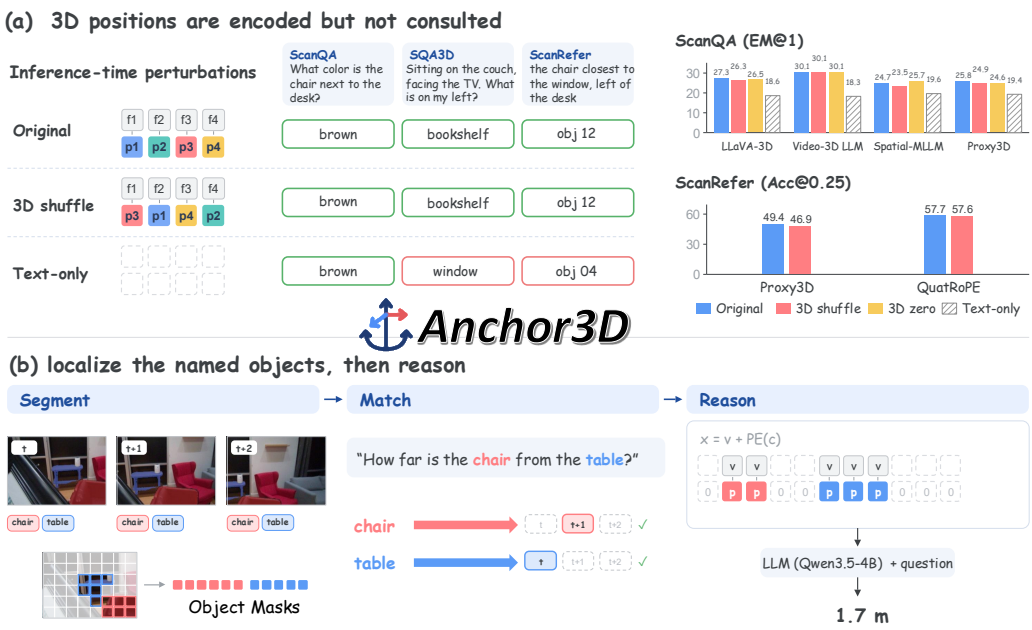

Figure 1: **Encoded geometry is not necessarily used.** Top: permuting 3D positions barely changes existing 3D LLM accuracy on 3D QA benchmarks. Bottom: Anchor3D explicitly binds the objects named in the question to segmented masks and passes only the patch tokens they cover, with their 3D positions, to the reasoning stage.

ing, moves from 57.7 to 57.6 under shuffling. **(ii) The geometry is nevertheless encoded.** A simple probing head on frozen hidden states recovers the relative offset Δxyz between two visual tokens even at the last layer of the LLM (R^2 of 0.61–0.94 per axis for four models), and this collapses when the 3D positions are set to zero. The information is present in the representation but does not influence the answer.

Together these results point to a failure of *binding* rather than of *encoding*. Existing QA and grounding benchmarks can largely be solved by matching the question to the semantic content of visual tokens or object proposals; the gradient rarely requires the model to identify *which* tokens a question refers to and read off *their* coordinates.

We therefore make question-conditioned localization an explicit bottleneck: Anchor3D keeps only the visual tokens of the objects named in the question and sets all others to zero. Trained with additional metric questions following R3D-Bench (Horton et al., 2026), it becomes sensitive to position shuffling, i.e., it now reads the geometry.

Our contributions are:

- An inference-time perturbation and probing study of five 3D LLMs, including a 3D-RoPE method, showing that permuting their 3D positions barely affects performance across the benchmarks we evaluate (ScanQA, SQA3D, and ScanRefer), even though a simple probing head can still decode relative 3D offsets from the hidden states of four of them: the geometry is encoded but not used.
- Anchor3D, which makes question-conditioned localization an explicit bottleneck: the objects named in the question are detected, segmented, and matched to their masks, and only their patch tokens reach the LLM.
- Anchor3D becomes sensitive to position shuffling, matches GPT-5.5 and Gemini 3.1 Pro on R3D-Bench with 4B parameters while outperforming all open-weight models without tool use, and improves ScanQA and SQA3D over the same pipeline.

2 Related Work

3D vision-language models and benchmarks. 3D vision-language benchmarks range from referring-expression grounding (ScanRefer, ReferIt3D, Multi3DRefer) and scene-level or situated QA (ScanQA, SQA3D) to spatial reasoning over videos (VSI-Bench) and quantitative 3D reasoning from egocentric RGB-D videos (R3D-Bench) (Chen et al., 2020; Achlioptas et al., 2020; Zhang et al., 2023; Azuma et al., 2022; Ma et al., 2022; Yang et al., 2025b; Horton et al., 2026). Models differ mainly in how they inject 3D information: point-cloud or scene features (Hong et al., 2023; Huang et al., 2023; Chen et al., 2024c; Zhu et al., 2024b), explicit object tokens (Huang et al., 2024; Kang et al., 2025; Chen et al., 2024d; Yang et al., 2025a), 2D or video features lifted with depth or reconstructed geometry (Qi et al., 2026; Zhu et al., 2024a; Zheng et al., 2025; Wu et al., 2025; Jiang et al., 2026), and increasingly explicit positional encodings such as QuatRoPE and GeoPE (Zhou et al., 2026; Yao & Yang, 2025). Whether the injected geometry is actually used for prediction is rarely verified.

Shortcuts and metric spatial reasoning. VQA models are known to exploit language priors (Agrawal et al., 2018; Cadene et al., 2020). In spatial reasoning, Real-3DQA, VSI-Bench-Debiased, and SpatialReasoner expose text-only and 2D shortcuts, and Transcrib3D, VLM-Grounder, and See-Ground reach competitive 3D results without native 3D reasoning (Ma et al., 2026; Brown et al., 2025; Ma et al., 2025; Fang et al., 2024; Xu et al., 2024; Li et al., 2025). Another line improves quantitative spatial reasoning with supervision on distance, size, and depth (Chen et al., 2024a; Cheng et al., 2024; Liao et al., 2024; Cai et al., 2025; Ouyang et al., 2025). We connect the two: we diagnose that 3D positions are ignored, and use metric supervision to make the model rely on them.

Grounding before reasoning. Intermediate localization, pointing, or object identifiers have been used to tie language reasoning to visual evidence in 2D (Yang et al., 2023; Shao et al., 2024; Liu et al., 2024b; Deitke et al., 2025) and 3D (Huang et al., 2024; Kang et al., 2025; Chen et al., 2024d; Zhi et al., 2025; Linghu et al., 2026), and token-selection methods keep task-relevant visual tokens mainly for efficiency (Wang et al., 2026; Chen et al., 2024b; Zhang et al., 2025; Yang et al., 2026). We instead use question-conditioned token selection as a bottleneck whose purpose is to force the model to read the 3D positions of the selected tokens.

3 Do 3D LLMs Use Their 3D Positions?

We study five recent 3D LLMs that span the main ways of injecting geometry: LLaVA-3D (voxel-pooled patches with 3D positional embeddings), Video-3D LLM (frame tokens with 3D position encoding), Spatial-MLLM (spatial features from a feed-forward reconstruction model), Proxy3D (object proxies with 3D positional embeddings), and QuatRoPE (a quaternion-based 3D rotary encoding over point-cloud object tokens) (Zhu et al., 2024a; Zheng et al., 2025; Wu et al., 2025; Jiang et al., 2026; Zhou et al., 2026; Huang et al., 2024; Zemskova & Yudin, 2025). All models are evaluated with their released checkpoints; every perturbation below is applied at inference time only.

3.1 The scene helps, but is the help geometric?

We first establish that the scene is used at all. Table 1 compares each model with its full scene input (*Original*) against the same model given only the question (*Text-only*). On SQA3D the gap is 7.9–12.8 EM@1 points across all five models, and on ScanQA it is 5.1–11.8 EM@1 points across the four models evaluated. The scene therefore carries information the models exploit. Whether that information is the 3D geometry, or merely the appearance and semantics of what is visible, is the question we turn to next.

3.2 Perturbing 3D positions at inference

Perturbations. Every model above attaches 3D geometry to its visual tokens: LLaVA-3D, Video-3D LLM, Proxy3D, and QuatRoPE inject a 3D coordinate through a positional embedding or rotary encoding, and Spatial-MLLM fuses geometry features from a feed-forward reconstruction model (VGGT). We keep the visual features fixed and apply two perturbations (Appendix A.2): *3D position shuffle* randomly permutes the 3D positions (for Spatial-MLLM, the VGGT geometry features) among the visual tokens of a scene, so each token keeps its appearance but receives the geometry

Table 1: Robustness to position perturbations on ScanQA (val), SQA3D (val), and R3D-Bench (R3D). "Original" denotes results reproduced by us with the unperturbed input on the full evaluation sets. EM-R: refined exact match. R3D Avg: average over the 15 question types of R3D-Bench (accuracy for multiple choice, MRA for numeric questions), with loose answer parsing and parse failures excluded; per-type results are in Table 10. "–" denotes not evaluated.

Method	Perturbation	ScanQA				SQA3D		R3D
		EM@1	CIDEr	BLEU-4	ROUGE-L	EM@1	EM-R	Avg
LLaVA-3D	Original	27.3	94.3	15.7	46.1	55.9	58.7	18.0
	Text-only	18.6	66.2	10.7	34.8	45.8	48.0	–
	3D position shuffle	26.3	90.0	14.3	44.5	54.4	57.1	18.3
	3D zero position	26.5	90.3	14.8	44.6	55.0	57.9	17.3
Video-3D LLM	Original	30.1	102.0	15.8	49.1	58.5	61.3	22.3
	Text-only	18.3	65.9	10.0	34.4	45.7	48.3	–
	3D position shuffle	30.1	101.8	17.4	48.8	57.8	60.7	22.0
	3D zero position	30.1	102.5	17.4	49.2	58.3	61.3	22.3
Spatial-MLLM	Original	24.7	87.0	13.1	42.6	55.4	58.4	28.7
	Text-only	19.6	66.4	10.1	34.7	47.3	49.8	–
	3D position shuffle	23.5	82.2	11.0	40.9	53.2	57.2	28.0
	3D zero position	25.7	85.5	12.3	42.3	54.1	56.7	28.2
Proxy3D	Original	25.8	90.9	13.7	44.5	55.2	58.3	25.1
	Text-only	19.4	65.6	9.7	34.7	45.8	48.1	–
	3D position shuffle	24.9	87.5	12.8	43.2	56.2	59.3	23.8
	3D zero position	24.6	86.7	12.7	43.2	54.7	57.6	25.7
QuatRoPE	Original	–	–	–	–	53.3	56.3	–
	Text-only	–	–	–	–	45.4	47.4	–
	3D position shuffle	–	–	–	–	53.2	56.2	–
	3D zero position	–	–	–	–	54.4	56.7	–

of another token; *3D zero position* sets the 3D coordinate of every visual token to zero (for Spatial-MLLM, the VGGT geometry features), so that the tokens keep their appearance but carry no 3D position at all; for QuatRoPE, every token then receives the same rotation.

Question answering. Table 1 reports the results. Shuffling the 3D positions changes EM@1 by at most 1.2 points on ScanQA and 2.2 points on SQA3D across all evaluated models; Video-3D LLM's ScanQA EM@1 is unchanged, and Proxy3D's SQA3D EM@1 even rises by 1.0. For LLaVA-3D and Proxy3D, the ScanQA drops (1.0 and 0.9 EM@1) are a small fraction of their 8.7 and 6.4 point gaps to text-only inference; in other words, most of what the scene contributes survives when every token is told it is somewhere else in the room. Per question, shuffling flips at most 3.9% of ScanQA and 6.4% of SQA3D questions from correct to incorrect, and at most 2.9% and 5.6% from incorrect to correct (Appendix E.1). Removing the 3D positions altogether is equally harmless: with all coordinates set to zero, EM@1 changes by at most 1.2 points on ScanQA and 1.3 points on SQA3D. The same holds on R3D-Bench, whose questions ask for metric distances, lengths, and volumes: shuffling changes the average over its 15 question types by at most 1.3 points (Proxy3D), and by only 0.3 points for LLaVA-3D and Video-3D LLM, and setting all coordinates to zero changes it by at most 0.7 points (Appendix E.2). Whether the tokens are given the wrong 3D positions or none at all, the answers barely change.

Grounding. One might expect grounding to be different, since the output is itself a 3D box. Table 2 shows it is not. Proxy3D loses 2.5 Acc@0.25 under position shuffling, and QuatRoPE, whose contribution is precisely a 3D rotary encoding, loses 0.1. The breakdown is informative: on the *Unique* split, where the target class appears once, shuffling costs under one point; the small drop is concentrated on the *Multiple* split, where several same-class candidates must be disambiguated by spatial relations. Both models operate over object proposals, so the referring expression can be matched to the semantic content of a proposal and the box read off from the proposal itself, without consulting the injected coordinates.

Table 2: **Inference-time perturbation of 3D positions on ScanRefer.** Permuting the 3D positions of visual tokens barely matters for grounding. "–" denotes not evaluated.

		Acc@0.25			Acc@0.5		
Model	Setting	All	Uni.	Mult.	All	Uni.	Mult.
Proxy3D	Original	49.4	77.2	43.1	44.8	71.1	38.9
	Pos. shuffle	46.9	76.4	40.2	42.6	70.6	36.2
QuatRoPE	Original	57.7	88.7	50.7	52.2	81.8	45.4
	Pos. shuffle	57.6	88.9	50.4	51.9	81.9	45.1

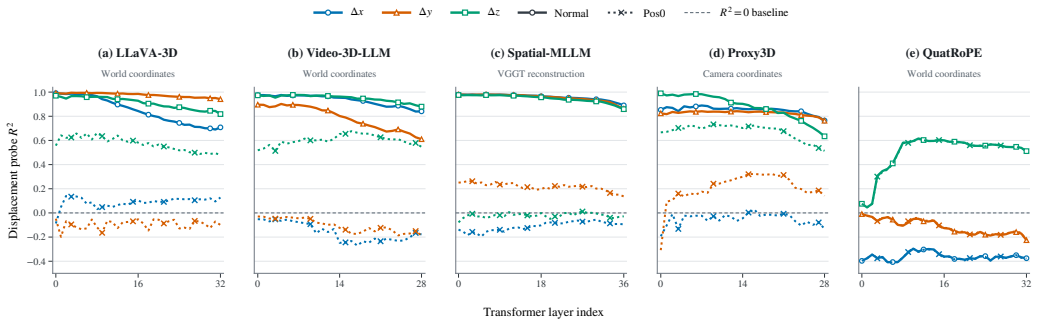

Figure 2: **Probing the 3D offset between two tokens from their hidden states** with a simple probing head, per axis and per layer. The dashed line marks R^2=0, i.e., predicting the mean offset. *Normal* (solid): original inputs. *Pos0* (dotted): 3D zero position. The offset remains readable in the deepest layers of every model except QuatRoPE.

3.3 The geometry is encoded but not read

Insensitivity to positions could mean that the positional signal is destroyed early in the network. We test this with a simple probing head, a small MLP trained on the frozen hidden states of each layer (Figure 2; Appendix B.4).

Relative-offset probe. The probing head maps the hidden states of two tokens to their offset Δxyz; we report R^2 per axis. The question is whether these offsets survive the transformer, and they do. For LLaVA-3D, Video-3D LLM, Spatial-MLLM, and Proxy3D, R^2 stays above 0.6 on every axis at every layer, and at the last layer it is still 0.61–0.94. QuatRoPE is the exception: Δx and Δy are not decodable at any layer ($R^2 < 0$), and only Δz is partially recovered, consistent with a rotary encoding that acts on queries and keys rather than being added to the hidden states.

Zero-position control. We retrain the probes on hidden states obtained with the 3D zero position of Section 3.2 (Pos0 in Figure 2). R^2 falls well below the original curves for the four models whose geometry enters the hidden states. Δx drops to near or below zero in all four; the remaining signal is largely confined to Δz for LLaVA-3D, Video-3D LLM, and Proxy3D, plausibly because height can partly be inferred from appearance (floors, tabletops, ceilings), with weaker residuals on Δy for Spatial-MLLM and Proxy3D ($R^2 \lesssim 0.3$). For QuatRoPE, the Pos0 curves nearly coincide with the original ones, as expected when the geometry does not enter the hidden states.

Taken together, Sections 3.2 and 3.3 show that the injected geometry survives inside the LLM in a form that a simple probing head can decode, yet the answer does not depend on it. We interpret this as a supervision problem rather than an architectural one: on ScanQA, SQA3D, and proposal-based ScanRefer, a decoder that matches the question to token semantics is already rewarded, and nothing forces it to identify *which* tokens the question refers to and to read their coordinates.

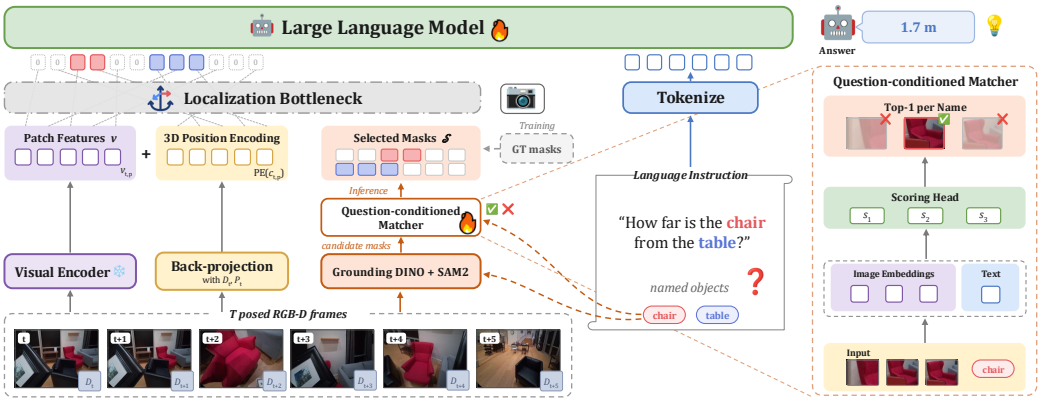

Figure 3: **Localize, then reason.** Grounding DINO and SAM2 produce candidate masks for each object named in the question; a matcher selects one mask per object; only the patch tokens covered by the selected masks, together with their 3D position encoding, reach the LLM, and all other visual tokens are set to zero.

4 Method: Localize, Then Reason

Section 3 shows that current 3D LLMs can answer by matching the question to the semantic content of visual tokens anywhere in the scene, without reading where those tokens are. Anchor3D removes this shortcut rather than adding a new geometric encoding: it performs the matching explicitly, and then hides everything that was not matched. Given a posed RGB-D sequence and a question, (i) the objects named in the question are detected and segmented in every frame with Grounding DINO (Liu et al., 2024a) and SAM2 (Ravi et al., 2024); (ii) a matcher selects one mask for each named object; and (iii) an LLM answers from the patch tokens covered by the selected masks only, with all other visual tokens set to zero. Figure 3 gives an overview.

4.1 Candidate Objects

A scene is given as T RGB-D frames $\{(I_t, D_t)\}_{t=1}^{T}$ with camera poses $\{P_t\}_{t=1}^{T}$. The object names are extracted from the question q with a vocabulary of object categories together with fixed templates and rules (Appendix C). Each name is detected in every frame by Grounding DINO, and the detected boxes are segmented by SAM2, giving a set of candidate masks $\mathcal{M}_t = \{M_{t,k}\}_k$ per frame. We do not associate masks across frames: an object visible in several frames yields several candidates, any of which can be selected in the next stage.

4.2 Question-conditioned Matching

For each name extracted from q, the matcher scores the candidates of that name in $\bigcup_t \mathcal{M}_t$ and keeps the top one; $\mathcal{S}$ is the set of kept masks. The matcher scores each name from its phrase alone and does not use the spatial relations stated in the question. The matcher is trained with ground-truth object supervision: for each training question, the candidates that belong to the ScanNet instances referred to by the question are positives and all others are negatives. The architecture and training details are given in Appendix C.

4.3 Reasoning over the Localized Tokens

3D-aware patch tokens. Each frame is encoded into patch tokens by the visual encoder, and each patch p of frame t is assigned a 3D world coordinate $\mathbf{c}_{t,p}$, obtained by back-projecting its pixels with D_t and P_t and average-pooling the resulting points. Since the visual encoder merges every two consecutive frames into one temporal patch (Section 5.1), the points of both merged patches are pooled together. The coordinate is encoded and added to the patch feature,

$$\mathbf{x}_{t,p} = \mathbf{v}_{t,p} + \mathrm{PE}(\mathbf{c}_{t,p}). \tag{1}$$

Localization bottleneck. Only the patches that overlap a selected mask are passed on:

$$\tilde{\mathbf{x}}_{t,p} = \begin{cases} \mathbf{x}_{t,p}, & \text{if patch } p \text{ of frame } t \text{ overlaps some } M \in \mathcal{S}, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{2}$$

Since a token covers the same patch location in two merged frames, it is kept if the patch in either frame overlaps a selected mask. If no name is extracted from q (e.g., SQA3D's *What is on my left?*), the bottleneck is bypassed and all tokens are kept. The remaining tokens are zeroed rather than removed, so the sequence length and the position ids of the text are unchanged. For questions posed relative to the observer (e.g., how far an object is from the viewer), the camera pose is additionally provided as a special token. The LLM, Qwen3.5-4B, receives $\{\tilde{\mathbf{x}}_{t,p}\}$ and the question and generates the answer autoregressively.

Training. The LLM is trained with ground-truth masks: $\mathcal{S}$ is replaced by the masks of the objects the question refers to, and the matcher output is used at inference on all benchmarks. The training data combine the training sets of ScanQA and SQA3D with metric questions that we build on ScanNet scenes following R3D-Bench (Horton et al., 2026), whose targets (distances, lengths, volumes) are determined by the 3D extent of specific named objects. ScanNet rooms and the ADT apartment differ in scale, so without normalization the position encoding would not learn the meaning of coordinate values that are either too large or too small for the other domain. We therefore normalize the coordinates of each scene before encoding them: the scene center is subtracted, and all three axes are scaled by the same factor $s = 20/L$, where L is the length of the longest side of the scene, so that this side spans $[-10, 10]$. Since L is measured from the points back-projected with the metric depth D_t and poses P_t, s is known for every scene at both training and inference. The question text is left unchanged, and the metric targets are expressed in the normalized frame: distances and lengths are multiplied by s and volumes by s^3, while all other answers are kept as they are. At inference, a predicted distance or length is divided by s and a predicted volume by s^3 to recover the value in the original metric units. Optimization details are given in Section 5.1.

5 Experiments

5.1 Setup

Benchmarks. Our primary benchmark is R3D-Bench (Horton et al., 2026), which asks 15 types of quantitative questions over egocentric RGB-D videos: multiple-choice comparisons (e.g., which of two objects is taller), metric distances and lengths, and volumes. Multiple-choice questions are scored by accuracy; numeric questions by mean relative accuracy (MRA), which averages over thresholds $\{0.50, \ldots, 0.95\}$ the fraction of predictions whose relative error is within $1 - \theta$ (Yang et al., 2025b). We additionally report ScanQA (CIDEr and EM) and SQA3D (EM).

Baselines. We compare against (a) proprietary and open-weight 2D VLMs given RGB frames only; (b) the same VLMs equipped with depth through a tool-use pipeline (CuTR + Tools); (c) the 3D LLMs analysed in Section 3; and (d) spatial VLMs such as SpatialRGPT (Cheng et al., 2024).

Implementation. We build our Anchor3D based on Qwen3.5-4B, which is then fine-tuned on the combination of training sets of SQA3D, ScanQA and self-built training set using ScanNet scenes (Appendix D), using the AdamW optimizer. The learning rates peak at 1e-5 for the LLM during the warmup phase and the vision encoder is kept frozen. All experiments are conducted with 4×A100-80G. And the training process takes approximately 1 day. Each scene is represented by 32 frames, with each frame's token budget 256. Since Qwen3.5 merges every two consecutive frames into one temporal patch, the LLM receives 4,096 visual tokens per scene.

5.2 Main results

Table 3 reports the results. Without tool use, Anchor3D reaches an average of 44.5 with 4B parameters. It outperforms all open-weight models and is on par with GPT-5.5 (44.7) and Gemini 3.1 Pro (44.6); only Gemini 3 Flash is higher (46.5). Fine-tuning Video-3D LLM, whose 3D position encoding Anchor3D reuses, on our training data yields 34.9, so the gain does not come from the training data alone.

Table 3: Main results on R3D-Bench across 15 spatial question types. **PF%**: parse failure rate (failures excluded from accuracy). **Depth**: whether the system has access to depth. Rows marked $*$ are evaluated by us. Gray rows are proprietary models, reported for reference only. **Bold** marks the best open-weight result with tool use, and without tool use (including end-to-end training).

			Multiple Choice (Acc ↑)						Distance (MRA ↑)								Volume (MRA ↑)				
	Depth	*PF% ↓*	*Gap Fit*	*Nearest*	*Top Higher*	*Wh. Taller*	*Wh. Longer*	*MC Avg*	*How Far*	*Far From Me*	*How Long*	*Much Taller*	*Much Longer*	*Fly Dist.*	*Walk Dist.*	*Dist Avg*	*Volume*	*Pour Left.*	*Pour Room*	*Vol Avg*	*Avg*
With tool use																					
CuTR + Tools (Q3-VL 235B-A22 Think)	✓	**0.0**	71.0	92.1	85.7	88.1	91.7	85.7	82.9	88.6	43.7	23.3	28.2	**80.9**	**58.5**	58.0	43.9	24.4	25.8	31.4	61.9
R3D pipeline (Qwen3-VL 235B-A22 Think)	✓	**0.0**	**95.2**	**95.6**	**90.2**	**92.5**	**95.2**	**93.7**	**89.6**	**95.0**	**82.2**	**66.0**	**58.5**	75.2	55.3	**74.6**	**46.0**	**32.2**	**33.9**	**37.3**	**73.5**
w/o tool use																					
Gemini 3 Flash	✗	0.1	80.6	39.5	63.4	82.8	81.1	69.5	34.6	24.6	63.0	46.7	40.7	47.1	38.9	42.2	33.0	6.2	16.0	18.4	46.5
Gemini 3.1 Pro	✗	0.3	83.9	38.6	66.5	85.5	80.7	71.0	29.9	23.7	57.1	43.1	37.3	39.2	34.3	37.8	27.3	10.1	11.6	16.3	44.6
GPT-5.5	✗	0.0	70.5	40.3	61.9	85.0	77.2	67.0	27.9	31.4	63.1	48.1	48.1	31.2	26.6	39.5	25.7	14.3	19.1	19.7	44.7
Qwen3-VL 235B-A22 Think RGB	✗	0.7	**74.2**	42.5	60.3	**80.6**	79.4	**67.4**	28.2	27.2	42.2	30.2	31.2	33.0	21.2	30.5	14.1	1.5	3.7	6.4	38.0
SpatialRGPT + Median	✓	9.1	50.0	24.0	53.1	70.6	68.8	53.3	16.1	29.9	**44.6**	7.3	11.3	0.0	0.0	15.6	**38.6**	0.0	1.1	13.2	27.7
Video-3D LLM*	✓	21.8	62.9	33.6	50.9	50.5	46.7	48.9	5.1	12.5	12.1	9.6	10.3	17.6	9.0	10.9	4.5	5.5	3.2	4.4	22.3
LLaVA-3D*	✓	6.4	50.0	23.4	50.9	44.8	32.4	40.3	4.2	10.7	17.4	6.4	3.3	7.1	8.4	8.2	3.6	**6.0**	1.3	3.6	18.0
Spatial-MLLM*	✗	0.4	53.2	38.1	50.0	62.2	60.7	52.8	30.0	31.1	26.1	11.3	7.1	15.0	7.4	18.3	28.4	3.9	6.2	12.8	28.7
Proxy3D*	✗	0.3	51.6	33.3	55.4	58.1	55.5	50.8	17.3	27.9	27.7	11.6	14.2	1.1	0.1	14.3	12.4	3.2	6.5	7.4	25.1
End-to-end training																					
Video-3D LLM*	✓	0.8	40.3	31.5	57.6	67.1	67.3	52.8	32.9	31.4	40.1	27.1	28.4	39.3	39.6	34.1	18.0	1.1	2.5	7.2	34.9
Anchor3D	✓	**0.0**	54.8	**60.5**	**61.2**	79.0	**81.1**	67.3	**42.5**	**35.7**	31.2	**36.6**	**35.0**	**48.0**	**48.4**	**39.6**	13.6	**6.0**	**33.8**	**17.8**	**44.5**

Relations versus extents. Anchor3D is the best model without tool use on Nearest, How Far, Fly Distance, Walk Distance, and Pour Room, i.e., on relations between objects and room-scale quantities, and weakest on the extent of a single object (Gap Fit, How Long, Volume, Pour Leftover). This mirrors the diagnosis of Section 3 in reverse: existing 3D LLMs are strong where appearance and language suffice, Anchor3D where the answer has to be read from 3D positions.

Positional resolution limits object extents. We attribute this weakness to the 3D position encoding rather than to the reasoning stage. Following previous RGB-D 3D LLMs, each token receives a single coordinate, the average of the back-projected points in its patch (Section 4.3). The resulting error of a few centimeters is negligible for distances of several meters, but a small object is covered by only a few patches, and its boundary patches, which the bottleneck keeps, also average in background points. The extent read from the tokens is thus biased by an amount comparable to the object itself, which MRA penalizes heavily.

Existing 3D QA benchmarks. Table 4 reports ScanQA and SQA3D. Anchor3D reaches 26.7 EM on ScanQA and 55.9 EM on SQA3D, comparable to LLaVA-3D and Spatial-MLLM but below Video-3D LLM. The controlled comparison is the *w/o matching* variant in Table 5, which shares our backbone, training data, and token budget. Removing the bottleneck lowers EM from 26.7 to 25.1 on ScanQA and from 55.9 to 51.2 on SQA3D.

5.3 Does the model now use 3D positions?

We apply the same inference-time perturbations to our model (Table 5); the shuffle permutes coordinates among the retained tokens only, so every token still receives a coordinate lying on a named object. We further ablate the matching bottleneck: *w/o matching* feeds all visual tokens to the LLM without the localization bottleneck, and *w/o metric data* trains on QA questions only. We also apply the shuffle to the *w/o matching* model, where it permutes positions among all visual tokens.

Anchor3D reads its 3D positions. Unlike the models of Section 3, Anchor3D is clearly affected by the position shuffle. Its R3D-Bench average falls from 44.5 to 33.5, and its EM falls by 1.6 points on ScanQA and 3.9 points on SQA3D. Video-3D LLM, whose 3D position encoding Anchor3D reuses unchanged, loses 0.0 and 0.7 EM on the same two benchmarks (Table 1), and on both benchmarks the drop of Anchor3D exceeds that of every model in Table 1. The *w/o matching* model, trained on the same data as Anchor3D, loses only 0.3 points on R3D-Bench and 1.0 and 0.6

Table 4: Results on ScanQA and SQA3D.

Model	ScanQA		SQA3D
	C ↑	EM ↑	EM ↑
3D LLM-based			
LEO	80.0	24.5	50.0
Chat-Scene	87.7	21.6	54.6
Descrip3D	93.7	22.7	55.7
Correspondence-based			
GPT4Scene (HDM)	90.9	28.2	57.4
Video-3D LLM	102.1	30.1	58.6
Spatial-MLLM-4B	91.8	26.3	55.9
3DRS	104.8	30.3	60.6
Representation-based			
LLaVA-3D	91.7	27.0	55.6
LEO-VL	100.4	22.6	60.8
Proxy3D (7B)	93.6	25.2	57.5
Ours	91.7	26.7	55.9

Table 5: **Positional sensitivity and ablations.** R3D-Bench: average over all 15 question types (accuracy for multiple choice, MRA for numeric questions; the Avg column of Table 3). ScanQA and SQA3D: EM. *Pos. shuffle* permutes 3D positions among the retained tokens at inference; *w/o match* removes the localization bottleneck and keeps all visual tokens; *w/o metric data* removes our R3D-Bench-style metric questions from training.

Setting	R3D-Bench Avg ↑	ScanQA EM ↑	SQA3D EM ↑
Ours (full)	**44.5**	**26.7**	**55.9**
Pos. shuffle	33.5	25.1	52.0
w/o match	32.6	25.1	51.2
w/o match + shuffle	32.3	24.1	50.6
w/o metric data	30.2	23.6	51.5

EM on ScanQA and SQA3D, so the change is not due to the training data. The same encoding goes from ignored to consulted once the tokens that reach the LLM are restricted to the objects named in the question.

Ablations. Both components are necessary. Without the bottleneck, the R3D-Bench average drops from 44.5 to 32.6; without metric training data, it drops to 30.2, and ScanQA and SQA3D also degrade (23.6 and 51.5 EM), so metric supervision helps general 3D QA as well. The shuffle rows show where the gain of the bottleneck comes from. With shuffled positions, the full model (33.5) is at the level of the *w/o matching* model, both with and without the shuffle (32.3 and 32.6). The bottleneck still removes the same distractor tokens in this setting, so its benefit does not come from a shorter or cleaner context alone; it comes from the 3D positions of the matched objects, and it nearly vanishes when those positions are wrong.

6 Conclusion

We asked whether 3D LLMs use the 3D positions they are given, and found that five recent models, including one built around a 3D rotary encoding, are insensitive to permuting those positions across the benchmarks we evaluate (ScanQA, SQA3D, and ScanRefer), while four of them still encode relative 3D offsets in a form that a simple probing head can decode. The geometry is present but not read, because the benchmarks reward matching the question to token semantics and never require binding named entities to their coordinates. We proposed to make this binding an explicit bottleneck, so that the LLM sees only the patch tokens of the objects named in the question together with their 3D positions, and trained it with metric supervision following R3D-Bench. Beyond the specific model, our diagnosis suggests that inference-time positional perturbation is a cheap and informative check that future 3D LLMs and benchmarks should report.

AI USE STATEMENT

In this work, we used generative AI tools to propose or refine hypotheses, design or provide feedback on research methodology and experiments, implement methods, support qualitative and thematic data analysis, and interpret results. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, assist with translation, or clean and reformat datasets; formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work, as it does not contain formal mathematical results. Additionally, we used generative AI tools for brainstorming, summarizing and analyzing existing literature, suggesting experimental parameters, creating and editing software code, creating or modifying scientific figures, drafting parts of the paper, editing the paper to improve readability, formatting references, and proposing a title and keywords.

We have reviewed all AI-assisted work. Hypotheses and experimental designs suggested by AI were critically assessed by the authors against prior work and our research goals, and all final decisions on hypotheses and methodology were made by the authors. AI-suggested experimental parameters were treated only as starting points and were validated through our own experiments. For the qualitative and thematic analysis, codes and themes proposed by AI were checked against the raw data by three authors, and disagreements were resolved through discussion. Interpretations of results suggested by AI were verified against the actual experimental outputs, and no claim in the paper goes beyond what our data support. AI-generated literature summaries were checked against the original papers, and every reference was manually verified to exist and to support the statement it is cited for. All AI-drafted or AI-edited text and figures were revised and approved by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All experiments use publicly available data (ScanNet, ScanQA, SQA3D, ScanRefer, and R3D-Bench) and publicly released model checkpoints. For the diagnosis, the five 3D LLMs are evaluated with their released checkpoints and without fine-tuning; Appendix A.1 and Table 6 specify, for each model, the tensor at which each perturbation is applied, and Appendix A.2 gives the exact definitions of the position shuffle, the 3D zero position, and text-only inference. The "Original" rows of Table 1 are re-run by us under the same evaluation code rather than copied from the original papers. The probing study is fully specified in Appendix B: the scene-level train/validation split (Appendix B.2), the distance-stratified pair sampling and the resulting pair counts (Appendix B.3, Table 7), the probe architecture (Appendix B.4), and the optimizer, schedule, training budget, and random seed, with no hyperparameter search (Appendix B.5). For Anchor3D, Section 4 describes each stage of the pipeline; Appendix C gives the matcher's object-name extraction, candidate generation, scoring function, label construction, training data, and checkpoint selection; and Section 5.1 lists the backbone, frame and token budgets, optimizer, learning rate, and compute. The evaluation metrics (accuracy and MRA on R3D-Bench, CIDEr and EM on ScanQA and SQA3D) and the treatment of parse failures are stated in Section 5.1 and Table 3.

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. Springer, 2020.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4971–4980. IEEE, 2018.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding, 2022. URL https://arxiv.org/abs/2112.10482.

Ellis Brown, Jihan Yang, Shusheng Yang, Rob Fergus, and Saining Xie. Benchmark designers should" train on the test set" to expose exploitable non-visual shortcuts. *arXiv preprint arXiv:2511.04655*, 2025.

Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. Rubi: Reducing unimodal biases in visual question answering, 2020. URL https://arxiv.org/abs/1906.10169.

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

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, Angel X. Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language, 2020. URL https://arxiv.org/abs/1912.08830.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, 2024b. URL https://arxiv.org/abs/2403.06764.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26418–26428. IEEE, 2024c.

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*, 2024d.

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

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 91–104. IEEE, 2025.

Jiading Fang, Xiangshan Tan, Shengjie Lin, Igor Vasiljevic, Vitor Guizilini, Hongyuan Mei, Rares Ambrus, Gregory Shakhnarovich, and Matthew R Walter. Transcrib3d: 3d referring expression resolution through large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9737–9744. IEEE, 2024.

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. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

Maxwell Horton, Wei Lu, Quan Tran, Yury Astashonok, Kirmani Ahmed, Babak Damavandi, Anuj Kumar, Xiao Zhang, and Seungwhan Moon. R3d: Quantitative 3d spatial reasoning for egocentric wearables. *arXiv preprint arXiv:2607.02921*, 2026.

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024.

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. *arXiv preprint arXiv:2311.12871*, 2023.

Jerry Jiang, Haowen Sun, Denis Gudovskiy, Yohei Nakata, Tomoyuki Okuno, Kurt Keutzer, and Wenzhao Zheng. Proxy3d: Efficient 3d representations for vision-language models via semantic clustering and alignment. *arXiv preprint arXiv:2605.08064*, 2026.

Weitai Kang, Haifeng Huang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Robin3d: Improving 3d large language model via robust instruction tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3905–3915. IEEE, 2025.

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3707–3717. IEEE, 2025.

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

Xiongkun Linghu, Jiangyong Huang, Ziyu Zhu, Baoxiong Jia, and Siyuan Huang. Scenecot: Eliciting grounded chain-of-thought reasoning in 3d scenes. In *International Conference on Learning Representations*, pp. 2754–2779, 2026.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024a. URL https://arxiv.org/abs/2303.05499.

Zuyan Liu, Yuhao Dong, Yongming Rao, Jie Zhou, and Jiwen Lu. Chain-of-spot: Interactive reasoning improves large vision-language models. *arXiv preprint arXiv:2403.12966*, 2024b.

Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso de Melo, Jianwen Xie, and Alan Yuille. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. *Advances in Neural Information Processing Systems*, 38:140751–140774, 2025.

Xianzheng Ma, Tao Sun, Shuai Chen, Yash Bhalgat, Jindong Gu, Angel Chang, Iro Armeni, Iro Laina, Songyou Peng, and Victor Prisacariu. Do 3d large language models really understand 3d spatial relationships? In *International Conference on Learning Representations*, pp. 32591–32610, 2026.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

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.

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*, pp. 157277–157302, 2026.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024.

Dongsheng Wang, Dawei Su, and Hui Huang. Zero-shot 3d question answering via hierarchical view-to-token transportation, 2026. URL https://arxiv.org/abs/2606.03100.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *Advances in neural information processing systems*, 38: 13569–13597, 2025.

Runsen Xu, Zhiwei Huang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Vlm-grounder: A vlm agent for zero-shot 3d visual grounding. *arXiv preprint arXiv:2410.13860*, 2024.

Jianing Yang, Xuweiyi Chen, Nikhil Madaan, Madhavan Iyengar, Shengyi Qian, David F Fouhey, and Joyce Chai. 3d-grand: A million-scale dataset for 3d-llms with better grounding and less hallucination. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29501–29512. IEEE, 2025a.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643. IEEE, 2025b.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models, 2026. URL https://arxiv.org/abs/2412.04467.

Yupu Yao and Bowen Yang. Geope: A unified geometric positional embedding for structured tensors. *arXiv preprint arXiv:2512.04963*, 2025.

Tatiana Zemskova and Dmitry Yudin. 3dgraphllm: Combining semantic graphs and large language models for 3d scene understanding, 2025. URL https://arxiv.org/abs/2412.18450.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, 2023.

Yiming Zhang, ZeMing Gong, and Angel X. Chang. Multi3drefer: Grounding text description to multiple 3d objects, 2023. URL https://arxiv.org/abs/2309.05251.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference, 2025. URL https://arxiv.org/abs/2410.04417.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding, 2025. URL https://arxiv.org/abs/2412.00493.

Hongyan Zhi, Peihao Chen, Junyan Li, Shuailei Ma, Xinyu Sun, Tianhang Xiang, Yinjie Lei, Mingkui Tan, and Chuang Gan. Lscenellm: Enhancing large 3d scene understanding using adaptive visual preferences. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3761–3771. IEEE, 2025.

Shengli Zhou, Minghang Zheng, Feng Zheng, and Yang Liu. Scalable object relation encoding for better 3d spatial reasoning in large language models. *arXiv preprint arXiv:2603.24721*, 2026.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024a.

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. Springer, 2024b.

A PERTURBATION PROTOCOL

This section specifies how the inference-time perturbations of Section 3.2 are implemented for each model. All experiments use the publicly released checkpoints without any fine-tuning; a perturbation modifies a single tensor at inference and leaves every other part of the forward pass unchanged.

A.1 WHERE EACH MODEL INJECTS 3D POSITIONS

The five models attach 3D geometry to their visual tokens at different points of the pipeline (Table 6). Our perturbations act on that coordinate tensor (for Spatial-MLLM, the geometry features) at the point where it enters the network, so that the perturbed model sees exactly the same visual features and the same sequence, with only the geometry changed.

Table 6: **How each model injects 3D positions.** The perturbations of Section 3.2 modify the coordinate tensor at the listed injection point. Spatial-MLLM has no explicit coordinate channel; for it, the perturbation acts on the VGGT geometry features instead.

Model	Visual token	Injection of the 3D coordinate	Perturbable
LLaVA-3D	voxel-pooled patch	3D positional embedding added to patch features before the projector	✓
Video-3D LLM	frame patch	3D position encoding added to patch features before the projector	✓
Proxy3D	object proxy	3D positional embedding added to each object token	✓
QuatRoPE	point-cloud object	quaternion rotary encoding applied to the query and key matrices of every attention layer	✓
Spatial-MLLM	frame patch	no explicit coordinate; geometry enters through VGGT geometry features, which are permuted or zeroed	✓

LLaVA-3D and Video-3D LLM. Both models compute a positional embedding from the 3D coordinate of each patch and add it to the patch feature before the multimodal projector. We replace the coordinate tensor that feeds this embedding; the projector, the LLM, and the text-side position ids are untouched.

Proxy3D. Proxy3D aggregates the scene into a set of object proxy tokens and adds a positional embedding computed from each proxy's 3D position. We replace the position of each proxy token before the embedding is computed. The proxy features themselves, which carry the appearance and semantics of the object, are unchanged.

QuatRoPE. QuatRoPE is a point-cloud model: following Chat-Scene and 3DGraphLLM (Huang et al., 2024; Zemskova & Yudin, 2025), it represents the scene as object tokens extracted from the point cloud, and injects geometry through a quaternion-based rotary encoding that rotates the query and key vectors of every attention layer as a function of the token's 3D coordinate. Here the perturbation replaces the coordinates used to compute the rotation, so that every attention layer sees the perturbed geometry. This is the strongest form of injection among the five models, and the one whose sole purpose is geometric. Since QuatRoPE does not officially report ScanQA results and takes point-cloud input rather than the RGB-D videos of R3D-Bench, we report neither its ScanQA nor its R3D-Bench performance ("–" in Table 1).

Spatial-MLLM. Spatial-MLLM does not attach a coordinate to its tokens. Its geometric information arrives through the geometry features of a feed-forward 3D reconstruction model (VGGT), which are combined with the 2D visual features of the same patch. Since there is no coordinate tensor, the 3D position shuffle permutes the VGGT geometry features among the visual tokens instead, leaving the 2D visual features unchanged. In the probing study, the probe target is the 3D coordinate of the token given by the VGGT reconstruction.

A.2 Definition of each perturbation

Let a scene provide N visual tokens with features $\{\mathbf{v}_i\}_{i=1}^{N}$ and 3D coordinates $\{\mathbf{c}_i\}_{i=1}^{N}$, where $\mathbf{c}_i \in \mathbb{R}^3$ is expressed in meters in the frame in which the model defines its positions: the world frame for LLaVA-3D, Video-3D LLM, and QuatRoPE, the camera frame for Proxy3D, and the VGGT reconstruction frame for Spatial-MLLM (Figure 2).

Position shuffle. We draw a single random permutation π of $\{1, \ldots, N\}$ and assign token i the coordinate $\mathbf{c}_{\pi(i)}$. The permutation is *global*: it ranges over all visual tokens of the scene at once, across frames, views, and objects, rather than within a frame or within an object. Each token therefore keeps its own appearance but is, in general, told that it lies somewhere else in the room, at a location that may belong to a different object or a different view. The sequence order and the features are unchanged. For Spatial-MLLM, the same permutation π is applied to the VGGT geometry features rather than to coordinates.

3D zero position. We assign every token the coordinate $\mathbf{c}_i = \mathbf{0}$, i.e., the origin of that frame, before the positional embedding or rotary encoding is computed. The tokens keep their features and order but no longer carry any information about where they are; for QuatRoPE, all tokens receive the same rotation, so attention no longer depends on their relative 3D positions. For Spatial-MLLM, the VGGT geometry features are set to zero instead.

Text-only inference. We remove the visual placeholder token (e.g., `<image>`) from the prompt, so that no visual tokens are inserted into the input sequence; the rest of the prompt, including the system prompt, the chat template, and the question, is kept exactly as in the original evaluation. The LLM therefore receives the question alone, with no scene features and, consequently, none of the 3D positional information that the models attach to their visual tokens: the additive 3D embeddings of LLaVA-3D, Video-3D LLM, and Proxy3D, the VGGT geometry features of Spatial-MLLM, and the rotary encoding of QuatRoPE. This corresponds to the natural text-only use of each released checkpoint and measures what the model can answer from the question and its language priors alone.

B Probing protocol

This section describes the probes of Section 3.3. We train a separate probe on the frozen hidden states of every layer of every model, for a total of 161 probes across the five models, and the same number again for the zero-position control (Appendix B.6).

B.1 Hidden states

For each scene we run the unperturbed model once and record the hidden state of every visual token at every layer, using the hidden-state outputs exposed by the backbone. Index 0 is the input embedding, which already contains the model's positional information (the additive 3D embedding for LLaVA-3D, Video-3D LLM, and Proxy3D); the last index is the output of the final normalization layer. Table 7 lists the number of probed layers per model. We use the full hidden width D without dimensionality reduction.

B.2 Scenes and split

Probes are trained on 90 ScanNet scenes and validated on 10 held-out scenes; the split is by scene, so no token pair in the validation set comes from a scene seen during training. We do not hold out

Table 7: **Probing data.** Number of probed hidden-state layers and number of token pairs used for training and validation. The nominal budget is 90 scenes × 6,000 pairs = 540,000 training pairs; smaller counts result from distance buckets that are empty or under-filled in some scenes (Appendix B.3).

Model	Probed layers	Train pairs	Val pairs
LLaVA-3D	33	518,961	58,800
Video-3D LLM	29	521,047	58,800
Spatial-MLLM	37	540,000	60,000
Proxy3D	29	406,539	42,443
QuatRoPE	33	497,595	55,883

a separate test set: the probes have no hyperparameters that are tuned on validation data and are trained for a fixed budget (Appendix B.5), so the validation set is used only for reporting.

B.3 Token-pair sampling

Within each scene with n visual tokens we enumerate all unordered pairs (a, b) with $a < b$ and compute their true 3D distance $\|\mathbf{c}_a - \mathbf{c}_b\|$. Because nearby pairs vastly outnumber distant ones, uniform sampling would be dominated by short offsets. We therefore stratify by distance into five buckets, $[0, 0.5)$, $[0.5, 1)$, $[1, 2)$, $[2, 5)$, and $[5, \infty)$ meters, and draw at most 1,200 pairs uniformly at random from each bucket, for a target of 6,000 pairs per scene. When a bucket contains fewer than 1,200 pairs, all of its pairs are used and the scene contributes fewer than 6,000; this happens mostly in the farthest bucket and most often for Proxy3D, whose small number of object tokens rarely spans more than 5 m, which explains the lower counts in Table 7.

Direction augmentation. Since pairs are enumerated with $a < b$, the raw offset $\mathbf{c}_a - \mathbf{c}_b$ has a non-zero mean under the token ordering. To remove this bias we store both $\boldsymbol{\delta}$ and $-\boldsymbol{\delta}$ for every pair and, during training, swap the two tokens of a pair with probability 0.5 in every batch, negating the target accordingly. At evaluation we report the forward and backward directions separately and average them.

B.4 Probe architecture

Relative-offset probe. Given the hidden states $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^D$ of a token pair at one layer, the probe input is the concatenation

$$\mathbf{x}_{ij} = [\,\mathbf{h}_i\,;\,\mathbf{h}_j\,;\,\mathbf{h}_i - \mathbf{h}_j\,;\,\mathbf{h}_i \odot \mathbf{h}_j\,] \in \mathbb{R}^{4D}, \tag{3}$$

which, e.g., for $D = 4096$ gives 16,384 dimensions. Each input dimension is standardized with the training-set mean and standard deviation and clipped to $[-5, 5]$; the clipping is inactive in practice. The probe is a two-hidden-layer MLP,

$$\mathrm{Linear}(4D, 512) \to \mathrm{GELU} \to \mathrm{Linear}(512, 512) \to \mathrm{GELU} \to \mathrm{Linear}(512, 3), \tag{4}$$

whose three outputs are the predicted offset $\hat{\boldsymbol{\delta}} = (\Delta x, \Delta y, \Delta z)$ in meters. Targets are standardized per axis with training-set statistics for the loss and mapped back to meters for reporting. We refer to this probe as the *simple probing head* in the main text.

B.5 Training

Every probe is trained independently, with no weight sharing across layers or models, using AdamW (learning rate 10^{-3}, weight decay 10^{-4}), a batch size of 4,096, and a cosine learning-rate schedule annealed to 10^{-5}. The loss is the mean squared error on the standardized targets. All probes are trained for a single epoch (about 100–130 steps, depending on the model) with random seed 0; no early stopping and no hyperparameter search are performed. This is a deliberately small training budget, so the reported errors should be read as upper bounds, and the reported R^2 as lower bounds, on what is recoverable from the hidden states: a longer-trained probe could only decode the geometry better, which would strengthen rather than weaken the conclusion of Section 3.3.

B.6 METRICS AND BASELINES

We report, on the validation scenes, the Spearman rank correlation between predicted and true values per axis, and the coefficient of determination R^2. The *zero-position control* (Pos0 in Figure 2) asks whether the probe reads the injected 3D positions rather than other cues: a separate probe is retrained from scratch, with the same pairs, architecture, and training protocol, on hidden states obtained under the 3D zero position of Appendix A.2, with the original coordinates as targets.

C MATCHER

The matcher selects, for each object named in the question, one candidate mask from the posed video. Grounding DINO, SAM2, and SigLIP are kept frozen; only a small scoring head is trained.

Object names. Object phrases are extracted from the question with a vocabulary of object categories together with fixed templates and rules, e.g., *How far is the white chair from the green pear?* $\rightarrow$ *white chair*, *green pear*. Attribute modifiers are kept; phrases are lower-cased and articles removed. Names come from the question text only.

Candidates. Each name is passed to Grounding DINO (Liu et al., 2024a), and the detected boxes are segmented by SAM2 (Ravi et al., 2024) in each of the $T{=}32$ frames. Candidates are generated per name, so an object can only be selected among the candidates of its own name. Masks are not tracked across frames; every (frame, mask) pair is a separate candidate.

Scoring. For each candidate we form two views: a context crop around the mask, and a masked crop in which pixels outside the mask are set to gray. A frozen SigLIP (Zhai et al., 2023) encodes both views and the text "a photo of {name}.", giving cosine similarities s^{ctx} and s^{mask}. A reliability head, an MLP ($12 \rightarrow 64 \rightarrow 1$), takes the two similarities, their differences from the mean and the maximum over the candidates of the same name, the Grounding DINO confidence, the SAM2 quality score, and the mask area, aspect ratio, fill ratio, and an image-border truncation flag. The final score is

$$\mathrm{score} = 40 \cdot \tfrac{1}{2}\big(s^{\mathrm{ctx}} + s^{\mathrm{mask}}\big) + 4 \cdot \tanh\big(\mathrm{MLP}(\cdot)\big), \tag{5}$$

so the head applies a bounded correction to the SigLIP ranking. At inference, each name selects its top-scoring candidate, and $\mathcal{S}$ is the set of selected masks. The matcher uses the name phrases only; it does not reason over the full question or impose joint constraints between the named objects.

Supervision. Labels are derived from the ground-truth oriented 3D boxes of ScanNet instances. The valid-depth pixels of a candidate mask are back-projected to world coordinates with D_t and P_t. The candidate is a positive for a name if the coordinate-wise median of these points lies inside the box of the referred instance, or if at least 30% of the points do; boxes are not enlarged. A name may have several positives, e.g., the same object in different frames. We train with a multi-positive ranking loss, a margin loss on hard negatives, and a KL term that keeps the score distribution close to that of the original SigLIP similarities. Ground-truth boxes define the labels only and are never used at inference.

Training and evaluation. The head is trained on 2,588 questions from 128 ScanNet training sequences, and the checkpoint (epoch 8) is selected on held-out ScanNet validation questions; no data from the evaluation domain is used. On the ADT questions of R3D-Bench, all selected masks satisfy the criterion above in 63.6% of cases.

D TRAINING DATA

Table 8 lists the composition of the Anchor3D training set.

Table 8: **Training data.**

Source	Questions	Share (%)
R3D-style, distance/size (7 types)	115,207	33.9
R3D-style, other types	118,625	34.9
SQA3D (train)	79,445	23.4
ScanQA (train)	26,515	7.8
Total	339,792	100.0

E Additional results

E.1 Correctness flips under position shuffle

Table 9 compares, question by question, the correctness of the prediction under the 3D position shuffle with that on the original input. *C→W* is the fraction of questions answered correctly under the original input and incorrectly under the shuffle, and *W→C* the reverse. Correctness is measured with EM@1.

Table 9: **Correctness flips under the 3D position shuffle** on ScanQA (val) and SQA3D (val), in % of questions. C→W: correct under the original input, incorrect under the shuffle. W→C: incorrect under the original input, correct under the shuffle. "–" denotes not evaluated.

Method	ScanQA		SQA3D	
	C→W	W→C	C→W	W→C
LLaVA-3D	3.1	2.0	6.4	4.9
Video-3D LLM	1.4	1.6	2.4	1.7
Spatial-MLLM	3.8	2.6	4.8	2.6
Proxy3D	3.9	2.9	4.6	5.6
QuatRoPE	–	–	0.1	0.1

E.2 Position perturbations on R3D-Bench

Table 10 gives the per-type R3D-Bench results behind the R3D Avg column of Table 1. Setting A is the released checkpoint with the original input; setting B applies the 3D position shuffle and setting C the 3D zero position of Appendix A.2. All settings use the same loose answer parsing, with parse failures excluded from accuracy. Across the four models, the average changes by between -1.3 and $+0.3$ points under the shuffle and by between -0.7 and $+0.6$ points under the zero position, and the MC, distance, and volume averages by at most 3.1 points under either perturbation.

Table 10: **3D position perturbations on R3D-Bench, per question type.** A: original input. B: 3D position shuffle. C: 3D zero position. Δ: difference to A in absolute points (for PF%, a positive Δ means more parse failures than under A). Scores use loose answer parsing with parse failures excluded from accuracy; the A rows are those of Table 3.

			Multiple Choice (Acc ↑)						Distance (MRA ↑)								Volume (MRA ↑)				
	Depth	*PF% ↓*	*Gap Fit*	*Nearest*	*Top Higher*	*Wh. Taller*	*Wh. Longer*	*MC Avg*	*How Far*	*Far From Me*	*How Long*	*Much Taller*	*Much Longer*	*Fly Dist.*	*Walk Dist.*	*Dist Avg*	*Volume*	*Pour Left.*	*Pour Room*	*Vol Avg*	*Avg*
Video-3D LLM (A)	✓	21.8	62.9	33.6	50.9	50.5	46.7	48.9	5.1	12.5	12.1	9.6	10.3	17.6	9.0	10.9	4.5	5.5	3.2	4.4	22.3
Video-3D LLM (B)	✓	21.4	59.7	34.8	50.9	51.5	45.5	48.5	5.0	11.6	13.4	10.3	9.6	18.5	7.7	10.9	3.7	5.5	3.2	4.1	22.0
Δ (B−A)		−0.4	−3.2	+1.2	0.0	+1.0	−1.2	−0.4	−0.1	−0.9	+1.3	+0.7	−0.7	+0.9	−1.3	0.0	−0.8	0.0	0.0	−0.3	−0.3
Video-3D LLM (C)	✓	22.2	62.9	34.9	50.4	51.3	45.5	49.0	5.6	12.7	14.8	10.3	10.2	17.5	6.1	11.0	4.1	5.3	3.0	4.1	22.3
Δ (C−A)		+0.4	0.0	+1.3	−0.5	+0.8	−1.2	+0.1	+0.5	+0.2	+2.7	+0.7	−0.1	−0.1	−2.9	+0.1	−0.4	−0.2	−0.2	−0.3	0.0
LLaVA-3D (A)	✓	6.4	50.0	23.4	50.9	44.8	32.4	40.3	4.2	10.7	17.4	6.4	3.3	7.1	8.4	8.2	3.6	6.0	1.3	3.6	18.0
LLaVA-3D (B)	✓	6.9	50.0	23.4	50.9	41.4	33.2	39.8	4.4	13.6	18.6	6.0	3.3	9.3	9.8	9.3	3.1	6.4	1.2	3.6	18.3
Δ (B−A)		+0.5	0.0	0.0	0.0	−3.4	+0.8	−0.5	+0.2	+2.9	+1.2	−0.4	0.0	+2.2	+1.4	+1.1	−0.5	+0.4	−0.1	0.0	+0.3
LLaVA-3D (C)	✓	8.8	50.0	23.7	50.9	44.6	32.7	40.4	3.0	8.3	19.8	6.4	3.5	2.8	3.6	6.8	2.6	6.5	1.5	3.5	17.3
Δ (C−A)		+2.4	0.0	+0.3	0.0	−0.2	+0.3	+0.1	−1.2	−2.4	+2.4	0.0	+0.2	−4.3	−4.8	−1.4	−1.0	+0.5	+0.2	−0.1	−0.7
Spatial-MLLM (A)	✗	0.4	53.2	38.1	50.0	62.2	60.7	52.8	30.0	31.1	26.1	11.3	7.1	15.0	7.4	18.3	28.4	3.9	6.2	12.8	28.7
Spatial-MLLM (B)	✗	0.5	51.6	35.1	51.3	61.5	58.7	51.7	29.6	29.9	24.7	11.9	6.1	13.6	6.7	17.5	31.4	3.6	4.8	13.3	28.0
Δ (B−A)		+0.1	−1.6	−3.0	+1.3	−0.7	−2.0	−1.1	−0.4	−1.2	−1.4	+0.6	−1.0	−1.4	−0.7	−0.8	+3.0	−0.3	−1.4	+0.5	−0.7
Spatial-MLLM (C)	✗	0.3	54.8	34.1	51.8	58.8	56.1	51.1	27.4	30.9	35.7	15.0	9.7	6.4	2.6	18.2	31.5	3.6	4.9	13.4	28.2
Δ (C−A)		−0.1	+1.6	−4.0	+1.8	−3.4	−4.6	−1.7	−2.6	−0.2	+9.6	+3.7	+2.6	−8.6	−4.8	−0.1	+3.1	−0.3	−1.3	+0.6	−0.5
Proxy3D (A)	✗	0.3	51.6	33.3	55.4	58.1	55.5	50.8	17.3	27.9	27.7	11.6	14.2	1.1	0.1	14.3	12.4	3.2	6.5	7.4	25.1
Proxy3D (B)	✗	0.4	50.0	28.9	50.4	56.4	52.6	47.7	16.3	27.2	26.9	11.2	13.3	1.0	0.1	13.7	11.9	3.6	6.8	7.4	23.8
Δ (B−A)		+0.1	−1.6	−4.4	−5.0	−1.7	−2.9	−3.1	−1.0	−0.7	−0.8	−0.4	−0.9	−0.1	0.0	−0.6	−0.5	+0.4	+0.3	0.0	−1.3
Proxy3D (C)	✗	0.9	54.8	32.9	53.1	62.1	56.6	51.9	19.6	30.6	25.6	10.2	10.1	2.1	0.4	14.1	16.9	4.2	6.7	9.3	25.7
Δ (C−A)		+0.6	+3.2	−0.4	−2.3	+4.0	+1.1	+1.1	+2.3	+2.7	−2.1	−1.4	−4.1	+1.0	+0.3	−0.2	+4.5	+1.0	+0.2	+1.9	+0.6