# OneCanvas: 3D Scene Understanding via Panoramic Reprojection

**Bartłomiej Baranowski**[1] **Dave Zhenyu Chen**[2] **Matthias Nießner**[1]
[1]Technical University of Munich [2]Huawei

Project page: `https://baranowskibrt.github.io/onecanvas/`
Code and model: `https://github.com/baranowskibrt/onecanvas`

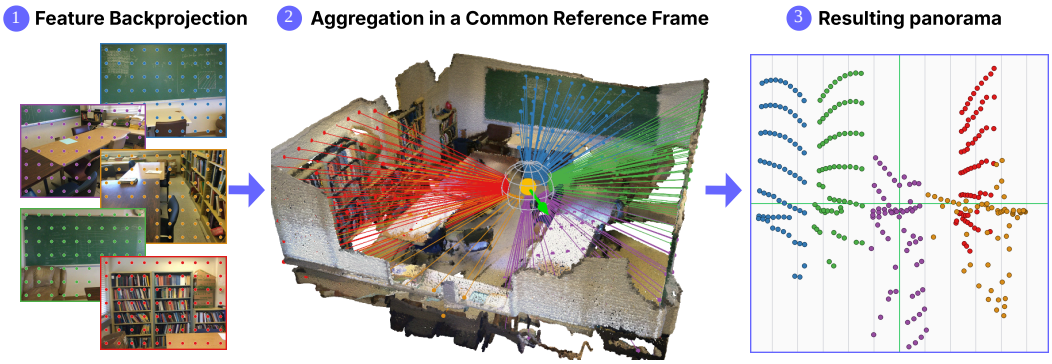



Figure 1: **OneCanvas (1)** backprojects per-frame patch features to 3D, **(2)** aggregates them in a common reference frame centered on a chosen viewpoint, and **(3)** places each patch at its continuous angular position on a panoramic canvas consumed by the VLM as one image.

## Abstract

Existing approaches to 3D scene understanding in Vision-Language Models (VLMs) either rely on complex, model-specific geometry encoders or large training budgets in pursuit of spatial reasoning. Instead, OneCanvas aggregates patch features from all views onto a single equirectangular panoramic canvas. Namely, each patch is unprojected to a 3D world coordinate using its depth and camera pose, then placed on the canvas at the continuous longitude and latitude of that point as seen from the canvas origin, with no rasterization or aggregation across overlapping views. A 3D position embedding of the patch's metric coordinates is added to its feature, restoring the depth lost when collapsing the world position to an angular canvas coordinate. Patches from all frames thus share one spatial coordinate system with no fusion or major architectural modifications of the backbone. The pretrained VLM consumes this representation as if it were an ordinary image. Because the canvas can be centered on any pose of interest, the same representation directly supports situated reasoning from a specific viewpoint, a common requirement in robotics and embodied AI. Thanks to this representation, we can also introduce a spatial pretraining curriculum: by procedurally placing patch features of objects, drawn from real images, at chosen 3D world positions on an otherwise empty canvas, we generate on-the-fly supervision spanning a broad range of spatial reasoning tasks, with answer distributions controlled to reduce spatial reasoning shortcuts. OneCanvas achieves state-of-the-art results on SQA3D and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

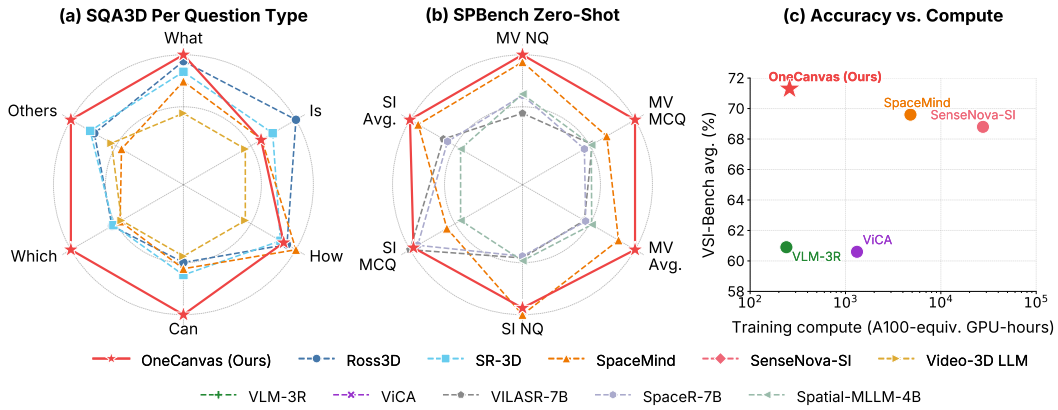



Figure 2: **OneCanvas benchmark comparison.** **(a)** Per-question-type accuracy on SQA3D and **(b)** zero-shot accuracy on SPBench (multi-view and single-image splits), shown as radar charts against competitors closest in benchmark performance. **(c)** Training compute (A100-equivalent GPU-hours, log scale) versus VSI-Bench accuracy.

VSI-Bench and generalizes to out-of-distribution data on SPBench, using an order of magnitude less training compute than its competitors closest in benchmark performance.

## 1 Introduction

Vision-language models have transformed how machines interpret images, video, and natural language jointly, but a great deal of practically valuable understanding is intrinsically three-dimensional. Robots that act in human environments, augmented-reality assistants that ground instructions in a user's physical surroundings, and autonomous agents that plan trajectories through unfamiliar spaces all need to answer questions about *where* things are, how far apart they are, which direction one must turn to face them, and what is or is not visible from a given vantage point. Building VLMs that answer such questions reliably from ordinary video of a scene is the goal of 3D scene understanding.

Most existing 3D-aware VLMs are built on top of pretrained 2D vision-language backbones with no native machinery for spatial reasoning. Adapting them to 3D therefore demands either inducing geometry implicitly from raw video, which is data-inefficient and tends to confound geometry with scene-specific texture priors, or augmenting the backbone with a dedicated geometry encoder. Even with these adaptations in place, the supplied geometry is often not actually used. Recent audits on situated 3D question answering find that text-only baselines match or approach state-of-the-art geometry-aware models on widely cited benchmarks Ma et al. [2026], indicating that the models fall back on statistical regularities of the question and scene class rather than reasoning over the geometric input they are given.

Existing 3D VLMs have nonetheless made substantial progress on 3D reasoning. One family augments the VLM with a dedicated geometric module, including point-cloud tokenisers Chen et al. [2024], Huang et al. [2024b], Wang et al. [2023], Zhu et al. [2024], depth-conditioned video position encodings Zheng et al. [2025b], Zhu et al. [2025], and fused geometry encoders such as VLM-3R Fan et al. [2026] and SpaceMind Zhao et al. [2025] that read 3D structure from monocular video and inject it into the visual backbone. A second family scales spatial supervision: SenseNova-SI Cai et al. [2026] and Cambrian-S Yang et al. [2026] curate large collections of spatial QA pairs to teach a geometrically vanilla VLM to reason about distance and direction by sheer exposure. Both directions deliver concrete improvements on every major spatial benchmark, and together they show that 3D reasoning is reachable from a pretrained VLM if one is willing to invest in either architecture or data.

Both directions also pay a heavy bill. Architectural fusion adds modules that must be jointly trained at scale to align a new geometry encoder with the language backbone, while data scaling demands

curating and sorting millions of high-quality spatial question-answer pairs. Neither directly closes the gap flagged by the audits above: methods in both families still see the scene only piecewise across frames and have no clear handle on which signals in the input actually encode the spatial relationships they are being asked to learn. Instead, we propose OneCanvas (Figure 1): each patch from each view is lifted to a 3D world coordinate using its depth and camera pose, then placed on a single equirectangular canvas at the longitude and latitude of that point as seen from a chosen canvas origin. These spherical coordinates fill the spatial axes of the patch's MRoPE position, while the source-frame index occupies the temporal axis. All patches from all views thus share one spatial coordinate system, and the pretrained VLM consumes the result as if it were an ordinary single-image input, with no major architectural modifications. This representation also enables a spatial pretraining curriculum that would be difficult to assemble in conventional 2D or video pipelines. By procedurally placing patch features of objects, drawn from real images, at chosen 3D world positions on an otherwise empty canvas, we generate on-the-fly supervision for tasks ranging from metric distances to directions, counting, observability, and navigation. Answer distributions are controlled to suppress the scene-statistic shortcuts recent audits have flagged Ma et al. [2026]. OneCanvas reaches state of the art on SQA3D Ma et al. [2023] (65.3 EM@1, 2.3 points above the previous best), VSI-Bench Yang et al. [2025a] (71.3 average), and SPBench Li et al. [2026] (72.1 zero-shot overall, 4.8 points above the next best method), while using an order of magnitude less training compute than its competitors closest in benchmark performance (Figure 2, with the full normalization methodology in Appendix C).

Our contributions are:

- **Panoramic feature reprojection.** A mechanism that aggregates patch features from all views onto a single panoramic canvas, matching the input format the VLM was pretrained on, allowing for flexible choice of canvas origin and orientation to suit different tasks.
- **Spatial pretraining curriculum enabled by the representation.** On-the-fly geometric supervision built by procedurally placing patch features of objects, drawn from real images, at chosen 3D positions on the canvas, with answer distributions controlled to reduce the dataset-statistic shortcuts.

## 2 Related Work

**3D-aware VLMs via architectural extensions.** Most existing 3D VLMs equip a pretrained language backbone with auxiliary modules that supply geometric information Ma et al. [2024a], Chen et al. [2024], Huang et al. [2024b], Wang et al. [2023], Zhu et al. [2024], Li et al. [2024], Zheng et al. [2025b], Fan et al. [2026], Zhao et al. [2025], Wang et al. [2025a], Cheng et al. [2025], Qi et al. [2026]. Early point-cloud-based 3D VLMs Hong et al. [2023], Chen et al. [2024], Huang et al. [2024b], Wang et al. [2023], Zhu et al. [2024], Li et al. [2024] pair a language backbone with a dedicated point cloud tokeniser and train end-to-end on 3D QA data, establishing the feasibility of LLM-based 3D scene reasoning. RGB-D-based 3D VLMs Zheng et al. [2025b], Zhu et al. [2025] instead treat the input as a video sequence and inject 3D position encodings computed from depth maps into the video representation, bridging the video and 3D domains without an explicit point-cloud encoder. HiSpatial Liang et al. [2026] extends this line by feeding metric-scale point maps as auxiliary inputs alongside RGB-D and supervising the VLM through a hierarchical curriculum from geometric perception to abstract spatial reasoning. With the emergence of geometry foundation models Wang et al. [2024a], Leroy et al. [2024], Wang et al. [2025b], Lin et al. [2026], more recent 3D VLMs extract 3D tokens directly from monocular video and fuse them with visual features before the LLM Fan et al. [2026], Zhao et al. [2025], Wang et al. [2025a], Cheng et al. [2025], Qi et al. [2026], Zheng et al. [2025a]. VLM-3R Fan et al. [2026] fuses implicit 3D tokens from a geometry encoder with visual features. SpaceMind Zhao et al. [2025] adds a Camera-Guided Modality Fusion module that combines a spatial encoder with a visual encoder. Ross3D Wang et al. [2025a] introduces a reconstructive visual instruction-tuning objective that supplies 3D-aware supervision during fine-tuning of a video VLM. SR-3D Cheng et al. [2025] is a region-promptable VLM that unifies single-view 2D and multi-view 3D inputs in a shared visual token space. GPT4Scene Qi et al. [2026] fine-tunes Qwen2-VL on video sequences of 3D scenes. SD-VLM Chen et al. [2025] introduces a sinusoidal depth positional encoding that adds depth information along the camera's $z$-axis. Loc3R-VLM Qu et al. [2026] pairs camera priors from a pretrained geometry foundation model with auxiliary BEV-layout and situation-modeling objectives. A parallel line uses reinforce-

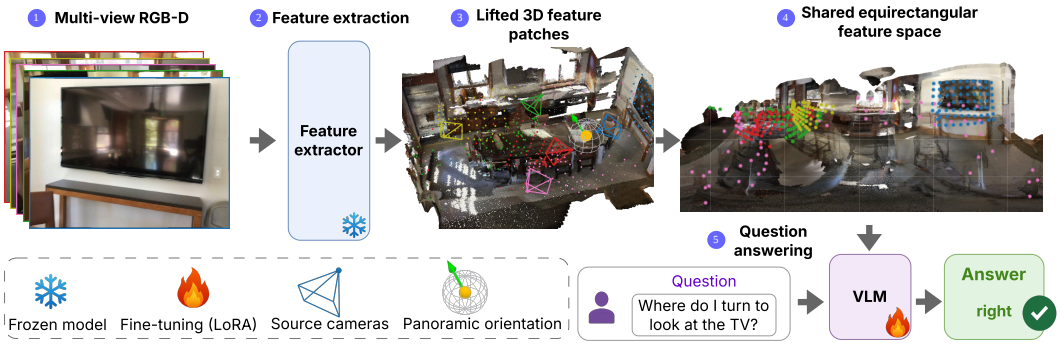



Figure 3: **Method overview.** Multi-view images are encoded by Qwen3-VL's frozen vision encoder, and lifted patches land at continuous positions on a shared equirectangular panorama. The VLM consumes this representation through its native attention layers.

ment learning with spatial reward signals Ouyang et al. [2025], Wu et al. [2025b], Batra et al. [2025], Chen et al. [2026], which is orthogonal to representation choice and could in principle be combined with our canvas. All of these methods introduce new architectural components, auxiliary objectives, or dedicated 3D encoders on top of the base VLM. In contrast, our approach introduces no new architectural components, no auxiliary loss terms, and no dedicated 3D encoder. 3D understanding emerges from how the input is prepared.

**Unified scene representations.** Panoramic image representations offer a 360° field of view that naturally preserves long-range spatial relationships, a property exploited for scene understanding Zheng et al. [2025c] and recently for 3D visual grounding. PanoGrounder Jung et al. [2025] renders multiple equirectangular panoramas from virtual cameras placed around a pre-reconstructed mesh or 3D Gaussian Splatting scene and processes them with a large VLM for grounding. PanoEnv Lin and Zheng [2026] studies VLM spatial intelligence on equirectangular images from synthetic environments, showing that geometric distortion in ERP images remains a challenge for off-the-shelf VLMs. The broader idea of unprojecting 2D features into 3D and reprojecting them onto a canonical 2D view is well-established in autonomous driving through BEV methods Philion and Fidler [2020], Ma et al. [2024b]. Lift3D T et al. [2024] lifts per-view features into a neural field and renders target views, and LiftProj Jia et al. [2025] uses a similar lift-then-project paradigm for panorama stitching. Our work brings this principle to VLM-based 3D scene understanding: per-view features are lifted with metric depth and camera poses, projected once onto a panoramic canvas, and consumed directly by the VLM, with the canvas origin chosen freely to support both scene-centered and pose-centered situated reasoning. Unlike PanoGrounder, LiftProj, and PanoEnv, we do not render an equirectangular image for a vision tower to re-encode: the tower sees the original perspective frames, and only the resulting patch positions are reparameterized. Unlike rendering or Lift3D's ray compositing, the canvas also retains occluded and co-located observations as separate tokens. Controlled comparisons are in Appendix B.2.

## 3 Method

Given a set of $K$ posed RGB images $\{I_k\}_{k=1}^{K}$ with metric depth maps $\{D_k\}$ and pinhole camera intrinsics $C_k = (f_x, f_y, c_x, c_y)$ (per-frame focal lengths and principal point), our goal is to construct a single panoramic representation that a pretrained VLM can ingest directly. The representation is built in two steps (Figure 3): (1) extract per-patch features from each view and lift them into 3D world coordinates (Section 3.1), and (2) reproject the lifted patches onto a shared equirectangular canvas, add a 3D position embedding of each patch's metric coordinates, and assemble the VLM's input sequence with its native 3D-RoPE position encoding (Section 3.2). Training proceeds in two stages (Figure 4): a stage-1 spatial pretraining phase (Section 3.3) followed by stage-2 adaptation on downstream target data (Section 3.4).

### 3.1 Feature Extraction and 3D Lifting

Each image $I_k$ is passed through Qwen3-VL's frozen vision encoder Bai et al. [2025] to obtain a patch-level feature map $F_k \in \mathbb{R}^{H_f \times W_f \times D}$ at the encoder's grid resolution $H_f \times W_f$ and per-patch feature dimension $D$. Each patch $(u, v)$ is lifted to world coordinates using its depth $z = D_k(u, v)$, intrinsics scaled to the feature-map resolution, and the camera-to-world pose $T_k \in \mathrm{SE}(3)$:

$$\mathbf{p}_{u,v}^{\text{world}} = T_k \begin{pmatrix} (u - c_x)\, z/f_x \\ (v - c_y)\, z/f_y \\ z \end{pmatrix}, \tag{1}$$

where $T_k \in \mathrm{SE}(3)$ acts as $T_k\mathbf{x} = R_k\mathbf{x} + \mathbf{t}_k$, and $(u, v, z)$ follow the OpenCV camera convention (x right, y down, z forward). After lifting all $K$ views we obtain a set $\mathcal{P} = \{(\mathbf{p}_i, \mathbf{f}_i, t_i)\}$ of lifted patches, with world coordinates $\mathbf{p}_i \in \mathbb{R}^3$, features $\mathbf{f}_i \in \mathbb{R}^D$, and a per-patch temporal index $t_i \in \{1, \ldots, K\}$ recording the position of the source frame in the input video. We carry $t_i$ through to panoramic placement (Section 3.2), where it becomes the patch's coordinate on the language model's temporal MRoPE axis.

### 3.2 Panoramic Canvas and Position Encoding

We place each lifted patch on an equirectangular canvas defined by its origin $\mathbf{c} \in \mathbb{R}^3$ and orientation $R \in \mathrm{SO}(3)$. The pair $(\mathbf{c}, R)$ is a free design choice and can be set to whatever frame is most natural for the task: an agent pose for situated reasoning, a viewpoint that maximizes scene coverage, or any other reference. Specific conventions used at training and inference are given in Section 4.1.

For each lifted patch, we compute the local offset $\mathbf{q}_i = R^\top(\mathbf{p}_i - \mathbf{c}) = (q_x, q_y, q_z)$ in the canvas frame and continuous spherical coordinates: longitude $\theta_i = \mathrm{atan2}(q_x, q_z)$ and latitude $\phi_i = \arctan(-q_y/\sqrt{q_x^2 + q_z^2})$. The image y-axis points downward in the camera frame, so $q_y$ measures the downward direction and the $-q_y$ in the latitude formula gives positive latitude upward.

**Continuous positions, not a pixel grid.** A natural alternative is to rasterize the lifted patches onto a fixed pixel grid and reduce cell contents through a hand-designed rule (averaging, depth-based selection, last-write). This collapses patches that project to the same or nearby angular position (for instance occluded surfaces along a common viewing ray or co-located observations from overlapping frames), destroying the distinction between surfaces and replacing a pretrained-attention decision with a fixed reduction. We instead keep each lifted patch as its own input token at its continuous $(\phi, \theta)$ position. Co-located patches remain distinct tokens, disambiguated by the 3D position embedding described below, and the model resolves overlap through attention rather than a pre-baked rule. Rasterization, rendered-pixel, and longitude-seam controls are reported in Appendix B.2 and Appendix B.5.

**Position encoding.** The VLM uses its stock 3D Rotary Position Embedding (3D-RoPE) Bai et al. [2025] unchanged. Only the *content* of the position IDs differs from the standard image/video setting. We set $W$ to the longitude $\theta_i \in [-\pi, \pi]$, $H$ to the latitude $\phi_i \in [-\pi/2, \pi/2]$, and $T$ to the source frame index $t_i \in \{1, \ldots, K\}$, with all three linearly rescaled to $[0, 100]$ to stay within the effective positional distribution the model was pretrained on. $T$ retains its standard "temporal / frame-ordering" semantics from video. Equirectangular distortion and the longitude seam are discussed in Appendix B.5.

**3D position embedding.** A 2D VLM picks up only a class-conditioned scale prior from pixels, not a metric reading of the actual layout. We supply explicit per-patch metric position by encoding the canvas-frame offset $\mathbf{q}_i = (q_x, q_y, q_z)$ as the concatenation of (i) per-axis sinusoids $\{\sin(\omega_k q_a), \cos(\omega_k q_a), q_a\}$ for $a \in \{x, y, z\}$ at 16 log-spaced frequencies $\omega_k \in [0.1, 100]$ rad/m, (ii) the same form on the radial primitives $\|\mathbf{q}_i\|$ and $r_{xz} = \sqrt{q_x^2 + q_z^2}$ at 8 log-spaced frequencies $\nu_k \in [0.1, 10]$ rad/m, and (iii) the unit-ray direction $\mathbf{q}_i/\|\mathbf{q}_i\|$ as raw passthrough. These radial primitives directly expose quantities relevant to pairwise distance and horizontal-radial reads such as floor area, rather than requiring the model to synthesise them from per-axis components. The 136-channel encoding is projected by a 2-layer MLP with hidden width 64 and added to the patch feature through a learned scalar gate. Injecting metric position in feature space rather than as additional RoPE dimensions keeps 3D-RoPE's pretrained angular and temporal semantics intact, and lets the VLM attend to metric position through the same content-attention pathway it uses for any other feature.

### 3.3 Spatial Pretraining Curriculum

Spatial reasoning is difficult to acquire from downstream QA alone. The supervision is indirect, target-task distributions are narrow, and scene-appearance cues dominate the gradient before any geometric reading forms, so a model fits the per-template answer marginal long before it learns to read the input. Once that prior is absorbed it is hard to displace, since switching to a geometric readout requires a transient drop in accuracy that the QA gradient is not strong enough to reward. We therefore want supervision in which the answer follows causally from the input geometry, with no language-prior or scene-memorisation shortcut. The panoramic canvas is the natural input format for it: arbitrary 3D content can be placed at any world position with metric coordinates carried in the per-patch embedding, so the geometric answer can be read directly from the placements through the same path the model uses on a real scene.

**Curriculum objects.** Placing recognisable real objects on the canvas can weaken the supervision we are trying to provide, since doors tend to be the same height and bathrooms a fairly consistent size, so a model that has seen enough scenes can often answer geometric questions from class identity alone. In our default formulation we populate a box-sized area of the canvas with a single feature patch sampled from a real scene, and refer to that placement in the prompt by inserting the same patch where a class name would appear. This inline copy keeps the MRoPE position of its slot in the prompt text, like any other token, rather than the longitude/latitude canvas position carried by its counterpart on the canvas. It also receives the additive 3D position embedding. Removing object identity as a shortcut makes the tasks rely primarily on the known geometry of the placements and their references. The patch is drawn from a precomputed pool of activations harvested from training-source scenes of ScanNet Dai et al. [2017], ScanNet++ Yeshwanth et al. [2023], and ARKitScenes Baruch et al. [2021], so the vision encoder treats it as ordinary visual content. An alternative class-labelled scheme based on EmbodiedScan Wang et al. [2024b] object features, and a head-to-head VSI-Bench comparison, are reported in Appendix A.

**Task types.** To learn geometry-based spatial understanding we construct a compact set of tasks generated on-the-fly across six families:

- **Metric measurement.** Surface-to-surface distance between two placed objects, relative-distance comparisons (which of several candidates is closer to a reference, matching VSI-Bench's relative-distance task), and floor-area readout as a room-size signal.
- **Egocentric direction.** "Left / right / front / back" queries from a chosen viewpoint, at three difficulty levels, plus an $o$'clock-direction variant and a camera-perspective variant matching downstream SI-style queries.
- **Multi-turn navigation.** $N$-turn path queries between two placed targets, $N \in \{1, 2, 3, 4\}$.
- **Observability.** When a placement first becomes visible along the frame sequence (appearance-order), and whether a target is visible from a given pose.
- **Counting.** Counts of placed objects under divisibility conditions (parity, divisibility by three) or location conditions (placements on the agent's left, right, front, or back).
- **Multi-target bounding-box readout.** Two to four placed objects, with the prompt asking for each one's 3D bounding box in canonical order.

### 3.4 Two-Stage Training

**Stage 1: spatial pretraining.** We train on the spatial pretraining curriculum alone, teaching the geometric reasoning skills downstream 3D-QA depends on under supervision designed to suppress the scene-prior shortcuts available in real-scene QA. Because each curriculum sample carries far fewer tokens than a full scene panorama, stage 1 is cheap to run and we can afford a long training schedule.

**Stage 2: target adaptation.** We *merge* the stage-1 LoRA weights back into the base language model and train a fresh, lower-rank LoRA adapter ($r$=64) on top, on a weighted mixture of downstream spatial QA (SQA3D, VSI-Bench-style data, and ViCA-family corpora). Merging freezes the stage-1 update into the base weights, and the lower rank of the new adapter limits how far stage 2 can drift from the stage-1 geometric reading while still allowing answer-format adaptation.




Figure 4: **Two-stage training.** Stage 1 trains LoRA Hu et al. [2022] adapters and the 3D position embedding on the spatial pretraining curriculum, where objects placed on an otherwise empty canvas and their inline references carry the geometric signal. Stage 2 merges the stage-1 adapter back into the base language model and trains a fresh, smaller adapter on real-scene downstream QA. Token embeddings stay frozen across both stages.

# 4 Experiments

## 4.1 Implementation Details

**Panoramic projection.** We sample $K{=}32$ frames per scene at evenly spaced timestamps along the video. The scene center is set to the annotated agent position for SQA3D and to the queried camera pose for SPBench single-image queries, and to the centroid of all camera positions otherwise. Training is run at $320{\times}240$ input resolution across all data sources. At evaluation we use $320{\times}240$ for SQA3D, where raising it gave no aggregate gain, and $640{\times}480$ for VSI-Bench and SPBench, where explicit numeric-measurement queries benefit from higher input resolution. For ScanNet scenes we crop 3% from each image edge (with intrinsics adjusted) to avoid spurious lifted points outside the scene. We use ground-truth depth and ground-truth camera poses whenever the dataset provides them, and fall back to estimates from a feed-forward reconstruction model on the small subset of scenes where the released ground-truth pose track is corrupted. During training only, we randomize the panorama's location and orientation per sample by drawing the scene center from the available camera positions and rotating the forward axis by a uniform random yaw.

**Training.** We build on Qwen3-VL-8B Bai et al. [2025], a vision-language model with a ViT-based vision encoder and a 3D RoPE position encoding scheme with temporal, height, and width dimensions. Training proceeds in two stages on 8 A6000 GPUs, with LoRA adapters targeting the attention and MLP projections of the language model. Stage 1 trains on the spatial pretraining curriculum with LoRA rank $r{=}256$ ($\alpha{=}512$, dropout 0.05), effective batch 16, learning rate $2{\times}10^{-5}$ with cosine schedule and 3% warmup. The 3D position embedding trains jointly at $50\times$ the base learning rate. In stage 1 it is rescaled in every forward pass to half the mean norm of the visual features, so it learns direction but not magnitude. Stage 2 starts with its gate set to that same scale and then learns the scale freely. During stage 1 we cap the placed-object patch count at 200 per sample, distributed uniformly across the placed boxes with a floor of 8 patches per box to retain shape, which keeps stage 1 fast while leaving the geometry signal intact. Samples also carry 6–12 unreferenced distractor boxes, so a placement must be identified by its inline patch rather than as the only object on the canvas (Appendix D). The stage-2 LoRA uses $r{=}64$, $\alpha{=}128$, dropout 0.05, and trains for 10k steps with effective batch 32 at the same learning rate, schedule, and warmup. Stage-2 training data is a weighted mixture of VLM-3R-VSIBench Fan et al. [2026], SQA3D Ma et al. [2023], and ViCA Feng [2025], mixed at approximately 70/15/15 by sampling weight.

## 4.2 Evaluations

We evaluate on three benchmarks: SQA3D Ma et al. [2023] (situated question answering), VSI-Bench Yang et al. [2025a] (eight spatial subtasks split between numerical and multiple-choice), and SPBench Li et al. [2026] (single-image and multi-view spatial reasoning, evaluated zero-shot).

**RGB-predicted geometry.** Without retraining, replacing ground-truth geometry with depth, poses, and intrinsics predicted from the evaluation RGB frames by Depth Anything 3 Lin et al. [2026] gives 64.9 on SQA3D, 70.0 on VSI-Bench, and 71.3 on SPBench, compared with 65.3, 71.3,

Table 1: **Evaluation on SQA3D Ma et al. [2023] test split.** Comparison of methods across question types. EM@1 and EM@R1 are the primary metrics. Per-question-type columns report strict EM@1. Best results in **bold**, second best underlined. Dashes indicate scores not reported by the method.

| Method | Per Question Type | | | | | | EM@1 | EM@R1 |
|---|---|---|---|---|---|---|---|---|
| | What | Is | How | Can | Which | Others | | |
| PQ3D Zhu et al. [2024] | 37.1 | 61.3 | 44.5 | 60.9 | 47.0 | 45.1 | 47.1 | 49.3 |
| 3D-VisTA Zhu et al. [2023] | 34.8 | 63.3 | 45.4 | 69.8 | 47.2 | 48.1 | 48.5 | 50.9 |
| LEO Huang et al. [2024b] | 39.0 | 63.9 | 44.9 | 66.2 | 47.7 | 51.1 | 50.0 | 52.4 |
| SIG3D Man et al. [2024] | 35.6 | 67.2 | 48.5 | 71.4 | 49.1 | 45.8 | 52.6 | 54.4 |
| Scene-LLM Fu et al. [2025] | 40.9 | 69.1 | 45.0 | 70.8 | 47.2 | 52.3 | 54.2 | 56.2 |
| ChatScene Huang et al. [2024a] | 45.4 | 67.0 | 52.0 | 69.5 | 49.9 | 55.0 | 54.6 | 57.5 |
| Video-3D LLM Zheng et al. [2025b] | 51.1 | 72.4 | 55.5 | 69.8 | 51.3 | 56.0 | 58.6 | 60.8 |
| GPT4Scene-HDM Qi et al. [2026] | 55.9 | 69.9 | 50.8 | 68.7 | 53.3 | 60.4 | 59.4 | 62.4 |
| VLM-3R Fan et al. [2026] | – | – | – | – | – | – | 60.7 | 63.4 |
| SpaceMind Zhao et al. [2025] | 54.1 | 74.8 | **61.7** | 71.0 | 51.9 | 53.6 | 61.1 | 63.8 |
| SR-3D Cheng et al. [2025] | 55.0 | 76.4 | 59.8 | 71.6 | 54.7 | 61.1 | 62.2 | – |
| Loc3R-VLM Qu et al. [2026] | – | – | – | – | – | – | 62.8 | 65.0 |
| Ross3D Wang et al. [2025a] | 56.0 | **79.8** | 60.6 | 70.4 | 55.3 | 60.1 | 63.0 | 65.7 |
| **OneCanvas** (Ours) | **56.6** | 74.7 | 60.2 | **75.4** | **72.6** | **65.7** | **65.3** | **68.4** |

Table 2: **Evaluation on SPBench Li et al. [2026].** All models are evaluated without using SPBench training data. Best results per section in **bold**, second best underlined.

| Method | Overall | SPBench-MV | | | SPBench-SI | | |
|---|---|---|---|---|---|---|---|
| | | NQ | MCQ | Avg. | NQ | MCQ | Avg. |
| Video-R1 Feng et al. [2025] | 43.8 | 32.5 | 53.0 | 42.8 | 27.7 | 62.0 | 44.9 |
| SpaceR-7B Ouyang et al. [2025] | 53.5 | 63.2 | 53.7 | 58.5 | 35.7 | 61.5 | 48.6 |
| VILASR-7B Wu et al. [2025b] | 54.0 | 56.2 | 59.6 | 57.9 | 36.6 | **63.7** | 50.2 |
| Spatial-MLLM-4B Wu et al. [2025a] | 52.5 | 63.7 | 58.9 | 61.3 | 38.1 | 49.3 | 43.7 |
| SpaceMind Zhao et al. [2025] | 67.3 | 76.2 | 70.5 | 73.8 | **66.3** | 53.2 | 59.7 |
| **OneCanvas** (Ours) | **72.1** | **79.0** | **91.8** | **81.5** | 62.8 | 62.7 | **62.8** |

and 72.1 using ground-truth geometry. The complete inference-only protocol and limitations are in Appendix B.1.

**Spatial question answering.** Table 1 reports results on SQA3D, comparing our method against point-cloud-based 3D VLMs and video-input approaches. We achieve the best overall accuracy. We obtain the best results on the question types that most directly test situated reasoning (*Which*, *Can*, *Others*), with the largest margin on *Which*. The large *Which* margin tracks the canvas-origin ablation (Table 5): agent-pose centering drives most of it, and *Which* is the most viewpoint-dependent question type. The remaining gap concentrates on *Is* (yes/no questions, where textual priors carry most of the signal) and *How* (predominantly counting, a capability the base 2D VLM is already heavily pretrained on).

Table 2 presents zero-shot results on SPBench, where every method is evaluated without SPBench training data. We top the leaderboard, with the largest gain concentrated on multi-view MCQ, consistent with the intuition that a single panoramic canvas gives multi-view questions a unified frame of reference rather than forcing the VLM to stitch several independent views. We also top the single-image split on average accuracy.

Table 3 shows results on VSI-Bench, where we again come out on top, with the largest margin on route planning, the subtask that most directly exercises multi-step reasoning over a unified panoramic canvas. We also lead on absolute distance, relative distance, and room size, and finish second on relative direction.

Table 3: **Evaluation on VSI-Bench Yang et al. [2025a].** Comparison across numerical and multiple-choice spatial reasoning subtasks. Best results per section in **bold**, second best underlined.

| Method | Avg. | Numerical Question | | | | Multiple-Choice Question | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Obj.Cnt. | Abs.Dist. | Obj.Size | Room Size | Rel.Dist. | Rel.Dir. | Route | App.Order |
| Spacer Ouyang et al. [2025] | 45.5 | 57.8 | 28.2 | 59.9 | 47.1 | 40.1 | 45.4 | 33.5 | 52.1 |
| ViLaSR Wu et al. [2025b] | 45.4 | 63.5 | 34.4 | 60.6 | 30.9 | 48.9 | 45.2 | 30.4 | 49.2 |
| Spatial-MLLM Wu et al. [2025a] | 48.4 | 65.3 | 34.8 | 63.1 | 45.1 | 41.3 | 46.2 | 33.5 | 46.3 |
| ViCA Feng [2025] | 60.6 | 68.8 | 57.0 | **79.2** | 75.1 | 58.5 | 42.6 | 34.5 | 68.8 |
| VLM-3R Fan et al. [2026] | 60.9 | 70.2 | 49.4 | 69.2 | 67.1 | 65.4 | 80.5 | 45.4 | 40.1 |
| VST Yang et al. [2025b] | 61.2 | 71.6 | 43.8 | 75.5 | 69.2 | 60.0 | 55.6 | 44.3 | 69.2 |
| Cambrian-S Yang et al. [2026] | 67.5 | 73.2 | 50.5 | 74.9 | 72.2 | 71.1 | 76.2 | 41.8 | **80.1** |
| SenseNova-SI Cai et al. [2026] | 68.8 | 72.0 | 53.5 | 76.8 | 72.8 | 69.6 | 80.8 | 48.5 | 76.4 |
| SpaceMind Zhao et al. [2025] | 69.6 | **73.3** | 61.4 | 77.3 | 74.2 | 67.2 | **88.4** | 44.3 | 70.6 |
| **OneCanvas** (Ours) | **71.3** | 68.9 | **62.3** | 75.1 | **76.8** | **73.1** | 88.0 | **60.8** | 65.7 |

Table 4: **Ablation study** on VSI-Bench. Each row lists the components present in that variant. The full model uses panoramic reprojection, 3D position encoding, and stage-1 spatial pretraining, while the bottom row is the base VLM with multi-view input and none of these components. All variants are LoRA-finetuned with matched training settings. Best results per column in **bold**, second best underlined.

| Configuration | Avg. | Numerical Question | | | | Multiple-Choice Question | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Obj.Cnt. | Abs.Dist. | Obj.Size | Room Size | Rel.Dist. | Rel.Dir. | Route | App.Order |
| Full model (ours) | **71.3** | 68.9 | **62.3** | 75.1 | **76.8** | 73.1 | 88.0 | 60.8 | **65.7** |
| Full w/o 3D PE | 70.4 | 69.3 | 58.1 | 76.1 | 71.3 | 70.3 | 88.6 | **66.0** | 63.8 |
| Panorama + 3D PE | 68.5 | 68.3 | 56.6 | **76.9** | 74.1 | **73.2** | **91.6** | 45.4 | 61.3 |
| Panorama only | 65.3 | 68.8 | 54.8 | 76.8 | 56.9 | 71.6 | 87.4 | 45.3 | 60.4 |
| Base VLM (multi-view) | 65.5 | **72.5** | 53.2 | 75.8 | 68.6 | 68.9 | 77.4 | 43.9 | 63.4 |

### 4.3 Ablation Studies

**Component ablation.** In Table 4 we ablate the key components of our method on VSI-Bench. To match the total LoRA capacity of the two-stage full model (stage 1 $r$=256 merged into the base, stage 2 $r$=64), the single-stage ablation variants are trained with $r$=256 ($\alpha$=512). Without the spatial pretraining stage we observe a large drop on the route task, confirming the value of the curriculum for this capability. We also perform better on the tasks that require estimating metric quantities, with the exception of object size, which relies much more heavily on text priors. Removing the positional encoding produces a large additional drop on the room size task, while the other metric questions stay relatively stable as the model can still infer sizes and distances from the scene's appearance. Removing only the 3D position embedding from the full model isolates the same effect with the curriculum intact: the drop is concentrated on the metric subtasks, while angular and categorical subtasks are unaffected, consistent with the 3D position embedding acting as a metric-scale auxiliary channel on top of the panoramic canvas. We also see how much the unified panoramic representation contributes compared to passing the images straight: tasks that require relating objects across different parts of the scene perform notably better with our method, with the largest improvement on relative direction. The exception is object counting, where the base VLM retains a small lead. We attribute this to the rearrangement itself, which moves the input away from the image layout the backbone was pretrained on. Counting depends less on spatial understanding than on how many tokens show the queried object and how concentrated they are Zhang et al. [2018], so the better spatial representation does not recover this cost (Appendix B.2).

**Canvas origin.** We evaluate the same checkpoint on SQA3D under four canvas origin strategies (Table 5). Using the agent's pose increases the results noticeably on most tasks, except on the *Is* and *How* categories, which are generally viewpoint agnostic. The random-camera origin is sampled from the raw camera positions used to construct the canvas and therefore remains inside the observed space. Only the outside-scene condition places the origin outside the room. Other camera placement strategies generally do not greatly influence the results, except for the outside-scene origin where the results drop consistently across the tasks.

Table 5: **Canvas origin ablation on SQA3D.** Same checkpoint evaluated under four canvas origin strategies. *Agent pose* tracks the agent's position and heading. The other three fix forward to world $+X$ with origins at the scene centroid, a raw camera position inside the observed space, or outside the scene. Per-question-type columns report strict EM@1. Best per column in **bold**, second best underlined.

| Canvas origin | Per Question Type | | | | | | EM@1 | EM@R1 |
|---|---|---|---|---|---|---|---|---|
| | What | Is | How | Can | Which | Others | | |
| Agent pose | **56.6** | **74.7** | 60.2 | **75.4** | **72.6** | **65.7** | **65.3** | **68.4** |
| Scene center | 54.6 | 74.2 | 59.8 | 68.9 | 56.7 | 58.9 | 61.2 | 64.3 |
| Random camera | 54.8 | 74.4 | **61.3** | 68.9 | 55.6 | 58.9 | 61.4 | 64.6 |
| Outside scene | 54.0 | 73.6 | 60.0 | 68.3 | 51.3 | 55.7 | 59.8 | 62.7 |

## 5 Conclusion

OneCanvas reframes 3D scene understanding as 2D spatial reasoning on a panoramic canvas. Multi-view RGB-D is reprojected onto a single equirectangular panorama with each patch carrying its metric 3D position through an additive 3D position embedding, and the pretrained VLM reads the result through its native attention. A spatial pretraining curriculum of objects placed on an otherwise empty canvas and referenced from the prompt warms up geometric reading before downstream fine-tuning. The combination sets a new state of the art on VSI-Bench and SQA3D, and tops the zero-shot SPBench leaderboard. The representation also carries over to other backbones (Appendix B.4).

**Limitations.** OneCanvas requires depth and camera poses, which pure RGB methods avoid, and pose-estimation failures can still degrade the canvas, though the inference-only Depth Anything 3 evaluation in Appendix B.1 narrows this gap. The single global panorama trades fine spatial precision for compactness and may limit very large or outdoor scenes. The pretraining curriculum is hand-authored, so adding a qualitatively new spatial skill requires writing a new task generator rather than collecting more data.


## Acknowledgments and Disclosure of Funding

This project was funded by the ERC Consolidator Grant Gen3D (101171131). We also thank Angela Dai for the video voice-over.


## References


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.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-D data. In *Advances in Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2021.

Hunar Batra, Haoqin Tu, Hardy Chen, Yuanze Lin, Cihang Xie, and Ronald Clark. Spatial-thinker: Reinforcing 3d reasoning in multimodal llms via spatial rewards. *arXiv preprint arXiv:2511.07403*, 2025.

Zhongang Cai, Ruisi Wang, Chenyang Gu, Fanyi Pu, Junxiang Xu, Yubo Wang, Wanqi Yin, Zhitao Yang, Chen Wei, Qingping Sun, et al. Scaling spatial intelligence with multimodal foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jieneng Chen, Wenxin Ma, Ruisheng Yuan, Yunzhi Zhang, Jiajun Wu, and Alan Yuille. Thinking with spatial code for physical-world video reasoning. *arXiv preprint arXiv:2603.05591*, 2026.

Pingyi Chen, Yujing Lou, Shen Cao, Jinhui Guo, Lubin Fan, Yue Wu, Lin Yang, Lizhuang Ma, and Jieping Ye. Sd-vlm: Spatial measuring and understanding with depth-encoded vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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*, pages 26428–26438, 2024.

An-Chieh Cheng, Yang Fu, Yukang Chen, Zhijian Liu, Xiaolong Li, Subhashree Radhakrishnan, Song Han, Yao Lu, Jan Kautz, Pavlo Molchanov, et al. 3d aware region prompted vision language model. *arXiv preprint arXiv:2509.13317*, 2025.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 31054–31065, 2026.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

Qi Feng. Visuospatial cognitive assistant. *arXiv preprint arXiv:2505.12312*, 2025.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 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. *Advances in Neural Information Processing Systems*, 36:20482–20494, 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 (ICLR)*, 2022.

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, 2024a.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024b.

Yuan Jia, Ruimin Wu, Rui Song, Jiaojiao Li, and Bin Song. Liftproj: Space lifting and projection-based panorama stitching. *arXiv preprint arXiv:2512.24276*, 2025.

Seongmin Jung, Seongho Choi, Gunwoo Jeon, Minsu Cho, and Jongwoo Lim. Panogrounder: Bridging 2d and 3d with panoramic scene representations for vlm-based 3d visual grounding. *arXiv preprint arXiv:2512.20907*, 2025.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

Hongxing Li, Dingming Li, Zixuan Wang, Yuchen Yan, Hang Wu, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. Spatialladder: Progressive training for spatial reasoning in vision-language models. In *International Conference on Learning Representations (ICLR)*, 2026.

Xiang Li, Jian Ding, Zhaoyang Chen, and Mohamed Elhoseiny. Uni3dl: A unified model for 3d vision-language understanding. In *European Conference on Computer Vision*, pages 74–92. Springer, 2024.

Huizhi Liang, Yichao Shen, Yu Deng, Sicheng Xu, Zhiyuan Feng, Tong Zhang, Yaobo Liang, and Jiaolong Yang. Hispatial: Taming hierarchical 3d spatial understanding in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations (ICLR)*, 2026.

Zekai Lin and Xu Zheng. Panoenv: Exploring 3d spatial intelligence in panoramic environments with reinforcement learning. *arXiv preprint arXiv:2602.21992*, 2026.

Xianzheng Ma, Brandon Smart, Yash Bhalgat, Shuai Chen, Xinghui Li, Jian Ding, Jindong Gu, Dave Zhenyu Chen, Songyou Peng, Jia-Wang Bian, et al. When llms step into the 3d world: A survey and meta-analysis of 3d tasks via multi-modal large language models. *arXiv preprint arXiv:2405.10255*, 2024a.

Xianzheng Ma, Tao Sun, Shuai Chen, Yash Bhalgat, Jindong Gu, Angel X Chang, Iro Armeni, Iro Laina, Songyou Peng, and Victor Adrian Prisacariu. Do 3d large language models really understand 3d spatial relationships? In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2603.23523.

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 (ICLR)*, 2023.

Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Yu Qiao, Ruigang Yang, and Xinge Zhu. Vision-centric bev perception: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10978–10997, 2024b.

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*, pages 13678–13688, 2024.

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.

Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision*, pages 194–210. Springer, 2020.

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 (ICLR)*, 2026.

Kevin Qu, Haozhe Qi, Mihai Dusmanu, Mahdi Rad, Rui Wang, and Marc Pollefeys. Loc3r-vlm: Language-based localization and 3d reasoning with vision-language models, 2026. URL https://arxiv.org/abs/2603.18002.

Mukund Varma T, Peihao Wang, Zhiwen Fan, Zhangyang Wang, Hao Su, and Ravi Ramamoorthi. Lift3d: Zero-shot lifting of any 2d vision model to 3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21367–21377, 2024.

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*, pages 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 Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024a.

Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, Xihui Liu, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023.

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

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025a.

Rui Yang, Ziyu Zhu, Yanwei Li, Jingjia Huang, Shen Yan, Siyuan Zhou, Zhe Liu, Xiangtai Li, Shuangye Li, Wenqian Wang, et al. Visual spatial tuning. *arXiv preprint arXiv:2511.05491*, 2025b.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis L Brown II, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, et al. Cambrian-s: Towards spatial supersensing in video. In *The Fourteenth 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 (ICCV)*, 2023.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Learning to count objects in natural images for visual question answering. In *International Conference on Learning Representations (ICLR)*, 2018.

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. *arXiv preprint arXiv:2511.23075*, 2025.

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 (NeurIPS)*, 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*, pages 8995–9006, 2025b.

Xu Zheng, Chenfei Liao, Ziqiao Weng, Kaiyu Lei, Zihao Dongfang, Haocong He, Yuanhuiyi Lyu, Lutao Jiang, Lu Qi, Li Chen, et al. Panorama: The rise of omnidirectional vision in the embodied ai era. *arXiv preprint arXiv:2509.12989*, 2025c.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 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*, pages 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*, pages 188–206. Springer, 2024.

# A Object Referencing: Alternative Class-Labelled Design

The main paper places *synthetic axis-aligned boxes* on the canvas whose per-patch features are drawn from a shared random pool of activations harvested from training-source scenes, and references each placement in the prompt via an inline copy of the same patch feature. The motivation is to disconnect object size and appearance from class identity, so geometry on the canvas becomes the principal signal for solving the curriculum (see the curriculum-objects discussion in the main method). A natural alternative is to populate the canvas with real objects harvested in-context from scene instance annotations Wang et al. [2024b], pasted onto the canvas with their original per-patch features, and referenced in the prompt by class name. Features are extracted within the original scene rather than re-encoded from cropped objects in isolation, since the latter would discard the surrounding-scene context each patch was encoded with. This variant adds text-to-object grounding as a side benefit, at the cost of re-introducing the class priors (typical door height, typical bathroom size) that the synthetic-box scheme is designed to suppress. Table 6 compares the two curriculum content choices on the full VSI-Bench test split, holding the backbone, canvas construction, MRoPE schedule, task families, family-flat weighting, stage-2 fine-tuning, and evaluation protocol fixed. The only thing that changes between rows is what populates the canvas in stage 1.

Table 6: **Curriculum content study on VSI-Bench (full test split).** Both rows share the same backbone, canvas construction, task families, stage-2 fine-tuning, and evaluation protocol. Only the stage-1 curriculum content differs. *Synthetic boxes*: axis-aligned boxes with shared-random per-patch features, referenced in the prompt by inline patch. *Scene-harvested objects*: real objects pasted from in-context scene crops with their original per-patch features, referenced by class name.

| Curriculum content | Avg. | Numerical Question | | | | Multiple-Choice Question | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Obj.Cnt. | Abs.Dist. | Obj.Size | Room Size | Rel.Dist. | Rel.Dir. | Route | App.Order |
| Synthetic boxes (main paper) | **71.3** | 68.9 | **62.3** | 75.1 | 76.8 | **73.1** | **88.0** | **60.8** | **65.7** |
| Scene-harvested objects | 70.2 | **69.8** | 61.1 | **76.5** | **79.6** | 67.8 | 86.1 | 59.8 | 60.8 |

# B Additional Experiments and Analyses

This section collects additional evaluations. The protocols are not interchangeable. The predicted-geometry, rasterization, and seam studies change only inference. The feature-versus-pixel and ERNIE comparisons are matched 2,000-step training runs. The seed replications use the complete 10,000-step schedule. The Qwen3.5 result uses the full two-stage recipe. We state these boundaries with each result.

## B.1 Geometry Predicted from RGB

We replace all ground-truth depth, poses, and intrinsics with estimates produced directly from RGB by Depth Anything 3 (DA3NESTED-GIANT-LARGE-1.1) Lin et al. [2026]. Reconstruction uses exactly the frames supplied to the VLM at evaluation: $K{=}32$ for SQA3D and VSI-Bench, and the number of views supplied with each SPBench question. It uses no extra views and no ground-truth geometry. This is an inference-only substitution using the paper checkpoint, with no retraining around the geometry estimator.

Predicted geometry reduces accuracy by 0.4, 1.3, and 0.8 points, respectively, while exceeding the prior-method results listed in the table on all three benchmarks. This establishes robustness to this specific feed-forward estimator, not invariance to reconstruction failure or the performance of a jointly trained geometry encoder. Because the estimator is called as a replaceable preprocessing module rather than trained jointly with the VLM, the result also does not require adapting the paper checkpoint to DA3.

Table 7: **Feed-forward predicted geometry.** Accuracy under the paper's full evaluation protocol. The previous-best row names the strongest prior result separately on each benchmark: Ross3D on SQA3D and SpaceMind on VSI-Bench and SPBench.

| Geometry | SQA3D | VSI-Bench | SPBench |
|---|---|---|---|
| Previous best | 63.0 | 69.6 | 67.3 |
| Ours, ground-truth geometry | **65.3** | **71.3** | **72.1** |
| Ours, RGB-predicted geometry | 64.9 | 70.0 | 71.3 |

### B.2 Continuous Features, Rasterization, and Rendered Pixels

**Inference-only rasterization.** We bin lifted patches into a fixed equirectangular grid, mean-pool the features, depth, and frame index of all patches in a cell, and assign the pooled token the cell-center angle. The $181\times362$, $90\times180$, and $45\times90$ grids have approximately square angular cells of $1^\circ$, $2^\circ$, and $4^\circ$. All rows use the paper evaluation protocol and the same checkpoint trained on the continuous canvas. The rasterized scores are therefore inference-only ablations. A model retrained for a grid could perform differently and was not evaluated.

Table 8: **Inference-only rasterization on VSI-Bench.** Token counts are the occupied-cell counts presented to the model.

| Canvas | Tokens | VSI-Bench |
|---|---|---|
| Continuous (ours) | 9,600 | **71.1** |
| Rasterized $181\times362$ | 7,144 | 68.2 |
| Rasterized $90\times180$ | 4,009 | 67.7 |
| Rasterized $45\times90$ | 1,563 | 65.8 |

Most of the loss appears at the finest grid. Appearance Order falls from 62.6 to 39.8 at $181\times362$, accounting for 2.8 points of the 2.9-point average gap. Replacing mean pooling with one original patch per occupied cell keeps the grid and token count fixed and scores 68.6 rather than 68.2. Thus averaging itself does not explain the gap. Both reductions retain only one token and one frame index per cell: repeated observations of a surface either have their frame indices averaged or all but one discarded, so first appearance is no longer recoverable. A raster grid may nevertheless extend the canvas to VLMs whose 1D positional encoding cannot assign arbitrary angular coordinates.

**Matched feature-versus-pixel training.** We also train two arms for 2,000 steps using the same base checkpoint, LoRA rank 256, effective batch 8, schedule, data mixture, and held-out VSI-Bench questions. They differ only in scene input: projected per-frame features versus a pre-rendered equirectangular z-buffer panorama.

Table 9: **Feature canvas versus rendered pixels.** Matched 2,000-step training runs on VSI-Bench. These are short-run comparisons, not the paper's full training schedule.

| Scene input | VSI-Bench |
|---|---|
| Feature canvas (ours) | **60.8** |
| Pixel panorama | 52.1 |

The 8.7-point difference is larger than the roughly one-point step-to-step variation observed in either run. It jointly reflects two differences and should not be read as isolating either one: z-buffer rendering collapses occluded and co-located surfaces to one surface per direction, and the rendered arm asks a vision tower pretrained on perspective images to re-encode an equirectangular image. Our canvas applies no visibility test, keeps these observations as separate tokens, and extracts their appearance features from the original perspective frames. It also permits changing the canvas origin without rendering again.

**Counting and depth coarsening.** The base VLM retains a small lead on object counting in Table 4. A depth-coarsening control does not support the explanation that reprojection inaccuracies fragment

or merge instances. Coarsening by up to $8\times$ produces a median patch displacement of 0.16 m and moves 29% of patches to a different surface, while counting scores remain similar across the four levels (68.4, 68.8, 69.3, and 68.9). We instead attribute the gap to the rearrangement itself, which moves the input away from the backbone's pretrained image layout, while counting depends mainly on how many tokens show the queried object and how concentrated they are Zhang et al. [2018].

### B.3 Training-Seed Replications

We trained three additional runs under different random seeds with every other setting fixed, retained all runs, and used the complete 10,000-step schedule rather than a rebuttal-window short run. As throughout the paper, checkpoints are selected by held-out generation accuracy.

Table 10: **Seed variance under full 10,000-step training.** Standard deviations are across the three reported seeds.

| | VSI-Bench | SQA3D (EM@1) | SPBench |
|---|---|---|---|
| Seed A | 71.0 | 66.2 | 73.4 |
| Seed B | 69.9 | 64.5 | 72.8 |
| Seed C | 70.3 | 64.7 | 73.4 |
| Mean $\pm$ std. | $70.4 \pm 0.6$ | $65.1 \pm 0.9$ | $73.2 \pm 0.3$ |

All three runs exceed the prior best of 69.6 on VSI-Bench, 63.0 on SQA3D, and 67.3 on SPBench. These three seeds quantify training-seed variation under one fixed recipe. They do not capture variation from data, hyperparameter, or model-family choices.

### B.4 Additional Backbones

**ERNIE-4.5-VL-28B-A3B.** We ported the representation outside the Qwen family. ERNIE runs end to end on real weights through canvas prefill, training, and generation, with the adapter confined to the language model, the vision tower unchanged, and no architectural change. The two arms below differ only in whether they use the canvas and are trained for 2,000 iterations with identical data, schedule, adapter shape, and effective batch.

Table 11: **Matched 2,000-step ERNIE-4.5-VL-28B-A3B comparison on VSI-Bench.** This is a stage-2-only short run without the spatial curriculum, so the controlled row difference, rather than the absolute accuracy or a comparison to the fully trained paper model, is the relevant quantity.

| | Obj. Cnt. | Abs. Dist. | Obj. Size | Room Size | Rel. Dist. | Rel. Dir. | Route | App. Order | Overall |
|---|---|---|---|---|---|---|---|---|---|
| Backbone alone | 50.6 | 26.9 | 70.5 | 42.2 | **52.3** | 44.0 | 21.3 | 13.9 | 40.2 |
| + canvas | **65.2** | **27.9** | **71.2** | **55.8** | 51.3 | **46.8** | **28.0** | **24.6** | **46.3** |
| $\Delta$ | +14.6 | +1.0 | +0.7 | +13.6 | -1.0 | +2.8 | +6.7 | +10.7 | +6.1 |

ERNIE is a general-purpose model rather than a spatial specialist, and these runs omit stage 1. The +6.1 overall difference therefore supports portability of the representation under this matched short protocol, but does not establish ERNIE's full-recipe performance.

**Qwen3.5-9B.** Using the full two-stage recipe and the paper's full evaluation protocol, Qwen3.5-9B reaches 69.42 on VSI-Bench, 67.18 EM@1 on SQA3D, and 69.81 on SPBench. This is a full-scale run with a newer Qwen generation and different vision stack, not the requested non-Qwen control. It is reported separately from the ERNIE comparison and does not replace the paper checkpoint or its 71.3, 65.3, and 72.1 main results.

### B.5 Distortion and the Longitude Seam

Patch features come from the original perspective frames, so the equirectangular mapping changes only where a patch is placed, not its features. Longitude is periodic, but its assigned position ID is not: patches at $+179^\circ$ and $-179^\circ$ are nearby in the scene but receive opposite ends of the longitude index range. The additive 3D position embedding remains continuous because it encodes each

patch's Cartesian offset from the canvas origin, and a uniformly random training yaw moves the seam across longitudes rather than systematically splitting one scene region. A localized angular-index discontinuity nevertheless remains.

We isolate it with an inference-only pairwise-distance probe. The two objects and answer remain fixed while the canvas orientation changes so that the seam lies either between the objects or away from them. Depth co-varies with angular separation $\Delta$ by construction, so each row is compared only internally.

Table 12: **Longitude-seam probe.** Mean absolute distance error in metres.

| $\Delta$ | Seam not between | Seam between |
|---|---|---|
| $15^\circ$ | 0.14 | 1.23 |
| $30^\circ$ | 0.15 | 0.50 |
| $60^\circ$ | 0.24 | 0.30 |
| $170^\circ$ | 0.69 | 0.71 |

The increase concentrates where the seam separates objects with small angular separation and becomes negligible as separation grows. Placing both objects immediately beside the seam without separating them gives 0.23 m error at $15^\circ$, isolating the effect to the wrap rather than proximity to the boundary. Under random canvas orientation this configuration affects 0.63% of VSI-Bench absolute-distance questions (one of 214 in our evaluation). At inference, the free canvas yaw can rotate the seam away from a region of interest. Settings where seam-straddling queries are common may instead require targeted curriculum examples or a wrap-aware positional encoding.

## C Training Compute Budget

The reproducible two-stage OneCanvas recipe uses 8 NVIDIA RTX A6000 GPUs (Ampere architecture, 48 GB GDDR6 each) with DeepSpeed ZeRO-2 and Flash Attention 2. Stage 1 (spatial pretraining) takes ≈25.2 h of wall-clock time, and Stage 2 (QA fine-tuning) takes ≈39.6 h, for a total of ≈64.8 h on 8 GPUs (518 raw A6000-GPU-hours). Table 13 compares this with the training costs reported by competing methods through an analytic A100-equivalent normalization. These are not measured cross-hardware runtime comparisons.

Table 13: **Training compute comparison.** Wall-clock training time and total compute for OneCanvas and competing methods, analytically normalized to A100-equivalent GPU-hours using the per-GPU-type peak-throughput factors in Table 14. OneCanvas numbers combine both training stages.

| Method | GPU type | # GPUs | Wall-clock | Raw GPU-h | A100-equiv. GPU-h |
|---|---|---|---|---|---|
| **OneCanvas (Ours)** | A6000 | 8 | ≈64.8 h | 518 | 259 |
| VLM-3R Fan et al. [2026] | H200 | 16 | 5 h | 80 | 240 |
| ViCA Feng [2025] | H100 | 8 | 55 h | 440 | 1,320 |
| SpaceMind Zhao et al. [2025] | H100 | 64 | 25 h | 1,600 | 4,800 |
| SenseNova-SI Cai et al. [2026] | H100$^\dagger$ | 128 | 72 h | 9,216 | 27,648 |

$^\dagger$ GPU type not stated in the paper, assumed H100.

**GPU normalization.** We convert raw GPU-hours to A100-equivalent GPU-hours by multiplying by the dense BF16 Tensor Core throughput ratio between the source GPU and the A100 SXM4. We use *dense* BF16 (no structured 2:4 sparsity) because that matches what training actually runs: BF16 numerics, no sparsity kernels. The factors are listed in Table 14 and are simply $\text{TFLOPS}_{\text{src}}/\text{TFLOPS}_{\text{A100}}$ rounded to one significant digit. This analytic conversion captures peak compute only and should be read as a normalized upper bound, not a measured speed ratio. In practice the achievable speedup on H100/H200 is bounded below the $3\times$ ratio by memory bandwidth (HBM ratio $1.67\times$ for H100, $2.4\times$ for H200) and by multi-node scaling efficiency at the 64- and 128-GPU jobs in Table 13. Under the more conservative bandwidth-bound conversion, SpaceMind and SenseNova-SI still use approximately $10\times$ and $59\times$ our normalized compute, respectively.

The H200 shares the GH100 compute die with the H100 and differs only in memory capacity (141 GB HBM3e vs. 80 GB HBM3) and bandwidth, so it carries the same dense BF16 through-

Table 14: **GPU normalization factors used in Table 13.** Dense BF16 Tensor Core TFLOPS (no 2:4 sparsity) are NVIDIA datasheet values, and HBM bandwidth is shown for reference but is not used in the conversion. The A100-equivalent factor is computed as $\text{TFLOPS}_{\text{src}}/\text{TFLOPS}_{\text{A100}} = \text{TFLOPS}_{\text{src}}/312$, rounded to one significant digit.

| GPU | Dense BF16 | HBM bandwidth | A100-equiv. factor |
|---|---|---|---|
| A6000 | 154.8 TFLOPS | 768 GB/s | $154.8/312 \approx 0.5\times$ |
| A100 SXM4 | 312 TFLOPS | 2,000 GB/s | $1.0\times$ |
| H100 SXM5 | 989 TFLOPS | 3,350 GB/s | $989/312 \approx 3.0\times$ |
| H200 SXM5 | 989 TFLOPS | 4,800 GB/s | $989/312 \approx 3.0\times$ |

put and the same conversion factor. The 989/312 ratio rounds down from 3.17 to 3.0, which mildly understates competitor compute and therefore biases the comparison *against* our method.

**Per-method calculations.** Table 15 applies the factors from Table 14 to the wall-clock training time reported by each method.

Table 15: **Per-method A100-equivalent compute.** A100-equivalent GPU-hours equal (# GPUs) $\times$ (wall-clock hours) $\times$ (A100-equiv. factor). # GPUs and wall-clock are taken from each method's paper, factors come from Table 14.

| Method | # GPUs | Wall-clock (h) | Factor | A100-equiv. GPU-h | vs. Ours |
|---|---|---|---|---|---|
| **OneCanvas (Ours)** | 8 | 64.8 | $0.5\times$ | 259 | $1\times$ |
| VLM-3R Fan et al. [2026] | 16 | 5 | $3\times$ | 240 | $0.9\times$ |
| ViCA Feng [2025] | 8 | 55 | $3\times$ | 1,320 | $5.1\times$ |
| SpaceMind Zhao et al. [2025] | 64 | 25 | $3\times$ | 4,800 | $19\times$ |
| SenseNova-SI Cai et al. [2026] | 128 | 72 | $3\times$ | 27,648 | $107\times$ |

The compute claim is scoped to methods near our accuracy, not to every baseline. SpaceMind (69.6 VSI-Bench) and SenseNova-SI (68.8) are the two closest methods to our 71.3 and use $19\times$ and $107\times$ the peak-normalized compute above. VLM-3R uses slightly less normalized compute than ours (240 versus 259 A100-equivalent GPU-hours), and ViCA uses $5.1\times$ ours, but their VSI-Bench scores are 60.9 and 60.6. At comparable compute, VLM-3R is 10.4 points lower. Thus the evidence supports lower training cost at comparable accuracy, not a universal compute advantage.

## D Spatial Pretraining Curriculum: Task Details

This section details the tasks that make up the stage-1 spatial pretraining curriculum. The curriculum organizes its task instances into six families and applies family-flat weighting: each family contributes an equal share of every minibatch, and every task within a family carries the same weight. The six families are Metric Measurement, Egocentric Direction, Multi-Turn Navigation, Observability, Counting, and Multi-Target Bounding-Box Readout.

All tasks are trained on a canvas containing only the placed objects, at procedurally sampled centres, dimensions, and yaws. No scene is ever loaded onto the canvas during stage 1, so the model only sees the placed objects.

Each placed box is realised as the set of canvas patches that fall within its volume, all carrying a single feature vector sampled from a precomputed pool of real ViT patch features. The pool is built once by running Qwen3-VL's frozen vision encoder on training-source scenes from ScanNet Dai et al. [2017], ARKitScenes Baruch et al. [2021], and ScanNet++ Yeshwanth et al. [2023], and box features are drawn independently from it at training time. Because the pool is scene-agnostic, a box cannot leak the current scene's identity, and multi-target tasks cannot shortcut by clustering on tokens from a shared source frame. Because the patches are real ViT outputs, their per-token feature distribution matches what the VLM sees on real scenes, so the stage-1 reading capability transfers to stage 2 where the canvas is populated with real-scene patches.

**Distractor boxes.** Samples add 6–12 boxes that the prompt never references. Each draws its own feature, disjoint from the task referents and from the other distractors, so neither the odd one out nor a coherent class is a usable shortcut. Floor-area tasks are sampled without distractors, since extra boxes would change the quantity being asked about.

We describe each family below, giving each task a short descriptive name followed by its precise input and output format.

### D.1 Metric Measurement

**Pairwise distance.** Two boxes are placed on the canvas with non-overlapping supports and a surface-to-surface separation of at least 5 cm. The prompt references each via an inline patch, and the model is asked to report the shortest distance between the two surfaces in metres, rounded to one decimal. The label is the scalar regression target.

**Closest-of-four.** Five boxes are placed on the canvas (one *target*, four *candidates*), with the winner-vs.-runner-up surface-distance gap constrained to at least 5 cm. The prompt references the target and the four candidates via inline patches, and the model picks which of the four candidates is closest to the target as a 4-way MCQ with answers {A, B, C, D}.

**Non-rectangular floor area.** A synthetic L- or T-shaped floor polygon is placed in the scene, rotated by a random yaw, and rendered as a thin slab of lifted points. The model reads off the floor area in square metres (one-decimal regression).

### D.2 Egocentric Direction

The world-frame direction tasks (the two-way, three-way, four-way diagonal, four-way cardinal, and clock-face variants below) place 3 boxes: a "ref" box at the agent position, a "fwd" box defining the facing direction, and a target box to classify. The prompt reads "If you are standing at [ref] and facing [fwd], . . .", with the bracketed names rendered as inline patches bound to the corresponding canvas patches. The camera-frame direction tasks place 2 boxes (target and pivot), with the canvas itself reoriented onto a real scene camera pose so no explicit facing reference is needed. The answer space varies by task.

**Two-way direction.** 2-way MCQ: {left, right}. The question is "is the target to the left or the right?".

**Three-way direction.** 3-way MCQ: {left, right, back}. The "back" bin activates when the target lies more than $135^\circ$ from the heading.

**Four-way diagonal direction.** 4-way diagonal MCQ: {front-left, front-right, back-left, back-right}. Sampling enforces an angular margin from the cardinal axes so the diagonal label is well-defined.

**Four-way cardinal direction.** 4-way cardinal MCQ {front, back, left, right} with $\pm 45^\circ$ bin boundaries.

**Camera-frame direction (easy / medium / hard).** Same three answer spaces as the world-frame two-way, three-way, and four-way diagonal variants, but the canvas is recentered and reoriented onto a real scene camera pose drawn from the frame sequence. The question is phrased from the camera's perspective, "From the camera's perspective, is X to Y's . . . ?".

**Clock-face bearing.** The answer is a clock-face bearing in $\{1, 2, \ldots, 12\}$, computed as the clockwise angle from the heading (12 = forward, 3 = right, 6 = back, 9 = left). Finer angular resolution than the cardinal or diagonal variants.

### D.3 Multi-Turn Navigation

The Navigation family carries the N-turn route-planning task at four turn counts ($N \in \{1, 2, 3, 4\}$), each carrying equal weight in the family.

**N-turn route planning ($N \in \{1, 2, 3, 4\}$).** A start box, a goal box, and $N$ intermediate waypoint boxes are placed at sampled positions on the canvas. The prompt references each via inline patches. The answer is the sequence of egocentric turn actions that takes the agent from start to goal, with one action per turn. The format is a fill-in MCQ where the question lists the actions as "[please fill in]" slots and the model selects one of 4 letter options. Each path-turn is drawn from {Turn Left, Turn Right}. Additionally, on the first turn we add a "Turn Back" option, so the initial move is drawn from {Turn Left, Turn Right, Turn Back}.

### D.4 Observability

**Appearance order.** Four boxes are placed at sampled positions on the canvas. Each box's per-patch MRoPE $T$ values are overridden to a controlled distribution: the four boxes are assigned consecutive first-appearance $T$-starts $\{k, k{+}1, k{+}2, k{+}3\}$ for a random $k$, and each box's remaining patch $T$ values are sampled uniformly in $[t_{\text{start}}, T_{\max}]$ with one patch pinned to $t_{\text{start}}$. The prompt references the four via inline patches and asks for the full first-time appearance order as a 4-way MCQ (correct permutation plus three random-permutation distractors). The construction isolates fine-grained $T$-axis resolution at the one-frame scale, with no scene content available to shortcut the answer.

**Line-of-sight visibility.** A yes/no task: given a viewer box $p_1$, a target box $p_2$, and a synthetic axis-aligned occluder box between them, does the line of sight from $p_1$ reach $p_2$? Visibility is computed from first principles as the fraction of $p_2$'s AABB reachable from $p_1$ by a segment not intersecting the occluder, and the label is "yes" if that fraction is at least $10\%$. The occluder is always present so box count cannot leak the label. Blocker side, along-segment position, and segment length are drawn from a shared distribution across both labels, with only the perpendicular offset of the occluder centre label-conditional. The task is *not* temporal. The features are pose-independent, and the task is listed under Observability because it tests the same "can the model resolve whether a region is reachable" capability as appearance order.

### D.5 Counting and Arithmetic Readout

**Object counting.** $N$ boxes are placed on the canvas with $N$ drawn uniformly over a wide integer range. All $N$ boxes share their feature draw with a single reference box that the prompt references via an inline patch: "How many [ref] are in the room?". The model emits the integer count. The shared feature gives the $N$ boxes a common identity that the model can read through content attention, while non-referenced distractor boxes carry their own different feature draws so they cannot be mistaken for the referent.

**Count parity.** Same setup as object counting but the question is "Is the number of [ref] in the room odd or even?". The answer is "even" or "odd". The placed count is sampled to make the two labels equiprobable, so the model cannot solve the task by guessing the marginal.

**Count divisibility-by-three.** Same setup but the question is "Is the number of [ref] in the room a multiple of three?". The answer is "yes" or "no". The placed count is sampled to make the two labels equiprobable. Phrased as a binary divisibility check rather than a three-way residue classification, which keeps the answer format consistent with the other yes/no tasks.

**Count-by-side.** A hybrid task. Given an origin pose with a forward heading and a reference box, the model counts how many same-feature boxes lie on a specified cardinal side from the agent's perspective. The side is drawn uniformly from {right, left, front, back} and the target count is drawn from $\{3, 4, 5, 6\}$ with weights $[30, 35, 25, 10]$. The answer is the integer. The task combines direction classification with counting.

### D.6 Multi-Target Bounding-Box Readout

**Multi-target bounding-box readout.** Two to four boxes are placed on the canvas at sampled centres, dimensions, and yaws. The prompt references each via an inline patch tagged with a letter: "Provide the 3D bounding box of each highlighted region, in the same order: A: [m1], B: [m2], . . .". The model emits the axis-aligned 3D bounding box (centre and metric extents) for each box in canonical near-to-far order.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state two contributions (panoramic reprojection as VLM input, and the spatial pretraining curriculum), both supported by the method (Section 3) and the experimental results on SQA3D, VSI-Bench, and SPBench (Section 4.2). The compute-efficiency claim made in the abstract is quantified in Appendix C.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations paragraph at the end of Section 5 discusses the requirement for depth and camera poses, sensitivity to pose-estimation errors, the tradeoff of using a single global panorama for very large or outdoor scenes, the hand-authored nature of the spatial pretraining curriculum, and the scope of the additional-backbone evaluations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present any theorems, lemmas, or formal theoretical results. The contribution is empirical, centered on a new input representation, training curriculum, and benchmark evaluation.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3 fully specifies the lifting, panoramic projection, and 3D position encoding. Section 4.1 reports the base model, training data, and the full set of training hyperparameters. Appendix D documents the spatial pretraining curriculum, and Appendix C reports the hardware and wall-clock budget. All training and evaluation datasets used (SQA3D, VSI-Bench, VLM-3R, ViCA, SPBench) are publicly accessible to other researchers.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The training and evaluation code, the on-the-fly curriculum generator, the trained model with both stage adapters, and instructions to reproduce the headline results are released at `https://github.com/baranowskibrt/onecanvas` and `https://huggingface.co/BaranowskiBrt/OneCanvas-Qwen3-VL-8B`. All training and evaluation datasets (SQA3D, VSI-Bench, VLM-3R, ViCA, SPBench) are publicly available from their original sources, and our code uses them through their official splits.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 reports the base model, panorama resolution, input frame count, training schedule, LoRA ranks, batch size, learning rate, and training data mixture. Evaluations use the public test splits of SQA3D, VSI-Bench, and SPBench. Curriculum task definitions and generation details are in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix B.3 reports three full 10,000-step replications under different training seeds, includes every run, and reports the mean and sample standard deviation for SQA3D, VSI-Bench, and SPBench. These error bars capture training-seed variation under the fixed recipe. They do not capture data, hyperparameter, or backbone variation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C reports the GPU type (8× NVIDIA RTX A6000, 48 GB), training framework (DeepSpeed ZeRO-2, Flash Attention 2), per-stage wall-clock hours, total raw A6000-GPU-hours, and an A100-equivalent normalization, with the methodology for the cross-architecture comparison explained in the same section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: The work uses publicly released indoor scene datasets under their stated licenses, does not involve human subjects, and we have reviewed and conformed to the NeurIPS Code of Ethics throughout the research.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper presents foundational research on a 3D scene-understanding input representation for VLMs evaluated on academic benchmarks. Positive impacts (more compute-efficient embodied and robotic perception) are indirect, and the method has no direct path to specific negative applications beyond those already inherent to general-purpose VLMs.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We release LoRA adapters fine-tuned on indoor scene QA datasets for spatial reasoning. The released artifacts inherit the safety profile of the base Qwen3-VL model and do not introduce additional misuse risk (no image generation, no scraped data, no high-risk capabilities).

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All used assets are cited at point of use: the Qwen3-VL Bai et al. [2025] base model, the ScanNet Dai et al. [2017], ScanNet++ Yeshwanth et al. [2023], and ARKitScenes Baruch et al. [2021] scene datasets, and the SQA3D Ma et al. [2023], VSI-Bench Yang et al. [2025a], VLM-3R Fan et al. [2026], ViCA Feng [2025], and SPBench Li et al. [2026] annotation datasets. Each is used under the terms of its original license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released model, with both stage adapters and the 3D position embedding, is documented by its model card at `https://huggingface.co/BaranowskiBrt/OneCanvas-Qwen3-VL-8B`, which states the base model, training procedure, evaluation commands and benchmark protocols, and license (MIT, with the Apache-2.0 base model). No new dataset is released. The spatial pretraining curriculum is generated procedurally at training time, and its generator ships with the released code (covered under Q5).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All training and evaluation data come from existing public 3D scene datasets and procedurally generated synthetic curriculum samples.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects, so no IRB approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as an important, original, or non-standard component of the core methodology, only for standard writing and editing assistance, so declaration is not required.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.