Lifting Unlabeled Internet-level Data for 3D Scene Understanding

Yixin Chen[1] Yaowei Zhang[1] Huangyue Yu[1] Junchao He[1,2] Yan Wang[1]
Jiangyong Huang[1,3] Hongyu Shen[1,4] Junfeng Ni[1,5] Shaofei Wang[1]
Baoxiong Jia[1] Song-Chun Zhu[1,3,5] Siyuan Huang[1]

[1] State Key Laboratory of General Artificial Intelligence, BIGAI [2] Beijing University of Posts and Telecommunications
[3] Peking University [4] Beijing Institute of Technology [5] Tsinghua University

hhttps://sv-pp.github.io/

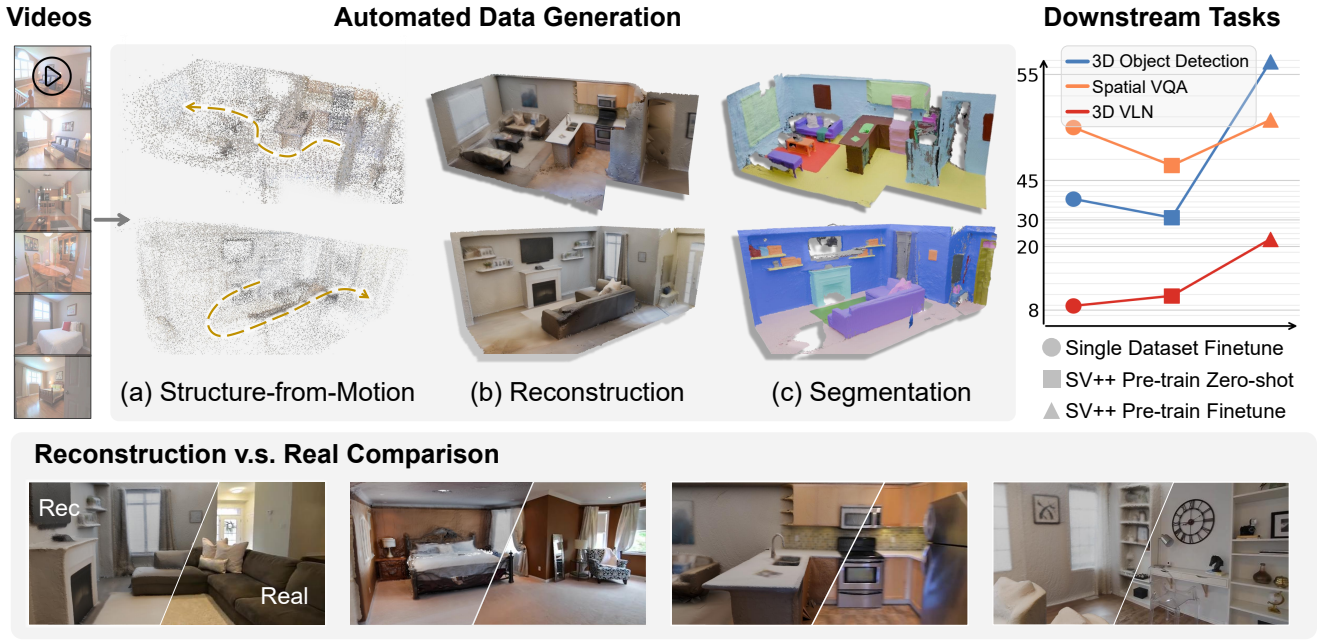


Figure 1. **Overview of SceneVerse++.** From unlabeled internet videos, we build automated data engines to create training data for comprehensive 3D scene understanding, realizing strong zero-shot performance on existing benchmarks, with further improvement after finetuning. This pinpoints future direction towards 3D spatial intelligence through improved automation on unlabeled, web-scale data.

Abstract

Annotated 3D scene data is scarce and expensive to acquire, while abundant unlabeled videos are readily available on the internet. In this paper, we demonstrate that carefully designed data engines can leverage web-curated, unlabeled videos to automatically generate training data, to facilitate end-to-end models in 3D scene understanding alongside human-annotated datasets. We identify and analyze bottlenecks in automated data generation, revealing critical factors that determine the efficiency and effectiveness of learning from unlabeled data. To validate our approach across different perception granularities, we evaluate on three tasks spanning low-level perception, i.e.*, 3D object detection and instance segmentation, to high-level reasoning,* i.e.*, 3D spatial Visual Question Answering (VQA) and Vision-Lanugage Navigation (VLN). Models trained on our generated data demonstrate strong zero-shot performance and show further improvement after finetuning. This demonstrates the viability of leveraging readily available web data as a path toward more capable scene understanding systems.*

1. Introduction

With the crucial role of 3D scene understanding in human and embodied intelligence, the field has made remarkable strides in recent years, spanning tasks from geometric perception (*e.g.*, depth estimation [25, 28, 32, 33, 36], camera

pose estimation [41, 91, 105, 106, 108]), semantic understanding (*e.g.*, 3D object detection [29, 58, 78] and segmentation [51, 93, 100]) to high-level reasoning (*e.g.*, 3D visual grounding [1, 17, 124] and spatial reasoning [3, 6, 73, 114]). The success of deep learning in this domain is fundamentally tied to the availability of large-scale, annotated, real-world 3D datasets [8, 27, 74, 115].

While methods [75, 106, 107] in 3D scene understanding continue to improve, progress in 3D scene data with high-quality annotations, on the contrary, has largely stagnated. Unlike 2D images [15, 92], which can be easily scraped and annotated from the web, capturing and labeling 3D data is far more challenging. The common procedure for 3D scene data curation involves recording thousands of frames with specialized hardware, *e.g.*, RGB-D sensors or LiDAR, reconstructing 3D meshes, and manually labeling 3D structures for dense semantic annotations. In fact, academia has not seen a quantitative leap in 3D data scaling since the pioneering ScanNet [27]; instead, efforts have focused on simplifying the procedure to get more scenes, *e.g.*, ARKitScenes [8] with 2x real-world sites at the cost of coarser scans and labeling, or improving data quality on a manageable number of scans, *e.g.*, ScanNet++ [115].

In this paper, we show that leveraging carefully designed data engines to generate training data from unlabeled, web-scale videos is a promising approach to address the scarcity of annotated 3D scenes. These data engines, often modularized, draw upon prior knowledge from existing foundation models [7, 62, 87] or scene-specific optimization methods that target particular aspects of general scene understanding, *e.g.*, reconstruction [53, 77, 79, 81], instance segmentation [37, 60, 112], and open-set semantics [54, 66, 83]. Since these submodular methods vary in representation, methodology, and technical focus, design choices for automatic data generation are non-trivial. The effectiveness of scaling generated data is task-dependent and strongly influenced by both quality and efficiency considerations.

To this end, we systematically analyze the bottlenecks in creating automated data engines for 3D scene understanding, provide guidelines on how to scale end-to-end (E2E) models, and pinpoint what submodular models should prioritize in future development. From Internet videos, we curate SceneVerse++ of 6,687 real-world scenes with images, camera poses, dense reconstructions, instance segmentations and high-level reasoning annotations. We demonstrate the effectiveness of internet-scale data by empowering three exemplar tasks in 3D scene understanding:

- 3D detection and segmentation. The models trained on SceneVerse++ realize strong zero-shot performance on ScanNet and ARKitscenes, and further significantly improve after finetuning (+20.6 for F1@.25).
- 3D spatial Visual Question Answering (VQA): Training on SceneVerse++ significantly improves the spatial reasoning performance of Vision-Language Models (VLMs), achieving zero-shot performance comparable to models trained on ground-truth 3D scenes.
- 3D Vision-Lanugage Navigation (VLN): We examine the zero-shot transfer from real-world videos to navigation in simulation, and demonstrate SceneVerse++ brings an extra 14% navigation success rate after finetuning.

2. Related Work

2.1. 3D Scene Understanding and Datasets

Early work in 3D scene understanding primarily focuses on tasks such as semantic segmentation [84, 109, 119], instance segmentation [51, 93, 100, 125], and object detection from images [11, 20, 29] or point clouds [5, 58, 75, 78]. Beyond geometry-centric perception, there has been growing interest in vision-language tasks within 3D scenes, including object referral [1, 17, 120], captioning [18, 19, 22, 118], spatial reasoning [6, 21, 46, 73, 114], and navigation [3, 45, 59, 86]. The shift is driven by the popularity of E2E VLMs [7, 23, 82, 101], offering advantages in multi-tasking [57] and scaling [50, 52, 98] in both model architecture and training data.

The success of these E2E models relies critically on 3D datasets [55, 74, 88, 102, 116, 122] with detailed annotation, such as pioneering ScanNet [27], later ARKitScenes [8] captured with portable devices, and ScanNet++ [115] with higher-quality scans. However, unlike their 2D counterparts, the scaling of the 3D datasets faces significant bottlenecks in capture and labeling costs that hinder further expansion. In the meantime, the internet contains orders of magnitude more unlabeled data that captures our 3D world.

In this paper, we advocate for advancing comprehensive 3D scene understanding by leveraging these unlabeled internet videos. We build upon methods that address intermediate problems in scene understanding, achieved by leveraging pre-trained models in a training-free [26, 71, 107, 112] or weakly-supervised manner [37, 68, 94, 96] to inject knowledge into 3D, *e.g.*, open-vocabulary 3D segmentation by lifting 2D results [100]. We build automated data engines on top of these submodules, leveraging their complementary strengths while mitigating limitations, achieving an efficiency-efficacy balance in internet-level data scaling.

2.2. Leveraging Internet-level Videos

Recognizing the scarcity of 3D datasets, an emerging direction is to harness video data to lift 2D content into 3D annotations for training. For instance, Miao et al. [76] proposes using existing 2D single-view datasets with estimated depth to generate 3D annotations. However, their data generation is bound to existing datasets [65, 95] with 2D segmentation annotations and operates at the single-image level, presenting a significant gap towards whole-

scene understanding. The abundant internet videos present an attractive, untapped resource, and recent work has begun to explore this direction, but mostly on training generative video [2, 44, 69, 72, 104] or novel-view synthesis (NVS) [30, 67, 70, 90] models. In pursuit of scalable 3D scene understanding [64, 97, 113], RoomTour3D [40] generates video instructions for navigation through summarization and candidate view selection, while NaVILA [24] incorporates real video trajectories into training to improve instruction-following in Vision-Lanugage Navigation (VLN). However, they remain confined to the navigation domain, without addressing broader spatial reasoning or scene understanding. Moreover, they often treat the multi-module data generation pipeline as given, offering little analysis of which components are most critical or where errors propagate. In contrast, our work addresses comprehensive 3D scene understanding tasks, from low-level perception to high-level reasoning, and provides systematic analyses, examining both the efficiency and efficacy of transforming internet-scale data for task-specific training.

3. Data Curation for SceneVerse++

Our work focuses on 3D scene understanding for static indoor scenes. The first step for task-specific 3D scene understanding is to curate internet videos and convert them to a basic 3D representation consisting of camera poses and sparse 3D geometry. Inspired by prior work on internet data processing [2, 40, 64], our data pipeline combines video curation with Structure-from-Motion (SfM) [41], encompassing shot splitting, filtering, key frame extraction, pixel matching, global bundle adjustment, and quality check.

We use TransNetV2 [99] to detect shots in long-form videos and discard very short clips. The filtering process removes low-quality or unsuitable content, including pure black screen, visual noise, humans [42], and outdoor scenes [123]. To handle potentially long-duration internet videos, we select keyframes based on parallax rather than uniform sampling [40], ensuring well-constrained triangulation with redundancy control. For sparse reconstruction and camera pose estimation, we adopt a dense pixel matching and bundle adjustment approach, which provides more robust camera poses and sparse point clouds than existing feed-forward methods [106, 107]. The overall pipeline resembles Mast3R-SFM [31], and we introduce optimized pseudo-track pixels to improve memory efficiency for long-term videos (*e.g.*, >300 frames) and incorporate relative image similarity to address the false-positive bias in existing pixel matching models [61]. Finally, we filter out the scenes with small spatial coverage, relatively empty space, or wrong SfM results. This can be achieved by existing VLMs [7, 82, 101], but we resort to human annotation (<10 seconds/scene) to ensure data quality for downstream tasks.

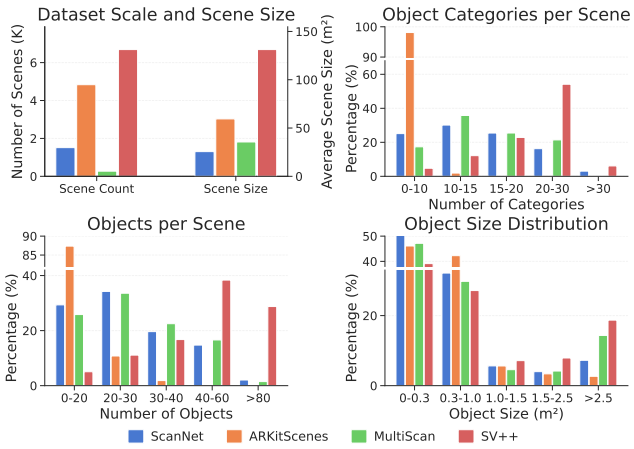


Figure 2. **Statistics comparison.** SceneVerse++ encompasses more scenes, larger areas, and greater object diversity compared with existing real-world datasets.

Statistics The dataset statistics are shown in Fig. 2, with comparison with ScanNet [27], MultiScan [74] and ARKitScenes [8]. Starting from 8,217 videos from the open internet platforms, we obtain 6,687 scenes, exceeding ARKitScenes captured with portable devices. SceneVerse++ contains multi-floor, multi-room scans from long-range videos, producing scenes significantly larger[1] than existing room-scale or lab-based datasets. More details about data curation, SfM methods, and examples with camera trajectories and sparse geometry are presented in *supplementary*.

4. SceneVerse++ for 3D Scene Understanding

In this section, we present how to leverage SceneVerse++ to generate training data and improve on three representative tasks in 3D scene understanding.

4.1. 3D Object Detection and Segmentation

Task and Benchmark The 3D object detection and segmentation task aims to localize distinct objects within a 3D scene, assigning each a precise geometric boundary and a semantic label. This task serves as a bridge between low-level 3D reconstruction and high-level scene understanding. In the following, we first introduce the data engine that generates 3D instance annotations, and then evaluate its effectiveness on real-world benchmarks.

Data Generation To obtain the complete reconstructed meshes and instance-level annotations from the sparse outputs of SfM, we design a reconstruction and segmentation pipeline, as illustrated in Fig. 3. It transforms internet images into 3D scenes, considering both efficiency and effectiveness in large-scale data generation.

[1] Scene area is approximated by the product of extents along x-y plane.

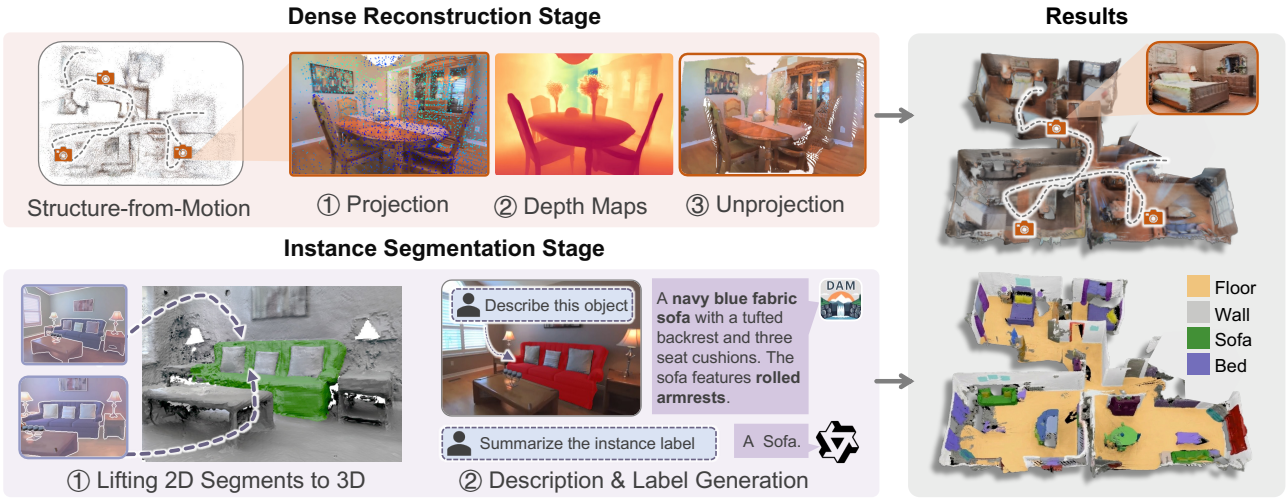


Figure 3. **Overview of data generation.** The pipeline leverages a modular design for automatic 3D reconstruction and segmentation.

Dense Reconstruction. Recent advances in 3D reconstruction introduce various approaches with different trade-offs between quality and efficiency. Neural rendering methods [16, 39, 48, 79–81, 117] produce photo-realistic rendering and recover detailed geometry, but they require dense computation for per-scene optimization, especially for large and complex environments. End-to-end reconstruction frameworks [106, 107] enable dense point cloud reconstruction directly from images, providing convenience and speed; however, they struggle with long videos due to memory constraints and often exhibit obvious artifacts in multi-view consistency and geometry distortion.

To balance efficiency and reconstruction quality, we design a reconstruction pipeline based on metric depth estimation that effectively leverages SfM outputs from Sec. 3. Specifically, we project the reconstructed sparse 3D points onto the image plane to obtain sparse depth maps, which serve as priors for PriorDA [110] to predict dense metric depth maps. The predicted depths are then fused using a Truncated Signed Distance Function (TSDF) representation to produce watertight 3D meshes. During fusion, unreliable large depth values are truncated, and radius- and statistical-based filters further remove floating noisy points. This design achieves stable, high-quality reconstructions with reduced computational cost, enabling efficient processing of large-scale internet videos while maintaining sufficient accuracy for downstream tasks. Qualitative results and computation time comparison are shown in Fig. 4.

Instance Segmentation. Recent advances in per-scene 3D segmentation have also explored different paradigms. For example, image-based approaches, such as the SAM series [56, 89], effectively identify 2D object masks across frames, but do not explicitly leverage 3D spatial information. When applied to long video sequences, they often produce duplicated instances due to incorrect cross-view associations. In contrast, feature-lifting methods [9, 37, 96] exploit spatial correspondences across multiple views through rendering [53, 77], but their performance is affected by the rendering quality and typically requires substantial computational resources and processing time for long videos.

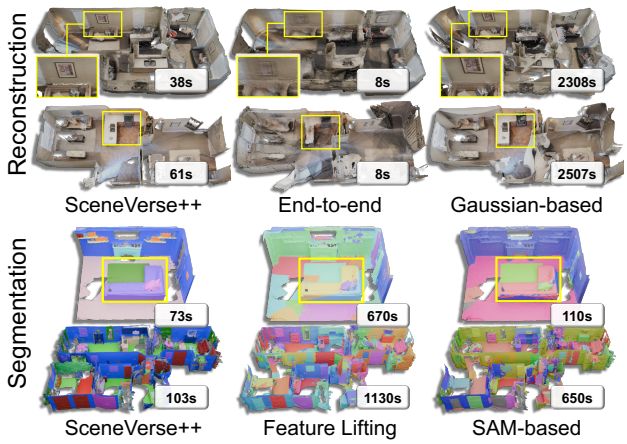


Figure 4. **Reconstruction and segmentation comparison**, where SceneVerse++ features a balance in quality and efficiency.

To overcome these challenges, we choose to lift 2D masks to 3D using the dense reconstruction results. Specifically, we first apply CropFormer [85] to obtain per-frame segmentation masks, which are then aggregated in 3D space based on neighboring-frame view consensus [112] and spatial agreement. Finally, we employ Describe Anything [63] and Qwen2-VL [7] to automatically generate textual descriptions for each 3D instance and align their semantic labels to the ScanNet category set. The segmentation comparison is shown in Fig. 4.

Statistics In practice, the average runtime for each scene is 71 seconds for dense reconstruction and 96 seconds for segmentation. On average, each scene in SceneVerse++ has 49 objects across 21 distinct categories, both surpassing existing datasets as shown in Fig. 2. This reflects the greater diversity of object types and richer scene compositions in our data. In addition, the object size distribution in SceneVerse++ closely aligns with that of real-world datasets, indicating that our reconstructed scenes preserve realistic scale and spatial relationships.

Performance We validate the effectiveness of our dataset on 3D object detection with SpatialLM [75] and 3D instance segmentation with Mask3D [93]. The quantitative results are summarized in Tab. 2.

- SpatialLM, derived from a Multimodal Large Language Model (MLLM), generates structured 3D scene descriptions for object detection and is originally trained on a synthetic dataset of over 12,000 indoor scenes. We adopt the same base model and evaluate on two real-world benchmarks. Without fine-tuning, the model trained on SceneVerse++ achieves slightly better detection on ScanNet and ARKitScenes than training only on synthetic data. When fine-tuned on ScanNet, the model pretrained on SceneVerse++ achieves a substantial improvement, *i.e.*, F1@0.25 of 58.6 *vs*. 38.0; this shows SceneVerse++ better captures real-world distributions and provides a better initialization. Training from scratch on ScanNet fails to converge, as the adapter linking 3D encoder [111] to MLLM requires significant pretraining [49, 75].
- The results on 3D instance segmentation using Mask3D reveal a different trend: the model pretrained solely on SceneVerse++ does not transfer well to ScanNet, but it consistently improves performance across all metrics after finetuning compared with training from scratch. This drop stems from Mask3D's reliance on segment-level masks obtained from a graph-based segmentation [35], which is highly sensitive to sensor and reconstruction pipelines. This highlights a key factor in model scaling - their susceptibility to domain-specific bias.

More details, additional experiments and ablations, and further discussions are provided in *supplementary*.

4.2. 3D Spatial VQA

Task and Benchmark Visual-spatial intelligence [114] requires the combination of visual perception, linguistic understanding, temporal reasoning, and spatial reasoning [38, 43]. Despite being a critical capability for future embodied agents to explore and perform tasks in the 3D world, it remains a challenging frontier for current VLMs. To investigate how SceneVerse++ can improve the spatial reasoning ability of VLM, we focus on 3D spatial Visual Question Answering (VQA), which requires a model to answer questions about 3D space by inferring spatial relations from 2D visual input. We evaluate on VSI-Bench [114], a 3D spatial understanding benchmark constructed from egocentric videos in ScanNet, ScanNet++, and ARKitScenes. It contains over 5,000 question-answering (QA) pairs spanning eight task types, presented as Multiple-Choice Answers (MCA) or Numerical Answers (NA). MCA performance is measured by mean accuracy, while NA performance is calculated using relative accuracy across multiple confidence thresholds.

Table 1. **Testing SpatialLM on 3D object detection.** Performance is reported under different pretraining and finetuning configurations with the same model architecture.

Benchmark	Pretrain	Finetune	F1@.25	F1@0.5
ARKitScenes	SpatialLM	-	35.1	21.2
	SceneVerse++	-	35.8	20.7
ScanNet	-	ScanNet	2.9	0.7
	SpatialLM	-	29.0	19.7
	SceneVerse++	-	30.9	21.3
	SpatialLM	ScanNet	38.0	28.7
	SceneVerse++	ScanNet	**58.6**	**45.4**

Table 2. **Testing Mask3D on 3D instance segmentation**. It presents reliance on data-specific bias that hurts model scaling.

Benchmark	Pretrain	Finetune	AP_{25}	AP_{50}	AP
ScanNet	-	ScanNet	36.1	31.8	22.8
	SceneVerse++	-	15.4	13.0	8.3
	SceneVerse++	ScanNet	**38.5**	**32.9**	**23.6**

Data Generation We generate general spatial QAs by transferring the geometry and semantic information in 3D scenes (Sec. 4.1) to 3D scene graphs [4, 50, 103], following VLM-3R [34]. Each node in the scene graph represents a distinct 3D object instance, and edges represent pairwise spatial relations. Leveraging these structured semantics, QA pairs are automatically generated for *Object Counting, Relative Distance, Relative Direction, Object Size, Absolute Distance*, and *Room Size* by designing task-specific templates [34]. For the *Route Planning* task, we generate QA pairs by employing a VLM [7] to summarize the navigation trajectories within 3D environments (introduced in Sec. 4.3). The summary is then transformed into fill-in-the-blank Multiple-Choice questions by masking specific actions. The *Appearance Order* task is not included following the setting of VLM3R.

Statistics Applying the automatic generation pipeline to the reconstructed scenes in SceneVerse++ yields 632K spatial VQA data following the VSI-Bench format. It comprises 391K samples for MCA and 241K samples for NA, respectively. More details on data generation and question type distribution are in *supplementary*.

Table 3. **Evaluation results on VSI-Bench.** Performance is reported on both the full set and ARKitScenes subset: 1) zero-shot test (-); 2) trained on SceneVerse++ (SV++); 3) trained on VLM-3R data from ScanNet and ScanNet++ (SN, SN++); and 4) trained on the combination of 2) and 3) (All). The figures of "SN, SN++" and "All" on the full set indicate in-domain (ID) results, while others are out-of-domain (OOD) results. SceneVerse++ is more effective in improving general spatial knowledge but less in domain-specific knowledge.

Model	Dataset Source	VSI-Bench Fullset$_{SN, SN++, ARKit}$ App. Ord.	Abs. Dist.	Obj. Cnt.	Rel. Dist.	Obj. Size	Room Size	Route Plan	Rel. Dir.	Avg.	Subset$_{ARKit}$ Avg.
Qwen2.5 -VL-3B	-	27.3	17.4	25.2	37.2	16.5	26.2	28.4	45.4	27.9	28.1
	SV++	26.1	30.2	61.8	49.3	49.8	43.9	33.6	47.8	42.8	48.0
	SN, SN++	32.4	39.6	67.4	48.9	64.0	53.8	38.7	44.9	48.7	49.0
	All	27.2	39.3	67.5	50.3	63.5	54.0	36.6	55.8	49.3	51.3
Qwen2.5 -VL-7B	-	34.5	21.0	41.5	38.6	50.5	36.7	29.4	41.0	36.6	39.4
	SV++	43.4	28.9	63.8	48.9	57.0	46.4	35.1	48.0	46.4	49.1
	SN, SN++	37.7	38.8	68.3	52.8	64.8	53.0	37.1	47.3	50.0	48.8
	All	29.8	38.3	67.1	51.7	65.8	53.5	41.2	57.3	50.7	50.5

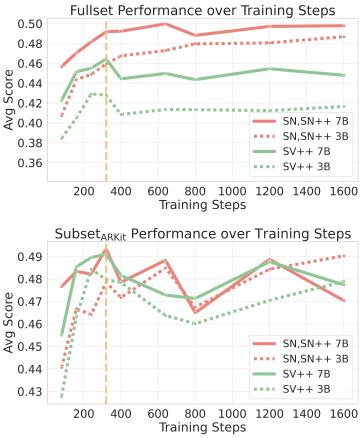


Figure 5. **Training dynamics.**

Performance We evaluate the performance of Qwen2.5-VL after LoRA fine-tuning [47] on VSI-Bench, which spans ScanNet (SN), ScanNet++ (SN++), and ARKitScenes (ARKit). Given the domain discrepancy between datasets, we regard training and testing on SN and SN++ as in-domain (ID), and out-of-domain (OOD) otherwise. For fairness, we sample 202K data from SceneVerse++ for training, comparable with 206K samples on SN and SN++ from VLM3R [34]. We report quantitative results in Tab. 3 and key observations as follows:

- **Spatial reasoning enhancement.** SceneVerse++ can improve the spatial reasoning capability of the base VLMs, yielding +14.9 for the 3B model and +9.8 for 7B on VSI-Bench full set. This highlights SceneVerse++ as a reliable and promising data source for advancing existing VLMs.
- **Domain generalization.** We observe comparable performance between SceneVerse++ and SN/SN++ on the VSI-Bench ARKit subset, indicating their comparable domain generalizability, despite that SN and SN++ have groundtruth annotations. This contrasts with the performance gap observed on the VSI-Bench full set, reflecting a larger-than-expected domain gap across datasets. Training on all data sources (All) further improves performance on both the full set and ARKit subset, showing the benefit of a broader domain covered in SceneVerse++.
- **Category-wise difference.** Per-category analysis reveals that SceneVerse++ delivers greater improvement on categories concerning *general spatial knowledge* such as *Relative Distance* and *Relative Direction*, which are less susceptible to domain-specific distribution. In contrast, it exhibits worse results on categories highly relying on *domain-specific knowledge* such as *Object Count* and *Room Size*, likely due to variations in object and scene distributions, as illustrated in Fig. 2.
- **Training dynamics.** We visualize the evolution of evaluation results within one training epoch in Fig. 5. A distinct turning point (green dashed line) emerges: model performance consistently improves before this point, after which in-domain training (SN, SN++ curves on full set) continues to rise while others plateau or decline. This provides further evidence of domain gap and overfitting to *domain-specific knowledge*, aligning with findings from concurrent works [12, 13].

4.3. 3D Vision-Lanugage Navigation (VLN)

Task and Benchmark The goal of VLN is to enable embodied agents to follow natural language instructions and navigate toward specified goals within 3D environments. Room-tour videos from the internet provide a valuable proxy for natural human navigation in real indoor spaces. Unlike prior work [24, 64], we focus on the key factors to provide rich, continuous trajectories that can bridge the gap between model navigation and real-world embodied behaviour. We adopt the widely used Room-to-Room (R2R) benchmark [3] built on Matterport3D [14] environments, where an agent receives a sequence of rendered egocentric observations and a goal-directed instruction as input, and outputs a sequence of discrete navigation actions. The action space consists of fixed translation and rotation steps, where movements are discretized into three distance bins of [25, 50, 75] cm and rotations into [15°, 30°, 45°].

Data Generation R2R establishes a controlled and standardized setting for instruction-following navigation, but its simulated trajectories differ from how humans naturally explore real environments. Specifically, VLN trajectories are goal-directed shortest paths with all forward-facing movements, whereas room-tour videos capture free-form exploration in the environment, often exhibiting *irregular camera motion*, *redundancy*, and *backtracking*, shown in Fig. 7. These discrepancies introduce challenges in deriving navigation-consistent trajectories from room-tour videos for VLN model learning. To bridge this gap, we analyze human motion patterns in real videos and design a three-stage pipeline to convert room-tour camera trajectories to navigation trajectories as in Fig. 6.

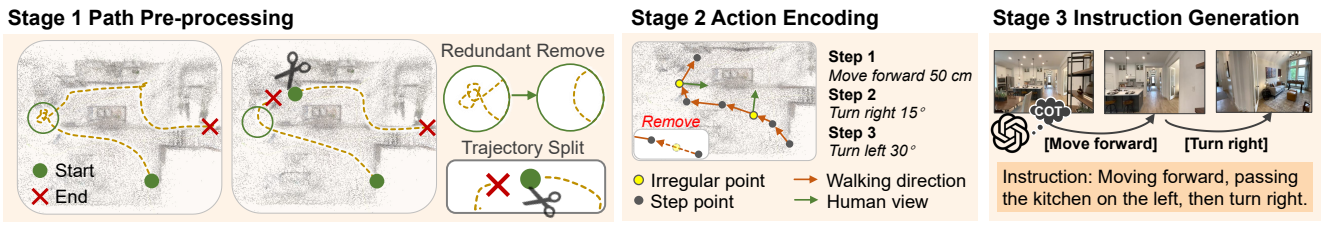


Figure 6. **Overview of the VLN data generation pipeline.** We construct VLN data from room-tour videos by (i) preprocessing trajectories to eliminate redundant local rotations and segmenting long paths into sub-paths suitable for instruction generation; (ii) converting camera transitions within each sub-path into R2R-style navigation actions; and (iii) generating instructions for each sub-path using VLMs.

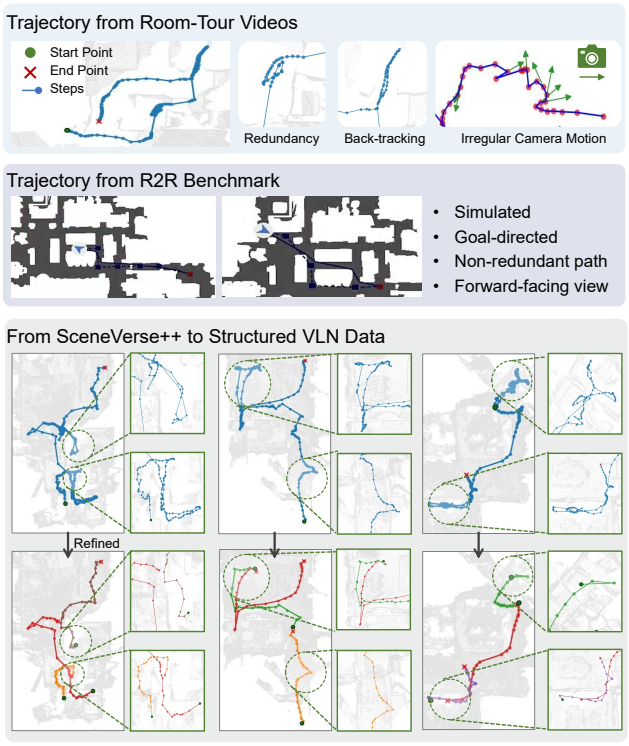


Figure 7. **Trajectory comparison.** Top: Room-tour videos show irregular and redundant camera motions. Middle: R2R trajectories are smooth and goal-directed. Bottom: raw videos are converted into VLN-compatible data. Different colors indicate sub-paths.

Path Pre-processing. The goal of this stage is to extract clean and coherent trajectories from room-tour videos using the SfM reconstructions described in Sec. 3. We first cluster camera positions within a 0.5m radius to merge nearby viewpoints, keeping one representative node per cluster to maintain trajectory continuity and remove redundant local rotations. Next, to eliminate backtracking, we split long trajectories into sub-paths. Specifically, we detect cluster centers along each trajectory and use them as potential break points—only when the two adjacent segments separated by a center exceed 15 steps do we perform a split. Finally, we filter out steps that involve rotations greater than 90° or translations larger than 70cm. The resulting trajectories are shown at the bottom of Fig. 7.

Action Encoding. Action encoding converts each structured trajectory into a sequence of discrete actions for VLN model training. We extract each node's 3D camera pose $(\mathbf{R}_i, \mathbf{t}_i)$ from the SfM reconstruction, project it onto the ground plane, and represent it as $p_i = [x_i, y_i, \theta_i]$, where (x_i, y_i) denotes the position and θ_i is the yaw angle derived from $\mathbf{R}_i$. As room-tour videos often contain irregular camera motions, we remove non-navigational "looking around motions" by removing actions whose viewing direction deviates from walking direction. Finally, the movement action is defined by the Euclidean distance between p_i and p_{i+1}, and rotation action by $\Delta\theta_i = \theta_{i+1} - \theta_i$. Both are discretized following the R2R convention for compatibility with existing VLN benchmarks.

Instruction Generation. We leverage VLM to generate natural language navigation instructions aligned with both motion and visual context, by providing both the corresponding images and encoded actions. The VLM first reasons about local motion changes using Chain-of-Thought (CoT) and then composes coherent instructions to describe the entire trajectory. To enhance linguistic diversity and improve generalization, we generate three stylistically varied instructions for each trajectory.

Statistics The VLN data derived from SceneVerse++ contains 9,631 trajectories, each averaging 12.8 meters in length and 15 steps. For each trajectory, we provide three instructions in formal, conversational, and narrative styles, averaging 42, 47, and 57 words, respectively. After discrete action encoding, forward and rotational movements account for 52% and 48%, reflecting a balanced motion distribution. R2R comprises 7,189 trajectories and 21,567 instructions averaging 29 words, collected from 29 simulated indoor scenes. Our dataset extends these benchmarks by incorporating richer linguistic diversity and natural, real-world motion patterns captured from internet videos.

Performance We validate the effectiveness of our constructed VLN dataset using LLaVA-Video [121] as the base model. All experiments are evaluated on the validation set

Table 4. **VLN evaluation under different training settings**. TR denotes trajectory refinement, and IE for instruction enrichment.

Pretrain	Finetune	SR↑	OS↑	SPL↑	Dist↓	PL
-	R2R	0.088	0.133	0.076	8.031	5.222
R2R + SceneVerse++	-	0.188	0.262	0.150	8.117	10.496
SceneVerse++	–	0.107	0.194	0.074	9.418	14.097
SceneVerse++	R2R	**0.228**	**0.315**	**0.191**	**7.65**	11.642
SceneVerse++ (w/o IE)	–	0.022	0.043	0.016	8.978	2.333
SceneVerse++ (w/o IE)	R2R	0.074	0.111	0.062	8.175	5.009
SceneVerse++ (w/o TR)	–	0.036	0.045	0.032	8.662	2.521
SceneVerse++ (w/o TR)	R2R	0.177	0.298	0.130	8.23	11.949

of R2R [3]. To ensure a fair comparison, we use the same number of training epochs across all experiments. Evaluation follows standard VLN metrics, including *Distance to Goal* (Dist.), *Success Rate* (SR), *Oracle Success* (OS), *Success-weighted Path Length* (SPL) and *Path Length* (PL).

- **Domain Transfer and Training Strategies.** We investigate how incorporating real-world video data affects VLN performance on R2R. As shown in Tab. 4, training on SceneVerse++ yields a modest SR improvement (0.107 *vs*. 0.088) under zero-shot evaluation compared with training on R2R alone. The substantially longer path length (14.1 *vs*. 5.2) reflects that the richer and more complex trajectories in room-tour videos offer more diverse and challenging experiences to learn navigation behaviors, compared with the shortest paths in R2R. Further fine-tuning on R2R significantly boosts SR to 0.228, demonstrating that large-scale video pretraining provides valuable visual and linguistic priors for navigation tasks. In contrast, directly mixing SceneVerse++ with R2R during training yields weaker results, suggesting that the visual gap between real videos and simulator-rendered scenes makes naive mix-training less effective.
- **Data Quality.** To further investigate the impact of data quality on VLN performance, we conduct ablation experiments on two core components in data generation: trajectory refinement (TR) and instruction enrichment (IE). As shown in Tab. 4, removing either component results in a clear performance drop. Even after fine-tuning on R2R, models pretrained on these ablated datasets fail to fully recover, *e.g*., SR decreases from 0.228 to 0.177 when TR is removed. These results demonstrate that raw internet videos alone are insufficient for effective VLN training; task-specific data processing is essential. We additionally include comparisons with YouTube-based VLN data from NaVILA [24] in *supplementary*, where SceneVerse++ enables stronger model performance due to higher quality.

5. Discussion and Conclusion

In this paper, we investigate pathways to advance comprehensive 3D scene understanding across multiple tasks by leveraging unlabeled internet videos. We develop automated data engines to generate training data and demonstrate that high-quality data can benefit downstream tasks. We further offer the following discussions on data generation, benchmarks, and model scaling. Limitations and future work are discussed in *supplementary*.

Scaling capability of models. In our experiments, we observe clear differences in how models scale. Models that depend on task-specific, pre-computed segments are more sensitive to data distribution shifts and hyperparameter changes, resulting in limited scalability and weaker generalization in 3D instance segmentation (Sec. 4.1). In contrast, models that operate directly on raw and widely available modalities, *e.g*., 3D voxels or RGB-based MLLMs, exhibit more robust scaling behavior. This contrast is less evident in two-dimensional settings due to the uniformity of image inputs, but becomes increasingly pronounced when scaling 3D understanding.

Fair evaluation of capability and benchmarks. Existing benchmarks may not fully reflect a model's true capability, *e.g*., VSI-Bench exhibits strong QA distribution bias [12] and VLMs overfit to data-specific cues for in-domain evaluation (Sec. 4.2). To ensure fair assessment, future evaluation should emphasize zero-shot testing on existing benchmarks, avoiding data contamination and minimizing data distribution gap, or more benchmarks that accurately measure 3D scene understanding and generalization in the wild.

Understanding data and task-specific biases. Effective data scaling requires not only high-quality data, but also a careful examination of data distribution and task-specific or benchmark-specific characteristics. Performance is strongly affected by factors that remain hidden without deeper analysis, *e.g*., the discrepancy between natural camera motion in real-world videos and goal-directed navigation trajectories (Sec. 4.3). Identifying such mismatches is essential to avoid biases and to ensure that scaled data provides meaningful improvements for the intended task.

Advancing automated data generation. Building an automated data generation pipeline reveals significant challenges in using existing models to produce high-quality data for 3D scene understanding from in-the-wild videos. Modules such as SfM, instance segmentation, and language grounding are typically trained on task-specific or small-scale benchmarks, limiting their generalization capabilities and introducing sequential errors when combined together for in-the-wild spatial understanding. As a result, substantial effort is required for careful model selection and non-trivial coordination across modules. We advocate that future development of these sub-modules should align with the broader goal of enabling robust in-the-wild 3D understanding, with evaluation based not only on task-specific performance but also on their contribution to reliable automated data generation pipelines.

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[2] Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025. 3

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 6, 8

[4] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *International Conference on Computer Vision (ICCV)*, 2019. 5

[5] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[6] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[7] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2, 3, 4, 5

[8] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *Proceedings of Advances in Neural Information Processing Systems Datasets and Benchmarks (NeurIPS Datasets and Benchmarks Track)*, 2021. 2, 3

[9] Yash Bhalgat, Iro Laina, João F Henriques, Andrea Vedaldi, and Andrew Zisserman. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 4

[10] Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations (ICLR)*, 2025. 3

[11] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[12] Ellis Brown, Arijit Ray, Ranjay Krishna, Ross Girshick, Rob Fergus, and Saining Xie. Sims-v: Simulated instruction-tuning for spatial video understanding. *arXiv preprint arXiv:2511.04668*, 2025. 6, 8

[13] Ellis Brown, Jihan Yang, Shusheng Yang, Rob Fergus, and Saining Xie. Benchmark designers should "train on the test set" to expose exploitable non-visual shortcuts. *arXiv preprint arXiv:2511.04655*, 2025. 6

[14] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *International Conference on 3D Vision (3DV)*, 2017. 6

[15] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[16] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 4

[17] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[18] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X Chang. D3net: a speaker-listener architecture for semi-supervised dense captioning and visual grounding in rgb-d scans. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[19] Sijin Chen, Hongyuan Zhu, Xin Chen, Yinjie Lei, Gang Yu, and Tao Chen. End-to-end 3d dense captioning with vote2cap-detr. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[20] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[21] Yixin Chen, I Guoxi Zhang, Yaowei Zhang, Hongming Xu, Peiyuan Zhi, Qing Li, and Siyuan Huang. Synergai: Perception alignment for human-robot collaboration. In *International Conference on Robotics and Automation (ICRA)*, 2025. 2

[22] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[23] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[24] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. In *Robotics: Science and Systems (RSS)*, 2025. 3, 6, 8, 4

[25] Jaeyoung Chung, Jeongtaek Oh, and Kyoung Mu Lee. Depth-regularized optimization for 3d gaussian splatting in few-shot images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[26] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. 2

[27] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[28] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[29] Zhipeng Ding, Xu Han, and Marc Niethammer. Votenet: A deep learning label fusion method for multi-atlas segmentation. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2019. 2

[30] Junting Dong, Qi Fang, Tianshuo Yang, Qing Shuai, Chengyu Qiao, and Sida Peng. ivs-net: Learning human view synthesis from internet videos. In *International Conference on Computer Vision (ICCV)*, 2023. 3

[31] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In *International Conference on 3D Vision (3DV)*, 2025. 3, 1

[32] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 1

[33] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *International Conference on Computer Vision (ICCV)*, 2023. 1

[34] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. *arXiv preprint arXiv:2505.20279*, 2025. 5, 6

[35] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 2004. 5

[36] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[37] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *International Conference on 3D Vision (3DV)*, 2022. 2, 4

[38] Howard Gardner. *Frames of mind: The theory of multiple intelligences*. Basic books, 2011. 5

[39] Antoine Guédon, Tomoki Ichikawa, Kohei Yamashita, and Ko Nishino. Matcha gaussians: Atlas of charts for high-quality geometry and photorealism from sparse views. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 4

[40] Mingfei Han, Liang Ma, Kamila Zhumakhanova, Ekaterina Radionova, Jingyi Zhang, Xiaojun Chang, Xiaodan Liang, and Ivan Laptev. Roomtour3d: Geometry-aware video-instruction tuning for embodied navigation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[41] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2, 3

[42] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision (ICCV)*, 2017. 3

[43] Annette Herskovits. *Language and spatial cognition*. Cambridge university press Cambridge, 1986. 5

[44] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 3

[45] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[46] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[47] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. 6

[48] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA)*, 2024. 4

[49] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 5

[50] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for

grounded scene understanding. In *European Conference on Computer Vision (ECCV)*. Springer, 2024. 2, 5

[51] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[52] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 2

[53] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA)*, 2023. 2, 4

[54] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[55] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[56] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *International Conference on Computer Vision (ICCV)*, 2023. 4

[57] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[58] Maksim Kolodiazhnyi, Anna Vorontsova, Matvey Skripkin, Danila Rukhovich, and Anton Konushin. Unidet3d: Multi-dataset indoor 3d object detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025. 2

[59] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. 2

[60] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[61] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision (ECCV)*, 2024. 3

[62] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, 2023. 2

[63] Long Lian, Yifan Ding, Yunhao Ge, Sifei Liu, Hanzi Mao, Boyi Li, Marco Pavone, Ming-Yu Liu, Trevor Darrell, Adam Yala, and Yin Cui. Describe anything: Detailed localized image and video captioning. *arXiv preprint arXiv:2504.16072*, 2025. 4

[64] Kunyang Lin, Peihao Chen, Diwei Huang, Thomas H Li, Mingkui Tan, and Chuang Gan. Learning vision-and-language navigation from youtube videos. In *International Conference on Computer Vision (ICCV)*, 2023. 3, 6

[65] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 2

[66] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2

[67] Kunhao Liu, Ling Shao, and Shijian Lu. Novel view extrapolation with video diffusion priors. *arXiv preprint arXiv:2411.14208*, 2024. 3

[68] Xi Liu, Chaoyi Zhou, and Siyu Huang. 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[69] Ruijie Lu, Yixin Chen, Yu Liu, Jiaxiang Tang, Junfeng Ni, Diwen Wan, Gang Zeng, and Siyuan Huang. Taco: Taming diffusion for in-the-wild video amodal completion. In *International Conference on Computer Vision (ICCV)*, 2025. 3

[70] Ruijie Lu, Yixin Chen, Junfeng Ni, Baoxiong Jia, Yu Liu, Diwen Wan, Gang Zeng, and Siyuan Huang. Movis: Enhancing multi-object novel view synthesis for indoor scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[71] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2

[72] Baorui Ma, Huachen Gao, Haoge Deng, Zhengxiong Luo, Tiejun Huang, Lulu Tang, and Xinlong Wang. You see it, you got it: Learning 3d creation on pose-free videos at scale. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[73] 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. 2

[74] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel Chang, and Manolis Savva. Multiscan: Scalable rgbd scanning for 3d environments with articulated objects. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3

[75] Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. Spatiallm: Training large language models for structured indoor mod-

eling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 2, 5

[76] Xingyu Miao, Haoran Duan, Quanhao Qian, Jiuniu Wang, Yang Long, Ling Shao, Deli Zhao, Ran Xu, and Gongjie Zhang. Towards scalable spatial intelligence via 2d-to-3d data lifting. In *International Conference on Computer Vision (ICCV)*, 2025. 2

[77] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 4

[78] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[79] Junfeng Ni, Yixin Chen, Bohan Jing, Nan Jiang, Bin Wang, Bo Dai, Puhao Li, Yixin Zhu, Song-Chun Zhu, and Siyuan Huang. Phyrecon: Physically plausible neural scene reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 4

[80] Junfeng Ni, Yu Liu, Ruijie Lu, Zirui Zhou, Song-Chun Zhu, Yixin Chen, and Siyuan Huang. Decompositional neural scene reconstruction with generative diffusion prior. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[81] Junfeng Ni, Yixin Chen, Zhifei Yang, Yu Liu, Ruijie Lu, Song-Chun Zhu, and Siyuan Huang. G4splat: Geometry-guided gaussian splatting with generative prior. In *International Conference on Learning Representations (ICLR)*, 2026. 2, 4

[82] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2, 3

[83] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[84] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[85] Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Wenbo Li, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High quality entity segmentation. In *International Conference on Computer Vision (ICCV)*, 2023. 4

[86] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[87] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 2

[88] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. In *Proceedings of Advances in Neural Information Processing Systems Datasets and Benchmarks (NeurIPS Datasets and Benchmarks Track)*, 2021. 2

[89] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4

[90] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[91] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[92] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[93] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *International Conference on Robotics and Automation (ICRA)*, 2023. 2, 5

[94] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv: Arxiv-2210.05663*, 2022. 2

[95] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[96] Hongyu Shen, Junfeng Ni, Yixin Chen, Weishuo Li, Mingtao Pei, and Siyuan Huang. Trace3d: Consistent segmentation lifting via gaussian instance tracing. In *International Conference on Computer Vision (ICCV)*, 2025. 2, 4

[97] Xuelun Shen, Zhipeng Cai, Wei Yin, Matthias Müller, Zijun Li, Kaixuan Wang, Xiaozhi Chen, and Cheng Wang. Gim: Learning generalizable image matcher from internet videos. In *International Conference on Learning Representations (ICLR)*, 2024. 3

[98] Mustafa Shukor, Enrico Fini, Victor Guilherme Turrisi da Costa, Matthieu Cord, Joshua Susskind, and Alaaeldin El-Nouby. Scaling laws for native multimodal models. *arXiv preprint arXiv:2504.07951*, 2025. 2

[99] Tomáš Souček and Jakub Lokoč. Transnet v2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020. 3, 1

[100] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2

[101] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2, 3

[102] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[103] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[104] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 3

[105] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[106] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3, 4

[107] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 4

[108] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[109] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019. 2

[110] Zehan Wang, Siyu Chen, Lihe Yang, Jialei Wang, Ziang Zhang, Hengshuang Zhao, and Zhou Zhao. Depth anything with any prior. *arXiv preprint arXiv:2505.10565*, 2025. 4

[111] Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 5

[112] Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 4

[113] Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. *arXiv preprint arXiv:2505.17006*, 2025. 3

[114] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 5

[115] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[116] Huangyue Yu, Baoxiong Jia, Yixin Chen, Yandan Yang, Puhao Li, Rongpeng Su, Jiaxin Li, Qing Li, Wei Liang, Zhu Song-Chun, Tengyu Liu, and Siyuan Huang. Metascenes: Towards automated replica creation for real-world 3d scans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2

[117] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 4

[118] Zhihao Yuan, Xu Yan, Yinghong Liao, Yao Guo, Guanbin Li, Shuguang Cui, and Zhen Li. X-trans2cap: Cross-modal knowledge transfer using transformer for 3d dense captioning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[119] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[120] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[121] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 7

[122] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[123] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 3

[124] 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 *International Conference on Computer Vision (ICCV)*, 2023. 2

[125] 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 (ECCV)*, 2024. 2

Lifting Unlabeled Internet-level Data for 3D Scene Understanding

Supplementary Material

A. Data Curation

As described in Sec. 3, we provide detailed information on how sparse reconstruction data are generated from Internet videos. The raw Internet data are collected from housing-tour videos on YouTube[1] and Bilibili[2], which contain a total of 8,217 videos, from which we obtain 6,687 reconstructed scene instances. The overall data processing pipeline consists of two main stages: preprocessing and reconstruction, as shown in Fig. S.2.

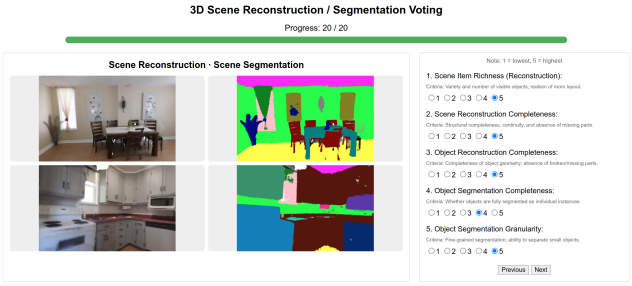
Preprocessing stage Internet videos often contain many shots rather than a continuous shot, which can significantly degrade the reconstruction quality if treated as a single sequence. To address this, we use TransNetV2 [99] to detect the shot boundary and split each video into multiple shots, each treated as an individual scene. Since each clip still includes a large number of redundant or noisy frames, we use parallax-based keyframe selection to retain representative frames, and employ detection models to filter out outdoor frames and frames that contain humans. To ensure both reconstruction efficiency and quality, long sequences are further subdivided based on the number of keyframes, with a maximum clip length of 300 frames and an overlap of 50 frames between adjacent clips.

Reconstruction stage To efficiently establish image correspondences, we use a loop and sequence pairing strategy. In the loop pairing strategy, we extract image features and compute feature distances to other images within a 100-frame range. The top 50 image pairs with feature distances greater than 0.4 are retained as valid loop pairs. In the sequence pairing strategy, the preceding and following 20 frames are used as sequential pairs. We then extract feature points [31] for each image pair and perform feature matching across the selected pairs to generate point correspondences. Finally, we use COLMAP to estimate the camera parameters and complete the sparse reconstruction.

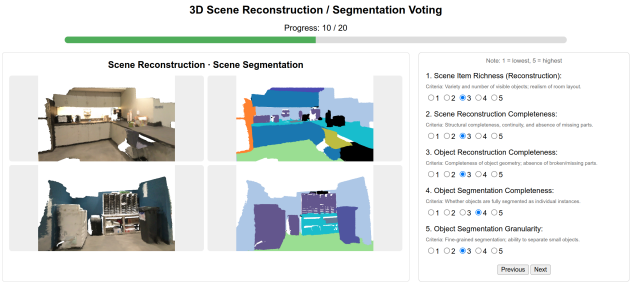
B. Data Quality Check

To assess the quality of data produced by our automated data engine, we perform a human evaluation on the reconstruction and instance segmentation. More specifically, we sample 10 scenes from SceneVerse++ and ScanNet, respectively, visualize their reconstruction and segmentation results side-by-side, and ask human subjects to rate each scene on a scale of 1 to 5, along the following 5 axes:

[1] http://youtube.com/
[2] http://bilibili.com/



(a) An example from SceneVerse++.



(b) An example from ScanNet.

Figure S.1. **Example of quality check.** The data samples from SceneVerse++ and ScanNet are mixed and anonymous.

- **Scene Item Richness**: diversity of abundance of visible items, and how well they reflect realistic indoor layouts.
- **Scene Reconstruction Completeness**: structural completeness of the reconstructed scene, including coherence and absence of major holes or missing regions.
- **Object Reconstruction Completeness**: integrity of individual object shapes, with no breaks, missing faces, or lost components.
- **Object Segmentation Completeness**: whether each object is segmented as a single, coherent instance without obvious omissions or incorrect splits.
- **Object Segmentation Granularity**: the fineness of segmentation, segmenting small objects accurately and avoiding unintended merging.

The results are in Tab. S.1. From the table, SceneVerse++ achieves quality comparable to or exceeding ScanNet across the above evaluation criteria, especially in the richness and completeness of reconstruction, which shows that our dataset captures diverse and real-world distributions. It also indicates that modern image-based reconstruction and segmentation methods, if properly adapted, have advanced to a point where they can surpass the sensor quality and reconstruction pipeline used in ScanNet capture in 2017. This highlights their potential for further scaling. The grading interface is shown in Fig. S.1.

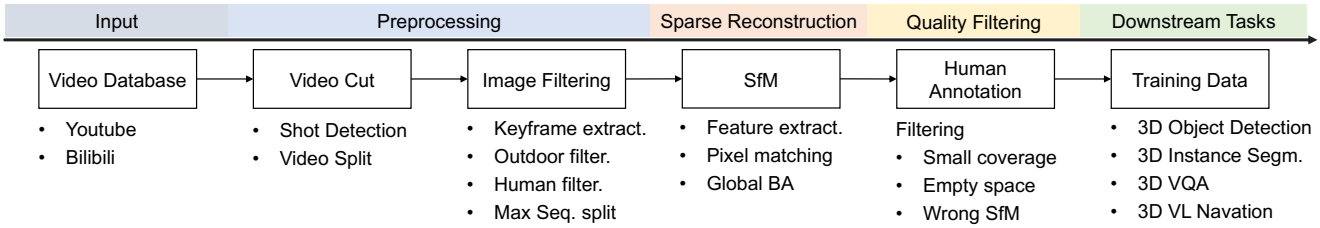


Figure S.2. **Overview of data curation pipeline**.

Table S.1. **The quality check of SceneVerse++ and ScanNet.**

Criterion	SceneVerse++	ScanNet
Scene Item Richness	4.43	3.68
Scene Reconstruction Completeness	4.25	3.09
Object Reconstruction Completeness	4.16	3.23
Object Segmentation Completeness	3.93	3.26
Object Segmentation Granularity	3.89	3.24
Average	4.13	3.30

Table S.2. **Evaluation sensitivity on different segment settings on 3D instance segmentation.** The model is trained with kThresh=10^{-2} and segMinVerts=20, and performance degrades if the distribution of the testing segments diverges from training.

kThresh	segMinVerts	AP_{25}	AP_{50}	AP
10^{-2}	20	36.1	31.8	22.8
10^{-1}	20	34.6	28.1	18.4
10^{-3}	20	35.9	30.4	21.3
10^{-2}	100	30.8	24.6	15.8
10^{-2}	500	17.9	12.6	7.2
10^{-1}	1000	11.2	7.7	4.2
10^{-3}	1000	10.9	7.5	4.1

C. Experiment Details

C.1. 3D Object Detection and Segmentation

3D Object Detection To better handle the large scenes in SceneVerse++, we adopt an additional spatial cropping augmentation during SpatialLM training. For each sample, one object is randomly selected, and the point cloud within a 3-meter radius of the object is extracted and used as the model input. SpatialLM is trained on SceneVerse++ using 8 NVIDIA A100 GPUs for 1000 epochs with a batch size of 1, requiring approximately 2 days. The model is then fine-tuned on ScanNet for another 1000 epochs with a batch size of 4, which takes about 12 hours. For supervision, we utilize 15 semantic categories selected from the ScanNet 20 labels.

3D Instance Segmentation In 3D instance segmentation experiments, we observe that the model trained on SceneVerse++ does not transfer well to ScanNet. One important reason is that Mask3D relies on the segment-level masks produced by a graph-based segmentation method, and different hyperparameters lead to noticeably different segment results. Two decisive hyperparameters, segmentation threshold (kThresh) and minimum segment size (segMinVerts), directly control the connectivity and granularity of segments. To illustrate this sensitivity, we provide further experiments by evaluating a model trained on ScanNet (with kThresh=10^{-2} and segMinVerts=20), on segments generated from different hyperparameter settings. As shown in Tab. S.2 and Fig. S.3, coarse segments fail to correctly isolate individual instances, while overly fine segments result in fragmented predictions and miss detections. This issue is more obvious during the training stage, where the mismatched segment distribution causes poor model transfer. These observations highlight a broader challenge in scaling 3D scene understanding: models sensitive to task-specific modalities and data distribution shifts exhibit limited scalability, whereas models operating directly on raw and widely available modalities may scale more robustly.

C.2. 3D Spatial VQA

Data Generation From the 3D reconstruction and instance segmentation results, we first construct the overall per-scene information, *i.e.*, the room size. Next, we automatically construct 3D scene graphs from point clouds. We first instantiate the graph nodes with the instance annotation from the point cloud and parameterize each node with the object centroid and size of the axis-aligned bounding box. Next, we traverse all the nodes to determine their spatial relationships, following Jia et al. [50]. We then save the counts for different object categories and generate the QAs accordingly.

- Object Count (Numerical Answers (NA)): Count the number of instances of a specified object category that has more than 1 instance within a room.
- Relative Distance (Multiple-Choice Answers (MCA)): Identify which of four candidate objects is closest in 3D space to a target object, which can be uniquely identified by its category.
- Relative Direction (MCA): Given a situation describing the observer's position and orientation, determine the rel-

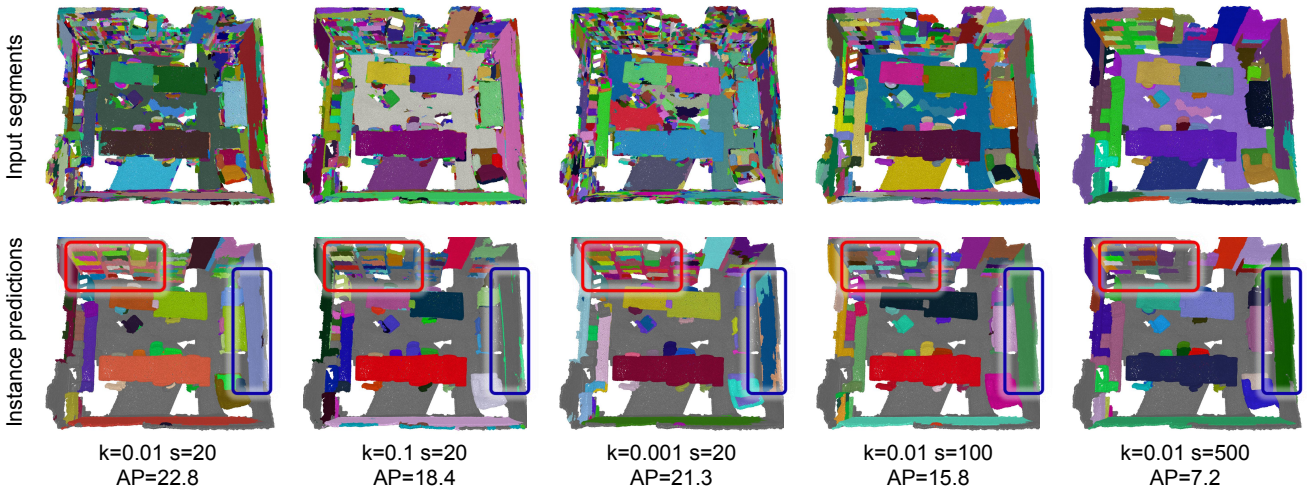


Figure S.3. **Qualitative example of sensitivity tests on different segment distributions.** We evaluate the model trained with k=0.01 and s=20 on other segment settings. The first row shows the input segments, and the second row shows the 3D instance prediction. As the segments become smaller, over-segmentation gradually appears (highlighted by the blue boxes). Conversely, as the segments become larger, under-segmentation becomes increasingly evident (see the red boxes). The AP is reported as the average over the whole ScanNet.

ative direction of a query object, which can be uniquely identified by its category.

- Object Size (NA): Estimate the length of the longest dimension of an object instance in centimeters.
- Absolute Distance (NA): Estimate the Euclidean distance between the closest points of two specified objects in meters. The two objects are randomly selected from the categories that have only one instance.
- Room Size (NA): Estimate the area of the room in square meters (numerical answer).
- Route Planning (MCA): We generate QA pairs by employing the navigation trajectories within 3D environments in the VLN task. The actions are masked to create multiple-choice, and the navigation summary is transferred to guidance via VLM. Detailed prompts are in Tabs. S.7 and S.8.

Dataset Statistics Applying the generation pipeline to SceneVerse++ yields 632K spatial VQA data following the VSI-Bench format. It comprises 391K samples for MCA and 241K samples for NA with 7 different question types. The number of each type of question is listed in Tab. S.3. In our experiment, we sampled a subset of 202K for training.

Training Configuration All experiments for 3D VQA fine-tuning were conducted using LoRA-based adaptation on an LLM backbone, with training performed on 4 × NVIDIA A100 GPUs. More Training Configuration and Reproducibility Details are provided in Tab. S.4.

C.3. 3D Vision-Lanugage Navigation (VLN)

Depth Scale Calibration We design a three-stage pipeline to convert room-tour camera trajectories to VLN trajectories in Sec. 4.3. In action encoding stage, we apply a scale calibration procedure during the action-encoding stage to ensure that movement distances computed from trajectories reflect real-world scale. This is necessary because the SfM reconstruction provides depth on an arbitrary scale, whereas VLN models require physically meaningful forward-motion distances. To estimate the correct scale, we identify video frames containing large and visually stable furniture (e.g., sofas, cabinets, refrigerators), whose depths are easier to estimate reliably. For each selected region, we obtain a robust monocular depth estimate using DepthPro [10]. In parallel, we extract the corresponding absolute (but unscaled) depth from the SfM reconstruction. By comparing these two depth values, we compute a depth-scale factor for each furniture instance. The scale factors are averaged across all selected samples to produce a stable calibration value, which is then applied to the entire reconstructed scene. Accurate depth calibration ensures that

Table S.3. **3D Spatial VQA Data Distribution.**

Task Type	Count
Object Relative Direction	155,199
Object Absolute Distance	137,397
Object Relative Distance	226,639
Object Size Estimate	44,050
Object Count	53,200
Route Plan	9,588
Room Size	6,684
Total	**632,757**

Table S.4. **Training Details for 3D Spatial VQA.**

Category	Setting
Hardware	4 × NVIDIA A100 GPUs
Precision	BF16
LoRA Rank	128
LoRA Scaling Factor	256
Per-device Batch Size	1
Gradient Accumulation Steps	32
Effective Batch Size	$4 \times 32 = 128$
Optimizer	AdamW
Learning Rate	2×10^{-5}
Weight Decay	0
Warmup ratio	0.03
LR Schedule	cosine
Epochs	5
Actual Training	Stop after 1 epoch
Random Seed	42

forward-motion distances derived from trajectories correspond to realistic navigation steps, improving the reliability of action encoding for VLN training. The prompt used for instruction generation is provided in Tab. S.6.

Training Configuration We train LLaVA-Video on 8 NVIDIA A100 GPUs. Zero-shot and mixed-training experiments are run for 1 epoch, while the pretrain–finetune setting uses 2 epochs of pretraining on SceneVerse++ and 1 epoch of fine-tuning on R2R. To ensure balanced exposure to actions across datasets, we apply label rebalancing: we count the occurrences of each action category across all episodes and select a reference frequency based on the median or maximum count. Actions below the reference are oversampled, and actions above the reference are subsampled. Finally, we use the total number of R2R training samples as the baseline and adjust other datasets accordingly to maintain comparable sample counts. Each epoch of training takes approximately 14 hours with a batch size of 2.

Comparison with Internet-Scale VLN Data To validate the effectiveness of our SceneVerse++, we compare it with the YouTube-derived VLN data from NaVILA [24], which contains roughly 20k trajectories, using Qwen-VL-7B [7] as the base model. We evaluate two settings, zero-shot and mixed-training with R2R, and report results in Tab. S.5. In the zero-shot setting, SceneVerse++ and NaVILA show similar performance (SR = 0.09). SceneVerse++ exhibits a larger path length (PL = 11.274), reflecting the inherently longer trajectories present in our data generation pipeline. In the mixed-training setting, SceneVerse++ yields clear improvements over NaVILA on key navigation metrics: higher Success Rate (0.32 vs. 0.29), higher SPL (0.258 vs. 0.213), and lower Distance-to-Goal (7.447 vs. 7.960). This indicates that SceneVerse++ provides more effective supervision for learning grounded navigation when combined with R2R.

Table S.5. **Comparison between NaVILA and SV++.** Experiments use Qwen2.5-VL-7B under zero-shot and mixed-training.

Data Source	Setting	SR↑	OS↑	SPL↑	Dist↓	PL
NaVILA	Zero-shot	**0.09**	0.132	**0.08**	**9.406**	8.505
SV++	Zero-shot	**0.09**	**0.145**	0.063	9.439	11.274
R2R + NaVILA	Mix	0.29	**0.424**	0.213	7.960	16.013
R2R + SV++	Mix	**0.32**	0.402	**0.258**	**7.447**	12.918

Notice that NaVILA contains roughly 2.5× more data than SceneVerse++, which may bias certain metrics in its favor. Despite this scale advantage, SceneVerse++ still achieves superior SR and SPL, suggesting that well-structured, navigation-aligned trajectories are more beneficial than raw data volume alone. These findings support the value of our data-generation pipeline while also underscoring the need to further explore domain differences and dataset scaling in future work.

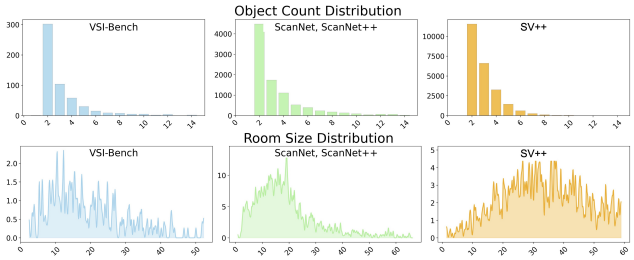


Figure S.4. **3D spatial VQA answer distribution.**

D. More Discussion

Why "Object Count" and "Room Size" performance drop in 3D spatial VQA? We believe the data distribution bias is the major factor here. Several pieces of evidence: 1) SceneVerse++ and ScanNet/ScanNet++ GT perform similarly on zero-shot experiment in Tab. 3; 2) From Fig. S.4, Object Count test distribution in VSI-Bench is highly biased at "2", where in-domain data (ScanNet / ScanNet++) has a much smaller divergence to this peak, showing potential benchmark overfitting:

$$D_{KL}^{obj\_cnt}(\text{VSI-Bench} \parallel \text{SceneVerse++}) = 1.04$$
$$D_{KL}^{obj\_cnt}(\text{VSI-Bench} \parallel \text{SN,SN++}) = 0.145.$$

Room size shows a larger domain gap:

$$D_{KL}^{room\_size}(\text{VSI-Bench} \parallel \text{SceneVerse++}) = 6.08$$
$$D_{KL}^{room\_size}(\text{VSI-Bench} \parallel \text{SN,SN++}) = 2.95,$$

where SceneVerse++ signatures multi-room scenes.

Data scaling analysis We provide scaling results for 3D detection and 3D VQA in Fig. S.5, where data $\sim \mathcal{O}(N_{\text{scenes}})$. Performance follows a *log-linear trend* in both cases, but VQA reaches saturation later. More effective scaling requires co-design involving model architecture, fair benchmarks, and data quality.

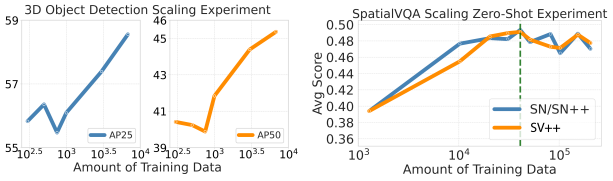


Figure S.5. **Data scaling effects.**

Per-scene computation overhead The average end-to-end per-scene running time is ∼0.59h, consisting of 0.27 GPU-hours (RTX 3090-level) and 0.32h CPU-hour (Xeon 14 vCPUs). Stage-wise, preprocessing and SfM take 69.8%, depth and 2D segmentation model inference takes 23.2%, dense 3D reconstruction 3% and 3D segmentation 4%. This overhead is manageable for large-scale data generation and could be further optimized.

E. Limitations and Future Work

Limited by computational resources, our experiments are bound to the minimal setting to examine the contribution of different data sources. In practice, 3D understanding capability also depends on the base model capacity, optimization strategy, and data mixture, *e.g.*, existing VLN systems often benefit from larger training corpora. Additionally, internet videos may contain privacy-sensitive content from public areas. Scaling such data requires careful adherence to ethical guidelines, regulatory frameworks, and responsible development. Future work includes iterative refinement of the generated data, integration with more advanced models to further enhance capability, and extending to dynamic videos that capture the 4D scene evolution.

Table S.6. **Prompts for Navigation instruction generation in SceneVerse++.**

You are an embodied AI agent making task summaries for a navigation task. Your goal is to generate faithful, human-readable navigation instructions.

— Input —

- A sequence of first-person images and a stepwise action sequence.
- Each image corresponds to the visual observation immediately before the action in that frame.
- Alignment is strictly one-to-one: `image[i]` always pairs with `action[i]`.
- An action entry may describe a single action or a composite action (e.g., "turn left and move forward"), but it still corresponds to the visual state in the paired image.

— Core reasoning rule —

- Always rely primarily on **visual observations** when determining how to move.
- Use actions only as fallback when the image is unclear.
- Maintain consistent spatial logic: if an object is on the left, turning left should bring it to the center view.

— Language and Output Style —

- Avoid first-person narration; use **third-person, objective** instructions such as "A sofa is on the right; turn right to face it."
- Avoid narrative openings (e.g., "The journey begins...").
- Use direct commands: "Turn right into the hallway.", "Walk straight past the sofa.".
- Always include all necessary turning/movement instructions.
- Mention only key orientation-relevant landmarks (sofa, table, doorway, window).

— Responsibilities —

0. Trajectory summarization:
   - Summarize overall motion, room types, and representative objects.
   - Briefly describe the starting location.
   - Provide a concise step-by-step movement description consistent with images.
   - End with a clear final position description.

1. Per-step reasoning:
   - Think in first-person as the agent (camera aligned with orientation).
   - Base reasoning on **visible evidence** in the current frame.
   - Mention only representative, orientation-relevant objects.
   - Use diverse spatial expressions: "to the right", "just ahead", "past the table", etc.
   - Ensure geometric consistency between viewpoint and actions.
   - If actions conflict with geometry, trust the image.

— Action rules —

- Actions may be single or composite (joined by "and").
- Allowed actions: `TurnLeft`, `TurnRight`, `MoveForward`, `Move`, `Stop`.
- "Move" alone means a small forward motion without rotation.
- Composite actions operate sequentially: turn first, then move.

— Special Requirement: Three Reformulations —

Rewrite the trajectory summary into **three distinct linguistic styles** with identical semantic content: Formal Instructional Style, Natural Conversational Style and Narrative Descriptive Style.

Guidelines:

- All three must preserve identical spatial logic and landmarks.
- No conflicts are allowed.
- All must fully cover the entire trajectory.

— Examples of the Three Styles —

- **Instruction 1 (Formal):** "Turn right into the hallway. Advance straight past the dining table. Enter the bedroom and stop in front of the bed."
- **Instruction 2 (Conversational):** "Take a right into the hallway and keep walking until you pass the dining table on your left. Go into the bedroom and stop by the bed."
- **Instruction 3 (Narrative):** "Turning right, you move into the hallway, the dining table sliding by on your left. The hall opens into a bedroom, where you halt just before the bed."

Table S.7. **Prompts to generate route plan VQA in SceneVerse++- part1.**

You are an AI assistant tasked with generating **Fill-in-the-blank Action Completion MCQ** for robot navigation. Your job is to output a multiple-choice question (with blanks) and its correct answer.

— Input —

A sequence of continuous key frames from a room-tour video (the frames are consecutive and represent a smooth camera/robot trajectory).

— High-level rule (priority order) —

1. ALL reasoning must be grounded purely on the **visual evidence from the frames**
2. Use visual cues such as object appearance/disappearance, relative positions, scaling, and viewpoint rotation to infer the robot's movements and turns.
3. When describing places, objects, or targets, use detailed and specific visual anchors — not just generic room names. For example: "the blue sofa on the right," "the black dining table ahead," "the kitchen counter with sink," or "the hallway with a white door at the end."
4. If any step or turn cannot be confidently inferred from visual evidence, skip or merge it rather than guessing. Do NOT fabricate movements.

— Core Procedure (must follow this order) —

1. **Construct a concise, numbered Trajectory Summary**:
   - Carefully analyze the continuous frame sequence to extract a minimal yet complete trajectory.
   - Each entry in the summary should be a single action step, such as: "1. Go forward until [object/room]", "2. Turn left", "3. Go forward until [object/room]".
   - Determine steps by tracking:
     - Appearance/disappearance or scaling of landmarks (for "Go forward")
     - Change in viewing direction or lateral movement (for "Turn left/right/back")
   - The summary should form a coherent navigation sequence from the starting viewpoint to the final destination.
   - Explicitly describe both start place and end place in visually grounded detail: e.g., "You are a robot beginning at the living room, facing the blue sofa." e.g., "You want to navigate to the kitchen with a table on your left."
   - When describing each "Go forward" anchor, be as specific as visually supported:
     - Include object appearance (color, size, material), or scene context (e.g., furniture type or nearby area). Example: "Go forward until the blue sofa."
   - Only include meaningful transitions — skip redundant minor movements or rotations that don't correspond to clear spatial change.
   - Ensure geometric reasoning consistency.
   - Make the trajectory alternate logically between "Go forward" and "Turn".
   - Example of a trajectory summary:
     - 1. Go forward until the 3-seater sofa (evidence: frames X–Y)
     - 2. Turn right (evidence: frames X–Y)
     - 3. Go forward until the dining table (evidence: frames X–Y)
     - 4. Turn left (evidence: frames X–Y)
     - 5. Go forward until the kitchen counter with sink (evidence: frames X–Y)
2. **QA Generation (based on Trajectory Summary only)**:
   - Normalize steps to alternate between "Go forward…" and "Turn …".
   - Determine where to place `[please fill in]` blanks:
     - Every turn step must become a blank.
   - "Go forward" must mention detailed visible landmarks.
   - Use the strict template:

```
Q: You are a robot beginning at [start place, with visual details].
You want to navigate to [end place, with visual details].
You will perform the following actions:
1. Go forward until [object/room]
2. [please fill in]
3. Go forward until [object/room]
...
N. Go forward until [object/room].
You have reached the final destination.
(Note: for each [please fill in], choose either
'turn back,' 'turn left,' or 'turn right.')
```

Table S.8. **Prompts to generate route plan VQA in SceneVerse++- part2**

3. Options generation:
- For each blank, permissible options:
  - 'turn back', 'turn left', 'turn right'
- If one blank: produce A–C.
- If >= 2 blanks: produce A–D.
- Each option is a full sequence of turns.

4. Correct answer determination:
- Must match turns inferred from visual trajectory summary.
- No guessing ambiguous turns.

— Output format (STRICT JSON SCHEMA — all keys required) —
- "trajectory summary": a list of strings.
- "question": full question string.
- "options": dictionary with keys A, B, C, D.
- "answer": one correct option.

— Strict behaviour notes (must obey) —
- Only use frame-based evidence.
- Build a reliable Trajectory Summary.
- The first step may be a Turn or a Go-forward action.
- Never guess ambiguous turns.