
Learning from Videos for 3D World: Enhancing MLLMs with 3D Vision Geometry Priors

Duo Zheng*, Shijia Huang*, Yanyang Li, Liwei Wang[#]

The Chinese University of Hong Kong

<https://lavi-lab.github.io/VG-LLM>

Abstract

Previous research has investigated the application of Multimodal Large Language Models (MLLMs) in understanding 3D scenes by interpreting them as videos. These approaches generally depend on comprehensive 3D data inputs, such as point clouds or reconstructed Bird’s-Eye View (BEV) maps. In our research, we advance this field by enhancing the capability of MLLMs to understand and reason in 3D spaces directly from video data, without the need for additional 3D input. We propose a novel and efficient method called the Video-3D Geometry Large Language Model (VG LLM). Our approach utilizes a 3D visual geometry encoder to extract 3D prior information from video sequences. This information is then integrated with visual tokens and input into the MLLM. Extensive experiments have shown that our method has achieved substantial improvements in various tasks related to 3D scene understanding and spatial reasoning, all directly learned from video sources. Impressively, our 4B model, which does not rely on explicit 3D data inputs, achieves competitive results compared to existing state-of-the-art methods, and even surpasses the Gemini-1.5-Pro in the VSI-Bench evaluations.

1 Introduction

Rapid advancements and impressive performance of Multimodal Large Language Models (MLLMs) [2, 24, 26, 29, 32, 39, 40, 43] have driven their applications in various fields, such as 3D scene understanding [11, 20, 21, 22, 35, 58, 63], vision-language-action models [3, 25, 36, 47], embodied navigation [55, 59, 61], and learning 3D knowledge from videos [35, 55, 58].

Efforts [35, 41, 58] have been made to improve the 3D spatial understanding capability of MLLMs by considering scenes as video sequences. For example, Video-3D LLM [41, 58] injects 3D coordinates into visual features at patch level to improve 3D perception. GPT4Scene [35] leverages BEV maps [28] rendered from reconstructed 3D point clouds for global awareness. However, a shared limitation of these approaches is their dependence on dense 3D data input (*e.g.*, depth maps and point maps), which are often hard to acquire in certain real-world scenarios. Although estimating 3D attributes directly from images is possible [38, 42], it can introduce estimation errors and therefore degrade the performance, thus restricting their practical applicability. This naturally leads to the question: “*Can MLLMs understand the 3D world directly from videos without any explicit 3D data input?*”

Recent research [50] has shown that MLLMs face difficulties in understanding 3D geometry from encoded visual representations. This issue arises because these MLLMs process video frames as separate tokens through a visual encoder, which fails to capture crucial 3D geometric information, such as correspondences across frames [30]. Consequently, the MLLM backbone has to infer the

*denotes equal contribution

[#]Corresponding Author

3D structure from the visual tokens to comprehend spatial relationships, which is both challenging and resource-intensive. This process often requires extensive supervision [6, 14, 54] and meticulous design to prevent issues such as catastrophic forgetting during fine-tuning [52]. These challenges highlight the critical need for methods that can incorporate 3D geometry priors into MLLMs.

In this work, we propose **Video-3D Geometry LLM (VG LLM)**, a novel framework designed to explicitly integrate 3D visual geometry priors into MLLMs. To achieve this, we introduce a 3D visual geometry encoder that enriches input visual sequences with additional geometric information. Specifically, input images are processed by both a conventional visual encoder and the newly integrated 3D visual geometry encoder. The features extracted by these encoders are fused at the patch level and subsequently passed to the MLLM backbone. As the 3D visual geometry encoder is pre-trained on tasks such as point map prediction on pairs or sequences of images [38, 42, 44], it embeds strong 3D perception prior knowledge and is able to capture correspondences across frames. By doing so, VG LLM can effectively incorporate 3D geometry priors into the model and become more robust to viewpoint transformations, significantly improving its spatial reasoning abilities.

Extensive experiments have been conducted on various 3D scene understanding and spatial reasoning tasks, where the model accepts video input. These 3D scene understanding tasks include 3D visual grounding [7], 3D dense captioning [8], and 3D video object detection [45]. For spatial reasoning tasks, we evaluate our model on VSI-Bench [50] and CV-Bench [40]. The experiments show that our fine-tuned model outperforms larger spatial-enhanced models by a substantial margin. The results uncover several interesting findings: (1) Without explicit dense 3D inputs, our approach outperforms many leading 3D input-based models, underscoring its effective 3D geometric understanding. (2) By implicitly modeling inter-frame correspondences within the visual representation, our 8B model has learned strong egocentric-alloentric transformation capabilities, leading to significant improvements of precision by **11.9%** and F1 by **10.7%** on 3D video object detection. (3) On tasks that require complex spatial reasoning skills, *i.e.*, VSI-Bench [50], our 4B model attains an impressive average score of **47.3%**, surpassing even the best proprietary model, Gemini-1.5-Pro [39]. Furthermore, our 8B model sets a new state-of-the-art performance with a score of **50.7%**. This highlights the great utility of 3D geometry modeling in broader scenarios.

2 Related Work

Multimodal Large Language Models (MLLMs). MLLMs [2, 24, 26, 29, 32, 39, 40, 43] have achieved significant progress in 2D image and video understanding. However, recent findings [31, 50] indicate a critical limitation: current MLLMs still struggle with complex visual spatial reasoning tasks. To address this challenge, some research efforts [2, 40, 43] have focused on enhancing the models’ ability to perceive and process spatial relationships through improved model representation. For example, Cambrian-1 [40] integrates self-supervised 2D visual representations with semantically rich features, aiming to provide a more comprehensive understanding of visual content that could benefit spatial reasoning. These methods focus on enriching the representation of each individual image, neglecting the 3D geometry information inherent in the continuous frames of a video. In contrast, our approach integrates 3D geometry priors from the video into the MLLM.

3D Large Language Models. Recent efforts [11, 13, 18, 20, 35, 58] have focused on enabling MLLMs to better understand 3D scenes. Previous work develops comprehensive 3D scene representations by using different types of 3D data. These include point cloud features [11, 13, 49], lifted multi-view image features [18, 20, 63], treating multi-view images as video sequences [35, 58]. The most closely related approaches are Video-3D LLM [58], which integrates positional information into the visual representation, and GPT4Scene [35], which constructs BEV maps through 3D reconstruction. In contrast to these methods, our model does not require any explicit dense 3D inputs.

Spatial Reasoning. Some work [5, 6, 14] has enhanced spatial understanding through large-scale synthesized VQA datasets for improved depth estimation. However, these methods primarily focus on static images, overlooking the dynamic and relational aspects inherent in complex spatial reasoning scenarios. To address this limitation, the Visual-Spatial Intelligence Benchmark (VSI-Bench) [50] was introduced, explicitly evaluating relational reasoning and egocentric-alloentric transformation abilities. Recognizing the scarcity of data for such complex scenarios, subsequent works like SAT [37] and SPAR [54] have proposed data generation pipelines to synthesize spatial QA datasets for supervised fine-tuning. In contrast, our model elicits 3D geometry information from the video and injects it into the model architecture, improving its spatial reasoning capabilities.

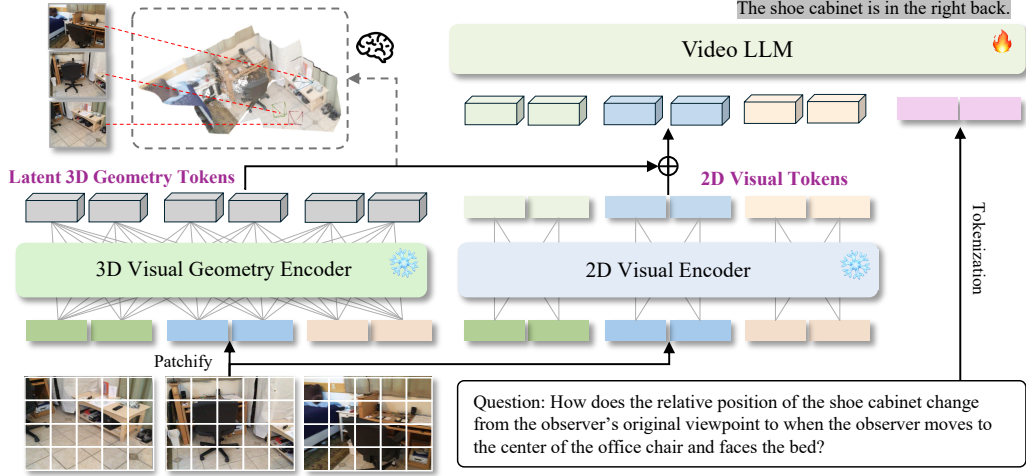


Figure 1: **The architecture of our VG LLM.** The 3D visual geometry encoder processes a sequence of images to produce globally geometry-aware visual features, while the 2D visual encoder extracts semantic-aware visual features from each individual image. 🧠 vividly shows that the latent 3D geometry tokens are able to recover the 3D scene if with a dense prediction head [42].

3 Method

We aim to model the 3D visual geometry in MLLMs without relying on explicit dense 3D input. To achieve this, as illustrated in Figure 1, we introduce a 3D visual geometry encoder to extract rich visual-geometric features from multiple images and feed them into the MLLM backbone. Section 3.1 outlines the architecture design of our model, and Section 3.2 details the training process.

3.1 Architecture

Preliminary. Given a sequence of RGB images $\{I_i\}_{i=1}^n$ and a natural language question Q , a conventional MLLM utilizes a 2D visual encoder to encode these images into image tokens $T_i^V \in \mathbb{R}^{\lfloor \frac{h}{p} \rfloor \times \lfloor \frac{w}{p} \rfloor \times c}$ first, where $I_i \in \mathbb{R}^{h \times w \times 3}$ and p is the patch size. Then an MLLM backbone accepts $\{T_i^V\}_{i=1}^n$ as its input and outputs the response. In this work, we choose Qwen2.5-VL [2] as the MLLM backbone. Note that Qwen2.5-VL additionally compresses the image tokens to reduce the computational cost. Qwen2.5-VL groups spatially adjacent 2×2 patches into a single image token, resulting in a smaller set of image token input to MLLM backbone, $T_i^{V'} \in \mathbb{R}^{\lfloor \frac{h}{2p} \rfloor \times \lfloor \frac{w}{2p} \rfloor \times c}$.

3D Visual Geometry Encoder. To model 3D geometric information like inter-frame correspondences within input frames, we employ a 3D visual geometry encoder to extract such information. This 3D visual geometry encoder produces 3D visual geometry features $T_i^G \in \mathbb{R}^{\lfloor \frac{h}{p} \rfloor \times \lfloor \frac{w}{p} \rfloor \times c}$ from all input images $\{I_i\}_{i=1}^n$ jointly. 3D visual geometry models [44, 38, 42] are natural candidates for the 3D visual geometry encoder, as they are trained to capture inter-frame correspondences and reconstruct 3D scenes without relying on additional 3D priors. These 3D visual geometry models comprise three key components: an encoder for per-image feature extraction, a fusion decoder for cross-frame interaction, and task-specific prediction heads for 3D attributes. Since we focus on feature extraction, which embeds 3D geometry prior information, rather than directly outputting 3D attributes, we leverage the encoder and the fusion decoder as our 3D visual geometry encoder. Specifically, we choose VGGT [42] to extract 3D visual geometry features given its superior performance in 3D tasks.

Visual Feature Fusion. We fuse both the image tokens $\{T_i^{V'}\}_{i=1}^n$ and the 3D visual geometry features $\{T_i^G\}_{i=1}^n$ before passing them into the MLLM backbone. We first transform each T_i^G into $T_i^{G'} \in \mathbb{R}^{\lfloor \frac{h}{2p} \rfloor \times \lfloor \frac{w}{2p} \rfloor \times c}$, which has the identical shape of $T_i^{V'}$, and then generate the geometry-augmented visual features $T_i^S = T_i^{G'} + T_i^{V'}$. The transformation of T_i^G aligns with the spatial merging strategy in Qwen2.5-VL, where we concatenate spatially adjacent 2×2 features in T_i^G and pass them to a two-layer MLP to output a single feature in $T_i^{G'}$.

To this end, the final visual features $\{T_i^S\}_{i=1}^n$ are concatenated with the text embeddings of the question Q into the MLLM backbone to produce the response.

3.2 Training

Our VG LLM offers a versatile framework designed to integrate 3D vision priors into MLLMs for various 3D tasks. In this work, we demonstrate the application of VG LLM to 3D scene understanding and spatial reasoning tasks.

3.2.1 Applying VG LLM to 3D Scene Understanding

To validate the efficacy of the model in understanding and reasoning about 3D scenes, we adapt it to several 3D scene understanding tasks, *i.e.*, 3D visual grounding, 3D dense captioning, and 3D video object detection. In contrast to previous work on understanding 3D scenes, our model relies solely on RGB images as input during both training and inference, without any 3D scene data as prerequisites. For 3D dense captioning, while we utilize pre-detected 3D proposals as the input, our model itself operates solely on RGB images. We directly learn all 3D scene understanding tasks through text generation, which obviates the need for task-specific heads while maintaining a simple next-token prediction objective during training. The unified text generation objective allows us to mix all tasks into a combined dataset for multi-task training. The detailed data format can be found in the appendix.

Coordinate System and Representation. Since we do not utilize any ground truth information of the 3D scene, we follow VGGT [42] to employ the coordinate system of the first frame as the base coordinate system. All coordinates are transformed into this base coordinate system (except for 3D visual grounding, which represents the bounding box in each frame’s coordinate system). All the numbers are expressed in plain text with 2 decimal places.

3D Visual Grounding. We follow previous work [54] to formulate the 3D visual grounding task as a 3D video grounding problem, which involves locating the frame index where the object appears and its 3D bounding box in the corresponding frame’s coordinates in one feed-forward pass. Unlike SPAR [54], which generates axis-aligned boxes, our method directly predicts 3D-oriented bounding boxes, making it more suitable for real-world scenarios. Specifically, given the 3D scene represented by frames $\{I_1, I_2, \dots, I_n\}$, and a natural language query Q , our model is designed to output both the frame index and the bounding box in the form $(x, y, z, w, h, d, \psi, \theta, \phi)$, where (x, y, z) is the center coordinate, (w, h, d) is the object size, and (ψ, θ, ϕ) is the rotation angles.

3D Dense Captioning. We follow the recent work [58, 63] that decomposes the 3D dense captioning task into two phases, *i.e.*, detecting 3D object proposals with an off-the-shelf detector and generating object descriptions based on object coordinates. Specifically, given a frame sequence $\{I_1, I_2, \dots, I_n\}$, we prompt our model to “describe the object located at (x, y, z) in detail”, where (x, y, z) is the box center of the object to be described. This setup ensures a fair comparison with prior work and effectively evaluates the model’s ability to understand positional and spatial relationships.

3D Video Object Detection. To investigate the capability of handling egocentric-alloentric transformation, we set up a 3D video object detection task based on the ScanNet [15]. Unlike previous benchmarks that focus on monocular or multi-view detection [45], our task requires models to detect all objects throughout the video in a unified coordinate system, without relying on explicit camera parameters or depth information. Since some objects aren’t visible in the first frame, the model must track changes between frames, estimate camera movement, and convert object locations from a global view to the camera’s perspective. Specifically, given continuous video frames $\{I_1, I_2, \dots, I_n\}$, the model is tasked to detect all objects $\{(b_1, c_1), \dots, (b_m, c_m)\}$ that appear in this video, where each b_i in the form $(x, y, z, w, h, d, \psi, \theta, \phi)$ represents its bounding box in the unified coordinate system and c_i is its category.

3.2.2 Instruction Tuning for Enhanced Spatial Reasoning.

Previous work [53, 62] has highlighted that existing pre-training and instruction tuning datasets often lack sufficient spatial-related phrases in their annotations, consequently hindering the spatial reasoning capabilities of MLLMs. To overcome this limitation, we leverage the SPAR-7M [54] dataset for instruction tuning. SPAR-7M is a comprehensive spatial reasoning dataset curated from three

richly annotated 3D datasets: ScanNet [15], ScanNet++ [51], and Structure3D [60]. It encompasses 33 diverse tasks, ranging from fundamental perception to mid-level viewpoint transformation and high-level scene imagination. As some work has shown that post-training on specific datasets can sometimes degrade the original performance on general benchmarks, we also incorporate a visual instruction tuning dataset, the LLaVA-Video-178K’s LLaVA-Hound split [57], into our training pipeline to preserve the generalization capability.

4 Experiments

In this section, we first present the implementation details of our model, followed by an evaluation of our model’s performance on 3D scene understanding tasks in Section 4.1, including 3D visual grounding, 3D dense captioning, and 3D multi-view detection. Next, we provide a comprehensive comparison with state-of-the-art methods on spatial reasoning benchmarks and generic multimodal benchmarks in Section 4.2. 3D scene understanding emphasizes 3D perception, while spatial reasoning focuses on interpreting and reasoning about spatial relationships in video. We train two models for 3D scene understanding and spatial reasoning tasks separately for fair comparison. Finally, we conduct a comprehensive analysis in Section 4.3 to reveal the effectiveness of our model’s core aspects: the integration of 3D geometry, the feature fusion strategy, and the data composition.

Implementation and Training Details. Our models are built upon two sizes of Qwen2.5-VL—3B and 7B [2], and integrated with VGGT-1B [42] as the 3D geometry encoder. We trained our models for one epoch on a mixed dataset, detailed as followed in this section, and employed Adam with a batch size of 64 and a warmup ratio of 0.03. During the warmup phase, the learning rate was gradually increased to $1e-5$ before linearly decaying to 0. In each training step, a batch was randomly sampled from a single source from the mixed dataset. During training, the MLLM’s visual encoder, the integrated 3D geometry encoder, and the multimodal connector are frozen, while the MLLM backbone remains unfrozen. All experiments were conducted on 8 H100 80G GPUs. For the 4B and 8B models, the training took 9 and 12 hours for 3D scene understanding, and 7 and 9 hours for spatial reasoning, respectively.

4.1 3D Scene Understanding Tasks

4.1.1 Setting

Datasets and Benchmarks. To evaluate the versatility of our model across diverse 3D scene understanding tasks, we employ a multi-task learning approach on a combination of datasets.

- *3D Visual Grounding.* We leverage the ScanRefer [7] dataset, which provides 36,665 object descriptions paired with axis-aligned bounding boxes across 562 indoor scans. We follow SPAR [54] to reformulate 3D visual grounding as *3D spatial-temporal video grounding*, aiming to locate the target object’s bounding box in camera coordinates along with its corresponding frame index. To determine the relevant appearance frame, we utilize the visible object annotations from EmbodiedScan [45]. We match the target 3D bounding box with those in EmbodiedScan based on their IoU and subsequently pick the optimal frame by comparing the 2D projection areas.
- *3D Dense Captioning.* We utilize the Scan2Cap benchmark [8] for 3D dense captioning, which requires generating descriptive captions for all objects within a scene. Following prior work [22, 58, 63], we use Mask3D-detected object proposals extracted by LEO [22] and task our model with generating captions conditioned on their center coordinates. To better leverage the visual geometry, we transform all object center coordinates to the coordinates of the first captured frame.
- *3D Video Object Detection.* For 3D video object detection, we curated a dataset from EmbodiedScan [45] consisting of consecutive frames and their corresponding visible object annotations in indoor scenes. Each sample comprises four consecutive frames sampled at 1 FPS, with all associated object instances transformed to the coordinate system of the initial frame. Consistent with EmbodiedScan, we divide the data into 958 training and 243 evaluation scenes. Within each scene, we randomly select 150/10 samples for training/evaluation.

Comparison Baselines. For the tasks of 3D visual grounding and dense captioning, our evaluation includes comparisons with both task-specific expert models and general-purpose 3D models. Specifically, for expert models, we compare our method against ScanRefer [7], MVT [23], and ViL3DRel [10] on ScanRefer [7]. On the Scan2Cap dataset [8], we include Scan2Cap [8], 3DJCG [4], D3Net [9], and Vote2Cap-DETR [12] for comparison. Furthermore, we compare our approach with the

generalist models, including Chat-3D v2 [21], Grounded 3D-LLM [13], LL3DA [11], LEO [22], LLaVA-3D [63], and Video-3D LLM [58]. For 3D video object detection, we compare our method with a baseline (Qwen2.5-VL-3B [2]) that does not incorporate 3D geometry information.

Evaluation Metrics. For ScanRefer [7], we report the accuracy at IoU thresholds of 0.25 and 0.5. For Scan2Cap [8], we use CIDEr (C), BLEU-4 (B-4), METEOR (M), and ROUGE (R) scores. The “@0.5” suffix indicates that these metrics are computed only for objects detected with an IoU of 0.5 or higher against the ground truth. Lastly, for 3D video object detection, we report the accuracy, recall, and F1 score for 20 common object classes at an IoU threshold of 0.25.

4.1.2 Results on 3D Visual Grounding

Model	3D Scene Input	Acc@0.25	Acc@0.5
ScanRefer [7]	✓	37.3	24.3
MVT [23]	✓	40.8	33.3
ViL3DRel [10]	✓	47.9	37.7
3D-LLM [20]	✓	30.3	-
Chat-3D v2 [21]	✓	35.9	30.4
Grounded 3D-LLM [13]	✓	47.9	44.1
ChatScene [21]	✓	55.5	50.2
LLaVA-3D [63]	✓	54.1	42.4
Video-3D LLM [58]	✓	58.1	51.7
SPAR [54]	✗	31.9 (48.8)	12.4 (43.1)
VG LLM-4B (Ours)	✗	36.4 (53.5)	11.8 (47.5)
VG LLM-8B (Ours)	✗	41.6 (57.6)	14.9 (50.9)

Table 1: **The quantitative results on ScanRefer.** The content in “()” indicates results with proposal refinement³.

Model	3D Scene Input	C@0.5↑	B-4@0.5↑	M@0.5↑	R@0.5↑
Scan2Cap [8]	✓	39.1	23.3	22.0	44.8
3DJCG [4]	✓	49.5	51.0	24.2	50.8
D3Net [9]	✓	62.6	35.7	25.7	53.9
Vote2Cap-DETR [12]	✓	61.8	34.5	26.2	54.4
LL3DA [11]	✓	65.2	36.8	26.0	55.0
Chat-3D v2 [21]	✓	63.9	31.8	-	-
Grounded 3D-LLM [13]	✓	70.2	35.0	-	-
LEO [22]	✓	72.4	38.2	27.9	58.1
Chat-Scene [21]	✓	77.1	36.5	-	-
LLaVA-3D [63]	✓	79.2	41.1	30.2	63.4
Video-3D LLM [58]	✓	80.0	40.2	28.5	61.7
VG LLM-4B (Ours)	✗	78.6	40.9	28.6	62.4
VG LLM-8B (Ours)	✗	80.0	41.5	28.9	62.6

Table 2: **The performance on Scan2Cap.** VG LLM captions 3D object proposals using RGB cues only.

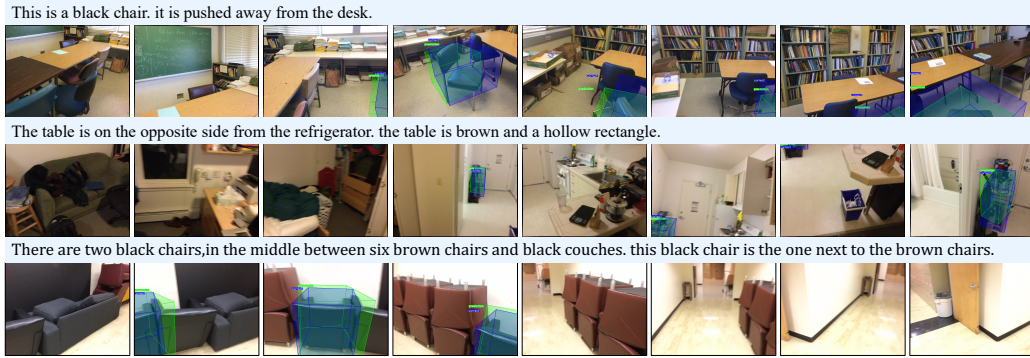


Figure 2: **Qualitative results for 3D visual grounding.** The ground truth and prediction are masked in blue and green, respectively. The predicted boxes are directly generated by our model without the refinement process.

In addition to the directly predicted bounding boxes, we also follow SPAR [54] to calculate the refined results with a proposal refinement process³.

VG LLM demonstrates strong 3D grounding capabilities from monocular RGB videos. The performance on ScanRefer is shown in Table 1. Our 8B model achieves an accuracy of 41.6% at an IoU threshold of 0.25, significantly outperforming SPAR’s 31.9% by 9.7 percentage points. Moreover, with the integration of a proposal refinement technique, Acc@0.25 increases substantially to 57.6%, making our results highly competitive with state-of-the-art methods like Video-3D LLM. This demonstrates that 3D visual grounding can be effectively approached in a video grounding manner.

Qualitative results for 3D visual grounding. Figure 2 illustrates the qualitative results on the 3D visual grounding task. For visualization purposes, the predicted and ground-truth 3D bounding boxes are projected onto their respective 2D image planes. Our method effectively handles spatial

³For proposal refinement, we compare the predicted box against all proposals detected by Mask3D and select the one with the highest IoU.

relationships between objects (e.g., away, opposite, next to), accurately identifying the corresponding frame index and generating the 3D-oriented bounding box.

4.1.3 Results on 3D Dense Captioning

Our model processes 3D object proposals generated by an off-the-shelf 3D object detector. Once these objects are localized, the model generates descriptions using only the RGB images and the center coordinates of the detected objects, without requiring explicit depth or geometric data as additional inputs. This approach achieves a score of 80.0 C@0.5 and 41.5 B-4@0.5, comparable to previous state-of-the-art methods such as LLaVA-3D and Video-3D LLM. These results validate the effectiveness of our model for 3D dense captioning when conditioned on 3D proposals.

4.1.4 Results on 3D Video Object Detection

Model	chair	cabinet	table	bin	couch	bed	bathtub	toilet	20 Common Classes		
									P ₂₅	R ₂₅	F1 ₂₅
4-Frame Setting											
Qwen2.5-VL-3B	37.7	10.2	35.0	23.1	39.0	64.8	32.4	68.8	32.6	27.9	30.0
+ Visual Geometry (VG LLM-4B)	49.7	13.1	41.3	39.2	44.6	71.2	33.5	83.4	41.7	35.7	38.2
$\triangle$ Improvement	+12.0	+2.9	+6.3	+16.1	+5.6	+6.4	+1.1	+14.6	+9.1	+7.8	+8.2
Qwen2.5-VL-7B	41.2	11.6	36.5	30.2	41.1	68.2	36.6	68.7	34.6	31.0	32.5
+ Visual Geometry (VG LLM-8B)	54.0	17.1	46.5	39.8	47.0	74.1	42.1	82.5	43.4	39.6	41.2
$\triangle$ Improvement	+12.8	+5.5	+10.0	+9.7	+5.9	+5.9	+5.5	+13.8	+8.8	+8.6	+8.7
6-Frame Setting											
Qwen2.5-VL-3B	32.8	7.8	31.3	20.9	32.2	58.8	36.5	66.1	27.8	24.1	25.7
+ Visual Geometry (VG LLM-4B)	41.6	12.4	39.8	33.1	45.0	70.2	33.8	80.6	39.7	34.0	36.4
$\triangle$ Improvement	+8.8	+4.6	+8.5	+12.2	+12.8	+11.4	-2.7	+14.5	+11.9	+9.9	+10.7
Qwen2.5-VL-7B	36.1	10.6	32.7	25.0	40.7	64.6	38.4	68.6	31.8	28.0	29.6
+ Visual Geometry (VG LLM-8B)	48.7	17.9	44.8	38.5	46.4	75.8	40.4	83.2	43.5	38.7	40.8
$\triangle$ Improvement	+12.6	+7.3	+12.1	+13.5	+5.7	+11.2	+2.0	+14.6	+11.7	+10.7	+11.2

Table 3: **The results on 3D video object detection.** The reported object categories follow the monocular detection in EmbodiedScan, and we report the average F1 score per category.



Figure 3: **Qualitative results for 3D video object detection.**

As shown in Table 3, we follow the monocular detection of EmbodiedScan [45] to evaluate the performance for categories that are common in daily life.

3D geometry significantly boosts cross-frame detection performance. Qwen2.5-VL [2], upon being fine-tuned on detection data, achieves promising performance on common object categories. Incorporating 3D geometry [42] yields a substantial improvement across all evaluation metrics. In the 4-frame setting, this approach elevates the average F1 score for the 4B model by 8.2% (from

30.0% to 38.2%) and for the 8B model by 8.7% (from 32.5% to 41.2%). This performance gain is attributed to the model’s improved ability to comprehend the egocentric-alloentric transformation between frames through the utilization of 3D geometry. Figure 3 provides qualitative results on the 3D video object detection task.

Integrating 3D geometry enhances the model’s robustness to variations in frame count. Table 3 shows that our method, although trained on 4-frame sequences, maintains strong performance when evaluated on longer sequences during inference. In contrast, the baseline’s performance drops noticeably, underscoring the robustness of our approach to variations in the number of frames.

4.2 Spatial Reasoning Benchmarks

		Obj. Count	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order
Model	Avg.	Numerical Answer				Multiple-Choice Answer			
Proprietary Models (API)									
GPT-4o	34.0	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5
Gemini-1.5-Flash	42.1	49.8	30.8	53.5	54.4	37.7	41.0	31.5	37.8
Gemini-1.5-Pro	45.4	56.2	30.9	64.1	43.6	51.3	46.3	36.0	34.6
Open-source Models									
InternVL2-8B	34.6	23.1	28.7	48.2	39.8	36.7	30.7	29.9	39.6
InternVL2-40B	36.0	34.9	26.9	46.5	31.8	42.1	32.2	34.0	39.6
LongVILA-8B	21.6	29.1	9.1	16.7	0.0	29.6	30.7	32.5	25.5
VILA-1.5-40B	31.2	22.4	24.8	48.7	22.7	40.5	25.7	31.5	32.9
LongVA-7B	29.2	38.0	16.6	38.9	22.2	33.1	43.3	25.4	15.7
LLaVA-NeXT-Video-72B	40.9	48.9	22.8	57.4	35.3	42.4	36.7	35.0	48.6
LLaVA-OneVision-72B	40.2	43.5	23.9	57.6	37.5	42.5	39.9	32.5	44.6
Spatial-Enhanced Models									
SAT-LLaVA-Video-7B	-	-	-	-	47.3	41.1	37.1	36.1	40.4
SPAR-8B	41.1	-	-	-	-	-	-	-	-
VG LLM-4B (Ours)	47.3	66.0	37.8	55.2	59.2	44.6	45.6	33.5	36.4
VG LLM-8B (Ours)	50.7	67.9	37.7	58.6	62.0	46.6	40.7	32.4	59.2

Table 4: The comparison with state-of-the-art models on VSI-Bench. *Spatial-Enhanced Models* are models that are specialized for spatial reasoning.

4.2.1 Setting

Datasets and Benchmarks. To fully leverage the 3D knowledge inherent in the 3D visual geometry encoder, we train our model on a dataset sampled from SPAR-7M [54] and the LLaVA-Hound split of the LLaVA-Video-178K [57]. In our work, we sample only 234K and 63K data points from SPAR-7M and the LLaVA-Hound split of LLaVA-Video-178K, respectively, which constitute 3% and 25% of the original datasets.

We first evaluate our method on a video spatial reasoning benchmark, VSI-Bench [50], which evaluates the egocentric-alloentric transformation and relational reasoning capabilities of MLLMs. Then we test our model on an image spatial-related benchmark *i.e.*, CV-Bench [40]. CV-Bench [40] assesses 2D understanding through spatial relationships and object counting, while its 3D evaluation focuses on depth ordering and relative distance perception.

Comparison Baselines. We compare our model with state-of-the-art proprietary and open-source MLLMs (e.g., GPT-4o, Gemini-1.5-Pro, LLaVA-NeXT-Video) [1, 24, 26, 27, 32, 34, 39, 40, 46, 57]. We also include two spatial-enhanced MLLMs, SAT-LLaVA-Video-7B [37] and SPAR-8B [54], which are fine-tuned on datasets targeted for spatial reasoning.

Evaluation Metrics. For VSI-Bench, we adopt accuracy for multiple-choice tasks and Mean Relative Accuracy (MCA) for numerical tasks. MCA calculates the average accuracy across a range of confidence thresholds, where a prediction is considered correct if its relative error is within a specified threshold. For CV-Bench and other generic multimodal benchmarks, we report accuracy.

4.2.2 Results on Spatial Reasoning Tasks

VG LLM achieves state-of-the-art performance on VSI-Bench. A detailed comparison with leading models is presented in Table 4. As shown in the table, our 4B model achieves an impressive average accuracy of 47.3%, outperforming all competitors, including the leading proprietary model, Gemini-1.5 Pro. Furthermore, our 8B model sets a new state-of-the-art performance with an average accuracy of 50.7%. These results underscore the model’s strong capability in understanding and reasoning from monocular videos.

Model	2D (%)	3D. (%)	Avg. (%)
<i>Proprietary Models (API)</i>			
GPT-4V [34]	64.3	73.8	69.1
GPT-4o [24]	74.8	83.0	78.9
Gemini-1.5-Flash [39]	70.9	71.8	71.4
Gemini-1.5-Pro [39]	77.1	77.6	77.3
<i>Open-source Models</i>			
Mini-Gemini-HD-34B [27]	71.5	79.2	75.4
LLaVA-NeXT-34B [26]	73.0	74.8	73.9
Cambrian-1-34B [40]	74.0	79.7	76.9
SAT-LLaVA-Video-7B [37]	73.0	83.8	78.4
SPAR-8B [54]	72.3	89.1	80.7
VG LLM-4B (Ours)	71.3	87.7	79.5
VG LLM-8B (Ours)	72.2	91.1	81.7

Table 5: **The comparison with state-of-the-art methods on CV-Bench.**

Benchmark	Qwen2.5-VL-3B	VG LLM-4B	Qwen2.5-VL-7B	VG LLM-8B
Video-MME _{w/o} sub. [17]	60.1	57.4	62.9	59.3
Video-MME _{w/} sub. [17]	60.9	59.8	61.1	63.9
BLINK [19]	46.6	48.9	55.9	51.5
TempCompass _{MC}	62.2	63.9	71.8	67.8
NextQA _{MC} [48]	77.3	74.8	81.4	79.3

Table 6: **Comparison of model performance on generic multimodal benchmarks.** To ensure a fair comparison, we evaluate Qwen2.5-VL with the resolution and sampling rate matched to those of VG LLM.

Our model demonstrates generalization across distinct data sources. To access the generalization capability of our method, we evaluate our approach on a spatial reasoning benchmark with different data sources, CV-Bench. This benchmark is constructed by repurposing the traditional CV datasets (*i.e.*, ADE20K, COCO, and Omni3D) to a vision-centric MLLM benchmark. As illustrated in Table 5, our method achieves the highest accuracy on 3D tasks at 91.1%. These results demonstrate our model can also generalize well across the spatial reasoning benchmarks with out-of-the-domain data sources.

4.2.3 Results on Generic Multimodal Benchmarks

Enhancing spatial understanding incurs negligible loss on general multimodal performance.

Our model integrates 3D geometry information and is fine-tuned on spatial reasoning datasets to bolster its spatial understanding capabilities. As presented in Table 6, these enhancements slightly compromise VG LLM’s performance on general multimodal benchmarks when compared to the baseline model. VG LLM-4B even achieves improvements on BLNK (+2.3%) and TempCompass_{MC} (+1.7%). This suggests that augmenting spatial reasoning capabilities can not only preserve but also enhance performance on specific multimodal tasks.

4.3 Analysis

4.3.1 3D Scene Understanding

Type	ScanRefer		Scan2Cap		3D Video Detection		
	Acc@0.25	Acc@0.5	CIDEr@0.5	BLEU-4@0.5	P ₂₅	R ₂₅	F1 ₂₅
No Additional Info. (Baseline)	31.9 (49.9)	9.3 (43.8)	58.0	36.3	32.6	27.9	30.0
Cross-Attn (1 Layer)	33.7 (51.3)	10.7 (45.2)	74.7	40.1	38.0	33.6	35.4
Cross-Attn (3 Layers)	34.4 (51.3)	10.5 (45.2)	75.7	40.2	38.5	44.0	35.4
Concat+MLP	27.7 (47.0)	6.8 (41.3)	75.7	39.9	37.1	32.5	34.4
Add (Ours)	36.4 (53.5)	11.8 (47.5)	78.6	40.9	41.7	35.7	38.2
Pred Camera Info.	32.1 (50.0)	9.9 (43.9)	56.8	36.3	33.3	28.1	30.3
Pred Depth Info.	32.3 (49.7)	9.7 (43.7)	57.1	36.1	32.1	27.3	29.3
Pred Point Info.	31.7 (49.7)	9.6 (43.8)	67.7	38.2	33.0	28.5	30.4
Pred (Depth + Camera) Info.	32.6 (49.8)	9.8 (43.7)	58.2	36.6	31.7	27.1	29.1
Latent 3D Geometry (Ours)	36.4 (53.5)	11.8 (47.5)	78.6	40.9	41.7	35.7	38.2

Table 7: **Ablation study of the effects of 3D visual geometry modeling.** All models are fine-tuned using the same training data and built upon Qwen2.5-VL-3B.

In this section, we conduct a further analysis on the feature fusion strategies and the types of added signals. To investigate the effect of feature fusion strategies, we have experimented with several feature fusion strategies, including 1) “Cross-Attn”, which consists of multiple blocks of cross-attention modules and MLPs with skip connections. 2D visual tokens serve as queries, while 3D visual tokens serve as keys and values. 2D positional embeddings are added to both the queries and keys; 2) “Concat+MLP” first concatenates the 2D and 3D visual tokens along the feature dimension, followed by an MLP to transform them into the text embedding space; 3) “Add” is the strategy employed in our paper, which directly adds the 2D and 3D visual tokens at a patch level.

Besides the visual geometry feature adopted in our model, VGGT can also directly predict the camera poses (*Pred Camera Info.*), the depth maps (*Pred Depth Info.*), and the point maps (*Pred Point Info.*) of each image frame. For these predicted spatial signals, we adopt a two-layer MLP to transform them and add to the corresponding 2D visual token. All models are trained and evaluated following the datasets and benchmarks in Section 4.1.

Directly adding the geometry features obtains the best results among all compared feature fusion strategies. As Table 7 illustrates, the “Cross-Attn” strategy significantly improves performance over the baseline. Moreover, increasing the number of layers to three yields even greater performance gains. While “Concat+MLP” performs worse than the baseline on ScanRefer, it outperforms it on Scan2Cap and 3D video object detection. Nevertheless, “Add” surpasses all other comparison methods despite its simplicity.

Incorporating point maps enhances 3D scene understanding, while camera or depth information offers no clear benefits. After fine-tuning on 3D downstream tasks, the baseline model Qwen2.5-VL shows enhanced 3D scene understanding. However, it continues to struggle with accurately localizing and describing 3D objects, as well as learning spatial transformations across different frames. As illustrated in Table 7, incorporating predicted camera or depth information alone provides no clear benefit. In contrast, the inclusion of predicted point information demonstrates a significant advantage for the 3D dense captioning task, boosting the CIDEr@0.5 score on Scan2Cap from 58.0 to 67.7 and the BLEU-4@0.5 from 36.3 to 38.2. It also provides a marginal improvement in 3D video object detection. These results indicate that current MLLMs still lack adequate spatial modeling in both data curation and architectural design.

Visual geometry features are more helpful than predicted spatial information. Although the point maps predicted directly by VGGT can enhance the spatial understanding ability of Qwen2.5-VL, these signals are relatively sparse and may contain errors. In contrast, directly using the visual geometry features provided by VGGT not only incorporates these spatial signals simultaneously but also reduces the impact of noise on performance, thereby achieving better results.

4.3.2 Spatial Reasoning

#	Data	Model	VSI-Bench								CV-Bench (3D)			
			Obj. Count	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order	Avg	Depth	Distance	Avg
1	∅	Qwen2.5-VL-7B	25.4	10.4	36.4	29.1	38.2	37.9	30.9	27.5	29.5	84.5	76.5	80.5
2	S1	Qwen2.5-VL-7B	68.6	36.0	57.9	59.7	45.8	38.9	30.4	60.2	49.8	87.0	87.2	87.1
3		VG LLM-8B	67.9	37.7	58.6	62.0	46.6	40.7	32.4	59.2	50.7	92.3	89.8	91.1
4	S2	Qwen2.5-VL-7B	71.3	48.3	68.5	65.5	65.1	77.5	40.2	16.7	56.6	66.8	74.5	70.7
5		VG LLM-8B	71.7	53.8	68.8	62.1	63.8	83.0	44.3	18.4	58.2	66.8	73.3	70.1
6	S1+S2	Qwen2.5-VL-7B	70.2	51.0	69.8	64.4	67.0	79.1	45.4	31.6	59.8	87.0	83.8	85.4
7		VG LLM-8B	71.4	56.8	69.0	69.1	67.9	83.2	47.4	32.5	62.2	92.0	89.5	90.8

Table 8: **Ablation study on spatial reasoning tasks.** ∅ indicates testing in a zero-shot setting. S1 is the training data detailed in Section 4.2.1, and S2 refers to the VLM-3R dataset.

We then investigate the effects of fine-tuning and visual geometry in Table 8. The table illustrates that fine-tuning on our mixed dataset (S1) leads to a substantial performance gain in measurement estimation (rows 1 vs. 2). In addition to the data used in this paper (S1), the VLM-3R dataset [16] (S2), which is specially curated for the task types in VSI-Bench, is also incorporated in our analysis. After incorporating the 3D geometry into the model architecture, the results show consistent gains with the different data compositions, *i.e.*, S1, S2 and S1+S2.

5 Conclusion

We present a novel framework to enhance MLLMs’ 3D spatial understanding capability, which incorporates a 3D visual geometry encoder to provide latent 3D geometric information given only video inputs. While being straightforward, our extensive experiments show that our model can outperform larger spatial-enhanced models on various 3D scene understanding tasks and spatial reasoning benchmarks, without relying on any explicit 3D scene input.

Acknowledgements This work is supported by National Key R&D Program of China (Project No. 2022ZD0161200, 2022ZD0161201). This work is also supported by Hong Kong Research Grant Council - Early Career Scheme (Grant No. 24200223).

References

- [1] Anthropic. Introducing the next generation of claude. <https://www.anthropic.com/news/claude-3-family>, 2024. 8
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv:2502.13923*, 2025. 1, 2, 3, 5, 6, 7
- [3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023. 1
- [4] Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. In *CVPR*, 2022. 5, 6
- [5] Wenxiao Cai, Yaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. *arXiv:2406.13642*, 2024. 2
- [6] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas J. Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *CVPR*, 2024. 2
- [7] Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. Scanrefer: 3d object localization in RGB-D scans using natural language. In *ECCV*, 2020. 2, 5, 6
- [8] Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. Scan2cap: Context-aware dense captioning in RGB-D scans. In *CVPR*, 2021. 2, 5, 6
- [9] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X. Chang. D³net: A unified speaker-listener architecture for 3d dense captioning and visual grounding. In *ECCV*, 2022. 5, 6
- [10] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Language conditioned spatial relation reasoning for 3d object grounding. In *NeurIPS*, 2022. 5, 6
- [11] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. LL3DA: visual interactive instruction tuning for omni-3d understanding, reasoning, and planning. In *CVPR*, 2024. 1, 2, 6
- [12] Sijin Chen, Hongyuan Zhu, Xin Chen, Yinjie Lei, Gang Yu, and Tao Chen. End-to-end 3d dense captioning with vote2cap-detr. In *CVPR*, 2023. 5, 6
- [13] Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Ruiyuan Lyu, Runsen Xu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv:2405.10370*, 2024. 2, 6
- [14] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. In *NeurIPS*, 2024. 2
- [15] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 4, 5
- [16] 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. 10
- [17] Chaoyou Fu, Yuhang Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv:2405.21075*, 2024. 9, 22
- [18] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual reasoning. In *WACV*, 2025. 2

- [19] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: multimodal large language models can see but not perceive. In *ECCV*, 2024. 9, 22
- [20] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. In *NeurIPS*, 2023. 1, 2, 6
- [21] Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. In *NeurIPS*, 2024. 1, 6
- [22] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *ICML*, 2024. 1, 5, 6, 22
- [23] Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3d visual grounding. In *CVPR*, 2022. 5, 6
- [24] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv:2410.21276*, 2024. 1, 2, 8, 9
- [25] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In *CoRL*, 2024. 1
- [26] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv:2408.03326*, 2024. 1, 2, 8, 9
- [27] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv:2403.18814*, 2024. 8, 9
- [28] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 1
- [29] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: on pre-training for visual language models. In *CVPR*, 2024. 1, 2
- [30] Benlin Liu, Yuhao Dong, Yiqin Wang, Zixian Ma, Yansong Tang, Luming Tang, Yongming Rao, Wei-Chiu Ma, and Ranjay Krishna. Coarse correspondences boost spatial-temporal reasoning in multimodal language model, 2024. 1
- [31] Benlin Liu, Yuhao Dong, Yiqin Wang, Yongming Rao, Yansong Tang, Wei-Chiu Ma, and Ranjay Krishna. Coarse correspondence elicit 3d spacetime understanding in multimodal language model. *arXiv:2408.00754*, 2024. 2
- [32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 8
- [33] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? In *ACL*, 2024. 22
- [34] OpenAI. GPT-4 technical report. *arXiv:2303.08774*, 2023. 8, 9
- [35] Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv:2501.01428*, 2025. 1, 2
- [36] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv:2501.15830*, 2025. 1

- [37] Arijit Ray, Jiafei Duan, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, Kuo-Hao Zeng, et al. Sat: Dynamic spatial aptitude training for multimodal language models, 2024. 2, 8, 9
- [38] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander G. Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. *arXiv:2412.06974*, 2024. 1, 2, 3
- [39] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024. 1, 2, 8, 9
- [40] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *NeurIPS*, 2024. 1, 2, 8, 9, 22
- [41] Haochen Wang, Yucheng Zhao, Tiancai Wang, Haoqiang Fan, Xiangyu Zhang, and Zhaoxiang Zhang. Ross3d: Reconstructive visual instruction tuning with 3d-awareness, 2025. 1
- [42] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. VGGT: visual geometry grounded transformer. *arXiv:2503.11651*, 2025. 1, 2, 3, 4, 5, 7
- [43] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv:2409.12191*, 2024. 1, 2
- [44] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 2, 3
- [45] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied AI. In *CVPR*, 2024. 2, 4, 5, 7, 22
- [46] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, et al. Cogvlm: Visual expert for pretrained language models. In *NeurIPS*, 2024. 8
- [47] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *ICLR*, 2024. 1
- [48] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021. 9
- [49] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *ECCV*, 2024. 2
- [50] 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. *arXiv:2412.14171*, 2024. 1, 2, 8, 22
- [51] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 5
- [52] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language models. *arXiv:2309.10313*, 2023. 2
- [53] Huanyu Zhang, Chengzu Li, Wenshan Wu, Shaoguang Mao, Yan xia, Ivan Vulić, Zhang Zhang, Liang Wang, Tieniu Tan, and Furu Wei. A call for new recipes to enhance spatial reasoning in mllms, 2025. 4

- [54] Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yujie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *arXiv:2503.22976*, 2025. 2, 4, 5, 6, 8, 9
- [55] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based VLM plans the next step for vision-and-language navigation. In *RSS*, 2024. 1
- [56] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024. 22
- [57] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv:2410.02713*, 2024. 5, 8
- [58] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d LLM: learning position-aware video representation for 3d scene understanding. In *CVPR*, 2025. 1, 2, 4, 5, 6, 22
- [59] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *CVPR*, 2024. 1
- [60] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *ECCV*, 2020. 5
- [61] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *AAAI*, 2024. 1
- [62] Shijie Zhou, Alexander Vilessov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Eric Xin Wang, and Achuta Kadambi. Vlm4d: Towards spatiotemporal awareness in vision language models. 2025. 4
- [63] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv:2409.18125*, 2024. 1, 2, 4, 5, 6, 22

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: Our abstract and introduction clearly state our main contributions. Our experimental results also support our claims.

Guidelines:

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

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: We have a section to discuss the limitations of our work. Please refer to Appx. [E](#).

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes]

Justification: We have included the detailed model design (Sec. 3.1), implementation details (Sec. 4) and evaluation details (Appx. A.1)

Guidelines:

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

5. Open access to data and code

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

Answer: [Yes]

Justification: We have provided instructions on data access and preparation, including how to access the raw, preprocessed, intermediate, and generated data. Please refer to Appx. B. The data and model weights will be publicly available upon paper publication.

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes]

Justification: We have included detailed experimental setup in Appx. A.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [No]

Justification: Error bars are not because it would be too computationally expensive.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

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

Answer: [Yes]

Justification: Yes, we report the type of computing resources, memory, and time required to reproduce our results in Sec. 4.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: Our research conforms in every respect with the NeurIPS Code of Ethics.

Guidelines:

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

10. Broader impacts

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

Answer: [Yes]

Justification: We have dedicated a section to discuss the societal impact of our work. Please refer to Appx. F.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [NA]

Justification: Our research builds upon existing datasets rather than creating new ones from the internet. So our work does not pose such risks to the best of our knowledge.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

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

Answer: [Yes]

Justification: All datasets used in our work are commonly used datasets with open access. We have adhered to their licenses and provided citations to give them credit.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [No]

Justification: The data and model weights will be publicly available upon paper publication.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

Answer: [NA]

Justification: Our research does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

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

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

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

Answer: [NA]

Justification: Our research does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The core method in our research does not involve LLMs as any components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Experimental Setting and Details

A.1 Evaluation Details.

We utilize the LMMs-Eval [56] to evaluate our method. LMMs-Eval offers flexible functionalities that support the customization of specific tasks and model interfaces, thereby standardizing the evaluation process of MLLMs. For inference, we employed greedy sampling to generate model outputs.

Spatial Reasoning and General Multi-modal Benchmarks. For benchmarks already integrated into LMMs-Eval, *e.g.*, VSI-Bench [50], Video-MME [17], and TempCompass [33], we utilize their original default configurations. For benchmarks not yet implemented in LMMs-Eval, *i.e.*, BLINK [19] and CV-Bench [40], we customize the task configurations and metrics according to their official implementations. For each input video in video-based benchmarks, we uniformly sample 32 frames from the entire duration.

3D Visual Grounding. For 3D visual grounding, the model is tasked with locating the frame index where the target object appears and the oriented bounding box within its coordinate system. Given a predicted frame index and bounding box, we first transform the bounding box to the world coordinate system. Then, the ground truth bounding box is extended to an oriented bounding box by appending (0, 0, 0) to represent its orientation. Finally, we calculate the Intersection-over-Union (IoU) between the predicted and ground truth bounding boxes. In the proposal refinement process, we utilize the Mask3D-detected object proposals extracted by LEO [22], and select the proposal that has the highest IoU with the predicted box.

3D Dense Captioning. For 3D dense captioning, we follow previous work [63, 58] to decompose the task into two stages, *i.e.*, detecting all object proposals and generating object descriptions based on object center coordinates. Specifically, we first leverage the Mask3D-detected object proposals extracted from LEO [22] to obtain the box’s center coordinates within the coordinate system of the reference frame. This can be achieved by multiplying the coordinates by the extrinsic matrix [45]. Then, the model is asked to generate descriptions based on the coordinates through greedy sampling. Lastly, we calculate the CIDEr, BLEU, Rouge, and METEOR scores utilizing the COCO caption evaluation toolkit⁴.

3D Video Object Detection. For each predicted bounding box, we find its best match among the unused ground truth boxes of the same category through greedy matching. Specifically, we calculate the IoU with all unused ground truth boxes of that category and select the one with the highest IoU if it exceeds a threshold (*e.g.*, 0.25). Once a ground truth is matched, it is marked as used and cannot be matched again. Finally, for each category, the precision, recall, and F1 score are calculated based on the number of true positives, false positives, and false negatives.

Additionally, we provide the prompt for 3D scene understanding tasks in Table 9.

B Data preparation

All datasets used in this research are publicly available, and we will provide the details for the data preparation in this section.

SPAR-7M. We follow the official codebase⁵ to mix data and draw visual markers on the input images. Since the navigation type contains images of varying lengths, we discard annotations of this type for simplicity.

LLaVA-Video-178K (LLaVA-Hound Split). For each input video, we sample frames with a sampling rate of 2 FPS, while constraining the total frame count between 4 and 8 through adaptive sampling.

3D Scene Understanding. For both ScanRefer and Scan2Cap, we uniformly sample 32 frames for each scene. For 3D video object detection, each entry in the training set contains 4 consecutive frames sampled at 1 FPS.

⁴<https://github.com/tylin/coco-caption>

⁵<https://github.com/fudan-zvg/spar>

3D Visual Grounding

X_{User} : Frame-0: Frame-1: Frame-2: Frame-3: ...

Localize the first clear frame in the video showing the object described in the text.

Text: There is a beige wooden bookshelf. placed next to another bookshelf.

Output a JSON dictionary with the frame index in "frame" and its 3D bounding box in "box_3d" in the frame's coordinates.

X_{Agent} :

```json

```
{ "frame": 12, "bbox_3d": [-0.63, -0.83, 2.43, 3.0, 0.59, 2.35, -2.32, 1.18, 3.05] }
```

```

3D Dense Captioning

X_{User} : ...

Carefully watch the video and describe the object located at [-0.89, -0.74, 2.45] in detail.

X_{Agent} : A white cabinet in the corner of the room. in the direction from the door and from the inside. it will be on the left, there is a small brown table on the left side of the cabinet and a smaller table on the right side of the cabinet.

3D Video Object Detection

X_{User} : ...

Detect the 3D bounding boxes in the camera coordinate system of the first frame.

Output a json list where each entry contains the object name in "label" and its 3D bounding box in "box_3d".

The 3D bounding box format should be [x_center, y_center, z_center, x_size, y_size, z_size, yaw, pitch, roll].

X_{Agent} :

```json

```
[
 { "label": "bag", "bbox_3d": [0.0, -0.3, 1.0, 0.26, 0.26, 0.15, 1.67, 0.96, -2.98] },
 ...
]
```

```

Table 9: **The prompt for 3D scene understanding tasks.** For 3D visual grounding and 3D video object detection, the output should be in JSON format, while for 3D dense captioning, it should be a natural language description.

C Detailed Results on 3D Video Object Detection

Detailed quantitative results for our 3D video object detection are presented in Table 10 and 11. As illustrated in the table, our method significantly outperforms the baseline that does not utilize visual geometry across most categories. Nevertheless, we observe that detecting small objects such as pillows, lamps, and backpacks remains challenging.

D More Visualization

Figure 4 shows more visualization results for 3D visual grounding. The first three examples are positive cases where our model successfully locates the 3D bounding boxes for different categories like chair, cabinet, and document organizer. While the model correctly identifies these objects, we notice that it still struggles with accurately predicting their orientation. The last two examples are negative cases. While the predictions might appear correct in the 2D projection, the model's inaccurate depth estimation causes some discrepancies in 3D space.

We provide more visualization results on 3D video object detection in Figure 5. From this figure, we observe that while the baseline performs decently in detecting objects in a given video, incorporating 3D geometry significantly improves both detection precision and recall. For instance, in the first example, the baseline's prediction for the desk mismatches the ground truth, whereas our approach improves the prediction box of the desk.

Object	Qwen2.5-VL-3B	VG LLM-4B	Qwen2.5-VL-7B	VG LLM-8B
chair	32.8	41.6	36.1	48.7
pillow	11.4	15.9	12.4	18.9
cabinet	7.8	12.4	10.6	17.9
table	31.3	39.8	32.7	44.8
lamp	5.6	15.6	8.0	15.9
couch	32.2	45.0	40.7	46.4
desk	31.6	38.5	34.4	32.3
stand	20.6	41.4	31.3	47.3
bed	58.8	70.2	64.6	75.8
backpack	28.9	37.5	28.6	40.4
bathtub	36.5	33.8	38.4	45.2
ottoman	0.0	2.8	2.7	5.3
dresser	31.2	41.1	36.4	44.5
bin	20.9	33.1	25.0	38.5
toilet	66.1	80.6	68.6	83.2
refrigerator	28.9	64.3	33.2	62.0
stove	37.8	68.5	55.9	69.5
microwave	11.5	21.1	7.8	25.8
monitor	15.5	22.1	20.3	31.7
computer	4.4	3.1	5.2	11.6

Table 10: **The detailed results on 3D video object detection in 6-frame setting.** We report the F1 score for all categories.

Object	Qwen2.5-VL-3B	VG LLM-4B	Qwen2.5-VL-7B	VG LLM-8B
chair	37.7	49.7	41.2	54.0
pillow	11.9	17.9	13.2	22.9
cabinet	10.2	13.1	11.6	17.1
table	35.0	41.3	36.5	46.5
lamp	4.4	16.4	9.0	18.9
couch	39.0	44.6	41.1	47.0
desk	33.7	39.0	34.5	41.0
stand	23.3	37.9	33.7	46.1
bed	64.8	71.2	68.2	74.1
backpack	27.4	42.0	33.3	41.4
bathtub	32.4	33.5	36.6	42.1
ottoman	3.2	3.8	0.0	0.0
dresser	33.1	40.3	38.2	49.2
bin	23.1	39.2	30.2	39.8
toilet	68.8	83.4	68.7	82.5
refrigerator	32.2	57.3	38.5	53.6
stove	62.5	68.8	61.2	68.6
microwave	25.9	32.1	21.3	33.3
monitor	22.3	29.1	21.3	34.0
computer	6.8	3.8	12.0	11.4

Table 11: **The detailed results on 3D video object detection in 4-frame setting.** We report the F1 score for all categories.

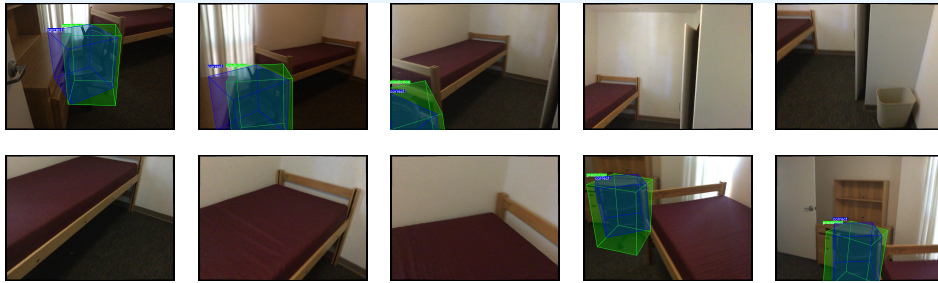
E Limitations

While our model is built upon 3B and 7B MLLM backbones, its capabilities remain constrained by the model’s inherent capacity. Scaling up the model size could potentially enhance both fundamental abilities and generalization performance. Furthermore, this work focuses exclusively on supervised fine-tuning, whereas emerging research demonstrates that reinforcement learning can significantly enhance model reasoning capabilities. Since VGGT is trained on commonly used 3D scene datasets, evaluating it on benchmarks built from that same data (such as ScanNet) poses a greater risk of overestimating its performance compared to evaluating it on out-of-distribution benchmarks. We leave these for future work.

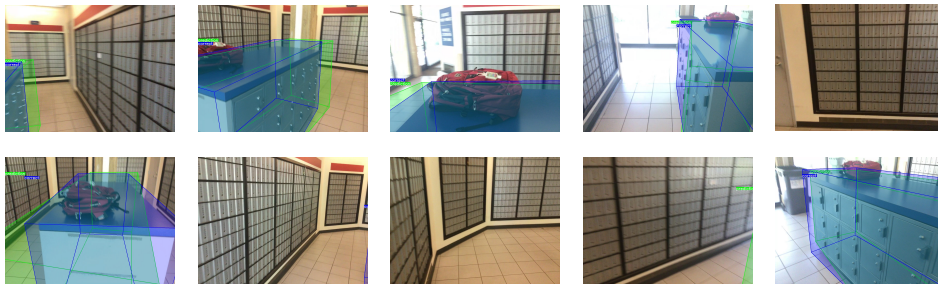
F Broader Impact

VG LLM is built upon Qwen2.5-VL and consequently shares many common challenges associated with multi-modal large language models (MLLMs). These challenges include hallucinations in visual understanding, inherited biases from the base models, and susceptibility to adversarial inputs. Despite these limitations, releasing VG LLM to the research community would be highly beneficial, as it could catalyze further advancements in 3D world understanding by leveraging Video LLMs and 3D vision geometry priors.

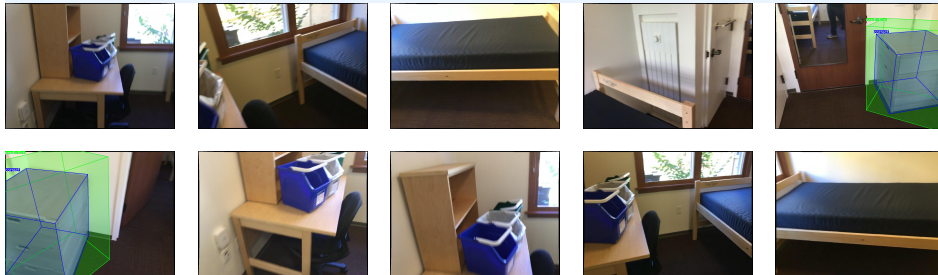
The office chair is next to the wooden table and next to the bed. the office chair has a curved backside and four legs.



The cabinet is in the middle of the room. the cabinet is a white box with a blue rectangular top.



A small document organizer with shelves. there is a door entry near it.



There is a door in the middle of the northern wall next to a metal cabinet. the door is shaped like a brown rectangle and as a dark yellow colored doorknob.

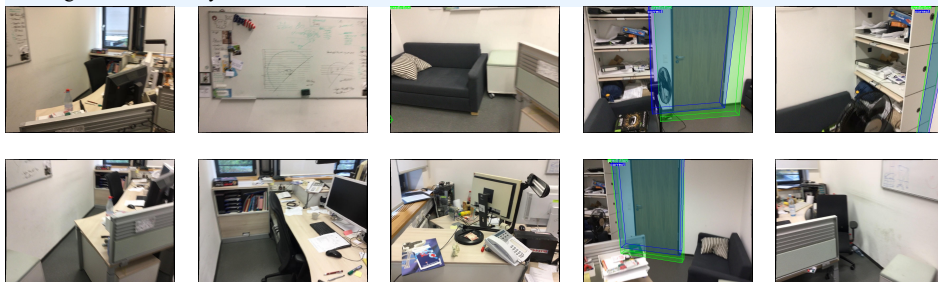


Figure 4: **Visualization of 3D visual grounding.** The first three examples are positive, whereas the last two are negative cases.

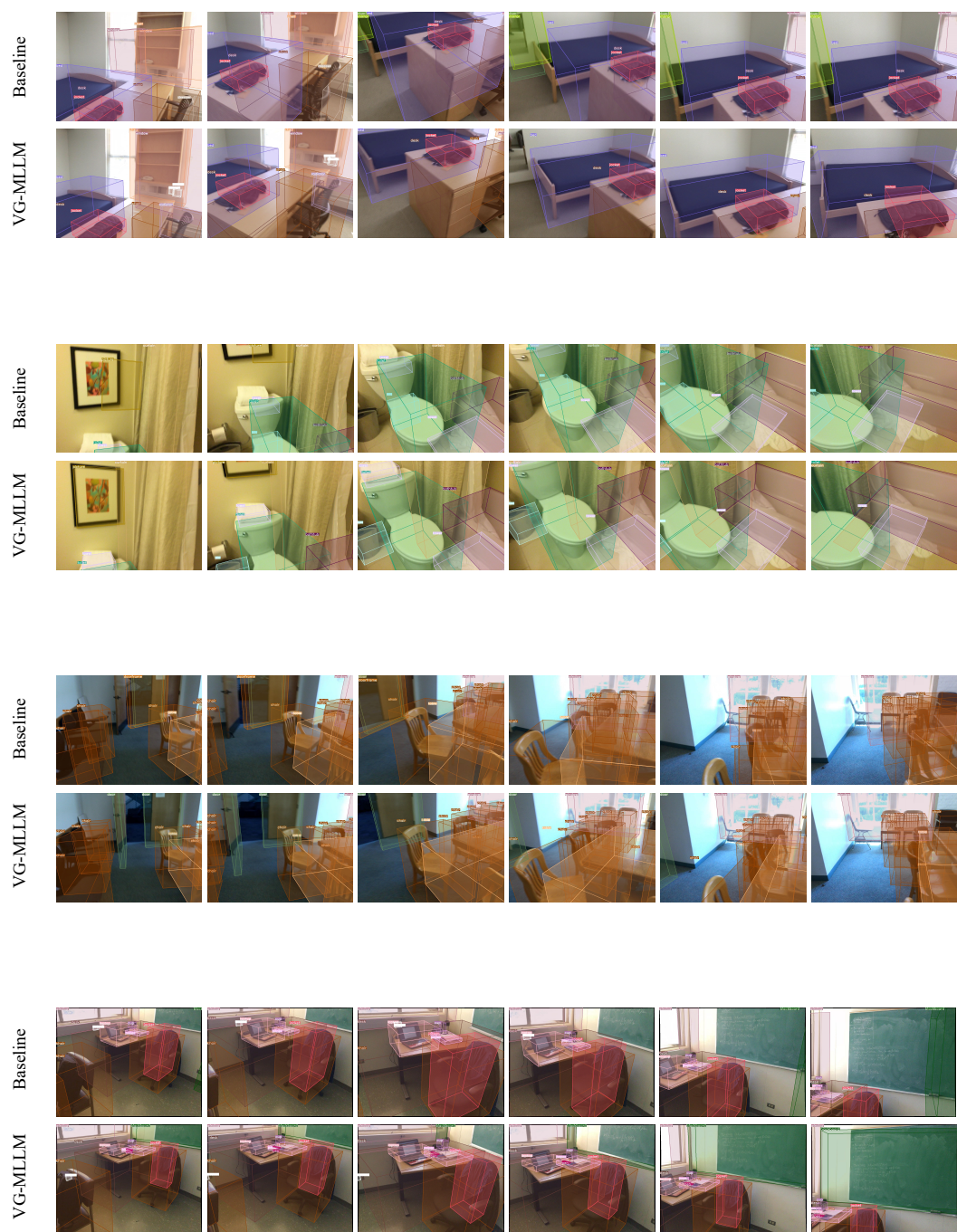


Figure 5: Visualization of 3D video object detection results.