

# Spatial-MLLM: Boosting MLLM Capabilities in Visual-based Spatial Intelligence

Diankun Wu \*  
Tsinghua University

Fangfu Liu \*  
Tsinghua University

Yi-Hsin Hung  
Tsinghua University

Yueqi Duan †  
Tsinghua University

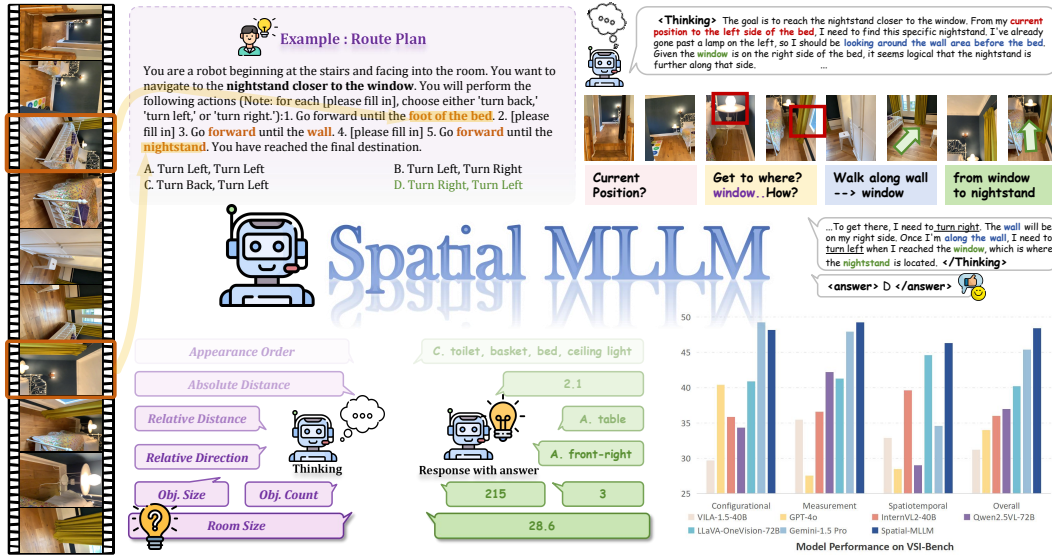


Figure 1: We propose *Spatial-MLLM*, a method that significantly enhances the visual-based spatial intelligence of existing video MLLMs. As shown, *Spatial-MLLM* is capable of understanding and reasoning about the underlying scene from video input, achieving state-of-the-art performance across a wide range of tasks.

## Abstract

Recent advancements in Multimodal Large Language Models (MLLMs) have significantly enhanced performance on 2D visual tasks. However, improving their spatial intelligence remains a challenge. Existing 3D MLLMs always rely on additional 3D or 2.5D data to incorporate spatial awareness, restricting their utility in scenarios with only 2D inputs, such as images or videos. In this paper, we present *Spatial-MLLM*, a novel framework for visual-based spatial reasoning from purely 2D observations. Unlike conventional video MLLMs which rely on CLIP-based visual encoders optimized for semantic understanding, our key insight is to unleash the strong structure prior from the feed-forward visual geometry foundation model. Specifically, we propose a dual-encoder architecture: a pretrained 2D visual encoder to extract semantic features, and a 3D spatial encoder—initialized from the backbone of the visual geometry model—to extract 3D structure features. A connector then integrates both features into unified visual tokens for enhanced spatial understanding. Furthermore, we propose a space-aware frame sampling strategy at inference time, which selects the spatially informative frames of a

\*Equal Contribution.

†Corresponding Author.

video sequence, ensuring that even under limited token length, the model focuses on frames critical for spatial reasoning. Beyond architecture improvements, we construct a training dataset from multiple sources and train the model on it using supervised fine-tuning and GRPO. Extensive experiments on various real-world datasets demonstrate that Spatial-MLLM achieves state-of-the-art performance in a wide range of visual-based spatial understanding and reasoning tasks. Project page: <https://diankun-wu.github.io/Spatial-MLLM/>.

## 1 Introduction

Multimodal Large Language Models (MLLMs) [1, 2, 3] have achieved significant progress in processing multimodal inputs to generate contextually aware and semantically coherent responses. While proprietary models such as Gemini [4] and GPT-4o [5] exhibit state-of-the-art performance, the open-source community continues to advance the field by improving these models’ ability to interpret diverse content modalities, including images [6, 7, 8], videos [9, 10, 11, 12, 13, 14], and audio [15, 16, 17]. Although these models excel at a wide range of 2D tasks, their capacity to perceive, understand, and reason about 3D scenes, *i.e.*, *3D spatial intelligence*, remains limited [18, 19].

The requirement of spatial understanding and reasoning typically arises in two scenarios. In the first scenario, the model has access to additional 3D or 2.5D data (*e.g.*, point clouds, camera parameters, or depth maps) alongside 2D visual inputs (*e.g.*, images or videos). These supplementary modalities enhance the model’s spatial awareness, enabling more accurate spatial reasoning. However, this setup limits the model’s applicability in many real-world scenarios where only monocular video of the scene is available, which is the second scenario. The model’s ability to perform spatial understanding and reasoning under such conditions is referred to as *visual-based 3D spatial intelligence* [18, 20]. A major challenge in this setting is that each frame provides only a partial observation of the scene, and no global representation (*e.g.*, the point clouds [21, 22, 23] or posed depth maps [24, 25]) is available as input. This requires the model to infer the global spatial layout from incomplete cues and internally integrate these partial observations into a coherent and implicit global representation, which demands strong spatial awareness. However, most existing video MLLMs pretrain their visual encoders on image-text pairs—primarily image-caption data [13, 14, 26]—following the CLIP [27] paradigm. This makes the visual encoder excel at capturing high-level semantic content but lack structure and spatial information when only 2D video inputs are available [28, 29, 30]. Consequently, current video MLLMs generally perform worse on spatial reasoning tasks than on other tasks, such as temporal understanding. Moreover, their performance still significantly lags behind human capabilities [18].

In this paper, we introduce *Spatial-MLLM*, a method that significantly improves the visual-based 3D spatial intelligence of existing video MLLMs. To address the limitations of visual encoders in general-purpose video MLLMs, our key insight is to unleash the strong structure prior provided by the feed-forward visual geometry foundation model [31, 32, 33, 34]. These models, typically trained on pixel-point pairs, complement the general-purpose video MLLM visual encoders that are trained primarily on image-text data [14]. Based on this insight, we design a dual-encoder architecture consisting of a 2D encoder—initialized from the visual encoder of a general-purpose video MLLM—to extract 2D semantic information, and a spatial encoder—leveraging the VGGT feature extractor [32]—to recover implicit 3D structural information from 2D video inputs. We then use a connector to integrate features from both branches into unified visual tokens. The resulting representation enables the Large Language Model (LLM) backbone to perform effective spatial reasoning without requiring explicit 3D data as input.

Furthermore, we fully exploit the additional information provided by the introduced feed-forward visual geometry model [32], and propose a space-aware frame sampling strategy at inference time, which selects the most spatially informative frames from the video sequence when the total number of input frames is limited (*e.g.*, due to the VRAM limitation). Specifically, we first feed a relatively large number of frames into the spatial encoder and decode the resulting 3D features into voxels. The frame selection task is then reformulated as a maximum coverage problem over these voxels, which we solve using a greedy algorithm. To train Spatial-MLLM, we construct a visual-based 3D spatial question-answering dataset and perform supervised fine-tuning on it. We further apply a simple cold-start [35] to help the model adapt to the correct reasoning format, and then train it using Group Relative Policy Optimization (GRPO) [36, 35] to enhance its long-chain-of-thought (long-CoT) spatial reasoning capability [37]. We conduct extensive evaluations on the VSIBench [18],

ScanQA [38], and SQA3D [39] benchmarks and demonstrate that the proposed spatial-MLLM achieves state-of-the-art performance in a wide range of visual-based spatial understanding and reasoning tasks.

In summary, our main contributions are:

- We introduce Spatial-MLLM, a method that significantly enhances the visual-based 3D spatial intelligence of existing video MLLMs, demonstrating strong spatial understanding and reasoning capabilities without requiring any 3D or 2.5D data input.
- We design a dual-encoder and connector that effectively integrates semantic information from a standard 2D visual encoder with structural information extracted by a spatial encoder, which is initialized using a feed-forward visual geometry foundation model.
- We fully exploit the additional information provided by the feed-forward visual geometry model and design a space-aware frame sampling strategy that selects spatially informative frames, thereby improving model performance under input length constraints.
- We train our model on the constructed dataset with a two-stage pipeline. Extensive experiments demonstrate that our method achieves state-of-the-art performance on a wide range of visual-based spatial understanding and reasoning tasks.

## 2 Related Work

### 2.1 MLLMs for Video Understanding

Multimodal Large Language Models have made significant progress in integrating vision and language. Early works such as BLIP-2 [2] and Flamingo [1] introduce token-level fusion (*e.g.*, Q-Former) and feature-level fusion (*e.g.*, cross-attention layers) to bridge modalities. Other approaches, including the LLaVA series [3, 40], MiniGPT-4 [41], and subsequent models [13, 42, 43], leverage MLPs to project visual features into the language space. Recent advancements in MLLMs have extended their capabilities from static images to videos, typically by introducing video-language alignment through large-scale pretraining [9, 44]. Later models, such as Qwen2.5-VL [14], enhance temporal reasoning via dynamic resolution and absolute time encoding. Although existing video MLLMs excel at capturing high-level semantics and temporal patterns, they struggle to interpret the underlying 3D scene from video input, which inspires our work to enhance their spatial understanding capabilities.

### 2.2 3D MLLMs for Scene Understanding

Recent advances in MLLMs have sparked interest in extending their capabilities from 2D to 3D scene understanding [23, 24, 25, 45, 46, 47, 22, 48, 49, 50, 51, 52]. LL3DA [23] extracts scene-level features from 3D point clouds using a Q-Former, while Grounded 3D-LLM [45] integrates 3D detectors to generate object proposals. Methods like Chat3D [46], LEO [47], and Chat-Scene [22] first segment 3D objects and encode object-centric features for LLM fusion. Alternatively, 3D-LLM [48] and Scene-LLM [49] aggregate CLIP features from pre-segmented multi-view object patches into 3D point representations, leveraging multi-view images and camera parameters. LLaVA-3D [24] projects 2D multi-view patch features into voxel space for 3D-aware aggregation, and GPT4Scene [50] enhances 3D reasoning by first reconstructing scenes and then using BEV images as input. While these methods advance 3D scene understanding, most of them require additional 3D or 2.5D input data that is difficult to acquire in real-world scenarios. In contrast, our approach only requires 2D videos as input.

### 2.3 Visual-based 3D Spatial Intelligence

Visual-based 3D spatial intelligence aims to enable video multimodal large language models (MLLMs) to perceive, infer, and reason about 3D structures and spatial relationships purely from 2D visual inputs. Despite recent advances, most existing video MLLMs are still primarily designed for 2D understanding tasks, and their extension to visual-based 3D reasoning, *e.g.*, 3D question answering [6, 53] and robotic manipulation [54], remains relatively underexplored. To address this limitation, a new wave of specialized benchmarks has emerged to systematically evaluate the spatial reasoning capabilities of these models [18, 20, 55, 56, 57, 58]. Among them, VSI-Bench [18] serves as a

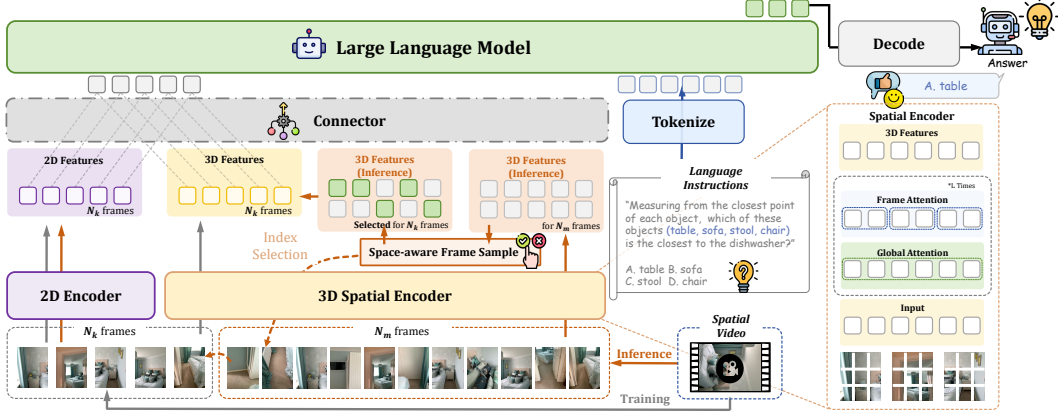


Figure 2: **Overview of Spatial-MLLM.** Our model is composed of a 2D visual encoder  $\mathcal{E}_{2D}$ , a 3D spatial encoder  $\mathcal{E}_{\text{Spatial}}$ , which is initialized from a feed-forward visual geometry foundation model, a connector, and a large language model backbone. At inference time, we incorporate a space-aware frame sampling strategy to select spatially informative frames when the number of input frames is limited due to GPU memory constraints.

pioneering benchmark that comprehensively assesses visual-spatial intelligence across multiple dimensions. STI-Bench [20] introduces physics-aware questions, such as velocity estimation, to quantify a model’s spatial and kinematic reasoning abilities. Ego-ST Bench [55] evaluates the model’s spatial intelligence from an egocentric perspective, while VLM4D [56] emphasizes motion dynamics, such as trajectory prediction, to probe 4D spatiotemporal interactions. Collectively, these benchmarks signify a shift toward a more holistic evaluation of visual-based spatial intelligence in MLLMs.

### 3 Method

In this section, we introduce Spatial-MLLM. Given a video of  $N$  frames depicting a scene, denoted as  $\mathcal{V} = \{\mathbf{f}_i\}_{i=1}^N$ , where  $\mathbf{f}_i \in \mathbb{R}^{H \times W \times 3}$ , Spatial-MLLM is designed to understand spatial relationships, perform spatial reasoning, and generate appropriate responses. We begin by describing the model architecture in Section 3.1, which comprises a 2D visual encoder, a 3D spatial encoder, a connector, and a large language model backbone. Then we present the space-aware frame sampling strategy in Section 3.2, which selects  $N_k$  spatially informative frames  $\{\mathbf{f}_i^k\}_{i=1}^{N_k}$ , where  $N_k \ll N$ . Finally, we introduce the training dataset construction process and two-stage training pipeline in Section 3.3.

#### 3.1 Spatial-MLLM Architecture

In this section, we present the architecture of Spatial-MLLM, which is shown in Figure 2. We adopt Qwen2.5-VL-3B [14] as our base model and explore strategies to enhance its spatial understanding and reasoning capability. Before diving into the details, we first briefly introduce the key insights that motivate our design.

**What hinders visual-based spatial intelligence in existing video MLLMs?** Existing video MLLMs [14, 13, 12] typically employ a pre-trained 2D visual encoder  $\mathcal{E}_{2D}$  to extract 2D patch features  $\mathbf{e}_{2D}$ . These features are then projected into visual tokens through a lightweight connection module. A large language model backbone  $f_\theta$  subsequently generates the final response by conditioning on both visual and textual tokens. A critical bottleneck in this process lies in the nature of the visual features extracted. The required type of information varies by task: high-level semantic representations are essential for 2D recognition and understanding, whereas fine-grained structural cues are crucial for spatial reasoning. However, the visual encoders used in current video MLLMs are primarily pre-trained on image-text datasets (mainly image-caption pairs) [14, 26] following the CLIP [27] paradigm. As a result, these models predominantly capture semantic content and often lack spatial awareness when no additional 3D or 2.5D data are available [28, 29, 30]. To address this, our key insight is to unleash feed-forward visual geometry foundation models [32], which are trained on pixel-point pairs and can recover rich 3D structural information from 2D inputs, which complements the semantic features extracted by the 2D visual encoder. We design a dual-encoder architecture that

exploits the strengths of both models and a connector to fuse semantic and structural information into unified visual tokens. Below, we introduce the core components of our design.

**Dual-Encoder.** The proposed dual-encoder consists of a 2D encoder  $\mathcal{E}_{2D}$  and a 3D spatial encoder  $\mathcal{E}_{\text{Spatial}}$ . For the 2D encoder branch, we adopt the same design as the visual encoder of Qwen2.5-VL [14] to encode input frames into semantically rich features:

$$\mathbf{e}_{2D} = \mathcal{E}_{2D} \left( \{\mathbf{f}_i\}_{i=1}^{N_k} \right), \quad \mathbf{e}_{2D} \in \mathbb{R}^{N_k' \times \lfloor \frac{H}{p_{2D}} \rfloor \times \lfloor \frac{W}{p_{2D}} \rfloor \times d_{2D}}, \quad (1)$$

where  $p_{2D}$  and  $d_{2D}$  denote the patch size and feature dimension of the 2D visual encoder, respectively. The two consecutive frames are grouped for video input, thus  $N_k' = \lceil N_k/2 \rceil$ .

For the spatial encoder branch, we utilize the feature backbone of VGGT [32]. Specifically, given  $N_k$  frames of the scene video, we first patchify the input and then extract 3D features with alternating frame-wise self-attention and global self-attention [59]. This process allows  $\mathcal{E}_{\text{Spatial}}$  to aggregate spatial information across different frames to get the final 3D features:

$$\mathbf{e}_{3D}, \mathbf{e}_c, \mathbf{e}_{\text{register}} = \mathcal{E}_{\text{Spatial}} \left( \{\mathbf{f}_i\}_{i=1}^{N_k} \right), \quad \mathbf{e}_{3D} \in \mathbb{R}^{N_k \times \lfloor \frac{H}{p_{3D}} \rfloor \times \lfloor \frac{W}{p_{3D}} \rfloor \times d_{3D}}, \quad (2)$$

where  $\mathbf{e}_{3D}$ ,  $\mathbf{e}_c$ , and  $\mathbf{e}_{\text{register}}$  represent the dense 3D feature, the camera feature for each frame, and the register tokens [60], respectively. We only use  $\mathbf{e}_{3D}$  in the feature fusion stage as it captures the dense structure information of the input frames.

**Connector.** After obtaining the 2D and 3D features, we use a connector to integrate the semantic and structural information from both branches. Specifically, we first align  $\mathbf{e}_{3D}$  with  $\mathbf{e}_{2D}$  in both spatial and temporal dimensions:

$$\mathbf{e}'_{3D} = \text{Rearrange}(\mathbf{e}_{3D}), \quad \mathbf{e}'_{3D} \in \mathbb{R}^{N_k' \times \lfloor \frac{H}{p_{2D}} \rfloor \times \lfloor \frac{W}{p_{2D}} \rfloor \times d'_{3D}}. \quad (3)$$

Here, the spatially and temporally adjacent information in  $\mathbf{e}_{3D}$  is aggregated into the feature channel dimension, enabling alignment with  $\mathbf{e}_{2D}$ . Next, we employ a lightweight connector to fuse the information to obtain the unified visual tokens:

$$\mathbf{e} = \text{Connector}(\mathbf{e}_{2D}, \mathbf{e}'_{3D}), \quad (4)$$

where  $\mathbf{e} \in \mathbb{R}^{S \times d_{\text{unim}}}$  denotes the final visual tokens and  $S = N_k' \times \lfloor \frac{H}{p_{2D}} \rfloor \times \lfloor \frac{W}{p_{2D}} \rfloor$  is the sequence length. In practice, we adopt a MLP-based design (detailed in Section B.2). Although more complex feature fusion methods, *e.g.*, cross-attention [59, 61, 51], could be applied, we find that this approach is effective to enhance the model’s spatial understanding and reasoning capabilities. We leave the exploration of more advanced fusion strategies for future work.

### 3.2 Space-Aware Frame Sampling

Due to GPU memory constraints, video MLLMs can process only a limited subset of frames from a scene video sequence. For example, in the VSI-Bench setup [18], only 8 to 32 frames are sampled as input to the video MLLM, while a typical scene video in VSI-Bench contains over 2,000 frames. A widely adopted solution is uniform frame sampling [13, 14, 18], which is effective for general-purpose video understanding. However, as spatial videos represent 3D scenes, the sampling strategy for spatial understanding tasks should focus on capturing most information of the underlying scene, which uniform sampling fails to achieve.

Benefiting from the feed-forward visual geometry foundation model, we design a straightforward space-aware frame sampling strategy at inference time. Specifically, given a scene video  $\mathcal{V} = \{\mathbf{f}_i\}_{i=1}^N$ , our objective is to select  $N_k$  frames,  $\{\mathbf{f}_i^k\}_{i=1}^{N_k}$  that have most coverage of the underlying scene. To achieve this, we first uniformly subsample  $N_m$  frames,  $\{\mathbf{f}_i^m\}_{i=1}^{N_m}$ , where  $N_m$  satisfies  $N_k < N_m < N$ , and is determined by the available GPU memory. In practice, we choose  $N_m = 128$  and  $N_k = 16$ . We then leverage  $\mathcal{E}_{3D}$  to extract their corresponding 3D features  $\mathbf{e}_{3D}^m$  and camera features  $\mathbf{e}_c^m$ . Subsequently, we use the pretrained camera head  $f_c$  and depth head  $f_d$  of the VGGT model [32] to decode a set of camera parameters and depth maps:

$$\{\mathbf{E}_i^m, \mathbf{K}_i^m\}_{i=1}^{N_m} = f_c(\mathbf{e}_c), \quad \text{and} \quad \{\mathbf{D}_i^m\}_{i=1}^{N_m} = f_d(\mathbf{e}_{3D}). \quad (5)$$

This allows us to calculate the voxels  $V(f_i^m)$  covered by each frame  $f_i^m$ , and formulate frame selection as a maximum coverage problem [62], *i.e.*, select  $N_k$  frames  $\{\mathbf{f}_i^k\}_{i=1}^{N_k} \subseteq \{\mathbf{f}_i^m\}_{i=1}^{N_m}$  such that the total number of unique covered voxels  $|\bigcup_{i=1}^{N_k} V(\mathbf{f}_i^k)|$  is maximized. In practice, we apply a greedy algorithm to accelerate computation [63, 25]. Once the  $N_k$  frames are selected, it is not necessary to recompute their 3D features  $\mathbf{e}_{3D}^k$  and the corresponding features from the precomputed set  $\mathbf{e}_{3D}^m$  can be directly reused. We provide the complete algorithm and detailed explanation in Section B.1.

### 3.3 Training

**Training Data Construction.** We first construct a visual-based 3D spatial question-answering dataset. The dataset has approximately 120k QA pairs and is constructed from three sources: the training set of ScanQA [38], SQA3D [39], as well as additional self-created spatial QA data. All items in our training dataset are derived from scenes in the training set of ScanNet [64] and are each represented as a quadruple  $\mathcal{I}_i = \langle \mathcal{Q}_i, \mathcal{A}_i, \mathcal{V}_i, \mathcal{M}_i \rangle$ , denoting the question, answer, video (scene) ID, and meta-information (*e.g.*, task type), respectively. For the self-created QA data, we follow the data processing pipeline proposed in VSI-Bench [18]. Specifically, we first convert ScanNet scenes into continuous video clips at 24 FPS and  $640 \times 480$  resolution. Then we generate spatial reasoning QA pairs leveraging the meta-annotations of Scannet. The generated QA pairs cover various spatial understanding and reasoning tasks, including object counting, object size, room size, absolute distance, appearance order, relative distance, and relative direction. Since the QA pair construction process is similar to that of VSI-Bench [18], we exclude the QA pair  $\mathcal{I}_i$  if its scene video  $\mathcal{V}_i$  is used in VSI-Bench (these videos are sourced from the validation set of Scannet) to prevent data leakage. Finally, the self-created data contains approximately 70k QA pairs in total. We provide additional details on training data construction in the section B.3. Figure 3 shows a brief summary of key statistics of the training dataset.

**Supervised Fine-tuning.** Leveraging the constructed training dataset, we first perform supervised fine-tuning (SFT) on our model. Since  $\mathcal{E}_{2D}$  and  $\mathcal{E}_{\text{spatial}}$  are pre-trained on large-scale image-text and pixel-point pairs, respectively, we freeze them to preserve their ability to extract rich semantic and structural information. We jointly train the connection module and the LLM backbone to enable the model to adaptively fuse 2D and 3D features and enhance its spatial understanding and reasoning capability. During this stage, we employ the standard cross-entropy loss  $\mathcal{L}_{ce}$  between the model-generated answers and the ground-truth annotations:

$$\mathcal{L}_{ce}(\theta) = - \sum_i \log P(o^{(i)} | o^{(1:i-1)}, q, \{f_j\}_{j=1}^{N_k}) \quad (6)$$

where  $\{f_j\}_{j=1}^{N_k}$  denotes input video frames,  $q$  denotes the system prompt and question,  $o^{(i)}$  represents the  $i$ -th token in the ground-truth answer, and  $o^{(1:i-1)}$  denotes the preceding answer tokens.

**RL Training.** Following the SFT stage, we first perform a simple cold start [35] to help the model adapt to the correct reasoning format. Then we train the model using Group Relative Policy Optimization (GRPO) [36] to enhance its long-CoT [37] spatial reasoning capability. During training, we first sample a set of output  $\{o_1, o_2, \dots, o_G\}$  for each question  $q$  from the policy model  $\pi_{\theta_{old}}$ . Then we optimize the policy model by maximizing the following objective:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{q, o_i} \left[ \frac{1}{G} \sum_{i=1}^G \min \left( \frac{\pi_{\theta}(o_i | q)}{\pi_{\theta_{old}}(o_i | q)} A_i, \text{clip} \left( \frac{\pi_{\theta}(o_i | q)}{\pi_{\theta_{old}}(o_i | q)}, 1 \pm \epsilon \right) A_i \right) - \beta \text{KL}[\pi_{\theta} \| \pi_{\text{ref}}] \right] \quad (7)$$

where  $A_i = \frac{r_1 - \text{mean}(r_1, r_2, \dots, r_G)}{\text{std}(r_1, r_2, \dots, r_G)}$  is the advantage function computed using the group rewards.

In GRPO, the design of the reward function is critical. In addition to a formatting reward applied to all task types, we introduce task-dependent reward modeling to ensure that it accurately reflects

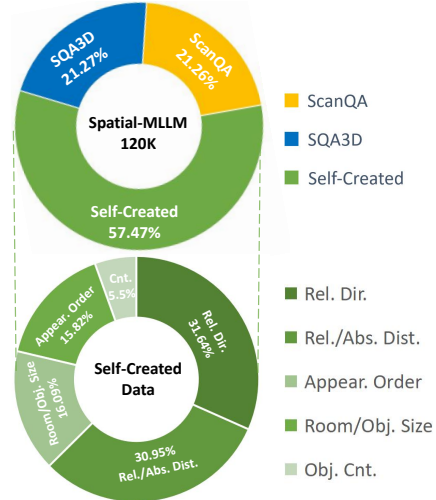


Figure 3: Basic statistic of training dataset.

Table 1: **Evaluation Results on VSI-Bench [18]**. For Spatial-MLLM and Qwen2.5-VL series [14], we use 16 frames as input and report micro average scores. For other open-source methods and GPT-4o [5], the number of frames is the same as VSI-Bench setting (ranging from 8 to 32 frames). For Gemini-1.5 Pro [4], it samples video frames at 1 FPS. **Bold** and underline denote the best-performing and second-best-performing open-source models, respectively.

Methods	Numerical Question				Multiple-Choice Question				Avg. Rank	
	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order		
<i>Proprietary Models</i>										
GPT-4o [5]	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5	34.0	7
Gemini-1.5 Pro [4]	56.2	30.9	64.1	43.6	51.3	46.3	36.0	34.6	45.4	2
<i>Open-source Models</i>										
InternVL2-40B [7]	34.9	26.9	46.5	31.8	42.1	32.2	<u>34.0</u>	39.6	36.0	6
LongVILA-8B [66]	29.1	9.1	16.7	0.0	29.6	30.7	32.5	25.5	21.6	12
VILA-1.5-40B [67]	22.4	24.8	48.7	22.7	40.5	25.7	31.5	32.9	31.2	9
LongVA-7B [68]	38.0	16.6	38.9	22.2	33.1	<u>43.3</u>	25.4	15.7	29.2	11
LLaVA-OneVision-72B [6]	43.5	23.9	<u>57.6</u>	37.5	<b>42.5</b>	39.9	32.5	44.6	40.2	4
LLaVA-Video-72B [12]	<u>48.9</u>	22.8	57.4	35.3	<u>42.4</u>	36.7	<b>35.0</b>	<b>48.6</b>	<u>40.9</u>	<u>3</u>
<i>Spatial-MLLM and Qwen2.5-VL Series</i>										
Qwen2.5-VL-3B [14]	24.3	24.7	31.7	22.6	38.3	41.6	26.3	21.2	30.6	10
Qwen2.5-VL-7B [14]	40.9	14.8	43.4	10.7	38.6	38.5	33.0	29.8	33.0	8
Qwen2.5-VL-72B [14]	25.1	<u>29.3</u>	54.5	<u>38.8</u>	38.2	37.0	<u>34.0</u>	28.9	37.0	5
<b>Spatial-MLLM-4B</b>	<b>65.3</b>	<b>34.8</b>	<b>63.1</b>	<b>45.1</b>	41.3	<b>46.2</b>	33.5	<u>46.3</u>	<b>48.4</b>	<b>1</b>

the proximity between the predicted and ground-truth answers. Specifically, we categorize the data into three types based on answer format: numeric answer questions, multiple-choice questions, and verbal answer questions. For numeric questions, we compute the mean relative accuracy [18]. For multiple-choice questions, we employ an exact match reward. For verbal answer questions, we use fuzzy matching based on Levenshtein distance. Further details on reward calculation are provided in Section B.5.

## 4 Experiments

### 4.1 Implementation Details

**Training details.** Spatial-MLLM is built on Qwen2.5-VL [14] and VGGT [32] and has approximately 4.9B parameters in total. We use the visual encoder of Qwen2.5-VL [14] to initialize  $\mathcal{E}_{\text{2D}}$ , and the LLM backbone of it to initialize  $f_{\theta}$ . We then use the feature backbone of VGGT [32] to initialize  $\mathcal{E}_{\text{spatial}}$ . During training, we use  $640 \times 480$  resolution and limit video frames to 16. In the SFT stage, we train the model using Adam optimizer [65] for one epoch. We set the global batch size to 16 and use a linear learning-rate schedule, with a peak value of  $10^{-5}$ . In the cold start stage, we first construct a small CoT dataset. Specifically, we prompt Qwen2.5-VL-72B [14] to generate multiple thinking processes and answers according to the scene video and question. Then we use the GT answer to filter a correct thinking-answer pair (more details are provided in Section B.4). We use a similar setting as in the SFT stage to train the model for 200 steps. In the RL stage, we perform 8 rollouts per question and set the default sampling temperature to 1. The KL divergence coefficient,  $\beta$ , is set to 0.04. Due to computational resource limitations, we train the model for 1,000 steps with a learning rate of  $1\text{e-}6$ . We show the training curve of SFT Stage and RL Stage in Figure 4.

**Inference Details.** During inference, we set  $N_m = 128$  and  $N_k = 16$  for space-aware frame sampling. Since spatial reasoning requires a certain level of determinism, we set the temperature to 0.1 and the top- $p$  to 0.001. The default input resolution from the scene video is  $640 \times 480$ .

### 4.2 Comparisons on VSI-Bench

**Setup.** VSI-Bench [18] contains more than 5,000 question-answer pairs derived from egocentric videos sourced from ScanNet [64], ScanNet++[69], and ARKitScenes[70]. The task types are divided into Multiple-Choice Answer (MCA) and Numerical Answer (NA). For the MCA tasks, we compute mean accuracy, and for the NA tasks, we calculate relative accuracy across confidence thresholds  $\mathcal{C} = \{0.5, 0.55 \dots, 0.95\}$ . We report the final average score and individual metrics on eight task types of VSI-Bench, including: (1) configurational reasoning tasks (object counting, relative direction, absolute direction, and route planning), (2) measurement estimation tasks (object size, room size, and absolute distance), and (3) spatiotemporal reasoning tasks (appearance order). For Spatial-MLLM



Table 2: **Evaluation Results on ScanQA [38] and SQA3D [39].** We use the val set of ScanQA and test set of SQA3D for evaluation following common practice [22, 47, 25]. **Bold** and underline denote the best-performing and second-best-performing models in each model category, respectively.

Methods	ScanQA (val)					SQA3D (test)		Video-Input Only
	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	EM-1	EM-R1	
Task-Specific Models								
ScanQA [38]	<u>30.2</u>	<u>10.1</u>	13.1	33.3	<b>64.9</b>	<u>47.2</u>	-	✗
SQA3D [39]	<b>30.5</b>	<b>11.2</b>	<u>13.5</u>	<u>34.5</u>	-	46.6	-	✗
3D-Vista [71]	-	-	<b>13.9</b>	<b>35.7</b>	-	<b>48.5</b>	-	✗
3D/2.5D-Input Models								
3D-LLM [48]	39.3	12.0	14.5	35.7	69.4	-	-	✗
LL3DA [23]	-	13.5	15.9	37.3	76.8	-	-	✗
Chat-Scene [22]	<u>43.2</u>	14.3	18.0	41.6	87.7	<u>54.6</u>	<b>57.5</b>	✗
3D-LLaVA [21]	-	<b>17.1</b>	<u>18.4</u>	<u>43.1</u>	<u>92.6</u>	<u>54.5</u>	<u>56.6</u>	✗
Video-3D LLM [25]	<b>47.1</b>	<u>16.2</u>	<b>19.8</b>	<b>49.0</b>	<b>102.1</b>	<b>58.6</b>	-	✗
Video-Input Models								
Qwen2.5-VL-3B [14]	22.5	3.8	9.7	25.4	47.4	43.4	45.9	✓
Qwen2.5-VL-7B [14]	27.8	3.0	11.4	29.3	53.9	46.5	49.8	✓
Qwen2.5-VL-72B [14]	26.8	<u>12.0</u>	13.0	35.2	66.9	47.0	<u>50.9</u>	✓
LLaVA-Video-7B [12]	<u>39.7</u>	3.1	<u>17.7</u>	<u>44.6</u>	<u>88.7</u>	<u>48.5</u>	-	✓
Oryx-34B [53]	38.0	-	15.0	37.3	72.3	-	-	✓
<b>Spatial-MLLM-4B</b>	<b>44.4</b>	<b>14.8</b>	<b>18.4</b>	<b>45.0</b>	<b>91.8</b>	<b>55.9</b>	<b>58.7</b>	✓

and Qwen2.5-VL series, we report micro average scores in Table 1 and macro average scores in Table 5.

**Baseline Models.** We compare our model with a broad range of video-input MLLMs. For proprietary model, we include GPT-4o [5] and Gemini-1.5 Pro [4] for comparison. For open-source video-input MLLMs, we compare our model with InternVL2 [7], LongVILA [66], VILA [67], LongVA [68], LLaVA-NeXT-Video [12], LLaVA-OneVision [6], and the Qwen2.5-VL [14] series. The parameter count of the baseline models is reported in Table 1.

**Results.** We present the quantitative results on VSI-Bench [18] in Table 1 and Table 5. Despite having 4.9B parameters, Spatial-MLLM significantly outperforms all proprietary and open-source MLLMs, including those with substantially larger parameter counts (*e.g.*, 32B or 72B). Among the remaining models, the best-performing one is the proprietary Gemini-1.5 Pro [4]. Notably, Spatial-MLLM is provided with only 16 input frames per video, while Gemini-1.5 Pro [4] samples videos at 1 FPS (*i.e.*, an average of 85 frames per video on VSI-Bench) according to its API instructions [18]. Despite the significantly lower number of input frames, Spatial-MLLM still achieves higher average accuracy than Gemini-1.5 Pro [4].

### 4.3 Comparison on ScanQA and SQA3D

**Setup.** ScanQA [38] and SQA3D [39] are two 3D question-answering benchmarks built upon ScanNet [64]. Since the authors did not provide a test set for ScanQA, we evaluate it using the validation set, which consists of 4,675 QA pairs focused on understanding spatial relationships such as object alignment and orientation, as well as the ability to accurately identify objects in 3D scenes based on textual questions. We follow standard practice [25, 50] by evaluating answer quality using the following metrics: CIDEr, BLEU-1, BLEU-4, METEOR, and ROUGE-L. For SQA3D, we evaluate the model on its test set, which contains 3,519 QA pairs. The task requires the model to first understand its position and orientation within the 3D scene, as described by text, then reason about its environment and answer a question under those conditions. Since SQA3D contains definitive answers, we use exact match accuracy (EM) and its refined version (EM-R) as evaluation metrics. We provide the evaluation results using additional metrics for both benchmarks in Section C.2.

**Baselines.** Since both the ScanQA [38] and SQA3D [39] benchmarks provide additional 3D annotations (*e.g.*, point clouds and depth maps of the scene), we compare Spatial-MLLM with several other model types in addition to video-input MLLM. These includes task-specific models designed for 3D question-answering tasks, such as ScanQA [38], SQA3D [39], 3D-VisTA [71], and LLMs that require point clouds or depth maps as input, such as Chat-Scene [22], Video-3D LLM [25], and 3D-LLaVA [21].



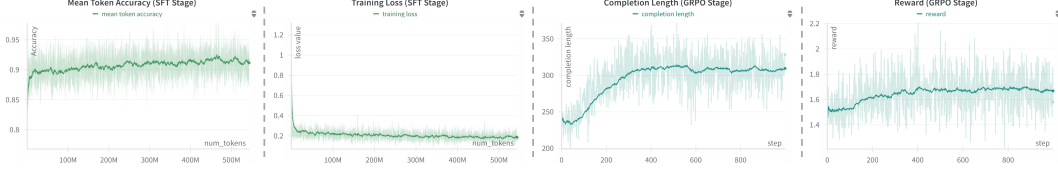


Figure 4: **Visualization of Training Curves in the SFT and RL Stages.** For the SFT stage, we present the mean token accuracy and loss curves. For the RL stage, we show the dynamics of completion length and reward.

**Results.** We present the quantitative results on the ScanQA [38] and SQA3D [39] benchmarks in Table 2. As shown, Spatial-MLLM significantly outperforms all video-input models across all metrics on both ScanQA and SQA3D. Our model also surpasses all task-specific models. Among models utilizing 3D or 2.5D input, only 3D-LLaVA [21] (on ScanQA) and Video-3D-LLM [25] (on ScanQA and SQA3D) achieve better performance than Spatial-MLLM. However, 3D-LLaVA requires additional point cloud input, and Video-3D-LLM depends on depth maps. Despite not relying on any additional 3D or 2.5D input, our model still outperforms other 3D-dependent models such as 3D-LLM [48], LL3DA [23], and Chat-Scene [22].

#### 4.4 Ablation Study and Analysis

**Ablation on Input Frame Number.** We evaluate the effect of the number of input frames on VSI-Bench across different models, including Spatial-MLLM, Gemini-1.5 Pro [4], and Qwen2.5-VL-3B [14]. The result is shown in Table 4. For the 1 fps setting of Gemini-1.5 Pro, we upload the entire video to the model following the VSI-Bench [18], where the video is sampled at 1 fps according to the API instructions. For the 0.1 fps and 0.25 fps settings, we first manually sample the video frames and then upload these sampled frames to the model. As shown, all models exhibit improved performance as the number of input frames increases, particularly when the number of frames is small.

**Effectiveness of Space-aware Frame Sampling.** We evaluate different frame sampling configurations in Table 4, including 8, 16, and 32 frames using uniform sampling and space-aware frame sampling. As shown, increasing the number of sampled frames improves performance for both space-aware frame sampling and uniform sampling. Compared with uniform sampling, space-aware frame sampling consistently outperforms it when the number of input frames is the same.

We further provide a visualization of our space-aware frame sampling in Fig. 5, which shows the point maps (predicted by the VGGT [32]) corresponding to the frames selected by different sampling strategies. As shown, the proposed space-aware frame sampling strategy consistently yields more spatial coverage than uniform sampling, which often overlooks transient regions that appear briefly in the video and tends to produce redundant viewpoints when the camera remains static.

##### Effectiveness of RL Training.

We evaluate Spatial-MLLM’s performance before and after GRPO training on VSI-Bench. The results are presented in the second (SFT + GRPO) and third (SFT) rows of Table 3. As shown, even though we conduct only small scale RL training (*i.e.*, 1,000 steps), the GRPO-trained model

Table 3: **Ablation Study.** We report micro average results for numerical questions and multiple-choice questions on VSI-Bench [18] in different settings.

Methods	Numerical	Multiple-Choice	Avg.
Spatial-MLLM	52.7	43.8	48.4
Spatial-MLLM (w/o sa sampling)	51.6	42.3	47.1
Spatial-MLLM (w/o sa sampling & GRPO)	51.5	40.4	46.1
Qwen2.5-VL-3B (SFT) <sup>‡</sup>	49.2	40.3	44.9
Qwen2.5-VL-3B (SFT) <sup>†</sup>	47.1	32.6	40.0
Qwen2.5-VL-3B	34.4	26.9	30.6

still achieves performance gains, suggesting that long chain-of-thought reasoning enhances the spatial reasoning capabilities required by VSI-Bench [18].

**Effectiveness of the Spatial-MLLM Architecture and Training Dataset.** We compare the supervised fine-tuned version of Spatial-MLLM, two supervised fine-tuned versions of Qwen2.5-VL-3B [14] (the base model of Spatial-MLLM) and original Qwen2.5-VL-3B model in Table 3. <sup>†</sup> denotes results obtained with the R1-V [72] training framework. <sup>‡</sup> denotes results which we further apply a question token mask during the loss computation process within R1-V [72], which aligns better with Spatial-MLLM training process. As shown 3, both SFT versions of Qwen2.5-VL-3B

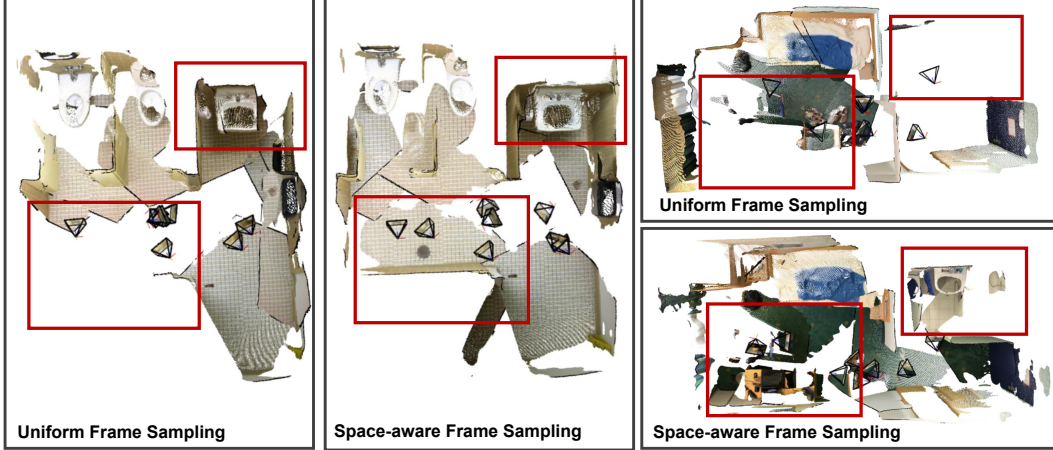


Figure 5: **Visualization of different frame sampling strategies.** For clarity of visualization, we set  $N_m = 128$  and  $N_k = 8$  in the visualization example.

Table 4: **Ablation on Input Frame Number.** We report micro average results on VSI-Bench [18] using different input frame numbers and frame rates (FPS). For Gemini-1.5 Pro, the input frame number is averaged over questions.

Methods	Frames	FPS	Numerical	Multiple-Choice	Avg.
Spatial-MLLM	8	N/A	50.8	41.2	46.1
	16	N/A	52.7	43.8	48.4
	32	N/A	53.1	45.3	49.3
Spatial-MLLM (w/o sa sampling)	8	N/A	48.2	39.2	43.8
	16	N/A	51.6	42.3	47.1
	32	N/A	52.4	44.2	48.4
Gemini-1.5 Pro [4]	12.2 (avg.)	0.1	43.1	35.7	39.5
	29.6 (avg.)	0.25	48.8	37.8	43.5
	117.1 (avg.)	1	49.7	44.0	46.9
Qwen2.5-VL-3B [14]	8	N/A	33.1	20.2	26.5
	16	N/A	34.4	26.9	30.6
	32	N/A	35.7	28.3	31.9

show improvements, indicating the effectiveness of our proposed dataset to enhance the model’s spatial reasoning capabilities. Furthermore, both models underperform compared to the supervised fine-tuned version of Spatial-MLLM, which validates the effectiveness of the proposed architecture.

## 5 Conclusion

We introduce Spatial-MLLM, a method that enables effective spatial understanding and reasoning from purely 2D visual inputs. By combining a semantic 2D encoder with a structure-aware spatial encoder initialized from a visual geometry foundation model, our dual-encoder design captures both semantic and spatial cues. Additionally, our proposed space-aware frame sampling strategy further enhances performance under limited input constraints. Trained on the collected dataset, our model achieves state-of-the-art results across multiple benchmarks.

**Limitations and Future Work.** Although Spatial-MLLM demonstrates significant improvements over previous video MLLMs across a wide range of visual-based spatial understanding and reasoning tasks, there remains room to scale Spatial-MLLM further in terms of model size and training data. Moreover, as this work primarily addresses visual-based spatial intelligence, we have trained and evaluated our model specifically on relevant datasets and benchmarks. An interesting direction for future work would be to explore how integrating spatial structural information might further benefit general video understanding and reasoning tasks.

## Acknowledgments

This work was supported in part by the Beijing Natural Science Foundation under Grant L252011, and by the National Natural Science Foundation of China under Grant 62206147.

## References

- [1] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, “Flamingo: a visual language model for few-shot learning,” *NeurIPS*, 2022.
- [2] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in *ICML*, 2023.
- [3] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *NeurIPS*, 2024.
- [4] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, *et al.*, “Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context,” *arXiv preprint arXiv:2403.05530*, 2024.
- [5] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, *et al.*, “Gpt-4o system card,” *arXiv preprint arXiv:2410.21276*, 2024.
- [6] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, Y. Li, Z. Liu, and C. Li, “Llava-onevision: Easy visual task transfer,” *arXiv preprint arXiv:2408.03326*, 2024.
- [7] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, *et al.*, “Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks,” in *CVPR*, 2024.
- [8] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, “Qwen-vl: A frontier large vision-language model with versatile abilities,” *arXiv preprint arXiv:2308.12966*, 2023.
- [9] B. Lin, B. Zhu, Y. Ye, M. Ning, P. Jin, and L. Yuan, “Video-llava: Learning united visual representation by alignment before projection,” *arXiv preprint arXiv:2311.10122*, 2023.
- [10] Z. Cheng, S. Leng, H. Zhang, Y. Xin, X. Li, G. Chen, Y. Zhu, W. Zhang, Z. Luo, D. Zhao, *et al.*, “Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms,” *arXiv preprint arXiv:2406.07476*, 2024.
- [11] R. Qian, X. Dong, P. Zhang, Y. Zang, S. Ding, D. Lin, and J. Wang, “Streaming long video understanding with large language models,” *arXiv preprint arXiv:2405.16009*, 2024.
- [12] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li, “Video instruction tuning with synthetic data,” *ArXiv*, vol. abs/2410.02713, 2024.
- [13] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K.-Y. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, “Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution,” *ArXiv*, vol. abs/2409.12191, 2024.
- [14] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, “Qwen2.5-vl technical report,” *ArXiv*, vol. abs/2502.13923, 2025.
- [15] R. Huang, M. Li, D. Yang, J. Shi, X. Chang, Z. Ye, Y. Wu, Z. Hong, J.-B. Huang, J. Liu, Y. Ren, Z. Zhao, and S. Watanabe, “AudioGPT: Understanding and generating speech, music, sound, and talking head,” *ArXiv*, vol. abs/2304.12995, 2023.
- [16] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang, “Salmonn: Towards generic hearing abilities for large language models,” *ArXiv*, vol. abs/2310.13289, 2023.
- [17] Z. Liu, Y. Dong, J. Wang, Z. Liu, W. Hu, J. Lu, and Y. Rao, “Ola: Pushing the frontiers of omni-modal language model with progressive modality alignment,” *ArXiv*, vol. abs/2502.04328, 2025.
- [18] J. Yang, S. Yang, A. W. Gupta, R. Han, F.-F. Li, and S. Xie, “Thinking in space: How multimodal large language models see, remember, and recall spaces,” *ArXiv*, vol. abs/2412.14171, 2024.
- [19] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Driess, P. Florence, D. Sadigh, L. J. Guibas, and F. Xia, “SpatialVLM: Endowing vision-language models with spatial reasoning capabilities,” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14455–14465, 2024.

- [20] Y. Li, Y. Zhang, T. Lin, X. Liu, W. Cai, Z. Liu, and B. Zhao, “Sti-bench: Are mllms ready for precise spatial-temporal world understanding?,” *ArXiv*, vol. abs/2503.23765, 2025.
- [21] J. Deng, T. He, L. Jiang, T. Wang, F. Dayoub, and I. Reid, “3d-llava: Towards generalist 3d llms with omni superpoint transformer,” *ArXiv*, vol. abs/2501.01163, 2025.
- [22] H. Huang, Y. Chen, Z. Wang, R. Huang, R. Xu, T. Wang, L. Liu, X. Cheng, Y. Zhao, J. Pang, *et al.*, “Chat-scene: Bridging 3d scene and large language models with object identifiers,” in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [23] S. Chen, X. Chen, C. Zhang, M. Li, G. Yu, H. Fei, H. Zhu, J. Fan, and T. Chen, “LI3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26428–26438, 2024.
- [24] C. Zhu, T. Wang, W. Zhang, J. Pang, and X. Liu, “Llava-3d: A simple yet effective pathway to empowering llms with 3d-awareness,” *arXiv preprint arXiv:2409.18125*, 2024.
- [25] D. Zheng, S. Huang, and L. Wang, “Video-3d llm: Learning position-aware video representation for 3d scene understanding,” *ArXiv*, vol. abs/2412.00493, 2024.
- [26] S. Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortsman, D. Ghosh, J. Zhang, E. Orgad, R. Entezari, G. Daras, S. Pratt, V. Ramanujan, Y. Bitton, K. Marathe, S. Mussmann, R. Vencu, M. Cherti, R. Krishna, P. W. Koh, O. Saukh, A. J. Ratner, S. Song, H. Hajishirzi, A. Farhadi, R. Beaumont, S. Oh, A. G. Dimakis, J. Jitsev, Y. Carmon, V. Shankar, and L. Schmidt, “Datacomp: In search of the next generation of multimodal datasets,” *ArXiv*, vol. abs/2304.14108, 2023.
- [27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning*, 2021.
- [28] Z. Tang, L. Lian, S. Eisape, X. Wang, R. Herzig, A. Yala, A. Suhr, T. Darrell, and D. M. Chan, “Tulip: Towards unified language-image pretraining,” *ArXiv*, vol. abs/2503.15485, 2025.
- [29] J. Qi, J. Liu, H. Tang, and Z. Zhu, “Beyond semantics: Rediscovering spatial awareness in vision-language models,” *ArXiv*, vol. abs/2503.17349, 2025.
- [30] B. Zhang, P. Zhang, X. wen Dong, Y. Zang, and J. Wang, “Long-clip: Unlocking the long-text capability of clip,” in *European Conference on Computer Vision*, 2024.
- [31] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, “Dust3r: Geometric 3d vision made easy,” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, 2023.
- [32] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, “Vggt: Visual geometry grounded transformer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.
- [33] Z. Li, R. Tucker, F. Cole, Q. Wang, L. Jin, V. Ye, A. Kanazawa, A. Holynski, and N. Snavely, “Megsam: Accurate, fast, and robust structure and motion from casual dynamic videos,” *ArXiv*, vol. abs/2412.04463, 2024.
- [34] Y. Wang, J. Zhou, H. Zhu, W. Chang, Y. Zhou, Z. Li, J. Chen, J. Pang, C. Shen, and T. He, “ $\pi^3$ : Scalable permutation-equivariant visual geometry learning,” 2025.
- [35] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J.-M. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B.-L. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D.-L. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S.-K. Wu, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W.-X. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X.-C. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y.-J. Zou, Y. He, Y. Xiong, Y.-W. Luo, Y. mei You, Y. Liu,

- Y. Zhou, Y. X. Zhu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. guo Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z.-A. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *ArXiv*, vol. abs/2501.12948, 2025.
- [36] Z. Shao, P. Wang, Q. Zhu, R. Xu, J.-M. Song, M. Zhang, Y. K. Li, Y. Wu, and D. Guo, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *ArXiv*, vol. abs/2402.03300, 2024.
- [37] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H. Chi, F. Xia, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," *ArXiv*, vol. abs/2201.11903, 2022.
- [38] D. Azuma, T. Miyanishi, S. Kurita, and M. Kawanabe, "Scanqa: 3d question answering for spatial scene understanding," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19107–19117, 2021.
- [39] X. Ma, S. Yong, Z. Zheng, Q. Li, Y. Liang, S.-C. Zhu, and S. Huang, "Sqa3d: Situated question answering in 3d scenes," *ArXiv*, vol. abs/2210.07474, 2022.
- [40] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.
- [41] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.
- [42] Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, and D. Cai, "Pandagpt: One model to instruction-follow them all," *arXiv preprint arXiv:2305.16355*, 2023.
- [43] R. Pi, J. Gao, S. Diao, R. Pan, H. Dong, J. Zhang, L. Yao, J. Han, H. Xu, L. Kong, *et al.*, "Detgpt: Detect what you need via reasoning," *arXiv preprint arXiv:2305.14167*, 2023.
- [44] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao, "Videochat: Chat-centric video understanding," *arXiv preprint arXiv:2305.06355*, 2023.
- [45] Y. Chen, S. Yang, H. Huang, T. Wang, R. Xu, R. Lyu, D. Lin, and J. Pang, "Grounded 3d-llm with referent tokens," *arXiv preprint arXiv:2405.10370*, 2024.
- [46] Z. Wang, H. Huang, Y. Zhao, Z. Zhang, and Z. Zhao, "Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes," *arXiv preprint arXiv:2308.08769*, 2023.
- [47] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, "An embodied generalist agent in 3d world," *arXiv preprint arXiv:2311.12871*, 2023.
- [48] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan, "3d-llm: Injecting the 3d world into large language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 20482–20494, 2023.
- [49] R. Fu, J. Liu, X. Chen, Y. Nie, and W. Xiong, "Scene-llm: Extending language model for 3d visual understanding and reasoning," *arXiv preprint arXiv:2403.11401*, 2024.
- [50] Z. Qi, Z. Zhang, Y. Fang, J. Wang, and H. Zhao, "Gpt4scene: Understand 3d scenes from videos with vision-language models," *arXiv preprint arXiv:2501.01428*, 2025.
- [51] Z. Fan, J. Zhang, R. Li, J. Zhang, R. Chen, H. Hu, K. Wang, H. Qu, D. Wang, Z. Yan, *et al.*, "Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction," *arXiv preprint arXiv:2505.20279*, 2025.
- [52] K. Ouyang, Y. Liu, H. Wu, Y. Liu, H. Zhou, J. Zhou, F. Meng, and X. Sun, "Spacer: Reinforcing mllms in video spatial reasoning," *arXiv preprint arXiv:2504.01805*, 2025.
- [53] Z. Liu, Y. Dong, Z. Liu, W. Hu, J. Lu, and Y. Rao, "Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution," *arXiv preprint arXiv:2409.12961*, 2024.
- [54] X. Wang, Y. Zhang, O. Zohar, and S. Yeung-Levy, "Videoagent: Long-form video understanding with large language model as agent," in *European Conference on Computer Vision*, pp. 58–76, Springer, 2024.
- [55] P. Wu, Y. Liu, M. Liu, and J. Shen, "St-think: How multimodal large language models reason about 4d worlds from ego-centric videos," *arXiv preprint arXiv:2503.12542*, 2025.

- [56] S. Zhou, A. Vilesov, X. He, Z. Wan, S. Zhang, A. Nagachandra, D. Chang, D. Chen, E. X. Wang, and A. Kadambi, “Vlm4d: Towards spatiotemporal awareness in vision language models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.
- [57] X. Linghu, J. Huang, X. Niu, X. S. Ma, B. Jia, and S. Huang, “Multi-modal situated reasoning in 3d scenes,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 140903–140936, 2024.
- [58] B. Yin, Q. Wang, P. Zhang, J. Zhang, K. Wang, Z. Wang, J. Zhang, K. Chandrasegaran, H. Liu, R. Krishna, et al., “Spatial mental modeling from limited views,” *arXiv preprint arXiv:2506.21458*, 2025.
- [59] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Neural Information Processing Systems*, 2017.
- [60] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, “Vision transformers need registers,” *ArXiv*, vol. abs/2309.16588, 2023.
- [61] X. Pan, S. N. Shukla, A. Singh, Z. Zhao, S. K. Mishra, J. Wang, Z. Xu, J. Chen, K. Li, F. Juefei-Xu, J. Hou, and S. Xie, “Transfer between modalities with metaqueries,” 2025.
- [62] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions—i,” *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
- [63] D. S. Hochbaum, *Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems*, p. 94–143. USA: PWS Publishing Co., 1996.
- [64] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, “Scannet: Richly-annotated 3d reconstructions of indoor scenes,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2432–2443, 2017.
- [65] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.
- [66] F. Xue, Y. Chen, D. Li, Q. Hu, L. Zhu, X. Li, Y. Fang, H. Tang, S. Yang, Z. Liu, E. He, H. Yin, P. Molchanov, J. Kautz, L. Fan, Y. Zhu, Y. Lu, and S. Han, “Longvila: Scaling long-context visual language models for long videos,” *ArXiv*, vol. abs/2408.10188, 2024.
- [67] J. Lin, H. Yin, W. Ping, Y. Lu, P. Molchanov, A. Tao, H. Mao, J. Kautz, M. Shoybi, and S. Han, “Vila: On pre-training for visual language models,” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26679–26689, 2023.
- [68] P. Zhang, K. Zhang, B. Li, G. Zeng, J. Yang, Y. Zhang, Z. Wang, H. Tan, C. Li, and Z. Liu, “Long context transfer from language to vision,” *ArXiv*, vol. abs/2406.16852, 2024.
- [69] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, “Scannet++: A high-fidelity dataset of 3d indoor scenes,” *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12–22, 2023.
- [70] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. Schwartz, and E. Shulman, “ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data,” in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.
- [71] Z. Zhu, X. Ma, Y. Chen, Z. Deng, S. Huang, and Q. Li, “3d-vista: Pre-trained transformer for 3d vision and text alignment,” *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2899–2909, 2023.
- [72] L. Chen, L. Li, H. Zhao, Y. Song, and Vinci, “R1-v: Reinforcing super generalization ability in vision-language models with less than \$3.” <https://github.com/Deep-Agent/R1-V>, 2025. Accessed: 2025-02-02.
- [73] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3d: A modern library for 3d data processing,” *ArXiv*, vol. abs/1801.09847, 2018.
- [74] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgb-d images,” in *European Conference on Computer Vision*, 2012.
- [75] S. Gupta, P. Arbeláez, and J. Malik, “Perceptual organization and recognition of indoor scenes from rgb-d images,” *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 564–571, 2013.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

### 1. Claims

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

Answer: [Yes]

Justification: The main claims made in the abstract and introduction (Sec. 1) accurately reflect the paper’s contributions and scope.

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 discuss possible limitations of our method in Sec. 5.

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: The 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: Please refer to our method (Sec. 3) and implementation details (Sec. 4) to reproduce our work step by step.

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: [No]

Justification: We will release the data and code upon acceptance.

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 specify all the details in Sec. 4.

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: We follow conventions observed in certain prior studies, and they don't include error bars in their experiments. To ensure a fair comparison, we similarly have not reported error bars in our paper.

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: We provide sufficient information on the computer resources in the Appendix.

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: The research conducted in the paper conforms 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 provide Broader impacts in the Appendix.

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: The paper poses no such risks.

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 code bases and datasets used in our work are open-source, and we strictly follow the licenses or terms of use.

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](https://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: [\[NA\]](#)

Justification: Our paper does not release new assets.

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: The paper does not involve crowdsourcing nor 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: [Yes]

Justification: Our experiment poses no potential risks to study participants. We also adhere to the NeurIPS Code of Ethics and the guidelines for our institution.

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 rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard 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.

## Technical Appendices and Supplementary Material

### A Broader Impacts

This work advances spatial reasoning in multimodal large language models by enabling 3D understanding purely from 2D visual inputs. Such capability may broaden the accessibility of spatially aware AI in domains such as robotics, autonomous systems, and visual content understanding, without requiring costly 3D data. As with other vision-language models, considerations of data privacy and ethical deployment remain important to ensure positive social outcomes.

### B Additional Method Details

#### B.1 Details of Space-Aware Frame Sampling

Our space-aware frame sampling algorithm consists of three stages: (1) Scene geometry preprocessing, (2) Voxelization and coverage calculation, and (3) Greedy maximum coverage selection. Beginning with the original video sequence  $\mathcal{V} = \{\mathbf{f}_i\}_{i=1}^N$ , we first perform uniform subsampling to obtain  $N_m = 128$  candidate frames  $\{\mathbf{f}_i^m\}_{i=1}^{N_m}$ . For each subsampled frame, we leverage the backbone and head of VGGT [32] to compute  $\{\mathbf{E}_i^m, \mathbf{K}_i^m\}_{i=1}^{N_m}$  and  $\{\mathbf{D}_i^m\}_{i=1}^{N_m}$  as illustrated in the main paper. Then we reconstruct 3D point maps  $\mathcal{P}_i^m$  through depth reprojection:

$$\mathcal{P}_i^m = \mathbf{D}_i^m \cdot \mathbf{K}_i^{-1}[\mathbf{u}|\mathbf{v}|1]^\top \cdot \mathbf{E}_i^{-1}, \quad (8)$$

where  $(\mathbf{u}, \mathbf{v})$  denote pixel coordinates. In practice, we also obtain a confidence value  $c(p) \in [0, 1]$  for each point  $p \in \mathcal{P}_i^m$  from the depth head. Although VGGT [32] can also directly decode point maps from 3D dense features, we find that using depth and camera produces more accurate results.

The voxelization and coverage calculation process first establishes a 3D bounding box encompassing all valid scene points:

$$\mathcal{P}_{\text{valid}} = \bigcup_{i=1}^{N_m} \{p \in \mathcal{P}_i^m \mid c(p) > 0.1 \wedge c(p) \geq \text{Percentile}(\{c(p)\}, 50\%)\}. \quad (9)$$

We then discretize the bounding box into voxels. To handle relative scales in VGGT [32] outputs, we use an adaptive way to set the voxel size  $\Delta$  to  $\frac{1}{\lambda}$  of the minimum dimension of the scene’s bounding box:

$$\Delta = \frac{1}{\lambda} \cdot \min(\max(\mathcal{P}_{\text{valid}}) - \min(\mathcal{P}_{\text{valid}})), \quad (10)$$

where  $\lambda$  is a hyperparameter and we set it to 20. Each frame’s voxel coverage  $V(\mathbf{f}_i^m)$  is then calculated by discretizing its valid points:

$$V(\mathbf{f}_i^m) = \left\{ \left\lfloor \frac{p - \min(\mathcal{P}_{\text{valid}})}{\Delta} \right\rfloor \mid p \in \mathcal{P}_i^m \cap \mathcal{P}_{\text{valid}} \right\}. \quad (11)$$

Finally, we can formulate frame selection as the typical maximum coverage problem [62]:

$$\max_{\mathcal{S} \subseteq \{1, \dots, N_m\}} \left| \bigcup_{i \in \mathcal{S}} V(\mathbf{f}_i^m) \right| \quad \text{s.t.} \quad |\mathcal{S}| = N_k, \quad (12)$$

In practice, we set  $N_k = 16$  and use a greedy approach [63, 25] to iteratively select the frame that provides the maximum new coverage, which is illustrated in Algorithm 1.

#### B.2 Details of Feature Fusion

Both the 2D and 3D encoders use a spatial patch size of 14. The 2D encoder further reduces the token sequence length by merging tokens spatially ( $2 \times 2$  adjacent tokens) and temporally (every 2 consecutive frames). As a result, the 2D encoder outputs exactly one-eighth the number of tokens compared to the 3D encoder (we exclude register and camera tokens). To align these tokens, we first apply the same spatial-temporal merging strategy as used in the 2D encoder. After merging, we rearrange the tokens into sequence, ensuring the two sets of tokens are precisely aligned in both position and number. Then we project both tokens into language model’s hidden dimension with a two-layer MLP and fuse them by element-wise addition.



---

**Algorithm 1** Greedy Maximum Coverage Sampling

---

**Input** Frame voxel sets  $\{V(\mathbf{f}_i^m)\}_{i=1}^{N_m}$ , target selection size  $N_k$

**Output** Selected frame indices  $\mathcal{S} \subseteq \{1, \dots, N_m\}$

```
1:  $\mathcal{S} \leftarrow \emptyset$  ▷ Selected frames
2:  $\mathcal{C} \leftarrow \emptyset$  ▷ Covered voxels
3:  $\mathcal{R} \leftarrow \{1, \dots, N_m\}$  ▷ Remaining candidates
4: for  $t \leftarrow 1$  to  $N_k$  do
5:   if  $\mathcal{R} = \emptyset$  then
6:     break ▷ No remaining candidates
7:   end if
8:    $i^* \leftarrow \underset{i \in \mathcal{R}}{\operatorname{argmax}} |V(\mathbf{f}_i^m) \setminus \mathcal{C}|$  ▷ Max coverage gain
9:   if  $|V(\mathbf{f}_{i^*}^m) \setminus \mathcal{C}| = 0$  then
10:    break ▷ No additional coverage
11:  end if
12:   $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}$  ▷ Update selection
13:   $\mathcal{C} \leftarrow \mathcal{C} \cup V(\mathbf{f}_{i^*}^m)$  ▷ Update covered voxels
14:   $\mathcal{R} \leftarrow \mathcal{R} \setminus \{i^*\}$  ▷ Remove from candidates
15: end for
16: return  $\mathcal{S}$ 
```

---

### B.3 Details of Dataset Construction

We follow a similar approach to that used in [18] to construct the self-created part of training dataset. Specifically, the construction involves three main processes: video preprocessing, metadata computation, and QA pair generation.

**Video Preprocessing.** In this stage, we extract frames from the raw ScanNet [64] scans and convert them into videos at 24 FPS with a resolution of  $640 \times 480$ .

**Metadata Computation.** In this stage, we extract spatial and semantic metadata from raw ScanNet scans and their associated semantic annotations. First, we align each raw scene mesh using the provided axis alignment matrices and convert it to the Open3D [73] point cloud. At the room level, we compute the room size using the alpha-shape algorithm and determine the center coordinates. At the object level, we generate oriented bounding boxes (OBBs) for each valid object instance and assign semantic labels from the annotations, excluding structural elements (*e.g.*, *walls*, *floors*) and ambiguous categories (*e.g.*, *otherstructure*). To ensure consistency across categories, we remap the original ScanNet semantic labels to a new label set based on the NYU40 classes [74, 75] (which we manually add and remove some categories to align with VSIBench [18]). In addition, we collect the projected 2D semantic annotation of each scene video for the appearance order task. The final metadata for each scene includes: (1) room size and center coordinates; (2) the projected 2D semantic annotation of the scene video; (3) object instances and their OBB parameters, including rotation matrices, extents, and centers; and (4) semantic labels for each object.

**QA Pair Generation.** Finally, we generate QA pairs of different tasks, including object counting, object size, room size, absolute distance, appearance order, relative distance, and relative direction.

- *Object counting (numerical)*: We first count how many times each object category appears in the scene, then randomly sample a category that appears at least twice. Question template: “How many <category>(s) are in this room?”
- *Object size (numerical)*: We randomly sample a unique object in the scene and take the longest side of its oriented bounding box (OBB) as the ground-truth length (in cm). Question template: “What is the length of the longest dimension (length, width, or height) of the <category>, measured in centimeters?”
- *Room size (numerical)*: We use the pre-computed room size (in  $\text{m}^2$ ) as the ground-truth value. Question template: “What is the size of this room (in square meters)?”
- *Absolute distance (numerical)*: For a pair of objects, we uniformly sample points inside each OBB and take the minimum Euclidean distance between the two point clouds as the ground-truth

(in m). Question template: “Measuring from the closest point of each object, what is the direct distance between the  $\langle \text{category\_A} \rangle$  and the  $\langle \text{category\_B} \rangle$  (in meters)?”

- *Appearance Order (multiple choice)*: We calculate the first appearance timestamp of each category, which is the timestamp when its visible pixel count exceeds a predefined threshold. Using these timestamps, we generate the correct order of appearance among the categories, along with other options. Question template: What will be the first-time appearance order of the following categories in the video:  $\langle \text{category\_A} \rangle$ ,  $\langle \text{category\_B} \rangle$ ,  $\langle \text{category\_C} \rangle$ ,  $\langle \text{category\_D} \rangle$
- *Relative distance (multiple choice)*: We use an “anchor” object that is unique in the scene and then select four additional objects while enforcing 15-30cm separation thresholds between options. Question template: “Which of these objects ( $\langle \text{category\_A} \rangle$ ,  $\langle \text{category\_B} \rangle$ ,  $\langle \text{category\_C} \rangle$ ,  $\langle \text{category\_D} \rangle$ ) is closest to the  $\langle \text{anchor\_category} \rangle$ ?”
- *Relative direction (multiple choice)*: For triple  $\{\text{position}, \text{facing}, \text{query}\}$  of unique categories, we compute the horizontal angle between the vectors  $\text{position} \rightarrow \text{facing}$  and  $\text{position} \rightarrow \text{query}$ . The angle is then discretized into directional classes (easy: left/right, medium: left/right/back, hard: front-left/front-right/back-left/back-right). Question template (easy example): “If I am standing by the  $\langle \text{position\_category} \rangle$  and facing the  $\langle \text{facing\_category} \rangle$ , is the  $\langle \text{query\_category} \rangle$  to the left or the right?”

#### B.4 Details of Cold Start

To align the model with the desired reasoning format, we perform a simple cold start for 200 steps before GRPO training. The key to this stage is the construction of a spatial reasoning dataset with chain-of-thought (CoT) annotations. The construction process is as follows:

**Subset Sampling.** We begin by sampling a subset  $\mathcal{D}_0 = \{\mathcal{I}_i\}_{i=1}^{N_s} = \{\langle \mathcal{Q}_i, \mathcal{A}_i, \mathcal{V}_i, \mathcal{M}_i \rangle\}_{i=1}^{N_s}$  from our training dataset.

**Multi-path CoT Generation.** For each item  $\mathcal{I}_i \in \mathcal{D}_0$ , we utilize Qwen2.5-VL-72B [14] to generate  $K$  independent reasoning processes  $\hat{\mathcal{T}}_i^{(k)}$  and corresponding answers  $\hat{\mathcal{A}}_i^{(k)}$ . We then compute a reward  $r_i^{(k)} = \text{Reward}(\hat{\mathcal{A}}_i^{(k)}, \mathcal{A}_i)$  for each reasoning-answer pair, where  $\text{Reward}(\cdot, \cdot)$  is the reward function described in Sec B.5. Consequently, we obtain a set of outputs  $\mathcal{O}_i = \{(\hat{\mathcal{T}}_i^{(k)}, \hat{\mathcal{A}}_i^{(k)}, r_i^{(k)})\}_{k=1}^K$  for each  $\mathcal{I}_i \in \mathcal{D}_0$ .

**Adaptive Filtering.** Since Qwen2.5-VL-72B [14] may generate incorrect reasoning processes and answers, we apply a filtering process based on the computed rewards. While using a global reward threshold is straightforward, it often results in an imbalance across question types in the selected subset. To mitigate this, we adopt an adaptive filtering strategy. Specifically, for each item  $\mathcal{I}_i \in \mathcal{D}_0$ , we first keep the output with the highest reward to get  $\hat{\mathcal{O}}_i = \{(\hat{\mathcal{T}}_i^{(k^*)}, \hat{\mathcal{A}}_i^{(k^*)}, r_i^{(k^*)})\}$  where  $k^* = \arg \max_k r_i^{(k)}$ . Let  $\hat{r}_i = r_i^{(k^*)}$  denote the maximum reward. We then categorize all items based on their question type and compute a question type-dependent threshold  $\tau_{t(i)}$ , where  $t(i)$  denotes the type of problem  $i$ . The item is added into the cold start set if and only if:

$$\hat{r}_i \geq \tau_{t(i)} \quad \text{and} \quad \hat{r}_i > 0,$$

where the type-dependent threshold satisfies  $\tau_{t(i)} := \text{Quantile}(\{\hat{r}_j \mid t(j) = t(i)\}, 0.5)$ . This rule preserves approximately the top 50% of generations per question type while discarding degenerate (zero-reward) outputs. In practice, we set  $N_s = 5000$  and  $K = 3$ , and finally we get 2459 items in the cold start set. We provide a pseudocode for this process in Algorithm 2:

#### B.5 Details of SFT and GRPO Training

**Reward Calculation.** Given predicted answer  $\mathcal{A}_{\text{pred}}$  and ground truth answer  $\mathcal{A}_{\text{gt}}$ , the reward function  $\text{Reward}(\mathcal{A}_{\text{pred}}, \mathcal{A}_{\text{gt}})$  consists of a format reward  $\mathcal{R}_{\text{fmt}}$  and a task-specific reward:

$$\text{Reward}(\mathcal{A}_{\text{pred}}, \mathcal{A}_{\text{gt}}) = \lambda_1 R_{\text{format}} + \lambda_2 \begin{cases} \mathcal{R}_{\text{MC}}, & \text{multiple-choice} \\ \mathcal{R}_{\text{MRA}}, & \text{numerical} \\ \mathcal{R}_{\text{Verbal}}, & \text{verbal} \end{cases} \quad (13)$$

---

**Algorithm 2** Cold Start Dataset Construction

---

**Input** Original dataset  $\mathcal{D}$

- 1: Qwen2.5-VL model  $M$
- 2: Reward function  $\text{Reward}(\cdot, \cdot)$
- 3: Sample size  $N_s$ , Paths per item  $K$

**Output** Filtered dataset  $\mathcal{D}_{\text{cold}}$

- 4: Initialize  $\mathcal{D}_0 \leftarrow \text{Sampling}(\mathcal{D}, N_s)$
  - 5:  $\mathcal{D}_{\text{cold}} \leftarrow \emptyset$
  - 6: **for** each item  $\mathcal{I}_i = \langle \mathcal{Q}_i, \mathcal{A}_i, \mathcal{V}_i, \mathcal{M}_i \rangle \in \mathcal{D}_0$  **do**
  - 7:   Generate  $K$  reasoning paths:  $\{\hat{\mathcal{T}}_i^{(k)}\}_{k=1}^K \leftarrow M(\mathcal{Q}_i, \mathcal{V}_i)$
  - 8:   Compute rewards:  $r_i^{(k)} \leftarrow \text{Reward}(\hat{\mathcal{A}}_i^{(k)}, \mathcal{A}_i), \forall k$
  - 9:   Select best path:  $k^* \leftarrow \arg \max_k r_i^{(k)}$
  - 10:   Record  $\hat{r}_i \leftarrow r_i^{(k^*)}, \hat{\mathcal{O}}_i \leftarrow (\hat{\mathcal{T}}_i^{(k^*)}, \hat{\mathcal{A}}_i^{(k^*)})$
  - 11: **end for**
  - 12: Group items by type:  $\{\mathcal{G}_t\} \leftarrow \text{GroupByType}(\{\hat{r}_i\})$
  - 13: **for** each question type  $t$  **do**
  - 14:   Compute threshold:  $\tau_t \leftarrow \text{Quantile}(\{\hat{r}_j | j \in \mathcal{G}_t\}, 0.5)$
  - 15: **end for**
  - 16: **for** each item  $\mathcal{I}_i \in \mathcal{D}_0$  **do**
  - 17:   **if**  $\hat{r}_i \geq \tau_{t(i)}$  **and**  $\hat{r}_i > 0$  **then**
  - 18:      $\mathcal{D}_{\text{cold}} \leftarrow \mathcal{D}_{\text{cold}} \cup \{\hat{\mathcal{O}}_i\}$
  - 19:   **end if**
  - 20: **end for**
  - 21: **return**  $\mathcal{D}_{\text{cold}}$
- 

where  $\lambda_1$  and  $\lambda_2$  are hyperparameters, both of which are set to 1 in our implementation. For *multiple-choice questions*, we implement exact match criterion:

$$\text{R}_{\text{MC}}(\mathcal{A}_{\text{pred}}, \mathcal{A}_{\text{gt}}) = \mathbb{I}(\psi(\mathcal{A}_{\text{pred}}) = \psi(\mathcal{A}_{\text{gt}})) \quad (14)$$

where  $\psi(\cdot)$  performs answer normalization through whitespace stripping and  $\mathbb{I}(\cdot)$  denotes the indicator function. For *numerical tasks*, we compute mean relative accuracy (MRA) [18]:

$$\text{R}_{\text{MRA}}(\mathcal{A}_{\text{pred}}, \mathcal{A}_{\text{gt}}) = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \mathbb{I}\left(\frac{|\alpha(\mathcal{A}_{\text{pred}}) - \alpha(\mathcal{A}_{\text{gt}})|}{|\alpha(\mathcal{A}_{\text{gt}})| + \epsilon} < \tau\right) \quad (15)$$

where  $\alpha(\cdot)$  normalizes numeric values,  $\epsilon = 10^{-8}$  prevents division by zero, and  $\mathcal{T} = \{0.50, 0.55, \dots, 0.95\}$  defines accuracy thresholds. For *verbal answer questions*, we compute a normalized similarity score using the Levenshtein ratio:

$$\text{R}_{\text{Verbal}}(\mathcal{A}_{\text{pred}}, \mathcal{A}_{\text{gt}}) = 1 - \frac{D_{\text{Lev}}(\phi(\mathcal{A}_{\text{pred}}), \phi(\mathcal{A}_{\text{gt}}))}{|\phi(\mathcal{A}_{\text{pred}})| + |\phi(\mathcal{A}_{\text{gt}})|} \quad (16)$$

where  $D_{\text{Lev}}$  denotes the Levenshtein edit distance, and  $\phi(\cdot)$  represents the text normalization function. In practice, we use the implementation provided by the *Levenshtein* library. In addition to the format and task-specific rewards, we also incorporate a reasoning length reward following Video-R1 [12], which encourages the model to perform more thinking before generating the final answer.

**Other Details.** Figure 6 presents the prompts used in the SFT and GRPO stages. For both stages, we adopt the default system prompt of Qwen2.5-VL [14], namely, "You are a helpful assistant." In the SFT stage, the user prompt consists of a question and a type template. In the GRPO stage, the user prompt comprises a question, a question post string, and a type template. We conduct all experiments on Intel(R) Xeon(R) Gold 6430 platform with 80G NVIDIA A800 GPUs.

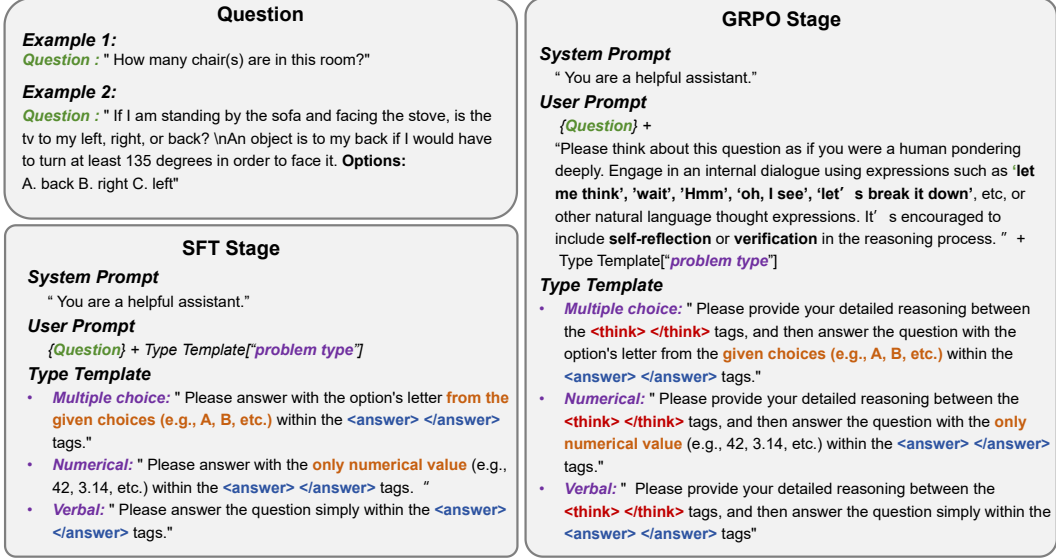


Figure 6: **Illustration of the prompts used in the SFT and GRPO stages.** We use the default system prompt of Qwen2.5-VL [14] (*i.e.*, "You are a helpful assistant") for both stages. In the SFT stage, the user prompt consists of a question and a type template. In the GRPO stage, the user prompt includes a question, a question post string, and a type template.

Table 5: Macro average scores on VSI-Bench [18] for Qwen2.5-VL [14] series and Spatial-MLLM.

Methods	Numerical Question				Multiple-Choice Question				Avg.
	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order	
Qwen2.5-VL-3B [14]	24.3	24.7	31.7	22.6	38.3	42.6	26.3	21.2	29.0
Qwen2.5-VL-7B [14]	40.9	14.8	43.4	10.7	38.6	40.1	33.0	29.8	31.4
Qwen2.5-VL-72B [14]	25.1	<u>29.3</u>	54.5	<u>38.8</u>	38.2	39.3	<u>34.0</u>	28.9	34.3
<b>Spatial-MLLM-4B</b>	<b>65.3</b>	<b>34.8</b>	<b>63.1</b>	<b>45.1</b>	41.3	<b>46.9</b>	33.5	<u>46.3</u>	<b>47.3</b>

## C Additional Experiments

### C.1 Additional Results on VSI-Bench

We present qualitative examples of Spatial-MLLM on the VSI-Bench [18] dataset in Figures 7 to 10. As illustrated, Spatial-MLLM is capable of reasoning with visual information across different task types and producing final answers accordingly. Furthermore, it demonstrates strong abilities in self-verification and task decomposition during the reasoning process.

### C.2 Additional Results on ScanQA and SQA3D

We present additional evaluation results on the ScanQA [38] and SQA3D [39] benchmarks in Table 6 and Table 7. As shown, our proposed method consistently outperforms all video-input models, including LLaVA-Video-7B [12] and Oryx-34B [53], both of which incorporate spatial reasoning datasets such as ScanQA [38] during training.

Despite having only 4.2 billion parameters, Spatial-MLLM significantly surpasses Qwen2.5-VL-72B [14] on the ScanQA benchmark, achieving substantial gains across multiple metrics—for instance, +2.3 EM-1, +17.6 BLEU-1, and +24.9 CIDEr. Similarly, on the SQA3D benchmark, Spatial-MLLM consistently outperforms Qwen2.5-VL-72B across all question types and overall performance, including improvements of +4.2 EM-1 and +7.8 EM-R1, with notable gains in the *Is* (+15.3) and *Which* (+13.9) categories.

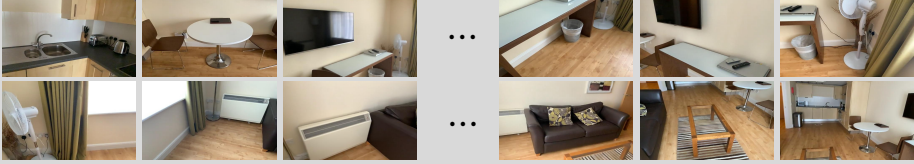
Table 6: **Additional evaluation results on ScanQA [38] for task-specific models, 3D/2.5D input models, and video-input models.** Reported metrics include EM-1, BLEU-1 to BLEU-4, ROUGE-L, METEOR, and CIDEr.

Methods	ScanQA (val)							
	EM-1	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	METEOR	CIDEr
<i>Task-Specific Models</i>								
ScanQA [38]	21.1	30.2	20.4	15.1	10.1	33.3	13.1	64.9
3D-Vista [71]	22.4	-	-	-	10.4	35.7	13.9	69.6
<i>3D/2.5D-Input Models</i>								
3D-LLM [48]	20.5	39.3	25.2	18.4	12.0	35.7	14.5	69.4
LL3DA [23]	-	-	-	-	13.5	37.3	15.9	76.8
Chat-Scene [22]	21.6	43.2	29.1	20.6	14.3	41.6	18.0	87.7
3D-LLaVA [21]	-	-	-	-	17.1	43.1	18.4	92.6
Video-3D LLM [25]	30.1	47.1	31.7	22.8	16.2	49.0	19.8	102.1
<i>Video-Input Models</i>								
Qwen2.5-VL-3B [14]	15.4	22.5	13.1	8.1	3.8	25.4	9.7	47.4
Qwen2.5-VL-7B [14]	19.0	27.8	13.6	6.3	3.0	29.3	11.4	53.9
Qwen2.5-VL-72B [14]	24.0	26.8	17.8	14.6	12.0	35.2	13.0	66.9
LLaVA-Video-7B [12]	-	39.7	26.6	9.3	3.1	44.6	17.7	88.7
Oryx-34B [53]	-	38.0	24.6	-	-	37.3	15.0	72.3
<b>Spatial-MLLM-4B</b>	26.3	44.4	28.8	21.9	14.8	45.0	18.4	91.8

Table 7: **Additional evaluation results on SQA3D [39] for task-specific models, 3D/2.5D input models, and video-input models.** In addition to the average EM-1 and EM-R1 across all questions, we also report the average EM-1 for different question types, including *What*, *Is*, *How*, *Can*, *Which*, and *Others*.

Methods	SQA3D (test)							
	What	Is	How	Can	Which	Others	Avg. (EM-1)	Avg. (EM-R1)
<i>Task-Specific Models</i>								
SQA3D [39]	31.6	63.8	46.0	69.5	43.9	45.3	46.6	-
3D-Vista [71]	34.8	63.3	45.4	69.8	47.2	48.1	48.5	-
<i>3D/2.5D-Input Models</i>								
Scene-LLM [49]	40.9	69.1	45.0	70.8	47.2	52.3	54.2	-
Chat-Scene [22]	45.4	67.0	52.0	69.5	49.9	55.0	54.6	57.5
Video-3D LLM [25]	51.1	72.4	55.5	69.8	51.3	56.0	58.6	-
<i>Video-Input Models</i>								
Qwen2.5-VL-3B [14]	34.8	52.1	39.8	52.7	45.6	47.0	43.4	45.9
Qwen2.5-VL-7B [14]	39.7	56.6	41.1	55.9	47.6	47.2	46.5	49.8
Qwen2.5-VL-72B [14]	41.7	56.3	41.5	55.6	44.5	48.0	47.0	50.9
LLaVA-Video-7B [12]	42.7	56.3	47.5	55.3	50.1	47.2	48.5	-
<b>Spatial-MLLM-4B</b>	45.9	71.6	55.1	69.5	52.0	53.0	55.9	58.7

**Video**



**Question**

"How many chair(s) are in this room?"

**Spatial-MLLM**

"<think>

**Let me think about this carefully.** In the images provided, I need to identify any chairs present in the room. First, I'll **scan the entire room** to ensure I'm not missing anything. The room appears to be a living space with a kitchen area, a couch, a dining area, and other furniture. **Now, let's break it down:**

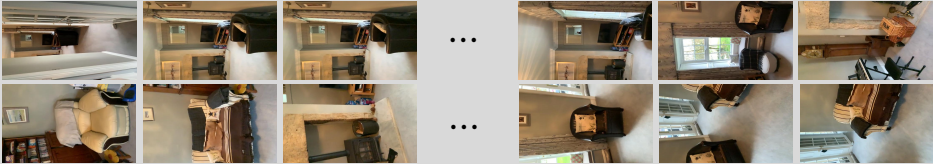
1. In the **kitchen area**, there is a small black stool or cushioned seat near the cabinets. This doesn't seem like a chair.
2. There's a couch in the **living room area**, but that's a sofa, not a chair.
3. In the **dining area**, there is a **small table with two chairs visible**. They are wooden and have simple designs. So, after a thorough review of the images, I can confirm that there are two chairs in the room. Oh, I see. The question was asking for the number of chairs, and now I'm confident that there are two.

</think>

<answer>2 </answer>"

Figure 7: Qualitative example on VSI-Bench [18].

**Video**



**Question**

"How many sofa(s) are in this room?"

**Spatial-MLLM**

"<think>

**Let me think about this for a moment.** The images show a living room with various pieces of furniture. As I scan through them, I notice a few clearly identifiable sofas.

First, in the initial image, there's a partially visible sofa near the **window**. It appears to be a **traditional** design, likely made of fabric and leather. Next, in the following image, there's another sofa positioned near the **fireplace**. This one seems to have a more **modern design**, possibly made of leather or a similar material. Then, in the third image, there's a third sofa visible in the background, near the **bookshelf**. It looks like it's a different style from the first two, possibly a **sectional** sofa. In the fourth image, there's a fourth sofa visible on the left side of the room. It appears to be a **single-seater** sofa, different from the others. **Let's break it down:**

- Traditional sofa: 1 - Modern sofa: 1 - Sectional sofa: 1 - Single-seater sofa: 1

**Oh, I see.** There are four distinct sofas visible in these images. **I should double-check to make sure I'm not missing any**, but after reviewing again, I confirm there are no additional sofas hidden or partially obscured. Final count: 4</think>

<answer>4 </answer>"

Figure 8: Qualitative example on VSI-Bench [18].

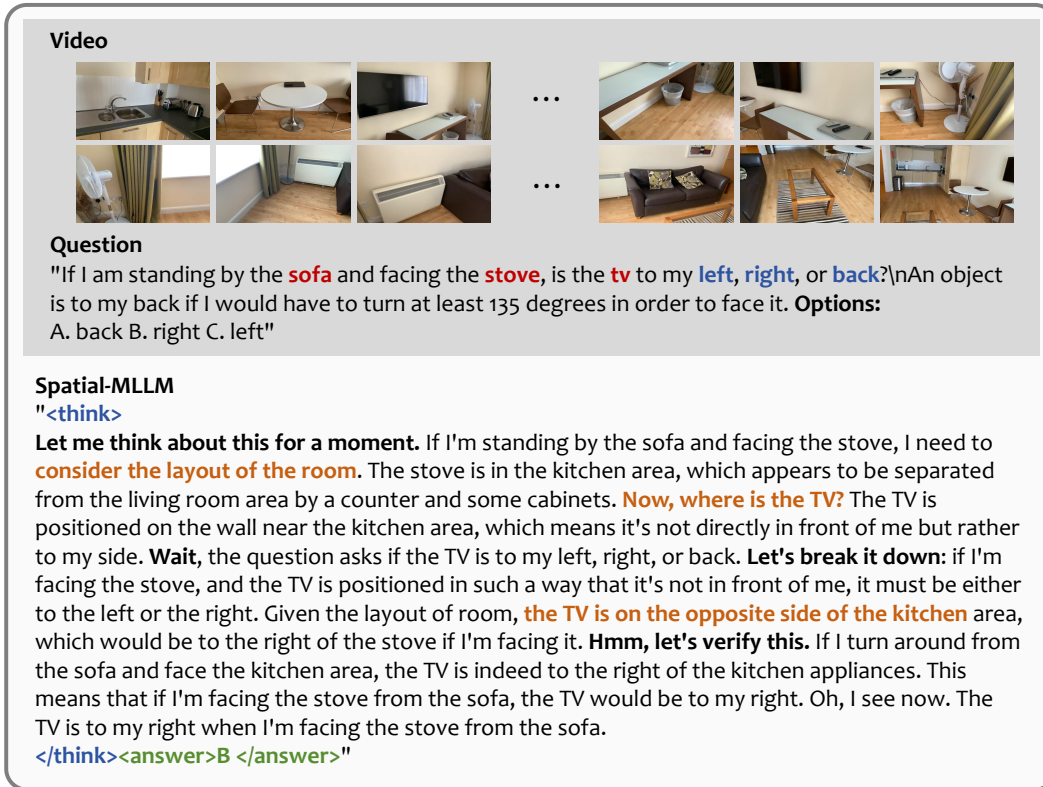


Figure 9: Qualitative example on VSI-Bench [18].

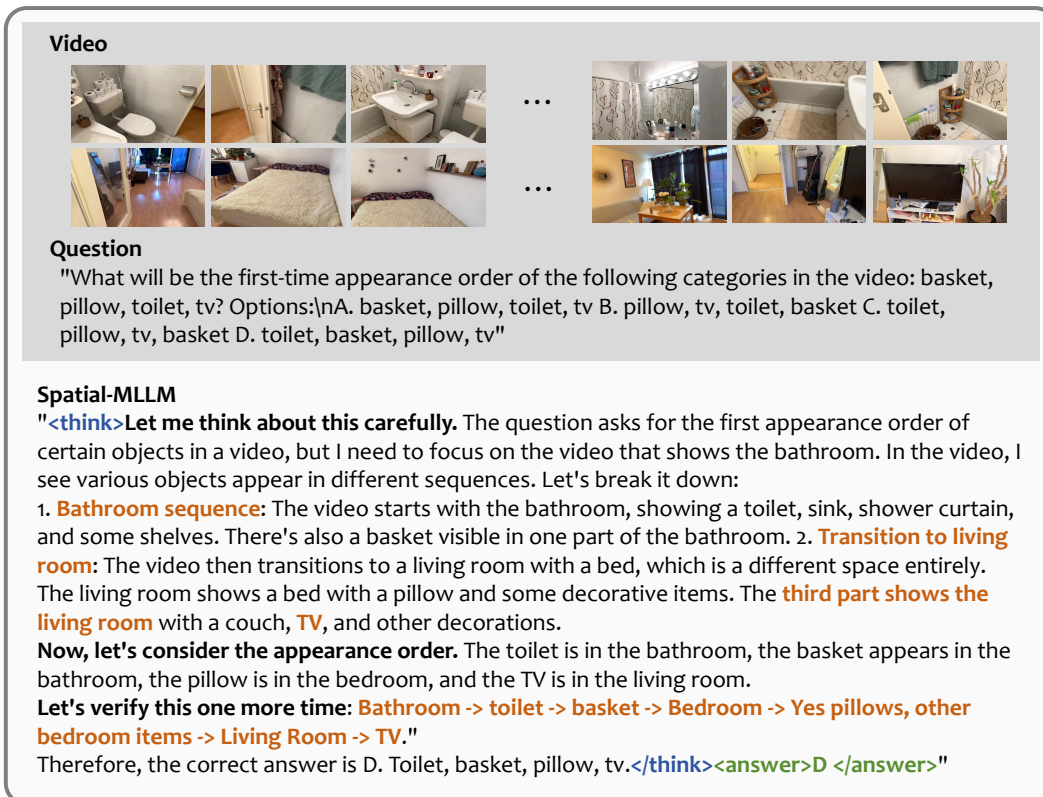


Figure 10: Qualitative example on VSI-Bench [18].