REASONING IN SPACE VIA GROUNDING IN THE WORLD

Anonymous authors
Paper under double-blind review

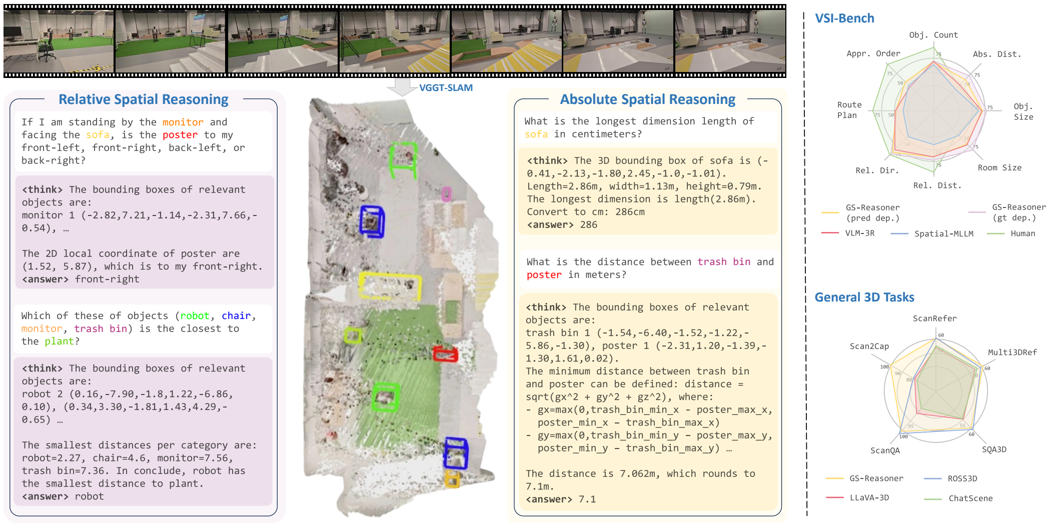
ABSTRACT

In this paper, we claim that 3D visual grounding is one of the cornerstones of spatial reasoning and introduce the *Grounded-Spatial Reasoner (GS-Reasoner)* to explore the effective spatial representations that bridge the gap between them. Existing 3D LLMs suffer from the absence of a unified 3D representation capable of jointly capturing semantic and geometric information. This deficiency is manifested either in poor performance on grounding or in an excessive reliance on external modules, ultimately hindering the seamless integration of grounding and spatial reasoning. To address this, we propose a simple yet effective *dual-path pooling* mechanism that tightly aligns geometric features with both semantic and positional cues, constructing a unified image patch-based 3D representation that encapsulates all essential information without increasing the number of input tokens. Leveraging this holistic representation, GS-Reasoner is the first 3D LLM that achieves autoregressive grounding entirely without external modules while delivering performance comparable to state-of-the-art models, establishing a unified and self-contained framework for 3D spatial reasoning. To further bridge grounding and spatial reasoning, we introduce the *Grounded Chain-of-Thought (GCoT)* dataset. This dataset is meticulously curated to include both 3D bounding box annotations for objects referenced in reasoning questions and step-by-step reasoning paths that integrate grounding as a core component of the problem-solving process. Extensive experiments demonstrate that GS-Reasoner achieves impressive results on 3D visual grounding, which in turn significantly enhances its spatial reasoning capabilities, leading to state-of-the-art performance.

1 INTRODUCTION

Visual-spatial intelligence encompasses the capability to perceive, interpret, and reason about 3D spaces, including the spatial layouts, object sizes, positions and their potential interactions. This skill is fundamental to various domains, such as embodied intelligence and autonomous driving. Accurately linking 3D objects with textual descriptions, a task known as 3D visual grounding, is a prerequisite for effective spatial reasoning. This aligns with human cognitive processes, where identifying relevant objects is a fundamental step before reasoning about their spatial relationships. Despite recent advancements in 3D large language models (LLMs) (Cheng et al., 2024; Cai et al., 2024; Zhou et al., 2025; Zheng et al., 2025; Wang et al., 2025b; Hong et al., 2023a; Chen et al., 2024a; Huang et al., 2023b; 2024; Zhu et al., 2024b), 3D LLMs still rely on pretrained 3D detectors or external decoders for grounding. This reliance not only limits their ability to fully understand 3D scenes but also impedes the cohesive integration of grounding and spatial reasoning. Therefore, a critical question arises: **How can we enable 3D LLMs to perform natural and effective grounding in an autoregressive manner, thereby enhancing their spatial reasoning capabilities?**

We identify two primary challenges in grounding enhanced spatial reasoning. The first challenge arises from the inherent complexity of 3D data. Unlike 2D images, point cloud-based 3D scenes encode rich spatial relations and depth cues that are difficult to capture and align with the semantic space of LLMs, especially given the scarcity of large-scale 3D datasets. Moreover, representing such fine-grained structures often requires a substantially larger number of tokens, further increasing modeling cost. Previous works (Hong et al., 2023a; Chen et al., 2024a) compress point cloud features with Q-former, while others (Fu et al., 2025; Huang et al., 2025b) adopt voxel-based representations to better preserve structure. However, these methods typically trade geometric fidelity for token efficiency, and the extracted point cloud features contain only limited semantic information, making



Relative Spatial Reasoning

```
If I am standing by the monitor and
facing the sofa, is the poster to my
front-left, front-right, back-left, or
back-right?
```

```
<think> The bounding boxes of relevant
objects are:
monitor 1 (-2.82,7.21,-1.14,-2.31,7.66,-
0.54), …

The 2D local coordinate of poster are
(1.52, 5.87), which is to my front-right.
<answer> front-right
```

```
Which of these of objects (robot, chair,
monitor, trash bin) is the closest to
the plant?
```

```
<think> The bounding boxes of relevant
objects are:
robot 2 (0.16,-7.90,-1.8,1.22,-6.86,
0.10), (0.34,3.30,-1.81,1.43,4.29,-
0.65) …

The smallest distances per category are:
robot=2.27, chair=4.6, monitor=7.56,
trash bin=7.36. In conclude, robot has
the smallest distance to plant.
<answer> robot
```

Absolute Spatial Reasoning

```
What is the longest dimension length of
sofa in centimeters?
```

```
<think> The 3D bounding box of sofa is (-
0.41,-2.13,-1.80,2.45,-1.0,-1.01).
Length=2.86m, width=1.13m, height=0.79m.
The longest dimension is length(2.86m).
Convert to cm: 286cm
<answer> 286
```

```
What is the distance between trash bin and
poster in meters?
```

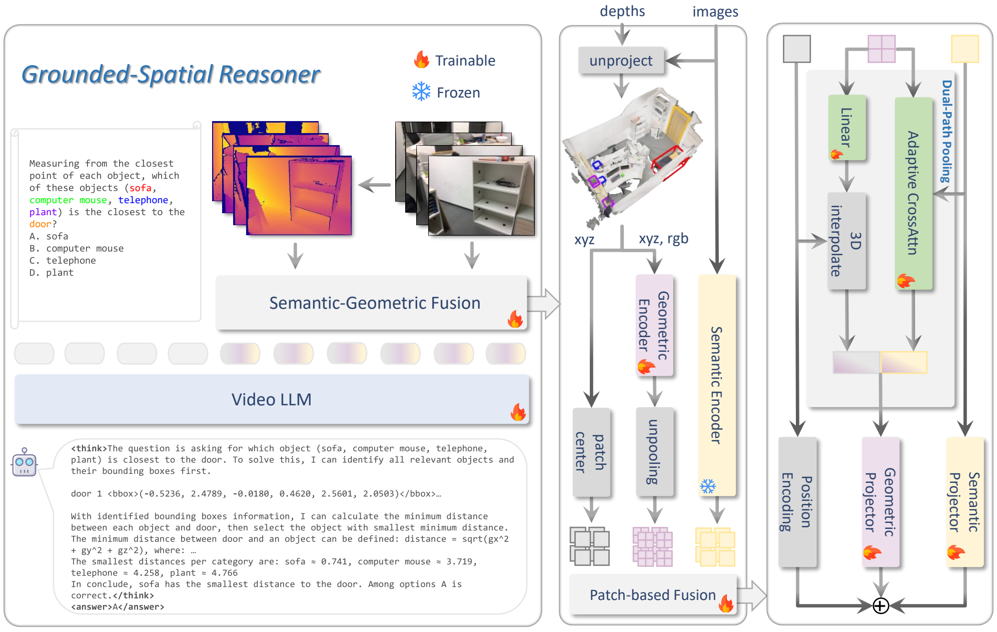
```
<think> The bounding boxes of relevant
objects are:
trash bin 1 (-1.54,-6.40,-1.52,-1.22,-
5.86,-1.30), poster 1 (-2.31,1.20,-1.39,-
1.30,1.61,0.02).
The minimum distance between trash bin
and poster can be defined: distance =
sqrt(gx^2 + gy^2 + gz^2), where:
- gx=max(0,trash_bin_min_x - poster_max_x,
  poster_min_x - trash_bin_max_x)
- gy=max(0,trash_bin_min_y - poster_max_y,
  poster_min_y - trash_bin_max_y) …

The distance is 7.062m, which rounds to
7.1m.
<answer> 7.1
```

Figure 1: We propose *GS-Reasoner*, which integrates visual grounding as an intermediate chain-of-thought for spatial reasoning. All the bounding boxes shown above are autoregressively derived by GS-Reasoner in the reasoning process. Notably, the showcased video is captured in the wild without sensory 3D inputs, highlighting the strong generalization capability of our model.

accurate grounding and reasoning difficult. More recent approaches (Zheng et al., 2025; Zhu et al., 2024b) encode 3D positional cues into video-based semantic features from vision foundation models, showing promising 3D reasoning benefits from visual LLM pretraining. Nevertheless, the geometric cues derived solely from 3D position encodings are weak, which constrains grounding performance. The second challenge lies in the lack of high-quality datasets that integrate grounding as an intermediate step for spatial reasoning. Existing 3D VQA datasets (Azuma et al., 2022; Ma et al., 2022) provide only short answers without grounding annotations or reasoning steps, making the combination of grounding and reasoning impossible. Additionally, these datasets fail to capture the contextual richness and structural complexity required for comprehensive spatial reasoning, further limiting progress toward robust 3D LLMs.

In this work, we propose a novel approach to address the identified challenges by introducing a comprehensive 3D scene representation and a GCoT dataset for spatial reasoning. Our 3D scene representation integrates semantic features from vision foundation models, geometric features encoded by a point cloud encoder, and 3D positional information. The key idea is to unify these heterogeneous signals within an image patch-based representation. Specifically, we pool the geometric features of point maps in a dual-path way to align them with the corresponding semantic feature and 3D position of the image patch, and subsequently fuse them into a unified hybrid representation. This hybrid representation preserves the strong generalization ability of LLMs gained from visual-semantic pretraining, while the incorporation of geometric information significantly strengthens its 3D scene comprehension. As a result, GS-Reasoner can accurately locate objects without relying on any external modules, which provides a natural intermediate step for spatial reasoning. To train models capable of handling both tasks, we construct the GCoT dataset. It includes precise 3D bbox annotations for objects mentioned in reasoning questions, along with step-by-step reasoning paths that embed grounding as a core component of problem solving. By structuring the tasks in this way, the dataset encourages models to first identify relevant objects before addressing complex spatial reasoning, yielding a more interpretable and cognitively aligned approach to learning spatial reasoning.

- We propose a semantic-geometric hybrid 3D scene representation that endows LLM with strong geometric priors, firstly enabling LLM to autoregressively perform 3D visual grounding with impressive results.
- We introduce the GCoT dataset, which bridges the gap between grounding and spatial reasoning, enabling GS-Reasoner to first ground objects and then reason about their spatial relationships in a manner aligned with human cognition.
- We demonstrate the effectiveness of GS-Reasoner through extensive experiments, showcasing its remarkable performance in both 3D visual grounding and spatial reasoning tasks.



```
Measuring from the closest
point of each object, which
of these objects (sofa,
computer mouse, telephone,
plant) is the closest to the
door?
A. sofa
B. computer mouse
C. telephone
D. plant
```

```
<think>The question is asking for which object (sofa, computer mouse, telephone,
plant) is closest to the door. To solve this, I can identify all relevant objects and
their bounding boxes first.

door 1 <bbox>(-0.5236, 2.4789, -0.0180, 0.4620, 2.5601, 2.0503)</bbox>…

With identified bounding boxes information, I can calculate the minimum distance
between each object and door, then select the object with smallest minimum distance.
The minimum distance between door and an object can be defined: distance = sqrt(gx^2
+ gy^2 + gz^2), where: …
The smallest distances per category are: sofa ≈ 0.741, computer mouse ≈ 3.719,
telephone ≈ 4.258, plant ≈ 4.766
In conclude, sofa has the smallest distance to the door. Among options A is
correct.</think>
<answer>A</answer>
```

(a) GS-Reasoner Overview (b) Semantic-Geometric Fusion (c) Patch-based Fusion

Figure 2: **Overview of GS-Reasoner framework**. Our method builds a semantic-geometric hybrid 3D scene representation, enabling 3D LLM to perform 3D visual grounding autoregressively, which allows grounding to be integrated as a chain-of-thought within the spatial reasoning process.

2 Related Work

3D Large Language Models for 3D Understanding. Recent advances in MLLMs have enabled 3D LLMs that integrate 3D information for tasks such as 3D VQA, visual grounding, and captioning. Early work 3D-LLM (Hong et al., 2023a) introduces a Q-Former to align point cloud features with LLMs, followed by studies (Chen et al., 2024a;b; Zhu et al., 2024a; Deng et al., 2025) constructing 3D representations with controllable token lengths. Voxel-based approaches (Fu et al., 2025; Huang et al., 2025b) balance token efficiency and geometric fidelity, while object-centric methods (Huang et al., 2024; 2025b; Yu et al., 2025) improve 3D scene understanding but lack global context. Recent works (Zheng et al., 2025; Zhu et al., 2024b; Wang et al., 2025b) propose encoding 3D positional information into visual features extracted by vision foundation models, achieving promising results on 3D tasks while maintaining the generalization ability of visual LLMs. Despite these advances, existing 3D LLMs still struggle to jointly capture semantic and geometric information from 3D scenes, limiting performance on 3D visual grounding or forcing reliance on external modules.

Video-Language Models for Spatial Reasoning. The goal of visual-based spatial intelligence is to equip video MLLMs with the ability to understand and reason about 3D spatial structures directly from video data. While Video-Language Models (VLMs) (Lin et al., 2023; Li et al., 2024; Bai et al., 2025; Liu et al., 2024; Chen et al., 2024c) perform well on video-language tasks, they still show limited results on recent spatial reasoning benchmarks (Yang et al., 2025). Spatial-MLLM (Wu et al., 2025a) and VLM-3R (Fan et al., 2025) enhance spatial reasoning by incorporating geometric features from recent developed visual geometry models (e.g., VGGT (Wang et al., 2025c)) and constructing large-scale spatial reasoning QA pairs for training. However, the constrained answer formats, such as single-choice selections or short numerical responses, potentially limit the ability of MLLMs to fully exploit the rich 3D information encoded in the geometric features of visual geometry models.

3 GS-Reasoner Framework

3.1 Overview

Given a sequence of N RGB images $\{I_i \in \mathbb{R}^{3\times H\times W}\}_{i=1}^{N}$ of a 3D scene and a spatial reasoning query Q, our goal is to build a model that can first identify all objects potentially relevant to Q and then perform step-by-step spatial reasoning in an autoregressive manner to derive the final answer. Depth maps $\{D_i \in \mathbb{R}^{H \times W}\}_{i=1}^{N}$, camera intrinsics $K \in \mathbb{R}^{3 \times 3}$, and extrinsics $\{T_i \in \mathbb{R}^{4 \times 4}\}_{i=1}^{N}$ are assumed available or can be estimated using visual geometry methods (Maggio et al., 2025).

As illustrated in Fig. 2 (a) and (b), the proposed GS-Reasoner framework comprises three main components: a semantic encoder, a geometric encoder, and a video LLM. The semantic encoder extracts rich semantic features from the input RGB images using a pre-trained vision foundation model. Meanwhile, the depth maps are back-projected into point maps $\{P_i \in \mathbb{R}^{3 \times H \times W}\}_{i=1}^{N}$, which are subsequently transformed into geometric and 3D positional features. Specifically, the geometric encoder processes the aggregated sensor point cloud $\mathcal{P} = \cup_{i=1}^{N} P_i$, where $\mathcal{P} \in \mathbb{R}^{M \times 3}$ denotes M 3D points, to capture structural information of the scene. Since the geometric features are permutation-invariant and thus lack explicit positional cues, we further position-encode the 3D coordinates of points. Finally, the semantic, geometric, and positional features are fused into a unified semantic-geometric hybrid 3D scene representation. This hybrid representation, together with the text query Q, is fed into the video LLM to perform autoregressive object grounding and spatial reasoning, ultimately producing the final answer.

We format the output of GS-Reasoner in a Chain-of-Thought (CoT) manner. All intermediate reasoning is enclosed within the “`<think>...</think>`” block: the model first analyzes the query, and then lists the 3D bounding boxes of all relevant objects in the following format “`OBJECT_NAME OBJECT_COUNT <bbox>(x1, y1, z1, x2, y2, z2)</bbox>...`”. If object bounding boxes are considered unhelpful for answering during question analysis, they are omitted. Each tuple “`(x1, y1, z1, x2, y2, z2)`” denotes the coordinates of two opposite corners of an axis-aligned 3D bounding box expressed in the world coordinate frame (units: meters). After grounding, the model carries out step-by-step spatial reasoning using all available information. Finally, it emits a concise final answer enclosed in “`<answer>...</answer>`”. This autoregressive output format improves interpretability while remaining flexible, enabling GS-Reasoner to be applied to various 3D visual grounding and spatial-reasoning tasks without changing the architecture.

3.2 Semantic-Geometric Hybrid 3D Scene Representation

In this section, we describe the construction of a comprehensive 3D scene representation that seamlessly integrates semantic and geometric information. Building on Video LLM, our goal is to enhance its spatial understanding capabilities by incorporating richer geometric cues, without increasing the input token count or compromising its language comprehension. Inspired by recent works (Zheng et al., 2025; Zhu et al., 2024b) that augment image patch features with 3D positional encoding, we design our 3D scene representation using image patches as the basic building block.

Geometric Feature Extraction. The first challenge arises when extracting per-patch geometric features from point maps. Instead of processing points independently within each patch—which often contain very few points and thus provide limited context for effective feature learning—we process the point cloud $\mathcal{P}$ as a whole. Specifically, we first partition the point maps $\{P_i \in \mathbb{R}^{3 \times H \times W}\}_{i=1}^{N}$ into patches of size $p \times p$, aligning with the image patch size used in the semantic encoder. To reduce computational cost, we uniformly sample K points from each patch, resulting in subsampled point maps denoted as $\{P_i^{sub} \in \mathbb{R}^{3 \times K \times H' \times W'}\}_{i=1}^{N}$, where $H' = \frac{H}{p}$ and $W' = \frac{W}{p}$. The collection of all sampled points across patches forms the input point cloud $\mathcal{P}$, which is subsequently processed by a point cloud encoder to extract geometric features. We adopt the encoder-only Point Transformer v3 (PTv3) (Wu et al., 2024; 2025b) as our point cloud encoder owing to its efficiency and effectiveness. Given a point set $\mathcal{M} = (\mathcal{P}, \mathcal{F})$, where $\mathcal{F} \in \mathbb{R}^{c}$ denotes point attributes (e.g., position, color), PTv3 first serializes the input point cloud with space-filling curves and partitions the points into subsets $[\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_{n'}]$ according to their serialization order. The serialized subsets are then processed by a U-Net-like encoder-only architecture, where each layer employs serialized attention to capture both local and global context. Between layers, each subset $\mathcal{M}_i = (\mathcal{P}_i, \mathcal{F}_i)$ is pooled as follows,

$$\boldsymbol{f}'_i = \mathrm{MaxPool}(\{\boldsymbol{f}_j \boldsymbol{U} \mid \boldsymbol{f}_j \in \mathcal{F}_i\}), \qquad \boldsymbol{p}'_i = \mathrm{MeanPool}(\{\boldsymbol{p}_j \mid \boldsymbol{p}_j \in \mathcal{P}_i\}), \tag{1}$$

where $(\boldsymbol{p}'_i, \boldsymbol{f}'_i)$ denotes the position and features of pooled point aggregated from subset $\mathcal{M}_i$, and $\boldsymbol{U} \in \mathbb{R}^{c \times c'}$ is a linear projection. Collecting pooled points from n' subsets yields the point set $\mathcal{M}' = \{\boldsymbol{p}'_i, \boldsymbol{f}'_i\}_{i=1}^{n'}$ for the next stage of encoding. Unpooling is performed by preserving mapping relationships through the pooling layers, which allows point features to be projected back to the original resolution and concatenated with features from the previous encoding stage as,

$$\boldsymbol{f}_i^{up} = \text{concat}(\boldsymbol{f}_i, \boldsymbol{f}_j^{up}), \qquad \text{if } (\boldsymbol{p}_i, \boldsymbol{f}_i) \in \mathcal{M}_j. \tag{2}$$

By progressively unpooling across layers and mapping the features back to point maps, we obtain the final geometric feature maps $\{G_i \in \mathbb{R}^{C\times K\times H'\times W'}\}_{i=1}^N$, which are spatially aligned with the inputs.

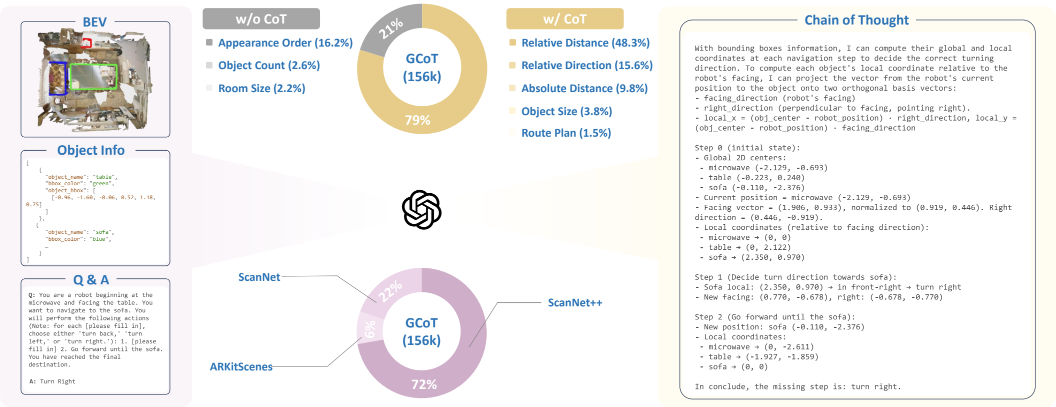
Dual-Path Pooling. With extracted geometric feature maps $\{G_i \in \mathbb{R}^{C\times K\times H'\times W'}\}_{i=1}^N$ and point maps $\{P_i^{sub} \in \mathbb{R}^{3\times K\times H'\times W'}\}_{i=1}^N$, a straightforward strategy for deriving per-patch representations is to apply max pooling within each patch on the geometric feature maps and mean pooling on the point maps, following the design in PTv3. However, we observed that this naive approach results in poor grounding performance, which can be attributed to two key issues: **(1) semantic-geometric misalignment**. While processing point cloud as a whole enhances the receptive field and enables more accurate geometric feature extraction compared to treating points within each patch independently, it also leads to misalignment between the geometric features and semantic features in each patch, as the 3D points in a patch can interact with almost all the points in point cloud, whereas the semantic features are constrained to the information visible in the current image. The geometric features pooled by max pooling emphasize the most salient features without considering the semantic context of the patch, exacerbating this misalignment. **(2) position-geometric misalignment**. Traditional point cloud encoders typically group points using KNN (Qi et al., 2017a;b; Zhao et al., 2021; Wu et al., 2022) or serialization (Wu et al., 2024; 2025b), ensuring that points within a group are spatially close in 3D space. This spatial proximity allows naive pooling strategies to effectively preserve geometric information within the group. In contrast, 3D points within an image patch do not necessarily satisfy this condition, particularly when a patch contains both foreground and background elements, which can lead to large spatial distances among points. Consequently, directly applying max pooling to the geometric features within the patch may introduce geometric inconsistencies, while mean pooling the 3D points can produce positions that are far from both foreground and background objects. These issues can negatively impact the accuracy of predicted 3D bounding boxes.

To address these challenges, we propose a simple yet effective dual-path feature fusion module that aligns semantic, geometric, and positional information at the patch level. To mitigate semantic-geometric misalignment, we construct semantic-aligned geometric features via a lightweight cross-attention network. Each patch's semantic feature serves as the query, while the K geometric features within the patch serve as keys and values. The attention mechanism allows the network to selectively integrate the geometric features most relevant to the patch's semantic context. For the position-geometric misalignment, we directly sample the 3D point corresponding to each patch's center pixel for position encoding, and then interpolate the geometric features based on the position of the 3D point to obtain position-aligned geometric feature. This simple strategy ensures consistency between positional and geometric information: if the sampled point is on the foreground, the interpolated features mainly come from foreground points, and vice versa. Finally, the semantic-aligned and position-aligned geometric features are concatenated and projected to produce the final patch-level geometric features, which are then combined with the projected semantic feature and sampled 3D point positional encoding to obtain the final patch-level hybrid feature.

4 GCoT: Grounded Chain-of-Thought Dataset

Recent works (Wu et al., 2025a; Fan et al., 2025; Ouyang et al., 2025) attempt to improve the spatial reasoning ability of MLLMs by constructing large-scale QA pairs with 3D object annotations. However, the answers in these datasets are typically restricted to single choices or short numerical values. Such limited supervision narrows the learning space of MLLMs, thereby reducing their learning efficiency and resulting in less interpretable outcomes. In fact, spatial reasoning is largely grounded in the locations and size relationships of relevant objects, indicating that identifying objects and reasoning about their geometry are fundamental steps, which is essentially a 3D visual grounding task. Introducing grounding as an intermediate step in spatial reasoning not only provides richer supervision but also improves interpretability, which motivates the construction of the GCoT dataset.

Fig. 3 presents an overview of the GCoT dataset. We first generate spatial reasoning QA pairs without CoT by following the dataset construction pipeline of (Yang et al., 2025; Fan et al., 2025), while preserving the object bounding box information used during generation. Leveraging the QA pairs, object bounding boxes, and bird's-eye views of the scenes, we then prompt GPT-4o (OpenAI et al.,



```
[
    {
      "object_name": "table",
      "bbox_color": "green",
      "object_bbox": [
        [-0.96, -1.60, -0.06, 0.52, 1.18,
0.75]
      ]
    },
   {
      "object_name": "sofa",
      "bbox_color": "blue",
      …
    }
]
```

Q: You are a robot beginning at the microwave and facing the table. You want to navigate to the sofa. You will perform the following actions (Note: for each [please fill in], choose either 'turn back,' 'turn left,' or 'turn right.'): 1. [please fill in] 2. Go forward until the sofa. You have reached the final destination.

A: Turn Right

```
With bounding boxes information, I can compute their global and local
coordinates at each navigation step to decide the correct turning
direction. To compute each object's local coordinate relative to the
robot's facing, I can project the vector from the robot's current
position to the object onto two orthogonal basis vectors:
- facing_direction (robot's facing)
- right_direction (perpendicular to facing, pointing right).
- local_x = (obj_center - robot_position) · right_direction, local_y =
(obj_center - robot_position) · facing_direction

Step 0 (initial state):
- Global 2D centers:
 - microwave (-2.129, -0.693)
 - table (-0.223, 0.240)
 - sofa (-0.110, -2.376)
- Current position = microwave (-2.129, -0.693)
- Facing vector = (1.906, 0.933), normalized to (0.919, 0.446). Right
direction = (0.446, -0.919).
- Local coordinates (relative to facing direction):
 - microwave → (0, 0)
 - table → (0, 2.122)
 - sofa → (2.350, 0.970)

Step 1 (Decide turn direction towards sofa):
- Sofa local: (2.350, 0.970) → in front-right → turn right
- New facing: (0.770, -0.678), right: (-0.678, -0.770)

Step 2 (Go forward until the sofa):
- New position: sofa (-0.110, -2.376)
- Local coordinates:
 - microwave → (0, -2.611)
 - table → (-1.927, -1.859)
 - sofa → (0, 0)

In conclude, the missing step is: turn right.
```

Figure 3: **Overview of Grounded Chain-of-Thought (GCoT) Dataset.** We first construct spatial QA pairs without CoT, and then prompt GPT-4o to generate CoT paths based on the bird's-eye view, object information, and QA pairs.

2024) to produce coherent CoT reasoning paths that lead to the final answers. The resulting dataset consists of 156k QA pairs, among which 79% contain CoT annotations. We omit CoT construction for the Appearance Order, Object Counting, and Room Size Estimation tasks, as these tasks do not require complex spatial reasoning. Additional details are provided in the Appendix A.

5 Experiments

We first describe the implementation details in Section 5.1 and report results on 3D visual grounding in Section 5.2. Since grounding forms the basis for spatial reasoning, we then evaluate our framework on spatial reasoning tasks in Section 5.3. Section 5.4 presents additional results on general 3D tasks, and Section 5.5 provides zero-shot evaluation and ablation studies to analyze the contributions of individual model components and the proposed GCoT dataset.

5.1 Implementation Details

Model Architecture. GS-Reasoner is developed on top of LLaVA-Video 7B (Zhang et al., 2024), an open-source video LLM based on Qwen2-7B (Team, 2024). For semantic encoding, we adopt SigLIP (Zhai et al., 2023), a vision transformer pre-trained on large-scale image–text pairs through contrastive learning. For geometric encoding, we employ Sonata (Wu et al., 2025b), an efficient point cloud encoder built upon PTv3 (Wu et al., 2024) and pre-trained in a self-supervised manner on large-scale point cloud datasets. We adopt sinusoidal positional encoding (Vaswani et al., 2017) to encode the 3D positions of image patches.

Training. GS-Reasoner is trained end-to-end for next-token prediction. We first pretrain on subsets of 3D visual grounding datasets, including ScanRefer (Chen et al., 2020), Multi3DRef (Zhang et al., 2023), SR3D, and NR3D (Achlioptas et al., 2020), to warm up object grounding, and then finetune on our proposed GCoT dataset, the remaining grounding data, and other 3D tasks (ScanQA (Azuma et al., 2022), SQA3D (Ma et al., 2022), Scan2Cap (Chen et al., 2021)). Data augmentation is important for training GS-Reasoner and we provide more details in Appendix B.1.

Inference. Unless otherwise specified, we uniformly sample 32 images from each scene as the model input during inference. For the 3D visual grounding task, ground-truth depth maps and camera parameters are provided to ensure a fair evaluation. For the spatial reasoning task, depth maps and camera parameters are estimated using VGGT-SLAM (Maggio et al., 2025), followed by metric alignment with Moge2 (Wang et al., 2025e). More details are provided in Appendix B.2.

5.2 Evaluation on 3D Visual Grounding

We evaluate our model on four widely used 3D visual grounding benchmarks: ScanRefer, Multi3DRef, SR3D, and NR3D. For single-object grounding (ScanRefer, SR3D, NR3D), we report Acc@25 and

Table 1: **Evaluation on 3D Visual Grounding.** GS-Reasoner achieves performance comparable to 3D LLMs using mesh proposals or external grounding, without any external components.

Methods	ScanRefer		Multi3DRef		SR3D	NR3D
	Acc@25	Acc@50	F1@25	F1@50	Acc@25	Acc@25
Expert Models						
3D-VisTA (Zhu et al., 2023)	51.0	46.2	-	-	56.5	47.7
PQ3D (Zhu et al., 2024c)	56.7	51.8	-	-	62.0	52.2
UniVLG (Jain et al., 2025)	63.5	56.4	-	-	73.0	56.3
Locate 3D (McVay et al., 2025)	61.1	50.9	-	-	68.2	56.1
3D LLMs + Proposals from Mesh PC						
Chat-Scene (Huang et al., 2024)	55.5	50.2	57.1	52.4	-	-
Inst3D-LMM (Yu et al., 2025)	57.8	51.6	58.3	53.5	-	-
Video-3D LLM (Zheng et al., 2025)	58.1	51.7	58.0	52.7	-	-
ROSS3D (Wang et al., 2025b)	61.1	54.4	59.6	54.3	-	-
SeeGround (Li et al., 2025b)	44.1	39.4	-	-	-	-
3D LLMs + External Grounding Module						
Grounded 3D-LLM (Chen et al., 2024b)	48.6	44.0	44.7	40.8	-	-
ReGround3D (Zhu et al., 2024a)	53.1	41.2	-	-	-	-
LLaVA-3D (Zhu et al., 2024b)	54.1	42.2	54.3	47.2	-	-
3D LLMs						
3D-LLM (Hong et al., 2023b)	30.3	-	-	-	-	-
GS-Reasoner	60.8	42.2	61.7	45.3	56.7	50.0

Table 2: **Evaluation on VSI-Bench.** GS-Reasoner achieves state-of-the-art performance on most tasks, with further gains using more accurate (ground-truth) depth.

Methods	Rank.	Avg.	**Numerical Question**				**Multiple-Choice Question**			
			Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order
Baseline										
Chance Level (Random)	-	-	-	-	-	-	25.0	36.1	28.3	25.0
Chance Level (Frequency)	-	34.0	62.1	32.0	29.9	33.1	25.1	47.9	28.4	25.2
VSI-Bench Perf. († = Tiny Set)										
†Human Level	-	79.2	94.3	47.0	60.4	45.9	94.7	95.8	95.8	100.0
†Gemini-1.5 Flash	-	45.7	50.8	33.6	56.5	45.2	48.0	39.8	32.7	59.2
†Gemini-1.5 Pro	-	48.8	49.6	28.8	58.6	49.4	46.0	48.1	42.0	68.0
†Gemini-2.0 Flash	-	45.4	52.4	30.6	66.7	31.8	56.0	46.3	24.5	55.1
Proprietary Models (API)										
GPT-4o	3	34.0	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5
Gemini-1.5 Flash	2	42.1	49.8	30.8	53.5	54.4	37.7	41.0	31.5	37.8
Gemini-1.5 Pro	1	45.4	56.2	30.9	64.1	43.6	51.3	46.3	36.0	34.6
Open-sourced VLMs										
InternVL2-40B	3	36.0	34.9	26.9	46.5	31.8	42.1	32.2	34.0	39.6
LongVILA-8B	9	21.6	29.1	9.1	16.7	0.0	29.6	30.7	32.5	25.5
VILA-1.5-40B	7	31.2	22.4	24.8	48.7	22.7	40.5	25.7	31.5	32.9
LongVA-7B	8	29.2	38.0	16.6	38.9	22.2	33.1	43.3	25.4	15.7
LLaVA-NeXT-Video-7B	4	35.6	48.5	14.0	47.8	24.2	43.5	42.4	34.0	30.6
LLaVA-NeXT-Video-72B	1	40.9	48.9	22.8	57.4	35.3	42.4	36.7	35.0	48.6
QWen2.5VL-7B	5	33.0	40.9	14.8	43.4	10.7	38.6	38.5	33.0	29.8
LLaVA-OneVision-7B	6	32.4	47.7	20.2	47.4	12.3	42.5	35.2	29.4	24.4
LLaVA-OneVision-72B	2	40.2	43.5	23.9	57.6	37.5	42.5	39.9	32.5	44.6
Specialized Spatial Reasoning Models										
Spatial-MLLM-4B	3	48.4	65.3	34.8	63.1	45.1	41.3	46.2	33.5	46.3
VLM-3R-7B	2	60.9	70.2	49.4	69.2	67.1	65.4	80.5	45.4	40.1
GS-Reasoner (pred dep.)	1	64.7	69.1	61.9	70.0	65.7	65.4	88.9	44.3	52.3
GS-Reasoner (gt dep.)	-	70.1	70.9	73.6	77.8	81.8	70.6	90.5	42.8	52.6

Acc@50, where a prediction is correct if its Intersection over Union (IoU) with ground truth exceeds 0.25 or 0.5, respectively. For multi-object grounding (Multi3DRef), we use the F1 score computed at IoU thresholds of 0.25 and 0.5. For fair comparison, we group the baselines into four categories: (1) Expert Models, specifically designed for 3D grounding and trained with both bounding box and mask supervision; (2) 3D LLMs + Proposals from Mesh Point Cloud, which select from proposals generated by detectors such as Mask3D (Schult et al., 2022); (3) 3D LLMs + External Grounding Module, which pair a 3D LLM with an auxiliary grounding module; and (4) 3D LLMs, which directly predict bounding boxes autoregressively without relying on any external modules or proposals.

As shown in Tab. 1, our model achieves superior performance compared with 3D-LLM (Hong et al., 2023b). Among other 3D LLM-based methods, it achieves state-of-the-art F1@25 on Multi3DRef, even surpassing methods that rely on proposals or external grounding modules. Moreover, on

Table 3: **Evaluation on General 3D Tasks.** GS-Reasoner outperforms state-of-the-art 3D LLMs on Scan2Cap and achieves comparable results on ScanQA and SQA3D.

Methods	Scan2Cap				ScanQA					SQA3D
	B-4 ↑	Rouge ↑	CIDEr ↑	Meteor ↑	B-4 ↑	Rouge ↑	CIDEr ↑	Meteor ↑	EM ↑	EM ↑
3D-LLM(flamingo) (Hong et al., 2023a)	-	-	-	-	7.2	32.3	59.2	12.2	20.4	-
3D-LLM(BLIP2-flant5) (Hong et al., 2023a)	-	-	-	-	12.0	35.7	69.4	14.5	20.5	-
LL3DA (Chen et al., 2024a)	36.8	55.1	65.2	26.0	13.5	37.3	76.8	15.9	-	-
Chat-3Dv2 (Huang et al., 2023a)	-	-	-	-	14.0	-	87.6	-	-	54.7
LEO (Huang et al., 2023b)	36.9	57.8	68.4	27.7	13.2	49.2	101.4	20.0	24.5	50.0
Scene-LLM (Fu et al., 2025)	-	-	-	-	12.0	40.0	80.0	16.6	27.2	54.2
ChatScene (Huang et al., 2024)	36.3	58.1	77.2	28.0	14.3	41.6	87.7	18.0	21.6	54.6
LLaVA-3D (Zhu et al., 2024b)	41.1	63.4	79.2	30.2	14.5	50.1	91.7	20.7	27.0	55.6
Video-3D LLM (Zheng et al., 2025)	42.4	62.3	83.8	28.9	16.2	49.0	102.1	19.8	30.1	58.6
ROSS3D (Wang et al., 2025b)	43.4	66.9	81.3	30.3	17.9	50.7	107.0	20.9	30.8	63.0
GS-Reasoner	47.6	69.2	101.0	32.1	16.2	49.2	102.6	19.8	29.9	59.9

ScanRefer Acc@50 and Multi3DRef F1@50, GS-Reasoner matches the performance of 3D LLMs with external grounding modules, despite using only noisy, incomplete sensor point clouds rather than high-quality mesh inputs. However, GS-Reasoner still lags behind 3D LLMs with proposals from mesh point clouds on these two metrics. We attribute this to two factors: (1) mesh point clouds are more complete and less noisy; and (2) conventional 3D detectors (e.g., Mask3D) are commonly trained with mask supervision, which is more conducive to precise object localization than bbox supervision, as also reported in (McVay et al., 2025; Jain et al., 2025). An interesting observation is that GS-Reasoner achieves comparable results to expert models on ScanRefer but falls behind on SR3D and NR3D, suggesting LLM-based methods are better at complex queries (as in ScanRefer), while expert models excel in precise localization for simpler descriptions (as in SR3D and NR3D).

5.3 Evaluation on Spatial Reasoning

We evaluate GS-Reasoner's spatial reasoning capability on VSI-Bench (Yang et al., 2025), which comprises over 5,000 QA pairs from egocentric videos in ScanNet (Dai et al., 2017), ScanNet++ (Yeshwanth et al., 2023), and ARKitScenes (Baruch et al., 2021). VSI-Bench provides two answer formats, multiple choice (MCA) and numerical (NA), and covers eight tasks spanning spatial and temporal reasoning. We follow the official VSI-Bench evaluation protocol for metric computation. The results in Tab. 2 demonstrate that GS-Reasoner achieves impressive performance, particularly on the Relative Direction and Absolute Distance tasks, which require complex spatial reasoning and precise 3D object localization. It also attains state-of-the-art results on the Appearance Order task, indicating that our semantic-geometric hybrid features effectively preserve temporal information from the original video while providing additional spatial cues. Moreover, performance consistently improves with more accurate depth input, with the average score exceeding 70 and yielding nearly a 10-point gain over the previous state of the art.

5.4 Evaluation on General 3D Tasks

We further present results on three established 3D vision-language understanding tasks: Scan2Cap, ScanQA, and SQA3D, following the official protocols and reporting performance in terms of CIDEr, BLEU, METEOR, ROUGE, and exact-match (EM) accuracy. The results in Tab. 3 show that GS-Reasoner sets a new state of the art for 3D dense captioning, achieving the best results on Scan2Cap across all metrics and significantly surpassing the previous leading method, ROSS3D (Wang et al., 2025b). We attribute these gains to explicitly predicting coordinates for 3D visual grounding, which forces the model to better capture geometric and positional cues, thereby improving dense captioning performance. However, GS-Reasoner does not achieve leading results on ScanQA and SQA3D. We believe the main reasons are the presence of many ambiguous questions in these datasets that do not clearly specify the target object, as well as a strong bias in answer distribution. These factors encourage the model to overfit to textual patterns instead of effectively exploiting 3D tokens. Recent studies (Huang et al., 2025a; Li et al., 2025a) have also shown that finetuning LLMs without 3D input can yield results comparable to the state of the art on ScanQA and SQA3D. Incorporating reconstruction constraints in the output (as done in ROSS3D (Wang et al., 2025b)) may help encourage the model to utilize 3D tokens, and we leave this for future research.

5.5 Analysis and Ablation Studies

Table 4: **Zero-shot 3D visual grounding.** We train the models exclusively on ScanNet and evaluate them on ScanNet++ and ARKitScenes for visual grounding, reporting all results in Acc@25. Note that GPT-4o is prompted to do 2D visual grounding and then back-project to 3D via depth map. Locate 3D is an expert model.

Methods	ScanRefer	LX3D	
	ScanNet	ScanNet++	ARKitScenes
GPT-4o VLM	59.9	60.5	26.8
Locate 3D	61.1	56.7	46.2
GS-Reasoner	60.8	51.0	45.6

Table 5: **Ablation Study on Data Aug. and 3D Representation.** We train models to autoregressively predict 3D bounding box coordinates using ScanRefer and Multi3DRef, and report results on ScanRefer.

Methods	**Data Aug.**	**Pos. Enc.**	**Geo.Pool.**	Acc@25	Acc@50
LLaVA-NeXT	✗	✗	✗	0.0	0.0
Video-3D LLM	✗	Avg	✗	15.4	3.5
	✓	Avg	✗	53.2	29.8
	✓	Avg	Max	57.5 +4.3	35.7 +5.9
	✓	Avg	Cross-Attn	58.9 +5.7	38.6 +9.8
	✓	Sample	Interpolate	59.3 +6.1	40.2 +10.4
GS-Reasoner	✓	Sample	Dual-Path	**60.8** **+7.6**	**42.2** **+12.4**

Table 6: **Ablation Study on Grounded CoT Mechanism.** We report results only for tasks in the GCoT dataset that include CoT annotations, to highlight the effectiveness of grounded CoT.

Methods	Avg.	**Numerical Question**		**Multiple-Choice Question**		
		Abs. Dist.	Obj. Size	Rel. Dist.	Rel. Dir.	Route Plan
LLaVA-NeXT-Video ft (w/o CoT)	52.3	45.1	64.3	58.9	60.7	32.5
GS-Reasoner ft (w/o CoT)	57.7 +5.4	50.8 +5.7	65.7 +1.4	62.3 +3.4	79.3 +18.6	30.4 -2.1
GS-Reasoner ft (Full)	**66.1** **+13.8**	**61.9** **+16.8**	**70.0** **+5.7**	**65.4** **+6.5**	**88.9** **+28.2**	**44.3** **+11.8**

Zero-shot Generalization. We evaluate the zero-shot generalization of GS-Reasoner on unseen 3D scenes. The model is trained solely on ScanNet data (ScanRefer, SR3D, etc.), and tested on the ScanNet++ and ARKitScenes validation splits of the Locate3D dataset (McVay et al., 2025) without finetuning. As shown in Tab. 4, GS-Reasoner achieves performance comparable to SOTA expert models on ARKitScenes and demonstrates strong results in novel scene spatial reasoning (Fig. 1).

Effectiveness of Data Augmentation and Semantic-Geometric Hybrid 3D Representation. We conduct ablation studies to assess the effectiveness of our data augmentation strategies and the proposed semantic-geometric hybrid 3D representation, using the 3D visual grounding task as the evaluation benchmark. We believe this task directly reflects the model's ability to jointly leverage semantic and spatial information from the input 3D scene. The results in Tab. 5 show that the model fails to accurately predict 3D bbox coordinates when only image input is provided (LLaVA-Next). Incorporating average position encoding (as in Video-3D LLM) still results in poor performance due to overfitting. Data augmentation brings notable improvements, yet the model continues to struggle with precise object localization, as indicated by the low Acc@50. Finally, by introducing geometric features from the geometric encoder and employing Dual-Path Pooling to progressively fuse position-aligned and semantic-aligned geometric features, we achieve substantial gains in both Acc@25 and Acc@50. These results demonstrate that the proposed hybrid 3D representation strengthens the model's understanding of 3D scenes and enables more accurate visual grounding.

Effectiveness of GCoT Dataset. We also ablate the impact of incorporating grounding into the chain-of-thought process on spatial reasoning performance. Specifically, we remove the CoT part from the answers in the GCoT dataset and train the model to directly predict the answer from the 3D scene and question. We report results for five tasks that incorporate grounding within the CoT process in Tab. 6. The results show that integrating grounding as part of the CoT process substantially improves performance across all tasks, particularly in object absolute distance, object relative direction, and route planning. This highlights the importance of not only providing the model with grounding capabilities but also guiding it to leverage grounding effectively to support spatial reasoning, demonstrating the necessity of the proposed GCoT dataset.

Table 7: **Computation Cost Comparison.** All values are reported in milliseconds.

Methods	2D Vision Enc.	3D Vision Enc.	Dual-Path	Total
LLaVA-NeXT-Video	429	-	-	470
GS-Reasoner	430	204	41	737

Computation Costs Evaluation. GS-Reasoner's computational overhead compared to the backbone LLaVA-Next-Video primarily arises from the construction of visual tokens. Specifically, GS-Reasoner requires encoding point clouds using Point Transformer and integrating them into image patch features. Therefore, we conduct a comparison of computational overhead during the visual token construction phase, using a single 4090 GPU with 32 input images. As shown in Tab. 7, GS-Reasoner introduces some additional computational overhead. However, since all visual tokens only need to be constructed once per response, this overhead is acceptable relative to the overall inference time. Furthermore, during subsequent reasoning, the semantic-geometric hybrid 3D representation constructed by GS-Reasoner does not increase the number of visual tokens, so the inference time remains comparable to that of LLaVA-Next-Video.

Table 8: **Ablation Study on Input Frames.** We investigate the impact of different numbers of input frames and sampling strategies on 3D visual grounding performance.

		ScanRefer		Multi3DRef	
Frames Num	**Sampling Strategy**	Acc@25	Acc@50	F1@25	F1@50
32	uniform	60.8	42.2	61.7	45.3
48	uniform	**61.7**	44.5	62.2	46.9
64	uniform	61.2	**45.1**	**62.5**	**48.0**
32	cdviews	61.1	43.4	61.9	46.2

Ablation Study on Input Frames. We conduct an ablation study to investigate the impact of different numbers of input frames and sampling strategies on 3D visual grounding performance. As shown in Tab. 8, model performance consistently improves with an increasing number of input images, and the improvements are more pronounced in Acc@50 (F1@50) compared to Acc@25 (F1@25), indicating that providing denser point clouds can indeed help enhance localization accuracy. Moreover, employing context-aware sampling strategies such as cdViews (Wang et al., 2025a) also yields certain improvements.

6 Conclusion

In this work, we present GS-Reasoner, a novel framework that integrates grounding into spatial reasoning as a chain-of-thought process. Built upon a hybrid semantic–geometric 3D scene representation, GS-Reasoner performs grounding without requiring any external modules, making it a natural intermediate step for spatial reasoning. The GCoT dataset further strengthens the model's ability to handle both tasks effectively.

Ethics Statement

Our work introduces a new dataset generated entirely using large language models (LLMs). As the dataset is synthetically generated, it does not contain any personally identifiable information or sensitive human data. Nevertheless, synthetic data may inherit biases present in the underlying LLM, and could potentially be misused for harmful or misleading purposes. To mitigate these risks, the dataset is intended solely for academic research, and will be released with clear guidelines on responsible usage. Users are encouraged to consider ethical implications when employing the dataset for downstream tasks.

Reproducibility Statement

To facilitate reproducibility, we will release the full dataset, preprocessing scripts, and detailed documentation upon acceptance. All experimental code, pretrained models, and evaluation protocols will also be made publicly available. The datasets used in our experiments are either publicly accessible or will be released as part of this work. We provide complete hyperparameter settings, training schedules, and random seeds in the paper and supplementary materials.

References

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *European conference on computer vision*, pp. 422–440. Springer, 2020. 6, 18

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19129–19139, 2022. 2, 6, 18

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

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 8, 16

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. *arXiv preprint arXiv:2406.13642*, 2024. 1

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *16th European Conference on Computer Vision (ECCV)*, 2020. 6, 18

Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26428–26438, 2024a. 1, 3, 8

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025. 19

Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Runsen Xu, Ruiyuan Lyu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024b. 3, 7

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024c. 3

Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3193–3203, 2021. 6, 18

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024. 1

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017. 8, 16

Jiajun Deng, Tianyu He, Li Jiang, Tianyu Wang, Feras Dayoub, and Ian Reid. 3d-llava: Towards generalist 3d lmms with omni superpoint transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3772–3782, 2025. 3

Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv preprint arXiv:2212.08320*, 2022. 19

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction, 2025. URL `https://arxiv.org/abs/2505.20279`. 3, 5, 16, 17

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual reasoning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2195–2206. IEEE, 2025. 1, 3, 8

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. 2022. 19

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023a. 1, 3, 8

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models, 2023b. 7

Haifeng Huang, Zehan Wang, Rongjie Huang, Luping Liu, Xize Cheng, Yang Zhao, Tao Jin, and Zhou Zhao. Chat-3d v2: Bridging 3d scene and large language models with object identifiers. *CoRR*, 2023a. 8

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024. 1, 3, 7, 8

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023b. 1, 8

Jiangyong Huang, Baoxiong Jia, Yan Wang, Ziyu Zhu, Xiongkun Linghu, Qing Li, Song-Chun Zhu, and Siyuan Huang. Unveiling the mist over 3d vision-language understanding: Object-centric evaluation with chain-of-analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24570–24581, 2025a. 8, 19

Jiangyong Huang, Xiaojian Ma, Xiongkun Linghu, Yue Fan, Junchao He, Wenxin Tan, Qing Li, Song-Chun Zhu, Yixin Chen, Baoxiong Jia, et al. Leo-vl: Towards 3d vision-language generalists via data scaling with efficient representation. *arXiv preprint arXiv:2506.09935*, 2025b. 1, 3

Ayush Jain, Alexander Swerdlow, Yuzhou Wang, Sergio Arnaud, Ada Martin, Alexander Sax, Franziska Meier, and Katerina Fragkiadaki. Unifying 2d and 3d vision-language understanding. *arXiv preprint arXiv:2503.10745*, 2025. 7, 8

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 20

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 3

Haoyuan Li, Yanpeng Zhou, Yufei Gao, Tao Tang, Jianhua Han, Yujie Yuan, Dave Zhenyu Chen, Jiawang Bian, Hang Xu, and Xiaodan Liang. Does your 3d encoder really work? when pretrain-sft from 2d vlms meets 3d vlms. *arXiv preprint arXiv:2506.05318*, 2025a. 8, 19

Rong Li, Shijie Li, Lingdong Kong, Xulei Yang, and Junwei Liang. Seeground: See and ground for zero-shot open-vocabulary 3d visual grounding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3707–3717, 2025b. 7

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 3

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024. 3

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022. 2, 6, 18

Dominic Maggio, Hyungtae Lim, and Luca Carlone. Vggt-slam: Dense rgb slam optimized on the sl (4) manifold. *arXiv preprint arXiv:2505.12549*, 2025. 4, 6, 18, 19

Paul McVay, Sergio Arnaud, Ada Martin, Arjun Majumdar, Krishna Murthy Jatavallabhula, Phillip Thomas, Ruslan Partsey, Daniel Dugas, Abha Gejji, Alexander Sax, et al. Locate 3d: Real-world object localization via self-supervised learning in 3d. In *Forty-second International Conference on Machine Learning*, 2025. 7, 8, 9

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, et al. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276. 5, 17

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025. 5

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017a. 5, 19

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b. 5, 19

Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, pp. 28223–28243. PMLR, 2023a. 19

Zekun Qi, Muzhou Yu, Runpei Dong, and Kaisheng Ma. Vpp: Efficient conditional 3d generation via voxel-point progressive representation. *Advances in Neural Information Processing Systems*, 36:26744–26763, 2023b. 19

Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction. In *European Conference on Computer Vision*, pp. 214–238. Springer, 2024. 19

Zekun Qi, Wenyao Zhang, Yufei Ding, Runpei Dong, Xinqiang Yu, Jingwen Li, Lingyun Xu, Baoyu Li, Xialin He, Guofan Fan, et al. Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. *arXiv preprint arXiv:2502.13143*, 2025. 20

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. 19

Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105*, 2022. 7

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 6

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6

Fengyun Wang, Sicheng Yu, Jiawei Wu, Jinhui Tang, Hanwang Zhang, and Qianru Sun. 3d question answering via only 2d vision-language models. *arXiv preprint arXiv:2505.22143*, 2025a. 10

Haochen Wang, Yucheng Zhao, Tiancai Wang, Haoqiang Fan, Xiangyu Zhang, and Zhaoxiang Zhang. Ross3d: Reconstructive visual instruction tuning with 3d-awareness. *arXiv preprint arXiv:2504.01901*, 2025b. 1, 3, 7, 8, 19

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025c. 3, 19

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10510–10522, 2025d. 19

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025e. 6, 18, 19

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *arXiv preprint arXiv:2505.23747*, 2025a. 3, 5

Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 5, 19

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4840–4851, 2024. 4, 5, 6, 19

Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22193–22204, 2025b. 4, 5, 6, 19

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10632–10643, 2025. 3, 5, 8, 16, 20

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12–22, 2023. 8, 16

Hanxun Yu, Wentong Li, Song Wang, Junbo Chen, and Jianke Zhu. Inst3d-lmm: Instance-aware 3d scene understanding with multi-modal instruction tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14147–14157, 2025. 3, 7

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023. 6

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025. 20

Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15225–15236, 2023. 6, 18

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 6

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16259–16268, 2021. 5, 19

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8995–9006, 2025. 1, 2, 3, 4, 7, 8

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*, 2025. 1

Chenming Zhu, Tai Wang, Wenwei Zhang, Kai Chen, and Xihui Liu. Scanreason: Empowering 3d visual grounding with reasoning capabilities. In *European Conference on Computer Vision*, pp. 151–168. Springer, 2024a. 3, 7

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024b. 1, 2, 3, 4, 7, 8

Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment, 2023. 7

Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. Unifying 3d vision-language understanding via promptable queries. *ECCV*, 2024c. 7

A Additional Dataset Details

Grounding Chain-of-Thought Dataset plays a crucial role in our training, guiding the model to learn how to incorporate 3D visual grounding as an intermediate step in spatial reasoning. Here, we provide additional details on the dataset construction.

A.1 Instruction QA Generation

We first construct spatial reasoning QA pairs without chain-of-thought annotations, following the dataset generation pipeline of (Yang et al., 2025; Fan et al., 2025). All data are sourced from existing large-scale 3D datasets, including ScanNet (Dai et al., 2017), ScanNet++ (Yeshwanth et al., 2023), and ARKitScenes (Baruch et al., 2021). To ensure consistency across datasets, we perform the following preprocessing steps for each scene:

- **Point Cloud.** We directly use the raw point cloud provided by each dataset. Since ScanNet++ and ARKitScenes do not guarantee alignment of the global coordinate system with the physical room structure (e.g., the XY plane may not align with walls or floors), we further apply axis alignment. Specifically, we estimate the gravity direction and compute the principal components of the point cloud to align the axes, yielding a transformation matrix for each scene.
- **Sampled Frame Data.** we uniformly sample 50 RGB frames, which serve as the basis for constructing frame metadata. These frames provide a consistent visual context for generating spatial reasoning questions and ensure coverage of diverse viewpoints within the scene.

Based on the preprocessed data, we further construct detailed metadata for each scene, consisting of the following components:

- **Scene Metadata.** This metadata is used for all spatial reasoning questions construction. We extract the axis-aligned bounding boxes (AABBs) of all object instances, either directly from mask annotations or by converting from oriented bounding box (OBB) annotations. In addition to bounding boxes, this metadata also includes global scene statistics such as the number of objects and room dimensions, which are later used to formulate numerical reasoning questions.
- **Frame Metadata.** This metadata is used specifically for appearance-based temporal reasoning questions. For each object, we determine its appearance time by recording the first frame in which its 2D mask area exceeds a given threshold. Consequently, the frame metadata of each scene contains the appearance time of all objects, enabling the construction of reasoning questions grounded in temporal visual evidence.

These two types of metadata provide the necessary information to generate a diverse set of spatial and temporal reasoning questions. Following the predefined question templates in (Yang et al., 2025), we iterate over the scene metadata to construct a large pool of candidate questions and their corresponding answers. The detailed procedures are as follows:

Spatial Reasoning QA.

- **Object Count.** For each object category with at least two instances in the scene, we generate counting questions by directly querying the number of instances.
- **Absolute Distance.** We randomly select pairs of objects that appear only once in the scene and compute their Euclidean distance, which serves as the basis for absolute distance queries.
- **Object Size.** For objects with a single instance, we compute the object size using the diagonal length of their AABB, and use this value to construct size-related questions.
- **Room Size.** We estimate the overall room size of each scene using the alpha-shape algorithm applied to the scene point cloud, allowing us to ask questions about scene-level spatial dimensions.
- **Relative Distance.** We randomly select a set of N objects ($3 \leq N \leq 5$), compute all pairwise distances, and identify the closest pair of objects. This enables the construction of questions that require comparative spatial reasoning.

- **Relative Direction.** We randomly select three objects with unique instances and compute their relative directions based on the centers of their AABBs. The resulting orientation relations form the basis of direction-based reasoning questions.

Temporal Reasoning QA. For temporal reasoning (i.e., appearance order), we randomly select four objects from each scene and determine their order of appearance using the frame metadata.

Route Planning QA. For route planning questions, we follow the procedure in VLM-3R (Fan et al., 2025) and employ the Habitat simulator to generate diverse navigation trajectories between two predefined points in each scene. The turning direction at each step is determined by computing the angle between consecutive anchor points along the trajectory. To identify relevant objects, we calculate their proximity to the trajectory by measuring the distance between anchor points and the 3D bounding boxes of scene objects provided in the scene metadata. Finally, we construct the QA pairs using predefined templates consistent with those in VLM-3R (Fan et al., 2025), where each question is grounded in the trajectory's turning direction and nearby objects.

A.2 Chain-of-Thought Generation

Building upon the generated QA pairs and their associated 3D object bounding boxes, we further create CoT annotations to guide the model in exploiting 3D visual grounding for spatial reasoning. Specifically, we employ GPT-4o (OpenAI et al., 2024) to generate CoT reasoning paths for each QA pair. To mitigate potential hallucinations from the LLM, we construct a bird's-eye-view map of each scene using the preprocessed point cloud. On this map, we explicitly annotate the 3D bounding boxes of relevant objects, with different colors denoting distinct object categories, consistent with the color keys used in the input bounding box information. The BEV map, together with the question, answer, and relevant object annotations, is then provided to GPT-4o to generate coherent and detailed CoT paths. For efficiency, we exclude tasks that do not require complex spatial reasoning or grounding (e.g., Room Size, Object Count, Appearance Order) from CoT generation, as these can be accurately answered without the need for step-by-step grounding-based reasoning.

A.3 Data Format

To train a model that can determine when to ground relevant objects and perform step-by-step reasoning autonomously, we prepend a *question analysis* part to each answer. This part explicitly guides the model to decide whether grounding is required before initiating the reasoning process. An example of the final data format for QA pairs with CoT is shown below:

QA w/ CoT Example

Question: [preprompt] If I am standing by the radiator and facing the table, is the telephone to the left or the right of the table? Options: A. left B. right

Answer: <think>The question is asking for the relative direction of the telephone with respect to my position (at the radiator) while facing the table. To solve this, I can identify all relevant objects and their bounding boxes first.

radiator 1 <bbox>(-1.9165, -1.0266, 0.0798, -1.6415, 0.9513, 0.6104)</bbox>, table 1 <bbox>(-0.9128, -1.7861, 0.0150, -0.5043, -1.0341, 0.5050)</bbox>, telephone 1 <bbox>(-1.4182, 0.9637, 0.9736, -1.1277, 1.2378, 1.0825)</bbox>

To determine the telephone's position relative to the table, I analyze spatial relationships through three steps:
Step 1: Observer Position and Orientation
Standing at the radiator's centroid:
- x = (-1.9165 + -1.6415)/2 = -1.779
- y = (-1.0266 + 0.9513)/2 = -0.038
Facing the table, whose centroid: ..
The telephone lies to the left of the table. Option A is correct. </think>
<answer>A</answer>

And example of the final data format for QA pairs without CoT is shown below:

QA w/o CoT Example

Question: [preprompt] What is the size of this room (in square meters)? If multiple rooms are shown, estimate the size of the combined space.

Answer: <think>The question is asking for the room size in square meters. To solve this, I can rely on the overall spatial information present in the video without identifying any object bounding boxes.
From the video, the room size is about 11.0 m^2.</think>
<answer>11.0</answer>

By structuring the data in this way, the model learns to autonomously decide when to ground relevant objects and perform step-by-step reasoning, without the need for additional prompting.

B Additional Implementation Details

B.1 Training Details

GS-Reasoner is trained end-to-end with cross-entropy loss for next-token prediction. We first pretrain on subsets of 3D visual grounding datasets, including ScanRefer (Chen et al., 2020), Multi3DRef (Zhang et al., 2023), SR3D, and NR3D (Achlioptas et al., 2020), to warm up object grounding, and then finetune on our proposed GCoT dataset, the remaining grounding data, and other 3D tasks (ScanQA (Azuma et al., 2022), SQA3D (Ma et al., 2022), Scan2Cap (Chen et al., 2021)). All parameters are learnable except those of the vision encoder. The LLM learning rate is fixed at $1e^{-5}$, while other modules use $1e^{-4}$ during pretraining and $1e^{-5}$ during finetuning. We use a batch size of 16 for pretraining and 256 for finetuning, set $K = 64$ in all experiments, and uniformly sample $N \in [16, 48]$ images per scene during training. Data augmentation is crucial for training GS-Reasoner, as the autoregressive objective tends to overfit object grounding under limited 3D data. We avoid conventional point cloud augmentations (e.g., jittering, elastic distortion) already covered in Sonata's pretraining, and instead focus on decoupling geometric and positional cues. Specifically, we apply Z-axis rotations of $[90°, 180°, 270°]$, random scaling within $[0.75, 1.25]$, and translations within $[-1, 1]$ meters, which alter bounding box positions and scales, forcing the model to exploit both cues for accurate predictions.

B.2 Inference Details

We develop a pipeline to recover metric depth and camera parameters from multi-view images, enabling spatial reasoning without any input beyond images. Specifically, we first use VGGT-SLAM (Maggio et al., 2025) to reconstruct dense depth maps and relative camera intrinsics and extrinsics from the multi-view images. We then apply MoGe-2 (Wang et al., 2025e) to estimate absolute-scale depth maps and per-image camera intrinsics independently. Since the intrinsics from these two methods may not be aligned, we avoid direct scale estimation in the depth dimension. Instead, we project all points into the camera coordinate system and compute a global scale factor s such that the scaled VGGT-SLAM point maps align with the corresponding MoGe-2 point maps across all views. Formally, s is obtained by solving the following optimization problem:

$$s^* = \arg\min_{s>0} \sum_{i=1}^{N} \sum_{j=1}^{M_i} \| s \cdot p_{i,j}^{\text{VGGT-SLAM}} - p_{i,j}^{\text{MoGe-2}} \|^2, \tag{3}$$

where $p_{i,j}^{\text{VGGT-SLAM}}$ and $p_{i,j}^{\text{MoGe-2}}$ denote the j-th point in the i-th view from VGGT-SLAM and MoGe-2, respectively, M_i is the number of valid points in view i, and N is the total number of views. Furthermore, we compute a per-scene axis-alignment transformation matrix based on the estimated camera poses and reconstructed point clouds.

C Additional Related Work

Point Cloud Representation Learning. Point cloud representation learning has been extensively studied for 3D understanding. Early works like PointNet (Qi et al., 2017a;b) use MLPs and symmetric functions to extract global features from point clouds. More recent methods, such as Point Transformer (Zhao et al., 2021; Wu et al., 2022; Qi et al., 2023b; Wu et al., 2024; 2025b), leverage attention mechanisms to capture local geometric structures and point relationships. ACT (Dong et al., 2022) pioneers cross-modal geometry understanding through 2D or language foundation models such as CLIP (Radford et al., 2021) or MAE (He et al., 2022). ReCon (Qi et al., 2023a; 2024) further proposes a learning paradigm that unifies generative and contrastive learning. Despite architectural differences, these methods share a common pipeline: points are grouped based on spatial distribution, features are extracted per group, and then aggregated into a global representation. The resulting sparse features can be upsampled to the original resolution for tasks such as semantic segmentation.

D Additional Experimental Analysis and Results

D.1 Analysis on General 3D Tasks

As shown in Tab. 3, GS-Reasoner does not achieve leading results on ScanQA and SQA3D. We believe the main reasons are the presence of many ambiguous questions in these datasets that do not clearly specify the target object, as well as a strong bias in answer distribution. These factors encourage the model to overfit to textual patterns instead of effectively exploiting 3D tokens. Recent studies (Huang et al., 2025a; Li et al., 2025a) have also shown that finetuning LLMs without 3D input can yield results comparable to the state of the art on ScanQA and SQA3D. Incorporating reconstruction constraints in the output (as done in ROSS3D (Wang et al., 2025b)) may help encourage the model to utilize 3D tokens, and we leave this for future research.

Table 9: **Ablation study on the impact of predicted depth maps and poses on spatial reasoning.**

Recon. Methods	Align Methods		Numerical Question				Multiple-Choice Question			
		Avg.	Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Order
-	-	70.1	70.9	73.6	77.8	81.8	70.6	90.5	42.8	52.6
VGGT-SLAM	MoGe-2	64.7	69.1	61.9	70.0	65.7	65.4	88.9	44.3	52.3
VGGT-SLAM	GT-Depth	67.8	69.4	68.9	76.5	78.8	65.2	87.6	43.4	52.9
VGGT	MoGe-2	59.6	66.8	55.6	65.5	63.1	58.3	77.3	34.0	51.6
CUT3R	-	56.8	66.8	55.3	60.2	50.7	59.8	76.6	34.0	50.7
TTT3R	-	58.0	67.5	57.6	60.2	46.5	62.8	82.9	35.1	51.5

D.2 Ablation Study on Predicted Depth Maps and Poses

We further evaluate the impact of different depth and pose estimation methods on the final spatial reasoning performance. Specifically, we consider the following geometry reconstruction methods: VGGT-SLAM (Maggio et al., 2025), VGGT (Wang et al., 2025c), CUT3R (Wang et al., 2025d), and TTT3R (Chen et al., 2025). For relative geometry reconstruction methods (VGGT-SLAM, VGGT), we use MoGe-2 (Wang et al., 2025e) for absolute scale recovery. We also experiment with using ground-truth depth for absolute scale recovery. The results are shown in Tab. 9, where the first row corresponds to using ground-truth depth and poses. From the experiments, we summarize two key factors that influence performance:

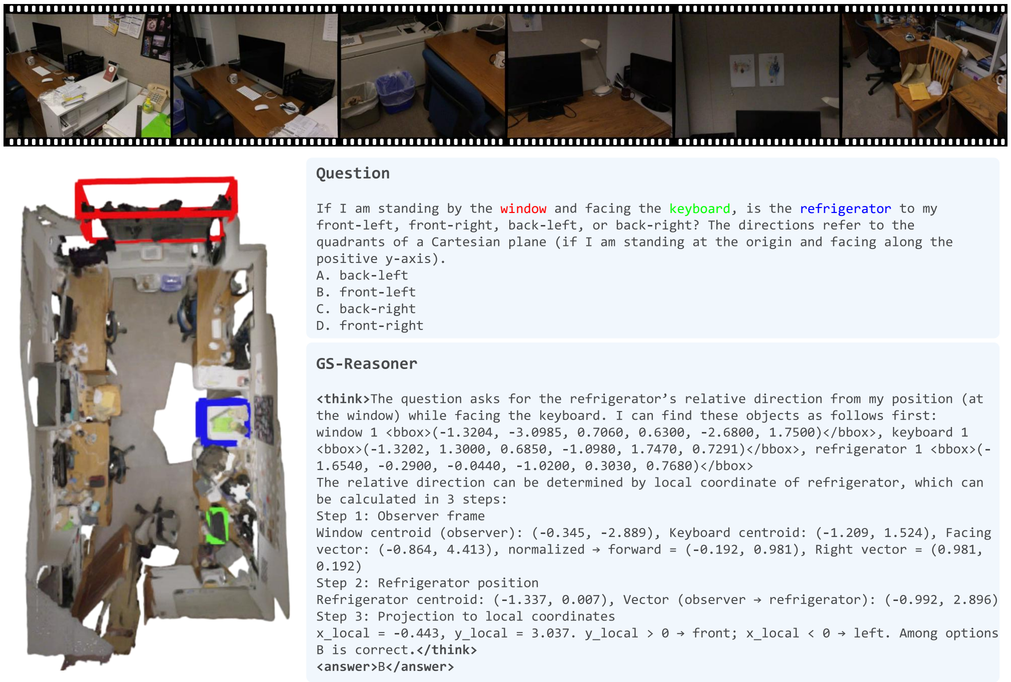
- **Accuracy of metric scale.** As observed, the most significant performance drop compared to using ground truth depth occurs in tasks related to absolute size estimation, such as room size and object absolute distance. During training, GS-Reasoner utilizes ground truth depth, which leads the model to prioritize 3D features in the input for these tasks. Consequently, inaccuracies in metric scale have a substantial impact on the results of such tasks. Notably, when using ground truth depth instead of MoGe2-predicted depth for metric alignment, there is a significant performance improvement.
- **Accuracy of pose.** The accuracy of pose affects the relative positional distribution of object point clouds within the scene, thereby influencing the model's understanding of spatial

relationships between objects. In the point clouds estimated by CUT3R, we observed significant pose errors, often resulting in completely incorrect scene layouts. This leads to a noticeable decline in GS-Reasoner's performance on tasks such as object relative direction. TTT3R improves upon CUT3R by reducing pose error accumulation during long-sequence inputs, resulting in some performance gains.

Beyond these two factors, other aspects have a relatively minor impact on spatial reasoning. For instance, the quality of depth details (e.g., sharpness) does not significantly affect GS-Reasoner.

D.3 More Qualitative Results

We present qualitative results of GS-Reasoner on VSI-Bench (Yang et al., 2025) as follows:



```
Question

If I am standing by the window and facing the keyboard, is the refrigerator to my
front-left, front-right, back-left, or back-right? The directions refer to the
quadrants of a Cartesian plane (if I am standing at the origin and facing along the
positive y-axis).
A. back-left
B. front-left
C. back-right
D. front-right
```

```
GS-Reasoner

<think>The question asks for the refrigerator's relative direction from my position (at
the window) while facing the keyboard. I can find these objects as follows first:
window 1 <bbox>(-1.3204, -3.0985, 0.7060, 0.6300, -2.6800, 1.7500)</bbox>, keyboard 1
<bbox>(-1.3202, 1.3000, 0.6850, -1.0980, 1.7470, 0.7291)</bbox>, refrigerator 1 <bbox>(-
1.6540, -0.2900, -0.0440, -1.0200, 0.3030, 0.7680)</bbox>
The relative direction can be determined by local coordinate of refrigerator, which can
be calculated in 3 steps:
Step 1: Observer frame
Window centroid (observer): (-0.345, -2.889), Keyboard centroid: (-1.209, 1.524), Facing
vector: (-0.864, 4.413), normalized → forward = (-0.192, 0.981), Right vector = (0.981,
0.192)
Step 2: Refrigerator position
Refrigerator centroid: (-1.337, 0.007), Vector (observer → refrigerator): (-0.992, 2.896)
Step 3: Projection to local coordinates
x_local = -0.443, y_local = 3.037. y_local > 0 → front; x_local < 0 → left. Among options
B is correct.</think>
<answer>B</answer>
```

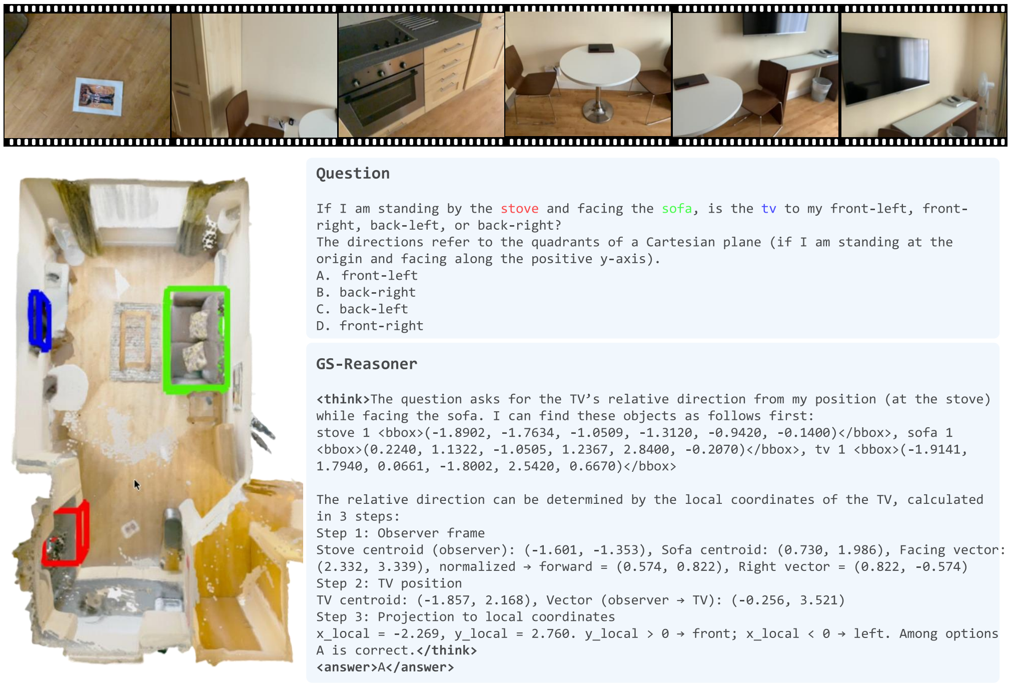
Figure 4: **Qualitative results on VSI-Bench.**

E Future Work

Spatial reasoning is a key aspect of robotics and embodied reasoning, especially for the vision-language-action (VLA) models (Kim et al., 2024; Qi et al., 2025; Zhang et al., 2025). Leveraging the strong spatial reasoning ability of GS-Reasoner in robotic tasks can substantially enhance the generalization and robustness of embodied reasoning. Future directions include jointly fine-tuning with GCoT data and action data, and employing GS-Reasoner as an embodied brain for planning and task decomposition.

F The Use of Large Language Models (LLMs)

In this work, we leverage LLMs to facilitate the construction of our *Grounded Chain-of-Thought (GCOT)* dataset. Specifically, the generation of CoT paths for spatial reasoning tasks is performed using LLMs, which allows us to capture rich intermediate reasoning steps that go beyond simple question-answer pairs.



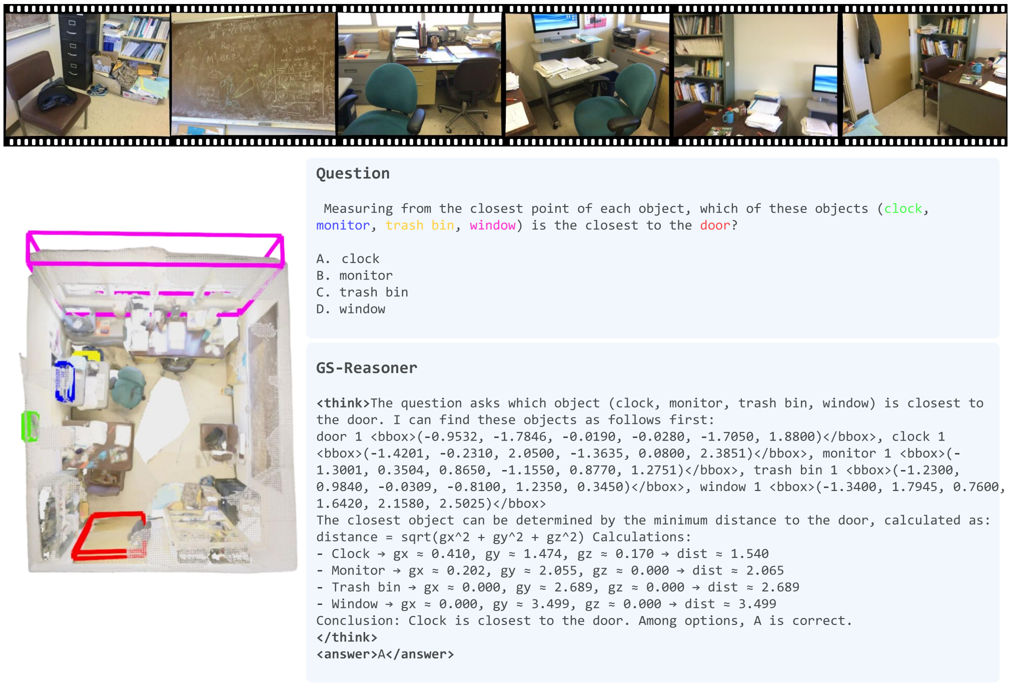
Question

```
If I am standing by the stove and facing the sofa, is the tv to my front-left, front-
right, back-left, or back-right?
The directions refer to the quadrants of a Cartesian plane (if I am standing at the
origin and facing along the positive y-axis).
A. front-left
B. back-right
C. back-left
D. front-right
```

GS-Reasoner

```
<think>The question asks for the TV’s relative direction from my position (at the stove)
while facing the sofa. I can find these objects as follows first:
stove 1 <bbox>(-1.8902, -1.7634, -1.0509, -1.3120, -0.9420, -0.1400)</bbox>, sofa 1
<bbox>(0.2240, 1.1322, -1.0505, 1.2367, 2.8400, -0.2070)</bbox>, tv 1 <bbox>(-1.9141,
1.7940, 0.0661, -1.8002, 2.5420, 0.6670)</bbox>

The relative direction can be determined by the local coordinates of the TV, calculated
in 3 steps:
Step 1: Observer frame
Stove centroid (observer): (-1.601, -1.353), Sofa centroid: (0.730, 1.986), Facing vector:
(2.332, 3.339), normalized → forward = (0.574, 0.822), Right vector = (0.822, -0.574)
Step 2: TV position
TV centroid: (-1.857, 2.168), Vector (observer → TV): (-0.256, 3.521)
Step 3: Projection to local coordinates
x_local = -2.269, y_local = 2.760. y_local > 0 → front; x_local < 0 → left. Among options
A is correct.</think>
<answer>A</answer>
```

Figure 5: **Qualitative results on VSI-Bench.**



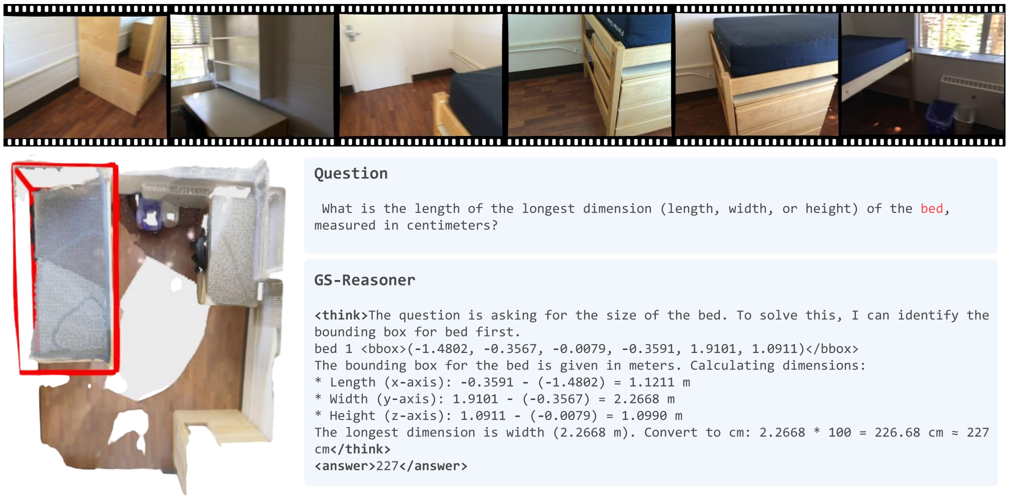
Question

```
 Measuring from the closest point of each object, which of these objects (clock,
monitor, trash bin, window) is the closest to the door?

A. clock
B. monitor
C. trash bin
D. window
```

GS-Reasoner

```
<think>The question asks which object (clock, monitor, trash bin, window) is closest to
the door. I can find these objects as follows first:
door 1 <bbox>(-0.9532, -1.7846, -0.0190, -0.0280, -1.7050, 1.8800)</bbox>, clock 1
<bbox>(-1.4201, -0.2310, 2.0500, -1.3635, 0.0800, 2.3851)</bbox>, monitor 1 <bbox>(-
1.3001, 0.3504, 0.8650, -1.1550, 0.8770, 1.2751)</bbox>, trash bin 1 <bbox>(-1.2300,
0.9840, -0.0309, -0.8100, 1.2350, 0.3450)</bbox>, window 1 <bbox>(-1.3400, 1.7945, 0.7600,
1.6420, 2.1580, 2.5025)</bbox>
The closest object can be determined by the minimum distance to the door, calculated as:
distance = sqrt(gx^2 + gy^2 + gz^2) Calculations:
- Clock → gx ≈ 0.410, gy ≈ 1.474, gz ≈ 0.170 → dist ≈ 1.540
- Monitor → gx ≈ 0.202, gy ≈ 2.055, gz ≈ 0.000 → dist ≈ 2.065
- Trash bin → gx ≈ 0.000, gy ≈ 2.689, gz ≈ 0.000 → dist ≈ 2.689
- Window → gx ≈ 0.000, gy ≈ 3.499, gz ≈ 0.000 → dist ≈ 3.499
Conclusion: Clock is closest to the door. Among options, A is correct.
</think>
<answer>A</answer>
```

Figure 6: **Qualitative results on VSI-Bench.**



Question

What is the length of the longest dimension (length, width, or height) of the bed, measured in centimeters?

GS-Reasoner

```
<think>The question is asking for the size of the bed. To solve this, I can identify the
bounding box for bed first.
bed 1 <bbox>(-1.4802, -0.3567, -0.0079, -0.3591, 1.9101, 1.0911)</bbox>
The bounding box for the bed is given in meters. Calculating dimensions:
* Length (x-axis): -0.3591 - (-1.4802) = 1.1211 m
* Width (y-axis): 1.9101 - (-0.3567) = 2.2668 m
* Height (z-axis): 1.0911 - (-0.0079) = 1.0990 m
The longest dimension is width (2.2668 m). Convert to cm: 2.2668 * 100 = 226.68 cm ≈ 227
cm</think>
<answer>227</answer>
```

Figure 7: **Qualitative results on VSI-Bench.**