SceneCOT:
Eliciting Chain-of-Thought Reasoning in 3D Scenes

Xiongkun Linghu[1], **Jiangyong Huang**[1,2], **Ziyu Zhu**[1,3],
Baoxiong Jia[1,†], **Siyuan Huang**[1,†]
[1]State Key Laboratory of General Artificial Intelligence, BIGAI
[2]Peking University, [3]Tsinghua University

Abstract

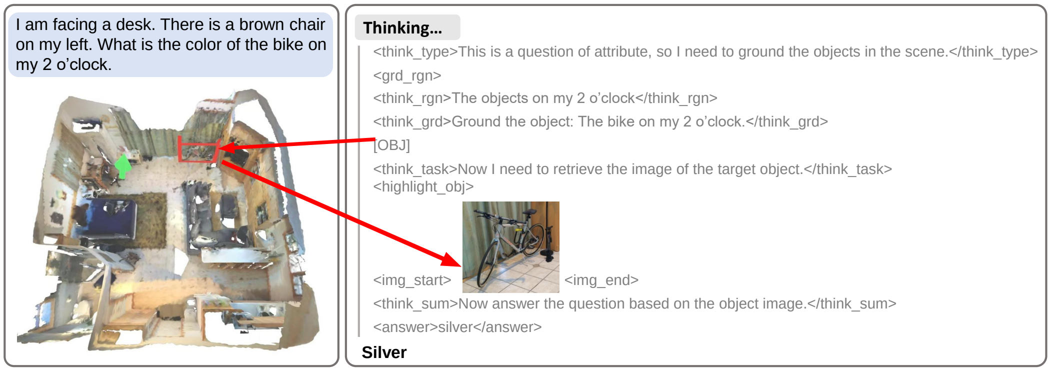
Existing research on 3D Large Language Models (LLMs) still struggles to achieve efficient and explainable reasoning, primarily due to the under-exploration of *the mechanism of human-like scene-object grounded reasoning*. This paper bridges the gap by presenting a novel framework. We first introduce a Chain-of-Thought reasoning method in 3D scenes (SceneCOT), decoupling a complex reasoning task into simpler and manageable problems, and building corresponding visual clues based on multimodal expert modules. To enable such a method, we build the first large-scale 3D scene Chain-of-Thought reasoning dataset SceneCOT-212K, including 212K high-quality data instances. Extensive experiments across various complex 3D scene reasoning benchmarks demonstrate that our new framework achieves state-of-the-art with clear interpretability. To our knowledge, this is the first attempt to successfully implement the COT technique for achieving human-like step-by-step reasoning for 3D scene understanding, where we show great potential in extending it to a wider range of 3D scene understanding scenarios.

1 Introduction

Understanding 3D scenes is a fundamental capability for developing human-level embodied agents [1, 2, 3, 4]. Despite increasing interest and extensive research in this area [5, 6, 7, 8, 9, 10], reasoning in complex 3D scenes remains highly challenging. Compared to image-based reasoning [11, 12, 13, 14], 3D reasoning, especially in an embodied, situated setting [15, 16] requires navigating larger environments, interpreting more intricate spatial relationships, and coping with partial observability due to limited perception during embodied interaction with the scene. We argue that addressing this multi-faceted problem, which spans perception, grounding, reasoning, and planning, requires a principled decomposition of the task into manageable subproblems, together with a structured approach for composing the reasoning process. However, developing such a reasoning model remains extremely challenging, and current research offers little explicit progress in this direction.

We approach this challenging problem by drawing insights from how humans naturally decompose and solve complex reasoning tasks in the 3D physical world. Consider the question: "Where can I grab two bottles of iced beers?" Human reasoning begins with task-oriented perception, first identifying the task as navigation, and then narrowing focus from the full scene to semantically relevant spatial regions (*e.g.*, the kitchen). Once in the potential region, humans engage in explicit semantic and attribute grounding, identifying the target object "beer" and filtering out irrelevant ones based on attributes (*e.g.*, "iced") to verify the quantity is correct. Next, they rely on spatial reasoning

[†] Corresponding author.


I am facing a desk. There is a brown chair on my left. What is the color of the bike on my 2 o'clock.

Thinking...

<think_type>This is a question of attribute, so I need to ground the objects in the scene.</think_type>
<grd_rgn>
<think_rgn>The objects on my 2 o'clock</think_rgn>
<think_grd>Ground the object: The bike on my 2 o'clock.</think_grd>
[OBJ]
<think_task>Now I need to retrieve the image of the target object.</think_task>
<highlight_obj>
<img_start> <img_end>
<think_sum>Now answer the question based on the object image.</think_sum>
<answer>silver</answer>

Silver

Figure 1: **Reasoning chain visualization of SCENECOT**. The reasoning chain mimics humans' recognition process: from high-level task recognition to low-level visual semantic understanding.

to localize referencing anchor objects to describe a path towards the final location. However, this multi-hop reasoning process, which spans task recognition, task-oriented scene perception, entity and attribute grounding, spatial relation reasoning, and final plan generation, is largely overlooked in current in 3D vision-language (3D-VL) models [17, 8, 18]. These models typically simplify the problem to a single-step question-answering task over the entire scene, omitting the critical intermediate reasoning steps. We argue that supporting such hierarchical multi-step reasoning is essential for solving complex 3D reasoning problems.

In fact, modeling such a multi-step reasoning process has been proven to be highly beneficial for solving complex reasoning problems in the language domain. In particular, the idea of Chain-of-Thought (CoT) [19] has led to substantial improvements in Large Language Models (LLMs), achieving superhuman performance in tasks such as math and science question answering [19, 20, 21]. CoT emulates the human thought process by decomposing a complex problem into simpler subproblems and solving them step by step, closely mirroring the aforementioned multi-hop reasoning process in 3D scenes. However, despite the clear need for such structured reasoning in 3D scenes, directly transferring CoT from natural language to 3D reasoning remains challenging due to the difficulty of aligning language with 3D representations. As a result, the application of CoT has so far been limited to Multi-modal LLMs (MLLMs) [22, 23] in 2D Visual Question-Answering (VQA) tasks, while its extension to 3D reasoning remains largely unexplored.

To address the challenges of 3D reasoning and incorporate insights from human thought process, we propose SCENECOT, a 3D-VL model equipped with CoT reasoning capabilities in 3D scenes. SCENECOT follows a hierarchical reasoning workflow from high-level task recognition to low-level visual-semantic grounding, enabling step-by-step problem decomposition. Specifically, given a 3D scene, an agent situation (*i.e.*, location and viewing direction), and a question, the 3D-CoT in SCENECOT is constructed under four steps: 1) the *task recognition and analysis* step identifies the task type and provides guidance for subsequent reasoning; 2) the *task-relevant region localization* step selects the most relevant sub-region of the scene given the task; 3) the *entity grounding* step reformulates the question into grounding queries of target objects and resolves them using expert models across 2D and 3D modalities; and 4) the *grounded reasoning* step translates the grounding outputs into textual context for final reasoning and answer generation. We provide an illustrative example of the SCENECOT method in Fig. 1.

To evaluate the effectiveness of the SCENECOT, we construct a 3D-CoT dataset, SCENECOT-212K, containing more than 212K 3D reasoning traces over existing 3D-QA datasets. Each reasoning trace is generated using a heuristic-based definition of task-oriented sub-regions, carefully selected grounding modalities and representations, and step-by-step reasoning sequences for final answer generation. We train SCENECOT on this dataset and observe significant performance improvements on two challenging 3D reasoning tasks: situated reasoning in MSQA [16], SQA3D [15], and grounding-QA coherence in Beacon3D [24], achieving new state-of-the-art results on these benchmarks. Furthermore, we conduct in-depth analyses of SCENECOT, providing insights into how leveraging CoT advances complex reasoning in 3D environments.

In summary, our contributions are as follows:

- We propose a novel Chain-of-Thought reasoning method SceneCOT, which decomposes the complex 3D scene reasoning problems into simpler and manageable problems, and reasons step by step from high-level task recognition to visual semantic understanding.
- We build the first 3D-CoT dataset SceneCOT-212K with rich stepwise annotation to enable our method, including more than 212K high-quality data instances for complex 3D reasoning.
- Empirical experiments demonstrate our method achieves state-of-the-art (SoTA) performances on the typical 3D scene reasoning tasks with clear interpretability. Our in-depth analyses also reveal how our method advances and provide valuable insights.

2 Related Work

LLMs for 3D Scene Understanding Understanding 3D scenes is crucial for achieving human-like intelligence, and there has been growing interest in leveraging LLMs for 3D scene understanding. 3D-LLM [17] introduces an LLM-based model that lifts multi-view 2D features into 3D space and aligns them with text embedding. PointLLM [25] targets object-level geometry understanding by leveraging a powerful point cloud encoder. LEO [8] aligns object-centric 3D representation with LLMs for 3D-VL tasks. Many follow-up works further enhance LLM-based 3D-VL models [26, 27, 28, 29, 30, 31]. For example, Grounded 3D-LLM [5] aligns point-level semantics with text; Chat-Scene [32] builds scene features using per-object 3D mask proposals. LLaVA-3D [33] stands on LLaVA [14] with 3D positional embeddings to enhance performance in 3D-VL tasks. Video-3D LLM [18] leverages a pretrained video-based MLLM to build temporal-spatial representations to understand 3D scenes. SplatTalk [34] applies 3D Gaussian Splatting [35] to convert posed multi-view RGB images into language-aligned 3D tokens for zero-shot 3D VQA tasks. Despite these advances, current 3D-VL models are limited by sparse supervision in a simple end-to-end training pipeline, with little exploration of intermediate reasoning. In particular, given the overfitting issues of 3D-VL models [36, 24], we argue that the potential of CoT reasoning remains largely untapped for 3D scene understanding, which is a necessary step towards more robust and intelligent 3D-VL models.

Reasoning Capability of LLMs and MLLMs LLMs have demonstrated remarkable reasoning capabilities in across diverse tasks such as math competitions and code generation [37, 21]. These capabilities can be further enhanced by CoT prompting [19], which decomposes complex reasoning into step-by-step subproblems. To unlock the reasoning potential of MLLMs, efforts have focused on bridging the domain gap between visual and textual modalities. For example, Flamingo [38] bridges frozen pretrained vision and language backbones through a Perceiver resampler [39] and interleaved cross-attention, aligning image embeddings with text tokens to improve multi-modal reasoning. Other works emphasize grounded vision-language reasoning, such as Shikra [40] and KOSMOS-2 [41]. OMG-LLaVA [42] proposes a framework to unify pixel-level, object-level, and region-level understanding. Some works studied step-by-step reasoning and integrated the visual search mechanism or Chain-of-Thought prompting for MLLMs. For exmaple, V$*$ [43] proposes a meta-framework to explicitly search the image sequentially, which has been proved helpful for the fine-grained visual recognition tasks, especially for the small object location. Video-of-Thought [44] proposes the first framework in video understanding to implement a human-like perception-to-recognition workflow. Visual COT [22] proposes a COT dataset to enable visual-based Chain-of-Thought training. This work proposes a framework to predict the object bounding box as the intermediate thought, then integrates the patch-level image tokens into the sequence, thus implementing visual-based COT. More recently, GPT-o3 and GPT-o3 mini [45] showcase impressive visual reasoning capabilities to solve a wide range of real-world problems. The models integrate multiple visual experts to build the visual clues that can accurately ground the text information in the images. However, these models rely solely on 2D inputs and lack direct access to 3D-aware visual context, limiting their ability to reason in spatially complex environments.

3 SceneCOT: Step-by-step Reasoning in 3D scenes

In this section, we present the design of SceneCOT, beginning with the formalization of step-by-step reasoning traces in common 3D reasoning tasks (Sec. 3.1). This structure reflects the human problem-solving process for complex understanding, as discussed in Sec. 1. We then describe the detailed learning and inference pipeline of SceneCOT, illustrating how it integrates and leverages 3D-CoTs to enhance reasoning capabilities in 3D scenes (Sec. 3.2).

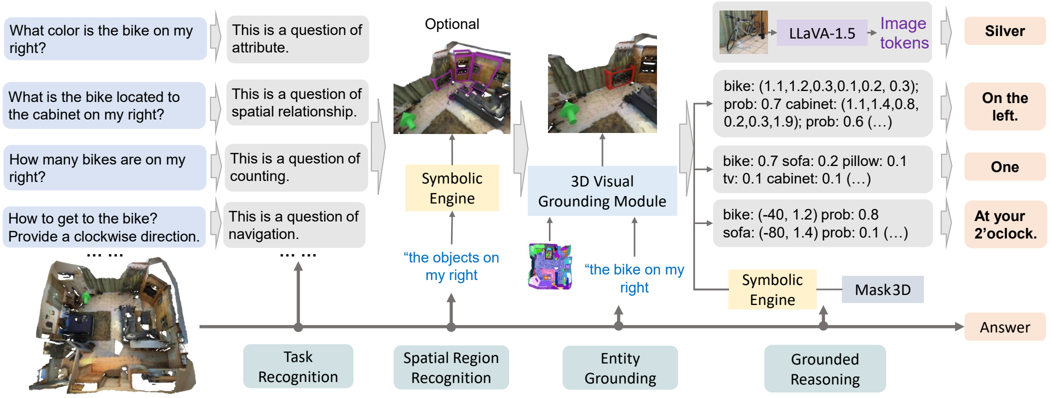

What color is the bike on my right?
What is the bike located to the cabinet on my right?
How many bikes are on my right?
How to get to the bike? Provide a clockwise direction.
… …
This is a question of attribute.
This is a question of spatial relationship.
This is a question of counting.
This is a question of navigation.
… …
Optional
Symbolic Engine
"the objects on my right
3D Visual Grounding Module
"the bike on my right
LLaVA-1.5
Image tokens
bike: (1.1,1.2,0.3,0.1,0.2, 0.3); prob: 0.7 cabinet: (1.1,1.4,0.8, 0.2,0.3,1.9); prob: 0.6 (…)
bike: 0.7 sofa: 0.2 pillow: 0.1 tv: 0.1 cabinet: 0.1 (…)
bike: (-40, 1.2) prob: 0.8 sofa: (-80, 1.4) prob: 0.1 (…)
Symbolic Engine
Mask3D
Silver
On the left.
One
At your 2'oclock.
Answer
Task Recognition
Spatial Region Recognition
Entity Grounding
Grounded Reasoning

Figure 2: **SCENECOT framework.** The diagram above illustrates the reasoning chain and the modular components involved in model training.

3.1 Chain-of-Thoughts in SCENECOT

Given a 3D scene, an agent's situation, and a question to be answered, we define the reasoning trace for answering the question as a concatenation of the following step-by-step descriptions:

1. **Task Recognition and Analysis:** The reasoning trace begins with the identification of the underlying task required to answer the question (*e.g.*, counting, navigation) along with initial analysis for solving it (*e.g.*, ground objects for counting). This information is critical for selecting relevant spatial regions, grounding task-specific entities, and guiding the overall reasoning process. It may also determine which specialized models (*e.g.*, detection, segmentation) to invoke in subsequent steps. We encapsulate this task-level guidance using the special token `<think_type>` within the reasoning trace.
2. **Task-relevant Region Localization:** Based on the task hints in the question and the agent's situation context, we significantly reduce the reasoning space by first providing region-level grounding for localizing the task-relevant subregion of the scene. Specifically, depending on the question, we discretize the surrounding space using either directional clues (*e.g.*, left, right, front, back) or a clock-based reference frame (*e.g.*, 1-12 o'clock). We annotate the selected region with a `<ground_rgn>` token and convert the task analysis from the previous step into region-specific instructions (*e.g.*, "list all objects in this subregion"), enclosed by `<think_rgn>` to indicate region-aware reasoning.
3. **Entity Grounding:** This step grounds the target objects relevant to answering the question. We first generate detailed grounding instructions that encode object semantics, attributes, and relational context. These instructions are enclosed by `<think_grd>`, and followed by a special `[OBJ]` token, which serves as a trigger for invoking specialized grounding modules to localize the referenced object(s).
4. **Grounded Reasoning:** Given the candidate object(s), we generate task-specific instructions, enclosed by `<think_task>`, that specify what information regarding these objects is necessary for downstream reasoning. These instructions guide the grounding model to retrieve or compute relevant information based on task requirements, such as retrieving 2D images for attribute recognition and exporting object class probabilities for object existence verification. We consider the following types of grounded object information, each annotated with distinct tags:

   - *Object probability* (`<obj_prob>`): For tasks such as counting and existence verification, we record the class probabilities of grounded objects, indicating the presence of objects.
   - *3D Object Location with Probability* (`<obj_loc_prob>`, `<obj_loc_plr_prob>`): For tasks that require spatial reasoning, we record object class probabilities along with their positions in 3D (`<obj_loc_prob>`). For navigation-related tasks, we represent object locations in a 2D polar coordinate (`<obj_loc_plr_prob>`) frame to facilitate direction-based reasoning.
   - *Object Image Tokens* (`<highlight_obj>`, `<img_start>`, `<img_end>`): For fine-grained attribute description tasks, we use `<highlight_obj>` to trigger image retrieval, inserting object-level image patches as visual tokens tagged by `<img>` to support appearance-based reasoning.

After obtaining the grounded object information necessary for solving the task, we include a summarization hint marked with `<thin_sum>` to guide the reasoning process, followed by the generation of the final answer, tagged with `<answer>`. An illustrative walkthrough of a sample reasoning trace for a situated question answering task from the MSQA dataset is shown in Fig. 1, with further details on each sub-process of the reasoning trace discussed in Appendix B.1.

3.2 SceneCOT Learning and Inference

We provide an illustrative explanation of how SceneCOT learns and performs inference with our defined 3D-CoTs in Fig. 2. At its core, SceneCOT is built upon a powerful MLLM, which serves as the primary reasoning engine. To support the step-wise reasoning structure introduced in Sec. 3.1, we incorporate three key modular components:

- A symbolic engine(spatial region recognition part) that interprets region-grounding commands tagged with `<ground_rgn>` (*e.g.*, "left", "4 o'clock") to filter out task-irrelevant regions.
- Specialized 3D-VL models for entity grounding, initialized with pre-trained weights and jointly updated during the training of SceneCOT.
- Off-the-shelf parsers and pre-trained 2D-VL models to extract object-specific grounding information (*e.g.*, location, coordinates, image patches) to support grounded reasoning as described in Sec. 3.1. These models are fixed and not updated during SceneCOT learning. We construct the above context using a predefined programming procedure that incorporates symbolic operations(regarded as part of the symbolic engine), referred to as Visual Clue Construction. The implementation details of this procedure are provided in Appendix A.2.

To train SceneCOT, we use a dataset of annotated reasoning traces as described in Sec. 3.1, jointly optimizing the reasoning engine and grounding modules under the following objective:

$$\mathcal{L} = \mathcal{L}_{\text{CoT}} + \mathcal{L}_{\text{ans}} + \mathcal{L}_{\text{ground}}, \tag{1}$$

where $\mathcal{L}_{\text{CoT}}$ and $\mathcal{L}_{\text{ans}}$ are causal language modeling losses for predicting the reasoning trace and final answer, respectively. $\mathcal{L}_{\text{ground}}$ is a cross-entropy loss applied only to the specialized grounding module for accurate object grounding. We train the MLLM model using LoRA.

In the inference stage, SceneCOT follows the reasoning steps specified by the predicted 3D-CoTs, invoking the appropriate modules to generate the final answer. Specifically, we use Mask3D [46] to an initial set of object proposals for the specialized grounding model to select. For special tokens that invoke function calls, the corresponding modules are executed externally, and their outputs are concatenated with prior predictions and fed back into SceneCOT for a new inference pass to complete the reasoning process. Additional details on model inference are provided in the Appendix A.3.

4 The SceneCOT-212K Dataset

To enable the learning of SceneCOT, we develop a large-scale 3D-CoTs dataset, SceneCOT-212K, containing 212K data instances to support step-by-step reasoning in 3D scenes. The dataset comprises two representative tasks in 3D scene reasoning: (1) Situated Reasoning and (2) Object-Centric Reasoning. It follows the standard 3D-CoTs structure as defined in Sec. 3. We construct the dataset through a two-step process: metadata collection and reasoning trace generation. Following MSQA, all the data instances in our dataset share a unified task space, including *Counting*, *Existence*, *Refer*, *Attribute*, *Spatial Relationship*, *Navigation*, *Room type*, *Affordance*, and *Description*. To feasibly adapt to SQA3D's task space, we also set several new task names but keep the same Visual Clue Construction types in MSQA.

4.1 Metadata Collection

For Situated Reasoning, we use MSQA and SQA3D as the data sources. Following the definition in Sec. 3.1, we collect the following components: (1) object instances within the corresponding sub-region, (2) the question type, and (3) the grounding text. For question types, we adopt the primary categorization defined in MSQA and construct the corresponding COT data using the official metadata. As Beacon3D emphasizes grounded reasoning, we treat it as part of the *Attribute/Description* subtasks within our unified task space.

Region-relevant and Question-relevant Objects Extraction First, we design a rule-based procedure to extract target objects based on the agent's location and orientation within the 3D scene. This begins by parsing the question and extracting directional cues using regular expression matching. In MSQA and SQA3D, directional information typically falls into two categories: cardinal directions (left, right, front, back) and clock-based directions (e.g., "at the 1–12 o'clock"). For instance, given the question "How many tables are on my right?", we extract all objects located to the right of the agent. This rule-based method ensures that answers can be accurately inferred from the object list within the corresponding sub-region. Secondly, for the target object entities, we design a rule-based method for *Existence*/*Counting* to ensure the correctness. For the remaining sub-tasks, we inherit the official annotations of the target objects in the released data. Since the target object IDs and question types are not available in SQA3D, we design a semi-automatic pipeline to extract such information using GPT-4o. The details of this pipeline can be found in Appendix B.1.

Data Generation of GQA3D For Beacon3D, we cannot directly construct the COT data due to the absence of metadata. To obtain a high-quality training set, we leverage Nr3D as our metadata source and construct a new grounded question answering dataset, GQA3D. Nr3D is a 3D Visual Grounding benchmark that primarily provides grounding texts and the IDs of target objects. We use GPT-4o to generate QA pairs based on the corresponding object images. All generated QA pairs are categorized as *Attribute* type in SceneCOT-212K. Implementation details are provided in Appendix B.2.

4.2 Reasoning Trace Generation

After collecting the metadata, we construct the full reasoning chain following the steps outlined in Sec. 3. 1) For the sub-tasks *Counting*, *Existence*, *Refer*, *Room Type*, and *Affordance*, we generate ground-truth thoughts using semantic labels and pseudo probabilities. Specifically, we randomly assign values between 0.5 and 1.0 to represent target objects, while assigning values between 0 and 0.5 to non-target objects. To control the token length of the input prompts, we cap the number of objects per training instance. 2) For the *Spatial Relationship* sub-task, we compute relative coordinates using the agent's location, orientation, and the positions of objects under the 3D rectilinear coordinate system. 3) For the *Navigation* sub-task, object locations are represented in a 2D polar coordinate system. To support the *Attribute* and *Description* sub-tasks, which require image retrieval, we construct an object image library. Object images are extracted from RGB frames in ScanNet using object positions and camera poses. We illustrate the overall data distribution in Sec. 4.2, with further implementation details provided in Appendix B.

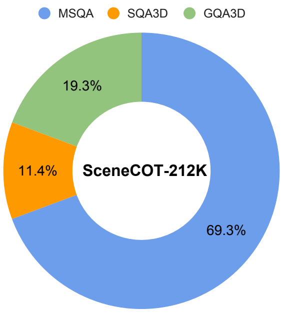


Figure 3: **Data Distribution of SceneCOT-212K.**

5 Experiments

In this section, we compare the overall performance of SceneCOT with previous state-of-the-art models and provide more valuable insights.

5.1 Experimental Setup

Task and Data. In our main experiments, we evaluate the effectiveness of our method on 3D scene reasoning tasks. We focus on two representative tasks: Situated Reasoning and Object-Centric Reasoning. For Situated Reasoning, we use MSQA and SQA3D as benchmark datasets. To ensure consistency across models, we follow the evaluation setting proposed in a recent work [34], using the pure text version of MSQA on ScanNet. We adopt the refined Version-2.1 of the dataset, as recommended by the authors [16]. For fair comparison, we re-implement the results of MSR3D and GPT-4o using the same refined data. Specifically, for MSR3D, we follow the official setup, including the use of the merged dataset and object mask proposals generated by Mask3D. For Object-Centric Reasoning, we adopt the Beacon3D benchmark, which provides a human-annotated grounding-QA chain and a comprehensive evaluation protocol focused on grounding-QA coherence. We follow the official evaluation setting and use the ground-truth object mask proposals. Notably, Beacon3D does

Table 1: **Experimental Results on MSQA and Beacon3D across multiple methods**. *: GPT-4o's input contains ground-truth object labels, locations, and attributes. †: The methods utilize the ground-truth object masks. ‡: our re-implementation on MSQA using the Version-2.1 dataset, official code and prompt of MSQA. In the table, TXT/PCD/IMG refers to the text/point cloud/image input of a 3D scene. In this table, *Spatial* includes *Spatial Relationship* and *Refer*. "S" indicates SQA3D, "M" indicates MSQA, "G" indicates GQA3D.

Methods	Scene Input	**MSQA**							**Beacon3D**		**SQA3D**	
		Count.	*Exist.*	*Attr.*	*Spatial*	*Navi.*	*Others*	Overall	Case	Obj.	EM	EM-R
GPT-4o*‡	TXT	32.3	79.3	**79.0**	37.0	31.7	**91.6**	52.3	57.1	20.2	–	–
LEO‡	PCD	32.5	88.5	58.7	44.2	39.6	81.4	54.8	43.2	7.8	50.0	52.4
MSR3D‡	PCD	32.3	**93.1**	50.0	46.5	**54.1**	75.6	54.2	–	–	–	52.9
PQ3D	PCD+IMG	–	–	–	–	–	–	–	27.8	3.5	47.1	–
SceneVerse	PCD	–	–	–	–	–	–	–	40.3	6.6	49.9	–
ChatScene	PCD+IMG	–	–	–	–	–	–	–	45.8	7.8	54.6	57.5
SplatTalk	IMG	19.6	60.3	44.0	35.8	35.5	61.8	41.8	–	–	47.6	49.9
SceneCOT (S)	PCD+IMG	–	–	–	–	–	–	–	–	–	37.7	41.0
SceneCOT (M+G)	PCD+IMG	**47.9**	82.1	49.6	47.2	51.6	80.3	**55.6**	58.9	**23.2**	–	–
SceneCOT (M+S+G)	PCD+IMG	47.0	84.2	43.8	**47.9**	47.9	78.7	54.0	**60.8**	22.6	**56.5**	**59.3**

not include a training set, making it well-suited for evaluating the generalization quality of our newly constructed dataset, GQA3D.

Evaluations and Baselines. As recommended by the authors of MSQA and Beacon3D, we adopt GPT-score as the primary evaluation metric. Following prior work, we also report Exact Match (EM) and Refined Exact Match Accuracy (EM-R) on SQA3D. For MSQA and SQA3D, we use predicted object masks and semantic labels, while for Beacon3D, we adhere to the official settings for baseline models and use ground-truth object mask proposals. We adopt two metrics to comprehensively evaluate the performance on Beacon3D: GPT-Score by "case" and GPT-Score by "object". "By case" averages the scores cross independent QA pairs. While "by object" is more strict: it averages the ratio of the objects for which all the QA pairs are correct. For baselines, we compare our method with several recent and strong models, including GPT-4o [47], MSR3D [16], PQ3D [7], Chat-Scene [32], SceneVerse [6], and SplatTalk [34]. Please note that our method **excludes** scene tokens from the LLM's input during reasoning, but exploits the visual clue through reasoning and Visual Clue Construction.

Implementation Details. We build SceneCOT based on LLaVA-1.5, an open-source multimodal LLM framework built upon Vicuna-7B. For the *Attribute* and *Description* sub-tasks, the selected object image is passed through a 2D vision encoder. The resulting image feature is then projected into the language embedding space via learnable projection layers. During training, we freeze these projection layers and apply LoRA to fine-tune the parameters of the LLM. For 3D visual grounding, we adopt a fine-tuned version of PQ3D as our expert model. This model has been trained on a subset of the domains provided in [6]. Additionally, we design a lightweight object mask predictor to estimate object logits. This module consists of a grounding head and a text embedding head, each implemented as a two-layer multilayer perceptron (MLP) with layer normalization. To compute the text embedding, we take the average of the token embeddings from the grounding text. During training, we fine-tune both the parameters of PQ3D and the object mask predictor. Following the strategy proposed in [48], we select 17 of the least frequently used tokens in Vicuna-7B's vocabulary to serve as special tokens. This approach allows us to avoid modifying the vocabulary itself, resulting in a more stable fine-tuning process. Our model is trained for 5 epochs using 4 GPUs.

5.2 Main Results on Situated Reasoning and Object-Centric Reasoning

In Tab. 1, we present the main results of various methods. As a novel reasoning framework, SceneCOT achieves SoTA performance across all three benchmarks. On SQA3D, our method outperforms Chat-Scene by a notable margin (56.5 vs. 54.6 on EM, 59.3 vs. 57.5 on EM-R). Notably, on Beacon3D, SceneCOT surpasses the previous SoTA by a significant margin (60.8 vs. 45.8 by case, 23.2 vs. 7.8 by object). In addition to the strong performance, our study yields several valuable observations.

SceneCOT significantly advances the state-of-the-art on *counting*—the most challenging sub-task in MSQA. This improvement is expected, as our method explicitly enumerates objects within the relevant sub-region based on semantic similarity to the target object mentioned in the question,

Table 2: **Experimental results on oracle data**. In our main results, we utilize Mask3D to provide object masks and semantic labels. In this table, we explore the upper boundary in two aspects: 1) perfect object masks and semantic labels, but still based on the predicted object probabilities. 2) Oracle ground-truth text-based thought.

Error sources	*Count.*	*Exist.*	*Attri.*	*Spatial.*	*Refer*	*Navi.*	*Others*	Overall
❶ + ❷	47.9	82.1	49.6	49.8	31.9	51.6	80.3	55.6
❷	73.3	86.5	63.3	49.9	67.2	55.4	81.8	64.9
–	98.8	100.0	63.3	55.7	84.9	87.2	86.8	78.1

offering a clear visual grounding for reasoning. SceneCOT also delivers strong performance on *Spatial* tasks, including both *Refer* and *Spatial Relationship*. However, it shows a performance drop on *Existence* and *Attribute* compared to LEO and MSR3D. We attribute this to grounding errors and discuss this in greater detail in Sec. 5.3.

More training data significantly enhances reasoning performance. We evaluate the impact of different training data mixtures using three configurations: (1) SQA3D alone, (2) SQA3D combined with GQA3D, and (3) SQA3D, GQA3D, and MSQA jointly. The results indicate that co-training substantially boosts performance on SQA3D, with Exact Match (EM) improving from 37.7 to 56.5. This gain is expected, as SQA3D contains a relatively small training set (approximately 24K examples), which limits the model's ability to learn complex reasoning and multimodal representations. GQA3D also benefits from co-training. In contrast, MSQA exhibits a slight performance decline when trained jointly with SQA3D, likely due to task distribution differences between the two datasets.

SceneCOT outperforms all baseline SoTA methods by a large margin on Beacon3D. Notably, our model achieves superior performance under the per-object evaluation metric, which is particularly challenging. This metric demands strong reasoning capabilities for individual objects, as it is a strict binary score: a score of 100 is assigned only if all answers related to a given object are correct; otherwise, the score is zero. These results highlight the effectiveness of the grounded reasoning chain in SceneCOT. Importantly, Beacon3D includes only a test set with high-quality human annotations. Therefore, the strong performance not only validates the effectiveness of our reasoning framework but also reflects the quality of our newly generated GQA3D dataset. A detailed analysis of QA-grounding coherence, which further illustrates the benefits of our approach, is provided in Appendix C.2.

5.3 Ablation Study and Analyses

Exploring The Upper Boundary of SceneCOT. In Tab. 1, the results on MSQA are based on predicted semantic labels and object masks, whereas during training, we use ground-truth visual clues. This raises a natural question: How would the model perform if the errors introduced by the Visual Clue Construction process were eliminated? To answer this, we conduct experiments using oracle data during the reference stage and analyze the impact of two main error sources:

❶ Semantic errors, including inaccurate semantic labels and object masks. These lead to noisy grounding features in 3D Visual Grounding. ❷ Grounding errors, where the grounding module predicts incorrect object probabilities, resulting in misleading visual clues. In the second row of Tab. 2, we remove the influence of error ❶ by providing perfect semantic labels and object masks. This significantly improves performance across all sub-tasks, especially for *Counting*, which relies on accurate multi-object recognition—a challenging task when using noisy masks. The third row further removes error ❷, giving the model perfect object probabilities, coordinates, and semantic labels. Under these ideal conditions, performance reaches near-maximum: *Counting/Existence* approach 100, and *Navigation/Refer* achieve 87.2 and 84.9, respectively. These results highlight the effectiveness of our 3D-CoTs design. For example, in *Navigation*, our use of a 2D polar coordinate system provides an abstracted yet informative representation of object-agent relationships. *Spatial Relationship* remains the most challenging, as it requires reasoning from numerical coordinates to natural spatial descriptions (e.g., "left of," "beside," or "adjacent to"). For instance, answering a question like "What is the spatial relationship between the table and the brown desk?" demands strong spatial reasoning based on agent-centered perspectives. Enhancing location representations and reasoning abilities could further improve performance in this area. In summary, accurate semantic labels, object masks, and object probabilities during inference all contribute significantly to model performance. We hope these findings offer valuable insights for building more advanced models for 3D scene reasoning.

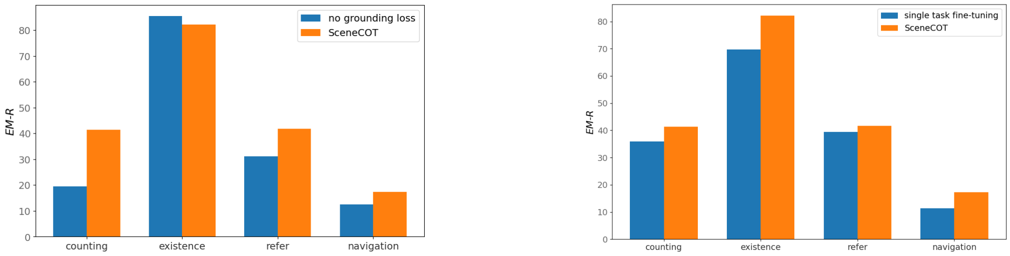
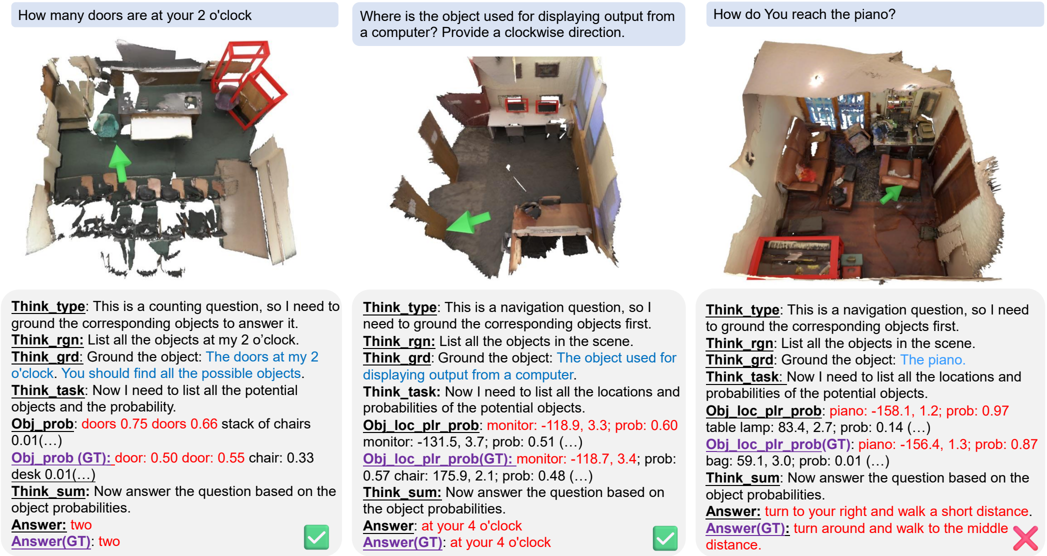


Figure 4: **Ablation study of grounding loss and training strategy**. We chose four indicating sub-tasks to verify the necessity of grounding loss and the usefulness of co-training. In this table, we adopt EM-R(EM-Refined) as the evaluation metric.

Grounding Loss Matters. In SCENECOT, we introduce a grounding module based on PQ3D and fine-tune it during training. To validate the importance of this design, we conduct an ablation study by removing the grounding loss term from the overall loss in Eq. (1), and evaluate the impact on MSQA. As shown in the left panel of Fig. 4, removing the grounding loss leads to a noticeable performance drop across several sub-tasks, including *Counting*, *Refer*, and *Navigation*. Interestingly, the performance on *Existence* does not follow this trend as clearly. We hypothesize that this is because the model can still rely on semantic labels during inference to answer existence-type questions.

Co-training Benefits Model Training. Our method designs multiple COT templates tailored to different tasks. To investigate whether co-training improves sub-task performance, we analyze results shown in the right panel of Fig. 4. The findings confirm that co-training yields clear benefits for most sub-tasks. During co-training, data from all sub-tasks jointly optimize the grounding module, leading to more accurate entity grounding and, consequently, improved final predictions.

5.4 Reasoning Chain Visualization


How many doors are at your 2 o'clock

Where is the object used for displaying output from a computer? Provide a clockwise direction.

How do You reach the piano?

Think_type: This is a counting question, so I need to ground the corresponding objects to answer it.
Think_rgn: List all the objects at my 2 o'clock.
Think_grd: Ground the object: The doors at my 2 o'clock. You should find all the possible objects.
Think_task: Now I need to list all the potential objects and the probability.
Obj_prob: doors 0.75 doors 0.66 stack of chairs 0.01(…)
Obj_prob (GT): door: 0.50 door: 0.55 chair: 0.33 desk 0.01(…)
Think_sum: Now answer the question based on the object probabilities.
Answer: two
Answer(GT): two
✓

Think_type: This is a navigation question, so I need to ground the corresponding objects first.
Think_rgn: List all the objects in the scene.
Think_grd: Ground the object: The object used for displaying output from a computer.
Think_task: Now I need to list all the locations and probabilities of the potential objects.
Obj_loc_plr_prob: monitor: -118.9, 3.3; prob: 0.60 monitor: -131.5, 3.7; prob: 0.51 (…)
Obj_loc_plr_prob(GT): monitor: -118.7, 3.4; prob: 0.57 chair: 175.9, 2.1; prob: 0.48 (…)
Think_sum: Now answer the question based on the object probabilities.
Answer: at your 4 o'clock
Answer(GT): at your 4 o'clock
✓

Think_type: This is a navigation question, so I need to ground the corresponding objects first.
Think_rgn: List all the objects in the scene.
Think_grd: Ground the object: The piano.
Think_task: Now I need to list all the locations and probabilities of the potential objects.
Obj_loc_plr_prob: piano: -158.1, 1.2; prob: 0.97 table lamp: 83.4, 2.7; prob: 0.14 (…)
Obj_loc_plr_prob(GT): piano: -156.4, 1.3; prob: 0.87 bag: 59.1, 3.0; prob: 0.01 (…)
Think_sum: Now answer the question based on the object probabilities.
Answer: turn to your right and walk a short distance.
Answer(GT): turn around and walk to the middle distance.
✗

Figure 5: **Visualization of qualitative examples of SCENECOT.** We select two indicating sub-tasks *Counting* and *Navigation* to illustrate the reasoning traces of SCENECOT. Left:SCENECOT correctly constructs the visual clue and reasons the correct answer. Middle: SCENECOT correctly answers the question based on the accurate relative location. Right: Even though the visual clue exactly matches the correct entity, the model summarizes to the wrong answer owing to the limited reasoning capability.

Finally, we present a case study in Fig. 5 to provide an intuitive understanding of the strengths of our framework. In the first example, the model grounds the doors located at the agent's 2 o'clock direction and constructs the corresponding visual clue `obj_prob`. Although the predicted labels differ slightly from the ground-truth label "door," the accurate object probabilities and semantic similarity guide the model to generate the correct final answer. In the second example, the model identifies all monitors in the scene, then uses the symbolic engine to compute their relative positions using 2D polar coordinates. This spatial information is integrated with object probabilities. After analyzing the full set of probabilities, the model infers the direction “4 o’clock” based on the angle value “–118.9°.” These two cases demonstrate clear and interpretable reasoning chains for challenging tasks, enabling easier diagnosis of error sources. In the third example, despite an accurate visual clue, the model fails in the final reasoning step. This reflects the observation in Tab. 2—even with oracle visual inputs, the performance on *Navigation* reaches only 87.2, indicating a gap due to limited reasoning capability in the base model. We argue that this performance ceiling can be lifted as foundation models like multimodal LLM continue to evolve. In summary, the visualizations illustrate both the advances and remaining limitations of SceneCOT. We hope these analyses contribute to the development of more robust and interpretable reasoning frameworks for 3D scene understanding.

6 Conclusion

In this paper, we present SceneCOT, an innovative framework for complex 3D scene reasoning that emulates the human reasoning process—from high-level task understanding to low-level visual-semantic grounding. To support this framework, we introduce a large-scale 3D CoT (3D-CoT) dataset, SceneCOT-212K, which contains rich and diverse step-by-step annotations of reasoning traces to facilitate the learning of chain-of-thought (CoT) reasoning in 3D environments. Through a carefully designed pipeline, we enable existing MLLMs to perform human-like CoT reasoning for solving complex 3D tasks. Experimental results demonstrate that our method achieves state-of-the-art performance on key 3D scene reasoning benchmarks, including both situated and object-centric reasoning tasks. Moreover, it provides more interpretable and explainable reasoning traces compared to prior approaches. Our in-depth analyses highlight the strengths of the proposed framework and offer insights into its underlying mechanisms. We also conduct comprehensive upper-bound studies to explore the performance ceiling of our method, offering guidance for the development of more advanced reasoning models in 3D scene understanding.

References

[1] Yixin Chen, Siyuan Huang, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[2] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1

[3] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[4] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1

[5] 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*, 2024. 1, 3

[6] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. *arXiv preprint arXiv:2401.09340*, 2024. 1, 7, 19

[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. In *European Conference on Computer Vision*, pages 188–206. Springer, 2024. 1, 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. In *International Conference on Machine Learning (ICML)*, 2024. 1, 2, 3

[9] Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. In *International Conference on Computer Vision (ICCV)*, 2023. 1

[10] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. In *International Conference on Learning Representations (ICLR)*, 2024. 1

[11] Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven CH Hoi. From images to textual prompts: Zero-shot vqa with frozen large language models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[12] Jianjiu Ou, Jianlong Zhou, Yifei Dong, and Fang Chen. Chain of thought prompting in vision-language model for vision reasoning tasks. In *Australasian Joint Conference on Artificial Intelligence*, 2024. 1

[13] Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. Visual chain-of-thought prompting for knowledge-based visual reasoning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2024. 1

[14] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 3

[15] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 2

[16] Xiongkun Linghu, Jiangyong Huang, Xuesong Niu, Xiaojian Ma, Baoxiong Jia, and Siyuan Huang. Multi-modal situated reasoning in 3d scenes, 2024. 1, 2, 6, 7

[17] 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 (NeurIPS)*, 2023. 2, 3

[18] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. *arXiv preprint arXiv:2412.00493*, 2024. 2, 3

[19] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 2, 3

[20] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2, 23

[21] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 2, 3

[22] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Unleashing chain-of-thought reasoning in multi-modal language models. *arXiv e-prints*, pages arXiv–2403, 2024. 2, 3

[23] Yuecheng Liu, Dafeng Chi, Shiguang Wu, Zhanguang Zhang, Yaochen Hu, Lingfeng Zhang, Yingxue Zhang, Shuang Wu, Tongtong Cao, Guowei Huang, et al. Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning. *arXiv preprint arXiv:2501.10074*, 2025. 2

[24] 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. *arXiv preprint arXiv:2503.22420*, 2025. 2, 3, 22, 24

[25] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023. 3

[26] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[27] Jianing Yang, Xuweiyi Chen, Nikhil Madaan, Madhavan Iyengar, Shengyi Qian, David F Fouhey, and Joyce Chai. 3d-grand: A million-scale dataset for 3d-llms with better grounding and less hallucination. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[28] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, 2025. 3

[29] Tao Chu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Qiong Liu, and Jiaqi Wang. Unified scene representation and reconstruction for 3d large language models. *arXiv preprint arXiv:2404.13044*, 2024. 3

[30] Zhuofan Zhang, Ziyu Zhu, Pengxiang Li, Tengyu Liu, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Siyuan Huang, and Qing Li. Task-oriented sequential grounding in 3d scenes. *arXiv preprint arXiv:2408.04034*, 2024. 3

[31] Zhangyang Qi, Ye Fang, Zeyi Sun, Xiaoyang Wu, Tong Wu, Jiaqi Wang, Dahua Lin, and Hengshuang Zhao. Gpt4point: A unified framework for point-language understanding and generation. *arXiv preprint arXiv:2312.02980*, 2023. 3

[32] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15459–15469, 2024. 3, 7

[33] 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*, 2024. 3

[34] Anh Thai, Songyou Peng, Kyle Genova, Leonidas Guibas, and Thomas Funkhouser. Splattalk: 3d vqa with gaussian splatting. *arXiv preprint arXiv:2503.06271*, 2025. 3, 6, 7

[35] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023. 3

[36] Weipeng Deng, Runyu Ding, Jihan Yang, Jiahui Liu, Yijiang Li, Xiaojuan Qi, and Edith Ngai. Can 3d vision-language models truly understand natural language? *arXiv preprint arXiv:2403.14760*, 2024. 3

[37] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 3

[38] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[39] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[40] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 3

[41] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 3

[42] Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. Omg-llava: Bridging image-level, object-level, pixel-level reasoning and understanding. *Advances in Neural Information Processing Systems*, 37:71737–71767, 2024. 3

[43] Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024. 3

[44] Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. Video-of-thought: Step-by-step video reasoning from perception to cognition. *arXiv preprint arXiv:2501.03230*, 2024. 3

[45] OpenAI. Introducing openai o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/, April 16 2025. 3, 23

[46] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105*, 2022. 5

[47] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 7

[48] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023. 7

[49] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 19

[50] Zhuofan Zhang, Ziyu Zhu, Pengxiang Li, Tengyu Liu, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Siyuan Huang, and Qing Li. Task-oriented sequential grounding in 3d scenes. *arXiv preprint arXiv:2408.04034*, 2024. 23

Appendix

A SceneCOT

A.1 Data Examples of Reasoning Traces

We provide examples of COT templates in SceneCOT cross all the sub-tasks, including the integral special tokens and textual context.

Question: You are climbing the shelf. There is a black and white monitor and a white and black monitor at your 5 o'clock. There is a brown door at your 2 o'clock. How many pillows are on your right in the middle distance?
Response:
<think_type>This is a counting question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects on my right in the middle distance</think_rgn>
<think_grd>Ground the object: The pillows on my right in the middle distance. You should find all the possible objects.</think_grd>
[OBJ]
<think_task>Now I need to list all the potential objects and the probability.</think_task>
<list_obj_prob>
<obj_prob>pillow: 0.74 pillow: 0.78 cabinet: 0.31 legs: 0.01 cabinet: 0.29 pillow: 0.14 pillow: 0.40 file cabinet: 0.31 wall: 0.04 whiteboard: 0.45</obj_prob>
<think_sum>Now answer the question based on the object probabilities.</think_sum>
<answer>two</answer>

Figure 6: A COT example of *Counting*.

Question: You are painting the door. There is a brown trash can in your 9 o'clock direction. There is a turquoise table with a smooth texture and a four-legged structure in your 6 o'clock direction. Is there a sink in the room?
Response:
<think_type>This is an existence question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects in the scene.</think_rgn>
<think_grd>Ground the object: The sinks in the room.</think_grd>
[OBJ]
<think_task>Now I need to list all the potential objects and the probability.</think_task>
<list_obj_prob>
<obj_prob>sink: 0.95 traffic cone: 0.24 folded table: 0.06 ladder: 0.20 trash can: 0.45 vacuum cleaner: 0.35 ladder: 0.19 ledge: 0.14 table: 0.43 floor: 0.23 recycling bin: 0.13 recycling bin: 0.05 rack: 0.11 folded table: 0.08 boxes: 0.46 trash can: 0.40 trash can: 0.36 folded table: 0.16 wall: 0.28 door: 0.48 doorframe: 0.37 wall: 0.28 sink: 0.04 table: 0.15 wall: 0.18 wall: 0.38 trash can: 0.26 wall: 0.32 wall: 0.18 wall: 0.04</obj_prob>
<think_sum>Now answer the question based on the object probabilities.</think_sum>
<answer>yes</answer>

Figure 7: A COT example of *Existence*.

Question: There is a window in your 7 o'clock direction. There is a black monitor with a smooth texture in your 3 o'clock direction. What object is brown and used for seating at your 6 o'clock?
Response:
<think_type>This is a refer question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects at my 6 o'clock.</think_rgn>
<think_grd>Ground the object: The brown object used for seating at your 6 o'clock.</think_grd>
[OBJ]
<think_task>Now I need to list all the potential objects and the probability.</think_task>
<list_obj_prob>
<obj_prob>chair: 0.98 wall: 0.30 floor: 0.33 chair: 0.48 chair: 0.44 table: 0.10 window: 0.26 backpack: 0.40</obj_prob>
<think_sum>Now answer the question based on the object probabilities.</think_sum>
<answer>the chair</answer>

Figure 8: A COT example of *Refer*.

Question: You are replacing the trash bag in the trash can. In front of you, there's a beige wooden door. To your left, at a middle distance, there's a dark rectangular suitcase with a hard shell. What is the state of the door on your right?
Response:
<think_type>This is an attribute question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects in the sub-region related to the question.</think_rgn>
<think_grd>Ground the object: The door on my right.</think_grd>
[OBJ]
<think_task>Now I need to retrieve the image of the target object.</think_task>
<highlight_obj>
<think_sum>Now answer the question based on the object image.</think_sum>
<answer>closed</answer>

Figure 9: A COT example of *Attribute*.

Question: A rack stand is at your 5 o'clock. At your 2 o'clock, there is a book rack with upright books and books. What is the usage of the books on your left?
Response:
<think_type>This is an affordance question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects in the sub-region related to the question.</think_rgn>
<think_grd>Ground the object: The books on my left.</think_grd>
[OBJ]
<think_task>Now I need to retrieve the image of the target object.</think_task>
<highlight_obj>
<think_sum>Now answer the question based on the object image.</think_sum>
<answer>reading, information, decoration</answer>

Figure 10: A COT example of *Affordance*.

Question: You are sitting at the desk. Behind you, there's a burgundy backpack. From where You are, how do You get to the door? Please provide a direction (behind, in front of, left, right) and the distance information (far/middle/near).

Response:

<think_type>This is a navigation question, so I need to ground the corresponding objects first.</think_type>

<grd_rgn>

<think_rgn>Now I need to list all the objects in the scene.</think_rgn>

<think_grd>Ground the object: The door.</think_grd>

[OBJ]

<think_task>Now I need to list all the locations(polar coordinate: angle, distance(/m)) and probabilities of the potential objects.</think_task>

<list_obj_loc_plr_prob>

<obj_loc_plr_prob>door: -86.1, 1.9; prob: 0.79 wall: -117.6, 2.7; prob: 0.01 window: 154.2, 2.0; prob: 0.25 fan: 152.1, 1.6; prob: 0.37 clothes: 96.4, 3.4; prob: 0.41</obj_loc_plr_prob>

<think_sum>Now answer the question based on the object probabilities.</think_sum>

<answer>turn right and walk to the middle distance.</answer>

Figure 11: A COT example of *Navigation*.

Question: You are hanging the picture. There is a blue tissue box at your 10 o'clock direction, with a smooth cardboard texture, dispensing tissues. A cluttered dark table is at your 7 o'clock direction. Where is the book located in relation to the desk?

Response:

<think_type>This is a spatial relationship question, so I need to ground the corresponding objects first.</think_type>

<grd_rgn>

<think_rgn>Now I need to list all the objects in the scene.</think_rgn>

<think_grd>Ground the object: The book and the desk.</think_grd>

[OBJ]

<think_task>Now I need to list all the locations and probabilities of the potential objects.</think_task>

<list_obj_loc_prob>

<obj_loc_plr_prob>book: -1.6,4.8,0.8,0.3,0.3,0.1; prob: 0.89 desk: -0.9,5.0,0.5,1.7,0.8,0.9; prob: 0.59 yoga mat: -2.5,4.6,0.7,0.6,0.2,0.2; prob: 0.46 shelf: -1.2,1.3,1.0,0.6,0.9,2.1; prob: 0.05 sink: 3.0,2.6,0.8,0.4,0.5,0.4; prob: 0.06 bag: -1.5,2.8,0.1,0.6,0.3,0.3; prob: 0.14 book: -1.6,4.8,0.8,0.3,0.3,0.1; prob: 0.17 kitchen island: 2.7,2.6,0.4,1.2,1.8,1.0; prob: 0.36 refrigerator: 2.8,4.6,0.9,1.0,0.4,1.7; prob: 0.11 lamp base: -1.4,2.2,0.0,0.2,0.2,0.1; prob: 0.33</obj_loc_plr_prob>

<think_sum>Now answer the question based on the object probabilities.</think_sum>

<answer>the book is placed within the area of the desk.</answer>

Figure 12: A COT example of *Spatial Relationship*.

A.2 Definition of the Symbolic Engine

Our symbolic engine serves two primary functions:

1. **Spatial Region Recognition**: It partitions the environment into sub-regions based on the agent's location and orientation.

2. **Visual Clue Construction**: It generates textual visual clues by integrating object probabilities, semantic labels, and spatial locations.

Question: What color is the printer on top of the nightstand?
Response:
<think_type>This is a appearance (grounded qa) question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects in the scene.</think_rgn>
<think_grd>Ground the object: The printer on top of the nightstand.</think_grd>
[OBJ]
<think_task>Now I need to retrieve the image of the target object.</think_task>
<highlight_obj>
<think_sum>Now answer the question based on the object image.</think_sum>
<answer>Black</answer>

Figure 13: A COT example of *GQA3D*.

Question: To your left, there's a brown wooden nightstand with a smooth texture and a box with a drawer. To your right, near you, there's a gray and white fabric cushion. Based on the objects around you, what type of room are You likely in?
Response:
<think_type>To your left, there's a brown wooden nightstand with a smooth texture and a box with a drawer. To your right, near you, there's a gray and white fabric cushion. Based on the objects around you, what type of room are You likely in?</think_type>
<think_type>This is a room type question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects in the scene.</think_rgn>
<think_grd>Ground the object: The objects around me that can indicate the type of the room.</think_grd>
[OBJ]
<think_task>Now I need to list all the potential objects and the probability.</think_task>
<list_obj_prob>
<obj_prob>stack of chairs 0.63 stack of chairs 0.62 bed 0.60 dresser 0.60 dresser 0.60 dresser 0.60 desk 0.60 stick 0.58 pillows 0.58 shelf 0.58 monitor 0.58 stick 0.57 shelf 0.57 backpack 0.57 trash can 0.56 cabinets 0.55 shoe 0.55 laundry hamper 0.54 clothing 0.54 pillows 0.54 shoe 0.54 shoe 0.54 shoe 0.54 trash can 0.54 trash can 0.53 shoe 0.53 window 0.49 window 0.49 window 0.49 doors 0.40</obj_prob>
<think_sum>Now answer the question based on the object probabilities.</think_sum>
<answer>a bedroom or dorm room</answer>

Figure 14: A COT example of *Room type*.

A.2.1 Spatial Region Recognition

In MSQA, there are two types of directional reference policies: 1) *Cardinal-Relative Directions*, such as left, right, front, and behind. 2) *Clock-based directions*, such as 1 to 12 o'clock positions. Here we provide some code fragments of our implementation.

1. Calculate the cardinal-relative sub-regions. This function is used to calculate the object lists of four directions. Since MSQA also requires the distance information in some scenarios, we also develop the function to support this feature.

```

# calculate the sub-graphs for Cardinal-Relative Directions

face_pt = np.array([np.cos(direction), np.sin(direction)])

stand_pt = np.array([stand_on_loc[0], stand_on_loc[1]])
pcds = np.array(pcds)
pcd_2d = pcds[:, :2]

```

Question: To your right, near you, there's a black legs and an orange pillow. In front of you, at a middle distance, there's a white wooden shelf. Can you describe the couch?
Response:
<think_type>This is a description question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects in the scene.</think_rgn>
<think_grd>Ground the object: The couch.</think_grd>
[OBJ]
<think_task>Now I need to retrieve the image of the target object.</think_task>
<highlight_obj>
<think_sum>Now answer the question based on the object image.</think_sum>
<answer>it's a black couch with yellow and orange pillows, made of fabric upholstery, designed for seating, with a smooth texture, cushioned with armrests, and it's in use.</answer>

Figure 15: A COT example of *Description*.

```
pcd_2d = pcd_2d - stand_pt
pcd_2d_norm = np.linalg.norm(pcd_2d, axis=1)
pcd_2d_norm[pcd_2d_norm == 0] = 1
pcd_2d_norm = np.expand_dims(pcd_2d_norm, axis=1)
pcd_2d = pcd_2d / pcd_2d_norm

### cal the angle between face_pt and pcd_2d
sum_all = np.dot(pcd_2d, face_pt)
sum_all[sum_all >= 1] = 1
sum_all[sum_all <= -1] = -1
angle = np.arccos(sum_all)
angle = angle / np.pi * 180

### cal the cross value between face_pt and pcd_2d
cross = np.cross(face_pt, pcd_2d)

front_mask = (angle < 30)
back_mask = (angle > 150)
left_mask = (cross > 0) & (angle > 30) & (angle < 150)
right_mask = (cross < 0) & (angle > 30) & (angle < 150)

front_list = get_inst_id(inst_label, front_mask)
back_list = get_inst_id(inst_label, back_mask)
left_list = get_inst_id(inst_label, left_mask)
right_list = get_inst_id(inst_label, right_mask)
```

2. Calculate the clock-based sub-regions. This function is used to calculate the clockwise information for each object.

```
def cal_clock_wise_direction(stand_on_pt, face_pt, inst_pcd):
    inst_pcd = inst_pcd[:,:2]
    # inst_loc = (inst_pcd.max(0) + inst_pcd.min(0)) / 2
    inst_loc = inst_pcd.mean(0)
    obj_dir = inst_loc - stand_on_pt
    if np.abs(obj_dir).sum() < 1e-6:
        return None, None

    obj_dir = obj_dir / np.linalg.norm(obj_dir)
    face_pt = face_pt / np.linalg.norm(face_pt)

    ### cal the angle between obj_dir and face_dir
    angle = np.dot(obj_dir, face_pt)
    angle = np.clip(angle, -1, 1)
    angle = np.arccos(angle)
    direct = np.cross(face_pt, obj_dir)
    if direct > 0:
        angle = 2 * np.pi - angle
    angle = angle / np.pi * 180

    clockwise_direction = round(angle / 30) % 12
    clockwise_direction = 12 if clockwise_direction == 0 else clockwise_direction
    clockwise_direction = int(clockwise_direction)

    return clockwise_direction, angle
```

3. We parse the text between <think_rgn> and </think_rgn> to obtain the user's directional instruction, then set query_type by matching it to known phrases like “on my left” or “at my 10 o'clock”.

```
# query: the object list of: cardinal direction-based and clock-based
# query type: four direction or clockwise or whole_scene
if query_type == 'cardinal' or query_type == 'clockwise':
    parse_words = ['left', 'right', 'front', 'back', 'behind'] if query_type == 'cardinal' else [f"{i} o'clock" for i in range(12)]
    direction = parse_direction_distance(question, parse_words)
    direction = direction if direction != 'behind' else 'back'
    distance = parse_direction_distance(question, ['far', 'middle', 'near'])
    if direction:
        obj_list, label_ids = get_obj_list(query, query_type, direction, distance)
        count_dict = Counter(obj_list)
    else:
        # no direction found, collect objects in all directions
        obj_list, label_ids = [], []
        distance = parse_direction_distance(question, ['far', 'middle', 'near'])
        for direction in query.keys():
            obj_list_temp, label_ids_temp = get_obj_list(query, query_type, direction, distance)
            obj_list += obj_list_temp
            label_ids += label_ids_temp
        count_dict = Counter(obj_list)
else:
    obj_list, label_ids = get_obj_list(query, query_type, '', '')
    count_dict = Counter(obj_list)
```

A.2.2 Visual Clue Construction

In the Grounded Reasoning step, the model generates the final answer by integrating both semantic information and a visual clue derived from object probabilities. We outline the algorithm for constructing this visual clue during the reasoning process. Given a prefix sequence $\mathcal{P}$ (i.e., the input_ids of the text prompt), the algorithm returns a modified sequence that incorporates relevant visual clues based on object probabilities, spatial locations, and semantic labels. This enhanced sequence is then fed back into the LLM to generate the final response. The component functions—build_obj_prob, build_obj_loc_prob, and build_obj_loc_plr_prob—are illustrated in Fig. 18 and Fig. 19. Additionally, Fig. 16 provides a visual comparison between the 2D polar coordinate system and the 3D spatial coordinate system used in our framework.

A.3 Implementation Details of the 3D Visual Grounding Module

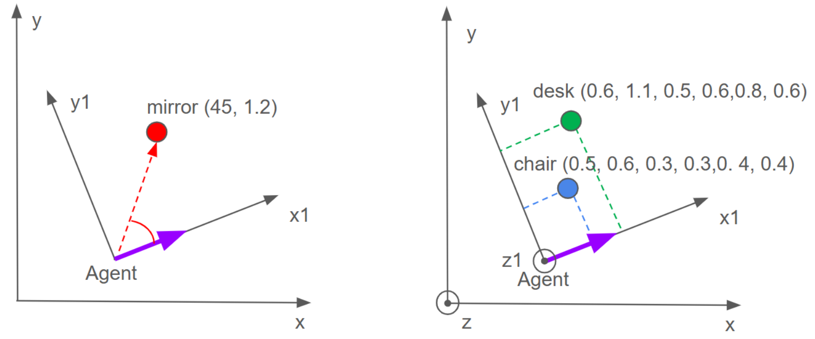
We illustrate the implementation of 3D Visual Grounding Module in Fig. 17. In the framework, the object feature is extracted by PQ3D, while the text embedding is extracted by the embedding tokenizer based on the grounding text. We train PQ3D based on the grounding data in [6], including the scenes in ScanNet, 3RScan, and MultiScan.

A.4 Inference

Recently, several state-of-the-art industrial LLMs—such as DeepSeek-V3 [49]—have adopted the Mixture-of-Experts (MoE) technique as a key performance-enhancing strategy. Inspired by this

Algorithm 1: Visual Clue Construction for Grounded Reasoning

```
Require: Object probabilities $\mathcal{P} \in R^{N\times 1}$, prefix sequence $S \in R^{N\times M}$, object locations and
         sizes $\mathcal{O}_L \in R^{N\times 6}$, object semantic labels $\mathcal{O}_{SL}(N \times 1)$ , object images
         $\mathcal{O}_I \in R^{N\times 3\times H\times W}$, maximum object number $K$
if LIST_OBJ_PROB_TOKEN_IDX in $S$ then
    obj_prob_content ← build_obj_prob($\mathcal{P}, \mathcal{O}_L, \mathcal{O}_{SL}, K$);
    index ← Index($S$, LIST_OBJ_PROB_TOKEN_IDX);
    new_sequence ← cat($S$[:index+1], get_text_embeddings(obj_prob_content));
else if LIST_OBJ_PROB_LOC_TOKEN_IDX in $S$ then
    obj_prob_loc_content ← build_obj_loc_prob($\mathcal{P}, \mathcal{O}_L, \mathcal{O}_{SL}, K$);
    index ← Index($S$, LIST_OBJ_PROB_LOC_TOKEN_IDX);
    new_sequence ← cat($S$[:index+1],
      get_text_embeddings(obj_prob_loc_content));
else if LIST_OBJ_PROB_LOC_PLR_TOKEN_IDX in $S$ then
    obj_prob_loc_plr_content ← build_obj_loc_plr_prob($\mathcal{P}, \mathcal{O}_L, \mathcal{O}_{SL}, K$);
    index ← Index($S$, LIST_OBJ_PROB_LOC_PLR_TOKEN_IDX);
    new_sequence ← cat($S$[:index+1],
      get_text_embeddings(obj_prob_loc_plr_content));
else if HIGHLIGHT_OBJ_TOKEN_IDX in $S$ then
    top_k_indices ← TopK($\mathcal{P}$);
    img ← $\mathcal{O}_I$[top_k_indices[0]];
    img_tokens ← Projector(Img_encoder(img));
    index ← Index($S$, HIGHLIGHT_OBJ_TOKEN_IDX);
    new_sequence ← cat($S$[:index+1],
      get_text_embeddings(IMG_START_TOKEN_IDX), img_tokens,
      get_text_embeddings(IMG_END_TOKEN_IDX));
return new_sequence
```


y
y1
mirror (45, 1.2)
x1
Agent
x
y
y1
desk (0.6, 1.1, 0.5, 0.6,0.8, 0.6)
chair (0.5, 0.6, 0.3, 0.3,0. 4, 0.4)
x1
z1
Agent
z
x

2D polar coordinate

3D rectangle coordinate

Figure 16: **Coordinate system comparison**. We provide a diagram of the two types of coordinate systems.

insight, we implement a minimal routing mechanism during the inference stage. Empirically, we observe that the model's performance on *Existence* and *Attribute* sub-tasks tends to degrade over time, while performance on other tasks improves. Motivated by this training dynamic, we introduce a simple two-expert selection strategy.

Specifically, the model with the highest overall validation performance is designated as **Expert-1**, while the model with the best validation performance on partial sub-tasks is designated as **Expert-2**. During inference, if the predicted question type comes from the ones that have a gradual decreasing trend in validation performance, the input prefix sequence is routed to Expert-2; otherwise, it is routed to Expert-1. This strategy feasibly utilizes the result of task recognition in the first reasoning step. How to balance the training dynamics for different tasks should be an important direction for future work.

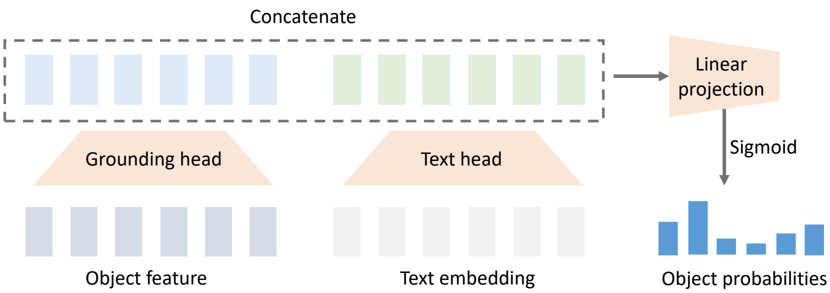


Figure 17: **3D Visual Grounding Module Design**.

Table 3: Hyperparameters for training.

Hyperparameter	Value
Optimizer	AdamW
Weight Decay	0.05
betas	[0.9, 0.999]
Learning Rate	3×10^{-5}
Warmup Steps	400
Type of GPUs	NVIDIA A100
Number of GPUs	4
Accumulate Gradient Batches	5
Batch Size/GPU (total)	2 (80)
gradient norm	5.0
epochs	5

Table 4: Hyperparameters for inference.

Hyperparameter	Value
Number of beams	5
maximum output length	256
repetition penalty	3.0
length penalty	1.0

As we apply LoRA to the LLM during training, this inference strategy introduces only an additional 330M parameters relative to a single base model, representing a reasonable trade-off between performance and deployment cost.

A.5 Hyperparameters for Model Training and Inference

We train the base LLM of SCENECOT in a single stage, initializing from the pretrained weights of LLaVA-1.5. The detailed hyperparameter settings are provided in Tab. 3 and Tab. 4.

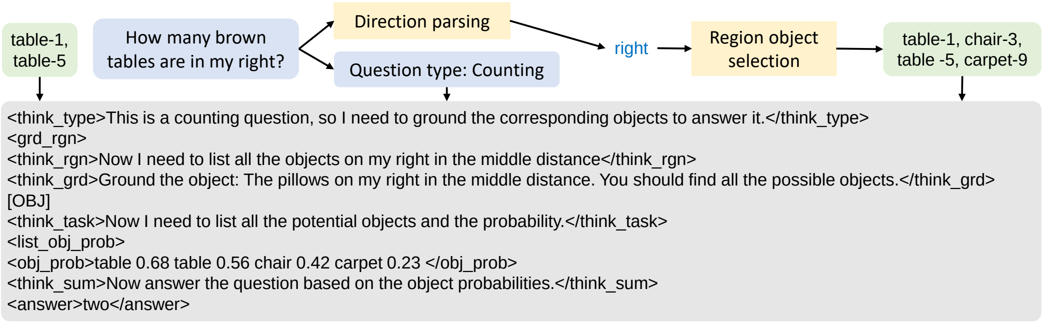
B SCENECOT-212K Dataset

B.1 Data Generation Pipeline of SCENECOT-212K

Our data generation pipeline is shared between Object-Centric Reasoning and Situated Reasoning, with Object-Centric Reasoning formulated as an *Attribute* sub-task under the broader Situated Reasoning framework. The overall process is illustrated in Fig. 18 and Fig. 19, using two representative sub-tasks as examples. Additionally, we design a pipeline to extract question types and target object IDs for SQA3D, as shown in Fig. 20.

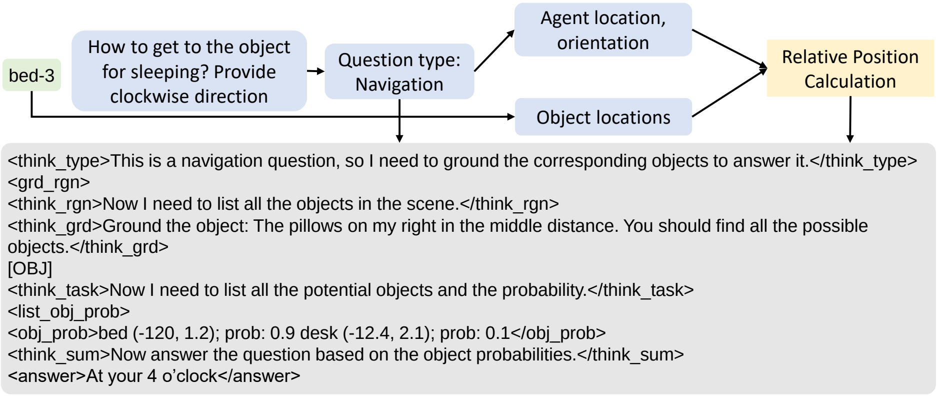
B.2 Data Generation Details and Examples of GQA3D

To enable the training of Object-Centric Reasoning in non-situated scenarios, we incorporate GQA3D as a key component of the overall dataset. The primary distinction from Situated Reasoning lies in the nature of the grounding text, which describes the target object from a global perspective rather than an egocentric view. Additionally, this task does not involve multi-object grounding or reasoning. We use GPT-4o to generate diverse question-answer pairs based on a given object image and its corresponding grounding text.



```
<think_type>This is a counting question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects on my right in the middle distance</think_rgn>
<think_grd>Ground the object: The pillows on my right in the middle distance. You should find all the possible objects.</think_grd>
[OBJ]
<think_task>Now I need to list all the potential objects and the probability.</think_task>
<list_obj_prob>
<obj_prob>table 0.68 table 0.56 chair 0.42 carpet 0.23 </obj_prob>
<think_sum>Now answer the question based on the object probabilities.</think_sum>
<answer>two</answer>
```

Figure 18: **Data Generation Procedure of *Counting*.**


bed-3
How to get to the object for sleeping? Provide clockwise direction
Question type: Navigation
Agent location, orientation
Object locations
Relative Position Calculation

```
<think_type>This is a navigation question, so I need to ground the corresponding objects to answer it.</think_type>
<grd_rgn>
<think_rgn>Now I need to list all the objects in the scene.</think_rgn>
<think_grd>Ground the object: The pillows on my right in the middle distance. You should find all the possible
objects.</think_grd>
[OBJ]
<think_task>Now I need to list all the potential objects and the probability.</think_task>
<list_obj_prob>
<obj_prob>bed (-120, 1.2); prob: 0.9 desk (-12.4, 2.1); prob: 0.1</obj_prob>
<think_sum>Now answer the question based on the object probabilities.</think_sum>
<answer>At your 4 o'clock</answer>
```

Figure 19: **Data Generation Procedure of *Navigation*.**

C Additional Experiments and Analyses

C.1 Additional Baseline Comparison

We provide additional experimental baseline results to further evaluate the advantages of SceneCOT. Specifically, we re-implemented LEO using the Version-2.1 dataset of MSQA. For a fair comparison, we also evaluated the performance of both LEO and MSR3D using predicted object masks generated by Mask3D. The results show that using predicted object masks leads to only a marginal performance drop for LEO and MSR3D. In contrast, our method experiences a significant performance boost when provided with ground-truth object masks and semantic labels. This indicates that SceneCOT has a higher performance ceiling, benefiting more from advances in visual grounding and semantic prediction modules, whereas LEO and MSR3D exhibit only limited gains under similar conditions.

C.2 Grounding and QA Coherence Analysis on Beacon3D

SceneCOT has superior performance on Beacon3D, a high-quality object-centric reasoning benchmark. To have a deeper understanding of the advantages of our method, we conducted a Grounding-QA coherence analysis as the author recommends [24].

The results in the Fig. 21 indicate that our method has the best QA-Grounding coherence. For example, SceneCOT has the highest ratio of "Good Coherence", which means QA and Grounding are both correct for an object. Besides, our method also has the lowest ratio of "Double Failure", which means QA and Grounding are both incorrect for an object. We then evaluate the metrics of R_1 and R_2, which evaluate the "Type 1" and "Type 2" ratio in the correct QA and wrong QA examples. The comparison on the left of the figure also reveals that feed-forward models such as LEO have a very high ratio of "Broken Coherence", exposing the overfitting for the question-answering data distribution instead of learning to ground and then reason. We also evaluate SceneCOT's Top-1 accuracy of object grounding in Beacon3D. The result is 59.0, which is very close to the performance on QA. We believe that better grounding modules will further enhance the performance of our method.

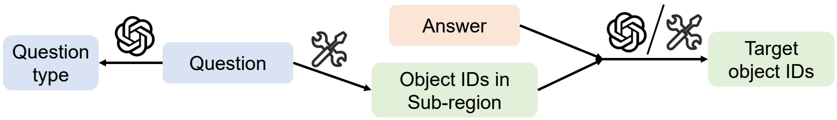


Figure 20: **Metadata generation for SQA3D**. In this pipeline, we extract the question type using ChatGPT. An off-the-shelf parser is used for extracting target objects. Then we feed the object list and the answer to another parser or ChatGPT to extract the target object IDs.

Table 5: **Metadata Examples of GQA3D**. GQA3D constructs QA pairs using the grounding text from Nr3D. The QA pairs are generated from the image of a target object.

Object Image	Grounding Text	QA pairs
	the tallest white box, the tall box to the left of the 2 boxes sitting on top of each other, the tall white box that is not stacked	Q: What material is ’the tallest white box’ made of? A: Cardboard type: appearance Q: Is ’the tall box to the left of the 2 boxes sitting on top of each other’ the same color as the door? A: No type: existence Q: Is there a fan next to ’the tall white box that is not stacked’? A: No type: existence

C.3 Additional Reasoning Chain Visualization Results

We present additional visualizations of the reasoning chains across multiple sub-tasks to provide a deeper understanding of our reasoning mechanism. In the first example, the model correctly identifies the target object based on the semantic grounding text (“The object used for sanitation”), and subsequently arrives at the correct answer by leveraging accurate visual cues. In the second example, the grounding module successfully locates the target objects (“The toilet and the picture”), which enables the model to reason effectively using object coordinates. However, in another case, the grounding module fails to identify the target object (“backpack”), resulting in an incorrect answer. We also include a video demonstration to intuitively showcase the entire workflow.

D Limitations

Though we propose a first step-by-step reasoning framework and have demonstrated its advantages on typical 3D scene reasoning tasks. There are also several limitations in our work.

First, our framework focuses on the tasks pre-defined in MSQA, which is limited in more complex scenarios such as embodied AI. For example, we do not consider the long-horizon tasks such as embodied task planning. Recently, SG3D [50] proposes a new benchmark to evaluate 3D-VL models’ capabilities of grounded task planning in embodied scenarios. We will consider extending our framework to this task in the future.

Second, SceneCOT-212K is built upon MSQA-ScanNet and Nr3D, which only contains the 3D scenes in ScanNet. Extending the dataset to more diverse real-world scenes is an important direction to unlock the real-world applications in the future.

Third, our thought design is still not perfect in partial sub-tasks. Even the we have demonstrated promising upper boundaries in some challenging sub-tasks, such as *Counting* and *Navigation*, our thought design still struggles with solving problems like *Spatial Relationship*. How to design better 3D-CoTs is another important direction to further increase the upper boundaries of the reasoning framework in 3D scenes. Besides, based on the recent practice in advanced reasoning LLMs [20, 45], exploring more learning algorithms such as Reinforcement Learning may also lead to more surprising capabilities for complex 3D scene reasoning.

Table 6: **Additional Baseline Results on MSQA and Beacon3D**. In the table, “GT” indicates the ground-truth object masks, “Pred” indicates the predicted object masks by Mask3D in MSQA. In this table, LEO’s performance on Beacon3D comes from the released paper [24]. In this table, “S” indicates SQA3D, “M” indicates MSQA, “G” indicates GQA3D.

Methods	Object Masks	MSQA							Beacon3D	
		Count.	*Exist.*	*Attr.*	*Spatial*	*Navi.*	*Others*	Overall	Case	Obj.
LEO	GT	33.4	88.2	57.9	**49.6**	44.0	**82.7**	55.0	43.2	7.8
LEO	Pred	32.5	88.5	**58.7**	44.2	39.6	81.4	54.8	–	–
MSR3D	GT	33.8	90.7	52.0	46.9	**55.3**	79.0	55.1	–	–
MSR3D	Pred	32.3	**93.1**	50.0	46.5	54.1	75.6	54.2	–	–
SceneCOT (M+G)	Pred	**47.9**	82.1	49.6	47.2	51.6	80.3	**55.6**	**58.9**	**23.2**

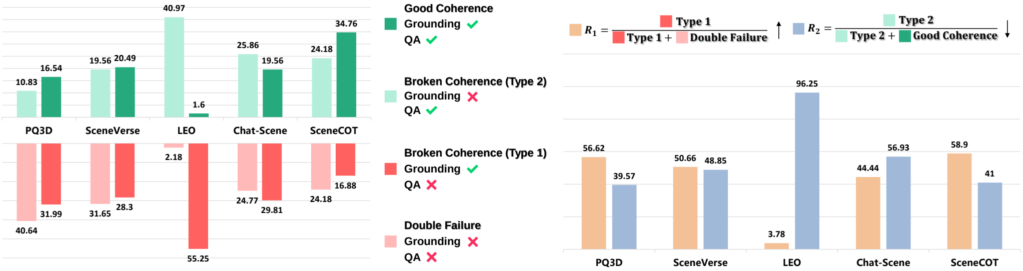


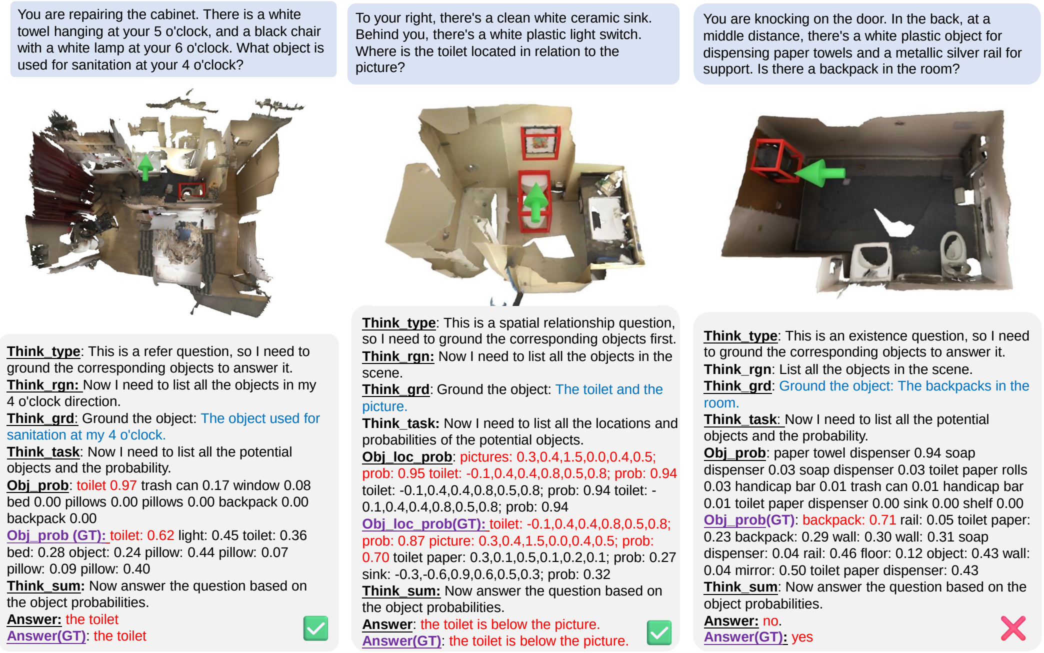
Figure 21: **Grounding and QA Coherence on Beacon3D**.

E Broader Impact

Understanding and reasoning in 3D scenes is a cornerstone capability for building intelligent embodied agents that can operate safely, reliably, and effectively in the physical world. Our proposed approach, SceneCOT, introduces a structured, interpretable framework for 3D scene reasoning by incorporating a Chain-of-Thought (CoT) paradigm into the 3D vision-language (3DVL) domain. By explicitly modeling intermediate reasoning steps such as task recognition, region localization, entity grounding, and grounded reasoning, SceneCOT offers a significant leap toward human-level understanding in complex 3D environments.

The potential societal benefits of this work are substantial. It enables advancements in a wide range of real-world applications, including domestic robotics, assistive technologies for individuals with disabilities, autonomous navigation systems, and intelligent agents for virtual and augmented reality. The stepwise, interpretable nature of our method also enhances transparency and safety, which are essential for deploying AI systems in human-centered environments.

At the same time, we acknowledge that powerful embodied AI systems can be misused in ways that may compromise privacy or safety, particularly in surveillance or military contexts. To mitigate such risks, we encourage responsible research practices, including dataset transparency, open evaluation protocols, and active engagement with the broader community on the ethical deployment of such systems. Our work serves as a foundation for building grounded and generalizable reasoning agents, and we hope it will inspire future research that aligns technological advancements with human values.



You are repairing the cabinet. There is a white towel hanging at your 5 o'clock, and a black chair with a white lamp at your 6 o'clock. What object is used for sanitation at your 4 o'clock?

Think_type: This is a refer question, so I need to ground the corresponding objects to answer it.
Think_rgn: Now I need to list all the objects in my 4 o'clock direction.
Think_grd: Ground the object: The object used for sanitation at my 4 o'clock.
Think_task: Now I need to list all the potential objects and the probability.
Obj_prob: toilet 0.97 trash can 0.17 window 0.08 bed 0.00 pillows 0.00 pillows 0.00 backpack 0.00 backpack 0.00
Obj_prob (GT): toilet: 0.62 light: 0.45 toilet: 0.36 bed: 0.28 object: 0.24 pillow: 0.44 pillow: 0.07 pillow: 0.09 pillow: 0.40
Think_sum: Now answer the question based on the object probabilities.
Answer: the toilet
Answer(GT): the toilet ✓

To your right, there's a clean white ceramic sink. Behind you, there's a white plastic light switch. Where is the toilet located in relation to the picture?

Think_type: This is a spatial relationship question, so I need to ground the corresponding objects first.
Think_rgn: Now I need to list all the objects in the scene.
Think_grd: Ground the object: The toilet and the picture.
Think_task: Now I need to list all the locations and probabilities of the potential objects.
Obj_loc_prob: pictures: 0.3,0.4,1.5,0.0,0.4,0.5; prob: 0.95 toilet: -0.1,0.4,0.4,0.8,0.5,0.8; prob: 0.94 toilet: -0.1,0.4,0.4,0.8,0.5,0.8; prob: 0.94 toilet: -0.1,0.4,0.4,0.8,0.5,0.8; prob: 0.94
Obj_loc_prob(GT): toilet: -0.1,0.4,0.4,0.8,0.5,0.8; prob: 0.87 picture: 0.3,0.4,1.5,0.0,0.4,0.5; prob: 0.70 toilet paper: 0.3,0.1,0.5,0.1,0.2,0.1; prob: 0.27 sink: -0.3,-0.6,0.9,0.6,0.5,0.3; prob: 0.32
Think_sum: Now answer the question based on the object probabilities.
Answer: the toilet is below the picture.
Answer(GT): the toilet is below the picture. ✓

You are knocking on the door. In the back, at a middle distance, there's a white plastic object for dispensing paper towels and a metallic silver rail for support. Is there a backpack in the room?

Think_type: This is an existence question, so I need to ground the corresponding objects to answer it.
Think_rgn: List all the objects in the scene.
Think_grd: Ground the object: The backpacks in the room.
Think_task: Now I need to list all the potential objects and the probability.
Obj_prob: paper towel dispenser 0.94 soap dispenser 0.03 soap dispenser 0.03 toilet paper rolls 0.03 handicap bar 0.01 trash can 0.01 handicap bar 0.01 toilet paper dispenser 0.00 sink 0.00 shelf 0.00
Obj_prob(GT): backpack: 0.71 rail: 0.05 toilet paper: 0.23 backpack: 0.29 wall: 0.30 wall: 0.31 soap dispenser: 0.04 rail: 0.46 floor: 0.12 object: 0.43 wall: 0.04 mirror: 0.50 toilet paper dispenser: 0.43
Think_sum: Now answer the question based on the object probabilities.
Answer: no.
Answer(GT): yes ✗

Figure 22: **Additional Reasoning Chain Visualization**. We provide more visualization results on *Refer*(Left), *Spatial Relationship*(Middle), and *Existence*(Right).