Task-oriented Sequential Grounding in 3D Scenes

Zhuofan Zhang[1,2*] **Ziyu Zhu**[1,2*] **Pengxiang Li**[1,3*] **Tengyu Liu**[1] **Xiaojian Ma**[1]
Yixin Chen[1] **Baoxiong Jia**[1] **Siyuan Huang**[1] **Qing Li**[1✉]
[1]State Key Laboratory of General Artificial Intelligence, BIGAI, China
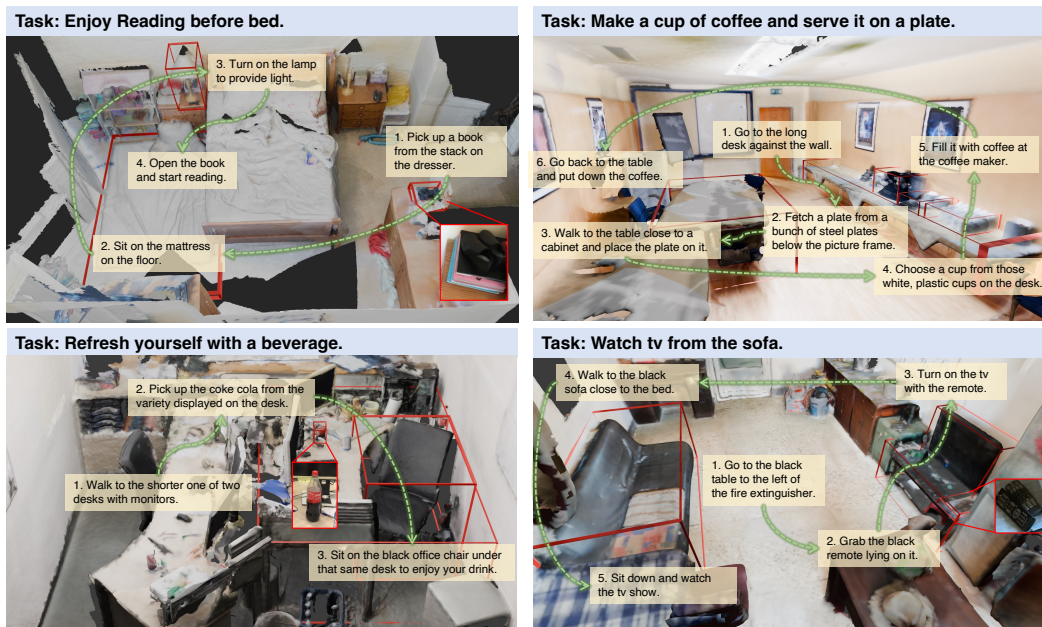[2]Tsinghua University [3]Beijing Institute of Technology
`sg-3d.github.io`

Figure 1: **The task-oriented sequential grounding task in 3D scenes (SG3D)**, wherein an agent is required to locate a sequence of target objects for detailed steps in a plan to complete daily activities. To solve this task, an agent must understand each step *in the context of the whole plan* to identify the target object, since a single step alone can be insufficient to distinguish the target from other objects of the same class. The dataset is available at `sg-3d.github.io`.

Abstract

Grounding natural language in physical 3D environments is essential for the advancement of embodied artificial intelligence. Current datasets and models for 3D visual grounding predominantly focus on identifying and localizing objects from static, object-centric descriptions. These approaches do not adequately address the dynamic and sequential nature of task-oriented grounding necessary for practical applications. In this work, we propose a new task: Task-oriented Sequential Grounding in 3D scenes, wherein an agent must follow detailed step-by-step instructions to complete daily activities by locating a sequence of target objects in indoor scenes. To facilitate this task, we introduce SG3D, a large-scale dataset containing *22,346 tasks with 112,236 steps across 4,895 real-world 3D scenes*. The

---

[*]Work done as an intern at BIGAI. ✉Corresponding author.

dataset is constructed using a combination of RGB-D scans from various 3D scene datasets and an automated task generation pipeline, followed by human verification for quality assurance. We adapted three state-of-the-art 3D visual grounding models to the sequential grounding task and evaluated their performance on SG3D. Our results reveal that while these models perform well on traditional benchmarks, they face significant challenges with task-oriented sequential grounding, underscoring the need for further research in this area.

1 Introduction

Grounding natural language in the physical 3D world is crucial for advancing embodied artificial intelligence (Embodied AI) [20, 55], where robots must follow human instructions to complete complex tasks [69]. Recent years have witnessed the collection of various datasets [31, 8, 3, 63, 55, 32] aimed at training and testing robust visual grounding models in 3D scenes [69, 70, 9, 23, 30]. While these datasets have driven progress in 3D visual grounding, they largely borrow practices from 2D visual grounding [8, 3] and focus primarily on identifying and localizing objects based on *object-centric* descriptions [17]. As illustrated in Fig. 2, these descriptions unnaturally detail the target objects' categories, attributes, and spatial relationships to distinguish them from other objects. However, a significant yet often overlooked gap exists between these standalone object-centric referrals and the *task-driven*, *sequential* object grounding commonly used in practical scenarios for Embodied AI [4, 48]. This gap is highlighted in Fig. 2, which compares object-centric and task-driven visual grounding in 3D scenes.

To bridge this gap, we propose a new task: Task-oriented Sequential Grounding in 3D scenes. In this task, an agent is instructed to accomplish a daily activity with detailed steps in an indoor scene, aiming to find a sequence of target objects for each step. To address this challenge, we constructed a large-scale dataset named SG3D. We compiled a set of RGB-D scans of realistic indoor scenes sourced from various 3D scene datasets, including ScanNet [49], ARKitScenes [7], 3RScan [53], etc. These scenes encompass a variety of room types, such as bedrooms, kitchens, offices, and living rooms. We represent these scenes using 3D scene graphs [5, 54] provided by SceneVerse [31], which describe the objects' categories, attributes, and spatial relations within the scenes.

We further designed an automated generation pipeline that utilizes these scene graphs and GPT-4 [2] to create diverse, high-quality daily tasks. Each task comprises a high-level description and a detailed plan, with the target object annotated for each step. To ensure the validity of the generated tasks, we conducted a human verification process to check if the tasks were appropriate for the scenes, if the plans were sufficient to accomplish the tasks, and if the target objects were correctly identified for each step. Invalid tasks were either filtered out or manually refined. Ultimately, the proposed SG3D includes *22,346 tasks* with *112,236 steps* across *4,895 real-world 3D scenes*, as exemplified in Fig. 1.

In our experiments, we adapted three state-of-the-art 3D visual grounding models to the sequential grounding task and evaluated them on the proposed SG3D. The models included 3D-VisTA [69], PQ3D [70], and LEO [27]. The results indicate that although these models excel on previous benchmarks, they struggle with the more complex and realistic grounding presented in the SG3D benchmark. This highlights the need for further research and development to improve performance in task-oriented sequential grounding scenarios for Embodied AI.

Our contributions are summarized as follows:

- We proposed a new task, Task-oriented Sequential Grounding in 3D scenes, to address the gap between object-centric and task-driven grounding required for practical Embodied AI applications.

- We constructed a large-scale dataset for this novel task, SG3D, which contains 22,346 tasks with 112,236 steps across 4,895 real-world 3D scenes.

- We adapted three state-of-the-art 3D visual grounding models (3D-VisTA, PQ3D, and LEO) to the sequential grounding task and evaluated them on SG3D. Experimental results indicate that these models struggle with task-oriented sequential grounding, highlighting the need for further advancements in this area.

2

Task: Wash hands before cooking.

| 1. Walk to the kitchen counter where the sink is located. | 2. Use the white soap from the dispenser above the counter. | 3. Wash your hands thoroughly in the sink. | 4. Dry hands with a paper towel from the dispenser on the wall. |

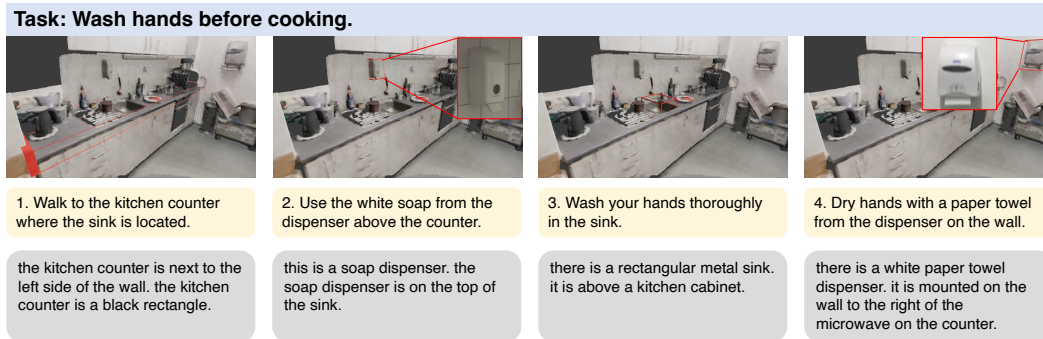| the kitchen counter is next to the left side of the wall. the kitchen counter is a black rectangle. | this is a soap dispenser. the soap dispenser is on the top of the sink. | there is a rectangular metal sink. it is above a kitchen cabinet. | there is a white paper towel dispenser. it is mounted on the wall to the right of the microwave on the counter. |

Figure 2: The comparison between task-oriented steps in SG3D (first row) and object-centric referrals in ScanRefer (second row) for the same target objects.

Table 1: **The comparison of SG3D with existing 3D visual grounding benchmarks.** SG3D expands the data scale of prior work by order of magnitude. "VG" stands for Visual Grounding, "SG" for Sequential Grounding, and and "MT" for Multiple Tasks. * Only new data is counted.

Dataset	Task	Referral type	Text Source	Quality Check	Scene	Obj.	Avg. Text Len.	Vocab.	Total
ScanRefer [8]	VG	Object-centric	Human	✓	1.5K	33K	20.3	4,197	52K
Nr3D [3]	VG	Object-centric	Human	✓	1.5K	33K	11.5	2,986	42K
Sr3D [3]	VG	Object-centric	Template	✓	1.5K	33K	9.7	158	84K
Multi3DRefer* [63]	VG	Object-centric	Template w/ Rephrasing	✓	1.5K	33K	15.1	7,077	20K
SceneVerse* [31]	MT	Object-centric	Human + GPT-3.5	✓	68K	1.5M	14.7	24,304	2.2M
SG3D	**SG**	**Task-oriented**	**GPT-4**	**✓**	**4.9K**	**123K**	**70.5**	**8,136**	**22K / 112K**

2 Related Work

3D Vision Language The field of 3D vision-language learning aims to establish a connection between natural language and the 3D physical world [69, 70, 33]. This connection enables embodied agents to comprehend their surrounding environment and communicate effectively with humans [69, 48]. Within this emerging domain, several benchmarks have been developed, focusing on tasks such as visual grounding [8, 3, 1], question answering [6, 65], and dense captioning [12]. While methods addressing single tasks have been proposed [23, 58, 42, 30, 64, 9], there is a growing trend towards unified models [69, 70, 13]. Additionally, open vocabulary approaches have gained traction in recent literature [46, 19, 52]. However, previous 3D visual grounding benchmarks are often *object-centric* and miss sequential information, whereas realistic grounding sentences are typically driven by *task-related* context [17]. In contrast to previous work, our benchmark provides more natural and informative language and introduces diverse *sequential* information.

Grounded Task Planning The field of Embodied AI focuses on the capabilities of agents to reason, plan, navigate, and act in 3D environments [16, 27, 4]. Grounded task planning is crucial as it enables these agents to execute human instructions effectively [40, 66]. Established benchmarks such as ALFRED [50], SAYPLAN [48], BEHAVIOR-1K [34], and TaPA [59] assess these abilities by measuring the success of the agents' overall task plans in synthetic environments. Other benchmarks, like LoTA-BENCH [14], EgoPlan-Bench [11], and G-PlanET [40], evaluate performance on a per-step basis, using rule-based or closed-set answer assessments. Specialized models [61, 62, 50] and foundation models [51, 57, 36, 39, 28, 22] have been utilized to accomplish grounded task planning. In contrast to previous work based on synthetic environments, our benchmark utilizes real 3D scenes. Moreover, by grounding each planned task to objects instead of low-level actions, we enable a wider range of actions and facilitate a more comprehensive analysis of results at each step.

3D Large Language Model Recent advancements in large language models (LLMs) have been significantly enhanced by integrating 3D spatial data, resulting in the development of 3D LLMs [43]. Existing works, such as LEO [24] and Chat3D [56], use object-centric or point-level representations to incorporate scene information into LLMs during instruction tuning [24, 60, 38, 21, 25]. LL3DA [10] employs a Q-former-like [35] structure to further improve LLMs' 3D scene perception. Additionally, recent models like LEO [27], 3D-VLA [67], and ManipLLM [37] have introduced action capabilities
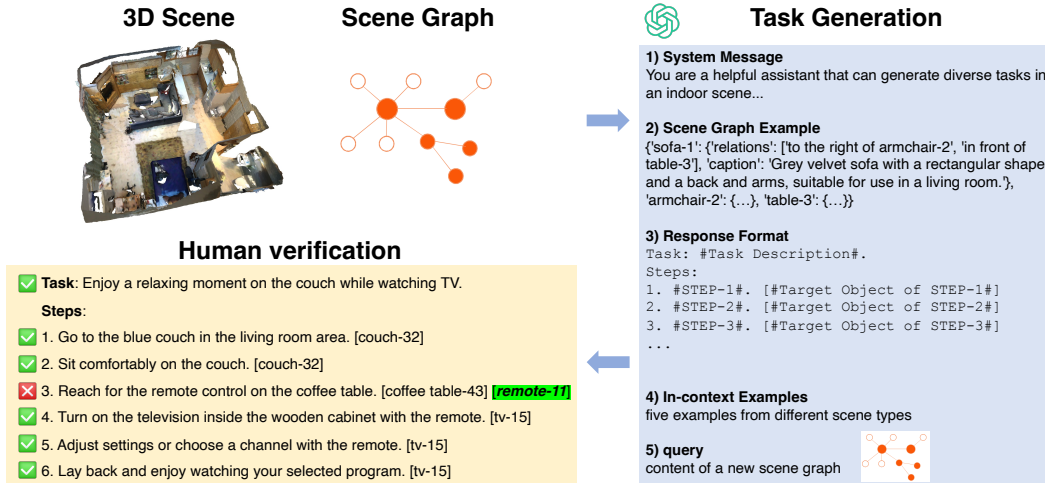
3

Figure 3: The pipeline of generating sequential grounding tasks in 3D scenes.

into 3D LLMs, enabling them to interact with and manipulate objects in 3D environments [27, 29, 41]. Our work enhances the capabilities of 3D LLMs by incorporating grounding abilities, which output specific objects alongside the text.

3 The 3D Sequential Grounding Benchmark (SG3D)

3.1 Task Definition

The problem of sequential grounding involves determining the relevance of objects in a given task. Specifically, given a 3D scene $\mathcal{S}$ and a task $\mathcal{T} = (t, \{a_1, ..., a_n\})$ where t is a high-level task description and $a_1, ...a_n$ are detailed steps of the task plan, a model is required to predict a sequence of objects $\mathcal{O} = \{o_1, ..., o_n\}$, $i.e.$, the model needs to learn a mapping $f : (\mathcal{S}, \mathcal{T}) \rightarrow \mathcal{O}$. Compared to prior work, the challenge in our task lies in grounding objects within sequential steps of a task plan.

3.2 Dataset Construction

As illustrated in Fig. 3, we leverage GPT-4 to generate tasks based on a 3D scene graph, followed by human verification.

3D Scenes Existing robotic task-planning approaches are typically evaluated in simulated environments [50, 34, 48], lacking observation of their effectiveness in real-world scenarios. To address this, we select reconstructed scenes as the 3D environment for our tasks. Specifically, we utilized real-world scenes from the SceneVerse dataset, incorporating scenes from ScanNet, ARKitScenes, HM3D, 3RScan, and MultiScan. In total, we curate 4,895 3D scenes in SG3D. Tab. 2 presents the number of scenes used in each dataset and the average number of object instances per scene.

Scene Graphs To generate the task data, we utilize 3D scene graphs from SceneVerse. Each node in the graph represents a 3D object instance within the scene, and each edge represents a spatial relationship between nodes, such as "near", "below", or "embedded". We enhance the scene graphs by adding object captions provided in SceneVerse, enriching the semantic attributes of the object nodes.

Task Generation Using the 3D scene graph, we prompt GPT-4 (*gpt-4-turbo-2024-04-09*) to generate diverse tasks. We ask GPT-4 to generate five tasks for each scene. Each task comprises a general description and several steps, while each step involves a target object that the agent must attend to. We carefully design the prompt and provide five examples from different room types to guide responses from GPT-4. Post-generation, we filter out tasks with illegal formats or exceed ten steps. The detailed prompt used for GPT-4 is provided in the Appendix.

Human Verification We manually verify the test set data to ensure data quality. Given the 3D scene mesh and the task generated by GPT-4, annotators follow these rules to judge each step's correctness:

1. If the step is unrelated to the task description, judge it as incorrect.
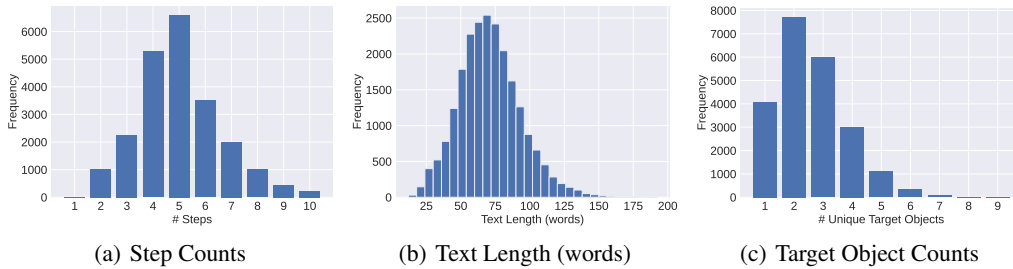
(a) Step Counts   (b) Text Length (words)   (c) Target Object Counts

Figure 4: Distributions of (a) step counts, (b) text length, and (c) target object counts per task.

Table 2: Dataset statistics of SG3D.

Dataset	#scenes	#obj. / scene	#tasks	#steps
3RScan [53]	472	31.5	2,194	11,318
ScanNet [15]	693	30.7	3,174	15,742
MultiScan [45]	117	40.8	547	2,683
ARKitScenes [7]	1,575	12.1	7,395	39,887
HM3D [47]	2,038	31.0	9,036	42,706
Total	4,895	25.1	22,346	112,336

2. If there is a missing step between step k and step $k + 1$, judge step $k + 1$ as incorrect.

3. If the step's description is insufficient to identify the target object, but the target object can be identified through context, judge it as correct; otherwise, judge it as incorrect.

We manually revised tasks with one incorrect step and dropped tasks with more than one incorrect step. The screenshot of the interface for verification is provided in the Appendix.

3.3 Dataset Analysis

In total, we collected data containing 22,346 tasks, encompassing 112,236 steps. Tab. 2 presents the statistics of task and step counts in our dataset. Each task description has an average length of 6.9 words, and each step has an average length of 12.7 words. The dataset was split into training and evaluation sets. For 3RScan, scenes from its training and evaluation splits were used as our training set, while scenes from its test split were used as the evaluation set. For other datasets, we adhered to the original split of the 3D scenes provided.

Fig. 4(a) illustrates the distribution of the number of steps per task, revealing an average of 5.03 steps per task. This underscores the complexity of our benchmark and the sequential nature of our data. Fig. 4(b) presents a histogram displaying the distribution of total text lengths for each task, including the task description and all associated steps, with an average of 70.5 words. This extended context poses a significant challenge for many text encoders, indicating the need for models capable of handling lengthy inputs. Additionally, we examine the number of distinct target objects involved in each task, as shown in Fig. 4(c). Unlike the step counts, the number of unique target objects per task considers target objects with the same ID across different steps as one object, resulting in an average of 2.59 unique objects per task. This finding indicates that multiple objects are typically involved in this process.

To illustrate the diversity of our dataset, we present three word clouds here. Fig. 5(a) and Fig. 5(b) depict the frequency of words in task descriptions and action steps, respectively. In the task descriptions, the terms "prepare" and "organize" are the most prevalent activities. In the action steps, "walk" and "place" are the most common actions, "table" is the most frequent object, and "white" is the most frequent adjective. This indicates that task descriptions tend to be abstract and demand-oriented, while action steps offer detailed, execution-oriented instructions. Fig. 5(c) highlights the most frequently occurring target object classes, including but not limited to "cabinet", "table", "chair", "sink", "bed", "shelf", demonstrating the association of different object classes with the task guidance.

(a) Task Description	(b) Action Step	(c) Target Object Labels

Figure 5: Word clouds of (a) task description, (b) action step, and (c) target object labels.

4 3D Sequential Grounding Models

We explore three typical models for this purpose: the dual-stream model 3D-VisTA [69], the query-based model PQ3D [70], and the 3D LLM LEO [27], with further details discussed below.

4.1 Architecture

Dual-stream model. In the dual stream model, we build upon the 3D-VisTA [69] baseline. In 3D-VisTA, the model employs a spatial transformer to process 3D object representations and extracts text features using BERT [18]. These object and text tokens are then combined and input into a unified transformer architecture to predict the target object.

Query-based model. Unlike the dual stream model, the query-based model employs a generalized decoding framework for vision-language tasks [71, 68]. PQ3D [70] is a prominent query-based model designed for 3D environments that unifies multiple representations and addresses various tasks through multi-task training. This model utilizes the CLIP text encoder to process textual inputs. For a fair comparison with other models, we limit our implementation to only the point feature branch for scene feature extraction.

3D LLM. The powerful reasoning capabilities of Large Language Models are highly advantageous for our task. We have adapted the recent 3D LLM LEO [27] to suit our needs. In addition to predicting actions for each step, our model also predicts a special token called [GRD]. This token enables integrated reasoning about both the previous and current step instructions. In order to predict the object, we concatenate the object token with the [GRD] token and pass them through the same grounding head as used in 3D-VisTA and PQ3D.

4.2 Training & Inference

During training, we optimize the three types of models previously discussed using the cross-entropy loss, which compares the predicted object scores $f(\mathcal{S}, \mathcal{T})$ with the ground truth scores $\mathcal{O}$, as defined in Eq. (1). In the case of the 3D LLM, following the methodology of LEO, we introduce an additional cross-entropy loss to provide supervision for action generation in text format.

$$\mathcal{L}_{grd} = \mathbb{E}_{(\mathcal{S},\mathcal{T},\mathcal{O})\sim\mathcal{D}}\text{CrossEntropy}(f(\mathcal{S}, \mathcal{T}), \mathcal{O}) \tag{1}$$

During the inference phase, the models are provided with the task description and detailed steps and predict the target object for each step. This setup applies to all models, facilitating a direct assessment of their ability to identify and prioritize the correct object based on sequential instructions.

5 Experiments and Results

5.1 Settings

Training Details We conduct training for all three model types across all available datasets for 50 epochs. For optimization, we employ the AdamW optimizer, setting the learning rate at 1e-4, with β_1 configured to 0.9 and β_2 to 0.999. Additionally, we apply a weight decay of 0.05. Specifically, for the PQ3D and 3D-VisTA models, we utilize a batch size of 32. For the LEO model, we reduce the batch size to 16 due to GPU memory limit. Furthermore, we use LoRA tuning [26] for the parameters of the LLM in LEO with a rank setting of 16.
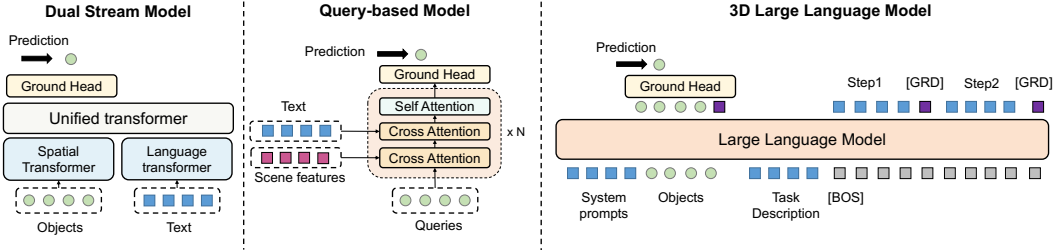
Figure 6: Sequential grounding models.

Evaluation Metrics We assess the grounding performance of all models using two key metrics: task accuracy (t-acc) and step accuracy (s-acc). Task accuracy refers to the average grounding accuracy over the total number of tasks t. A sample is considered correct if the grounded objects are accurately identified for all steps within a task. Conversely, step accuracy is calculated by averaging the accuracy across all individual steps a. Task accuracy evaluates the model's ability to consistently interpret and respond accurately across a sequence of text prompts. On the other hand, step accuracy focuses on the model's effectiveness at each individual step.

5.2 Quantative results

1.Previous 3D-VL models, such as 3D-VisTA and PQ3D, struggle to transfer to the sequential grounding task without fine-tuning. In the zero-shot setting, these models achieve relatively low step accuracies ranging from 22.8% to 34.6% and task accuracies ranging from 0.0% to 10.3% across all datasets. This indicates that the models' pre-training on non-sequential tasks is insufficient for handling the complexities of sequential grounding, highlighting the need for task-specific fine-tuning.

2. Fine-tuning greatly enhances performance but low task accuracy scores (< 40%) indicate that consistent sequential grounding remains a challenge. 3D-VisTA's t-acc increases from 8.3% to 30.6%, while PQ3D's t-acc improves from 7.8% to 26.8%. LEO, the 3D LLM model, achieves the best performance after fine-tuning, with a s-acc of 62.8% and a t-acc of 34.1%. Despite these improvements, all models' t-acc scores remain below 40%, indicating that current models still struggle to achieve consistent sequential grounding. This limitation highlights the need for further research and model design to effectively address the challenges posed by sequential grounding tasks.

3. The 3D LLM model, LEO, consistently outperforms the other models across all datasets, particularly in terms of task accuracy. LEO achieves the highest task accuracies 34.1%, compared to 3D-VisTA 30.6% and PQ3D 26.8%. This advantage can be attributed to LEO's 3D LLM architecture, which effectively captures and reasons about sequential dependencies in grounding tasks. Although LEO also enhances step accuracy, the improvement is less substantial compared to the significant gains observed in task accuracy.

4. Despite access to ground truth object labels, the performance of GPT-4 remains limited in the sequential grounding task. GPT-4 achieves a relatively high step accuracy of 73.4% when given access to ground truth labels. However, its task accuracy is only 46.6%, indicating that the model struggles to maintain consistency and correctness throughout the entire sequence of steps in a task. This result suggests that adapting large language models to sequential grounding tasks is a nontrivial challenge.

5.3 Ablation Study & Analysis

Effect of offering sequential information. To analyze the impact of sequential information, we remove multi-step action context during both the training and testing phases. The experimental results in Fig. 7 show that removing sequential information leads to a significant performance drop in task accuracy for both LEO and 3D-VisTA. LEO experiences an average t-acc drop of 3.4%, while 3D-VisTA has an even higher average drop of 5.0%. This suggests that models have learned to utilize sequential information during the grounding process. In contrast, PQ3D's performance drop is much more limited, with an average t-acc drop of only 0.8%. This can be attributed to PQ3D's reliance on

Table 3: **The grounding accuracy on SG3D.** "s-acc" denotes the grounding accuracy averaged over steps and "t-acc" denotes the grounding accuracy averaged over tasks. A task is considered correct if and only if all steps are correct. We run each experiment for three times and report error bars.

	Model Type	ScanNet		3RScan		MultiScan	
		s-acc	t-acc	s-acc	t-acc	s-acc	t-acc
Zero-shot							
3D-VisTA	*Dual-stream*	26.9	4.7	23.7	2.2	22.8	4.7
PQ3D	*Query-based*	29.7	4.1	24.6	2.9	23.2	0.0
GPT-4 w/ GT labels	*LLM*	69.2	38.1	73.1	37.0	73.7	32.6
Fine-tune							
3D-VisTA	*Dual-stream*	58.4 ± 0.1	21.1 ± 0.5	53.3 ± 0.8	14.9 ± 1.5	48.3 ± 3.4	$\mathbf{11.6} \pm 2.4$
PQ3D	*Query-based*	54.8 ± 0.8	17.8 ± 0.7	49.3 ± 1.3	9.9 ± 2.5	46.4 ± 2.1	4.7 ± 0
LEO	*3D LLM*	$\mathbf{61.2} \pm 1.0$	$\mathbf{25.7} \pm 1.7$	$\mathbf{55.8} \pm 0.6$	$\mathbf{16.0} \pm 1.8$	$\mathbf{52.7} \pm 1.6$	7.6 ± 1

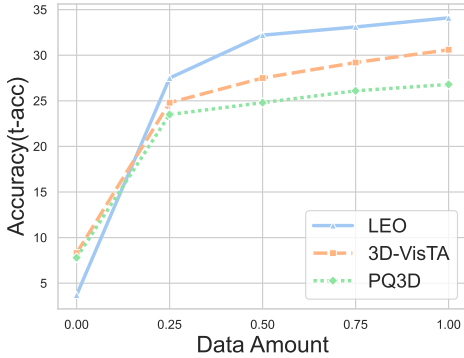|	Model Type	ARKitScenes		HM3D		OverAll	
		s-acc	t-acc	s-acc	t-acc	s-acc	t-acc
Zero-shot							
3D-VisTA	*Dual-stream*	30.8	9.0	25.3	10.3	26.9	8.3
PQ3D	*Query-based*	34.6	8.6	24.4	9.7	28.2	7.8
GPT-4 w/ GT labels	*LLM*	73.2	48.3	75.9	52.1	73.4	46.6
Fine-tune							
3D-VisTA	*Dual-stream*	68.8 ± 0.9	37.6 ± 1.1	59.6 ± 0.7	32.4 ± 0.8	60.9 ± 0.4	30.6 ± 0.7
PQ3D	*Query-based*	65.2 ± 0.5	32.1 ± 0.7	56.1 ± 0.3	30.0 ± 0.7	57.3 ± 0.1	26.8 ± 0.5
LEO	*3D LLM*	$\mathbf{69.6} \pm 0.4$	$\mathbf{41.5} \pm 1.5$	$\mathbf{61.5} \pm 1$	$\mathbf{35.7} \pm 1.3$	$\mathbf{62.8} \pm 0.7$	$\mathbf{34.1} \pm 1.2$

Figure 7: Ablation of sequential steps.

Figure 8: Data efficiency for training.

a CLIP text encoder, which may have difficulty comprehending long sentences, thereby overfitting to short, single-step instructions.

Training data. Fig. 8 shows that increasing the amount of training data improves the performance of all models. Notably, LEO demonstrates higher data efficiency, achieving comparable performance to PQ3D and 3D-VisTA using only 25% of the data. This advantage likely stems from LEO's foundation on a large language model, which has been pre-trained on extensive task-relevant information and acquired common sense knowledge.

5.4 Qualitative Results

Fig. 9 demonstrates that sequential grounding tasks require models to reason across sequential steps. The results from LEO show that after training, it has the ability to perform sequential grounding, as illustrated in tasks 1, 2, and 5. However, the model sometimes struggles to maintain sequential consistency, as seen in task 3. Additionally, task 4 presents a failure case where the model does not understand the concept of a diaper bin. Our dataset is task-oriented and requires common sense knowledge to solve effectively. The examples highlight the challenges and complexities involved in sequential grounding tasks, emphasizing the need for models to possess both sequential reasoning capabilities and relevant common sense knowledge to achieve consistent and accurate results.

8

Task 1:
Water the desk plant

Step1: *Go to the cabinet standing on the floor and open a drawer to find a watering can.*

Step2: *Fill the watering can using water from the radiator below the window.*

Step3: *Walk back to the desk supporting the green plant next to several monitors.*

Step4: *Carefully pour water into the pot of the small green plant.*

Step5: *Wipe any spilled water from the desk using a cloth from the cabinet*

Task 2:
Watch a movie on the television

Step1: *Walk to the dresser close to the organizer shelf and the instrument case.*

Step2: *Turn on the sleek black television above it*

Step3: *Sit on the bed decorated with a cozy blanket and two plush pillows.*

Step4: *Enjoy the movie on the television.*

Task 3:
Sanitize your hands.

Step1: *Move towards the sink below the soap dispenser aligned with the toilet seat cover dispenser.*

Step2: *Reach for the bottle of hand sanitizer aligned with the toilet seat cover dispenser inside the mirror.*

Step3: *Apply a generous amount of sanitizer to your hands.*

Step4: *Rub your hands together thoroughly to spread the sanitizer.* ✗

Task 4:
Check the time in the nursery.

Step1: *Go to the black diaper bin next to the white wardrobe closet.* ✗

Step2: *Look upward to see the clock with a red tag above the bin.*

Task 5:
Enjoy some nursery room decorations

Step1: *Walk to the baby crib near the changing table.*

Step2: *Look up to admire the baby mobile with a tan tent above the crib.*

Figure 9: **Qualitative results from LEO.** Red are predictions and green are ground-truth boxes.

6 Conclusion and Future Work

In this work, we introduce the task of Task-oriented Sequential Grounding in 3D scenes and present SG3D, a large-scale dataset designed to facilitate research in this area. Evaluations of state-of-the-art 3D visual grounding models on SG3D benchmark reveal the substantial challenges in adapting these models to sequential grounding tasks. These results emphasize the necessity for further research and model development. We encourage the community to move beyond traditional 3D visual grounding towards more practical, task-oriented applications, paving the way for more advanced and capable embodied agents.

Limitations The current dataset cannot be directly transferred to simulation platforms for robot manipulation, and the performance of the evaluated models is insufficient for reliable real-world deployment. To improve performance, future research can explore integrating advanced techniques such as chain-of-thought reasoning, reflection mechanisms, and 2D vision foundation models.

References

[1] Ahmed Abdelreheem, Kyle Olszewski, Hsin-Ying Lee, Peter Wonka, and Panos Achlioptas. Scanents3d: Exploiting phrase-to-3d-object correspondences for improved visio-linguistic models in 3d scenes. *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, 2024. 3

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

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

[4] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 2, 3

[5] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *International Conference on Computer Vision (ICCV)*, 2019. 2

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

[7] 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. 2, 5

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

[9] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Language conditioned spatial relation reasoning for 3d object grounding. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3

[10] 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. *arXiv preprint arXiv:2311.18651*, 2023. 3

[11] Yi Chen, Yuying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Ruifeng Xu, Ying Shan, and Xihui Liu. Egoplan-bench: Benchmarking egocentric embodied planning with multimodal large language models. *arXiv preprint arXiv:2312.06722*, 2023. 3

[12] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[13] 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)*, pages 18109–18119, 2023. 3

[14] Jae-Woo Choi, Youngwoo Yoon, Hyobin Ong, Jaehong Kim, and Minsu Jang. Lota-bench: Benchmarking language-oriented task planners for embodied agents. *arXiv preprint arXiv:2402.08178*, 2024. 3

[15] 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*, pages 5828–5839, 2017. 5

[16] Matt Deitke, Dhruv Batra, Yonatan Bisk, Tommaso Campari, Angel X Chang, Devendra Singh Chaplot, Changan Chen, Claudia Pérez D'Arpino, Kiana Ehsani, Ali Farhadi, et al. Retrospectives on the embodied ai workshop. *arXiv preprint arXiv:2210.06849*, 2022. 3

[17] 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. 2, 3

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 6

[19] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7010–7019, 2023. 3

[20] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022. 2

[21] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024. 3

[22] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv preprint arXiv:2309.16650*, 2023. 3

[23] Zoey Guo, Yiwen Tang, Ray Zhang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. Viewrefer: Grasp the multi-view knowledge for 3d visual grounding. In *International Conference on Computer Vision (ICCV)*, pages 15372–15383, 2023. 2, 3

[24] 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)*, pages 20482–20494, 2023. 3

[25] Yining Hong, Zishuo Zheng, Peihao Chen, Yian Wang, Junyan Li, and Chuang Gan. Multiply: A multisensory object-centric embodied large language model in 3d world. *arXiv preprint arXiv:2401.08577*, 2024. 3

[26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 6

[27] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 2, 3, 4, 6

[28] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning (ICML)*, pages 9118–9147. PMLR, 2022. 3

[29] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 4

[30] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 3

[31] 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. 2, 3

[32] Shunya Kato, Shuhei Kurita, Chenhui Chu, and Sadao Kurohashi. Arkitscenerefer: Text-based localization of small objects in diverse real-world 3d indoor scenes. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 784–799, 2023. 2

[33] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, pages 19729–19739, 2023. 3

[34] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning (CoRL)*, pages 80–93. PMLR, 2023. 3, 4

[35] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 3

[36] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212, 2022. 3

[37] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. *arXiv preprint arXiv:2312.16217*, 2023. 3

[38] Zeju Li, Chao Zhang, Xiaoyan Wang, Ruilong Ren, Yifan Xu, Ruifei Ma, and Xiangde Liu. 3dmit: 3d multi-modal instruction tuning for scene understanding. *arXiv preprint arXiv:2401.03201*, 2024. 3

[39] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023. 3

[40] Bill Yuchen Lin, Chengsong Huang, Qian Liu, Wenda Gu, Sam Sommerer, and Xiang Ren. On grounded planning for embodied tasks with language models. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, pages 13192–13200, 2023. 3

[41] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. *arXiv preprint arXiv:2403.03174*, 2024. 4

[42] Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[43] Xianzheng Ma, Yash Bhalgat, Brandon Smart, Shuai Chen, Xinghui Li, Jian Ding, Jindong Gu, Dave Zhenyu Chen, Songyou Peng, Jia-Wang Bian, et al. When llms step into the 3d world: A survey and meta-analysis of 3d tasks via multi-modal large language models. *arXiv preprint arXiv:2405.10255*, 2024. 3

[44] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *2nd Workshop on Mobile Manipulation and Embodied Intelligence at ICRA 2024*, 2024. A3

[45] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel Chang, and Manolis Savva. Multiscan: Scalable rgbd scanning for 3d environments with articulated objects. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:9058–9071, 2022. 5

[46] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–824, 2023. 3

[47] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 5

[48] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. In *7th Annual Conference on Robot Learning*, 2023. 2, 3, 4

[49] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision (ECCV)*, pages 125–141. Springer, 2022. 2

[50] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10740–10749, 2020. 3, 4

[51] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *International Conference on Computer Vision (ICCV)*, pages 2998–3009, 2023. 3

[52] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3

[53] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7658–7667, 2019. 2, 5

[54] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[55] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. *arXiv preprint arXiv:2312.16170*, 2023. 2

[56] Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023. 3

[57] 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. 3

[58] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19231–19242, 2023. 3

[59] Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*, 2023. 3

[60] 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

[61] Cheng-Fu Yang, Haoyang Xu, Te-Lin Wu, Xiaofeng Gao, Kai-Wei Chang, and Feng Gao. Planning as in-painting: A diffusion-based embodied task planning framework for environments under uncertainty. *arXiv preprint arXiv:2312.01097*, 2023. 3

[62] Xiaohan Zhang, Yifeng Zhu, Yan Ding, Yuke Zhu, Peter Stone, and Shiqi Zhang. Visually grounded task and motion planning for mobile manipulation. In *International Conference on Robotics and Automation (ICRA)*, pages 1925–1931. IEEE, 2022. 3

[63] 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*, pages 15225–15236, 2023. 2, 3

[64] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *International Conference on Computer Vision (ICCV)*, 2021. 3

[65] Lichen Zhao, Daigang Cai, Jing Zhang, Lu Sheng, Dong Xu, Rui Zheng, Yinjie Zhao, Lipeng Wang, and Xibo Fan. Towards explainable 3d grounded visual question answering: A new benchmark and strong baseline. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 3

[66] Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[67] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 3

[68] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2020. 6

[69] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *International Conference on Computer Vision (ICCV)*, pages 2911–2921, 2023. 2, 3, 6

[70] 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. *arXiv preprint arXiv:2405.11442*, 2024. 2, 3, 6

[71] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15116–15127, 2023. 6

A Appendix

A.1 Details of Dataset Construction

Detailed Prompt used in Task Generation   The prompt messages employed in the task generation process are depicted in Fig. A1, with the "System prompt" specifically illustrated in Fig. A2. Specific response examples, denoted as "<EXAMPLES>" in the system prompt, are presented in Fig. A3. We deliberately omit to show GPT-4 the corresponding scene graph for the provided response examples, as an overly long context increases the likelihood of errors.

> messages = [{'role': 'system', 'content': System prompt}, {'role': 'user', 'content': Scene graph of the scene to process}]

Figure A1: Prompts messages for GPT-4 task generation.

Details in Human Verification   Fig. A4 shows the interface used for human verification. The interface consists primarily of an interactive 3D mesh window and a right-hand column that displays task data. When a specific step is selected, the target object is highlighted within the mesh using a red bounding box. Users can rotate, translate, and zoom in or out within the 3D mesh window. Annotators mark each step with a tick or a cross. Following this verification process, tasks containing one incorrect step are manually revised by ourselves.

A.2 Additional Data Statistics

The statistics for task and step counts in the training and validation splits are presented separately in Tab. A1.

Table A1: Statistics of the training and evaluation splits for various datasets.

		Training Set	Evaluation Set	Train+Eval
3RScan	# tasks	2,056	138	2,194
	# steps	10,622	696	11,318
ScanNet	# tasks	2,731	443	3,174
	# steps	13,634	2,108	15,742
MultiScan	# tasks	504	43	547
	# steps	2,459	224	2,683
ARKitScenes	# tasks	6,952	443	7,395
	# steps	37,552	2,335	39,887
HM3D	# tasks	8,146	890	9,036
	# steps	38,833	3,873	42,706
Total	# tasks	20,389	1,957	22,346
	# steps	103,100	9,236	112,336

A.3 Implementation Details

During the training phase for 3D-VisTA and PQ3D, we concatenate t and $\{a_1, ..., a_p\}$ to predict the object o_p at each step p during gradient updates. For the training of LEO, which allows for multiple [GRD] tokens in a single forward pass, we concatenate t and $\{a_1, ..., a_n\}$ to predict all objects simultaneously. In the 3D LLM LEO model, object tokens are first processed by a grounding head, which consists of a two-layer multi-layer perceptron, before being fed into the LLM. To ensure a fair comparison among these three models, we employ the same PointNet++ encoder across all models. Beam search is utilized in LEO for generating action steps and the [GRD] token, with the beam width set to 4.

You are a helpful assistant that can generate diverse tasks in an indoor scene.

The scene is represented by a scene graph in the JSON dictionary format. Each entity in the scene graph denotes an object instance, named '<category>-<ID>'. The 'caption' describes the object's attributes, such as 'color', 'material', etc. The 'relations' describes the object's spatial relations with other objects. For example, from the scene graph:
```

'sofa-1': 'relations': ['to the right of armchair-2', 'in front of table-3'], 'caption': 'Grey velvet sofa with a rectangular shape and a back and arms, suitable for use in a living room.', 'armchair-2': 'relations': ['to the left of sofa-1'], 'caption': 'The armchair is made of leather, specifically black leather, and has a spherical shape.', 'table-3': 'relations': [], 'caption': 'The table is a rectangular wooden table with a brown finish, sometimes used as a dining table or coffee table, with a smooth wooden texture and various styles, including a sign or place setting on it, and can have plates or a white cloth on it.'
```

You can know that 'sofa-1' is grey, the 'armchair-2' is made of leather, the 'table-3' is made of wood, the 'armchair-2' is on the left of the 'sofa-1', the 'sofa-1' is in front of the 'table-3'.

Using the provided scene graph, design daily tasks that a person can do in this scene. Besides, decomposing every task into a sequence of steps that can be performed using the objects in this scene. For each step, give the target object that the person should attend to. Your output must follow the template below:
```
Task: #Describe the task using one sentence.#
Steps:
1. #The step must perform only one action. Split actions such as 'pick up xxx and place it xxx' into two separate steps. All objects, attributes, and relations must be explicitly listed in the given scene graph. Do not include the IDs of the objects, use ordinal words, attributes, and relations to refer to different object instances of the same category. Use pronouns ('it', 'them', 'here', and 'the other', etc.) as much as possible to make the step concise.# [#Use '<category>-<ID>' to denote the target object. Do NOT assume objects that do not exist in the scene graph! Each step must have exactly one target object. #]
2. ...
3. ...
...
```

Here are some examples:
```
<EXAMPLES>
```

Generate 5 different tasks involving different objects and separate these tasks by "===".

Figure A2: System prompt for GPT-4 task generation.

A.4 Social Impacts

Sequential grounding models significantly impact various societal domains. Enhanced 3D interaction capabilities in advanced embodied agents can substantially improve assistive technologies for individuals with disabilities, facilitating daily activities and enriching their quality of life. In healthcare, these models augment the functionality of robotic assistants, enabling more efficient navigation and operation within complex hospital environments.

However, there are notable negative social implications to consider. As the proficiency of sequential grounding technologies advances, so too does their ability to monitor and analyze detailed environmental and activity data. This escalation in capability may increase surveillance potential, potentially encroaching upon individual privacy rights, particularly if deployed in public or semi-public spaces without adequate regulatory frameworks and protective measures.

A.5 Generation ability of LEO

In this additional experiment, given the task description t, we ask LEO to generate both action steps $\{a_1, .., a_n\}$ and object $\{o_1, ..., o_n\}$. Since the action steps can be rearranged in various topological orders, we do not use a perfect match to measure the similarity between the predicted plan and the ground truth plan. Instead, we employ metrics from OpenEQA [44], which utilize GPT-4 to score the model's response based on ground truth. A score of 1 indicates no match, while a score of 5 indicates a perfect match. In our experiments, the GPT score on ScanNet is 2.1 ± 1.0, suggesting significant room for improvement. The prompts used for score computation are provided in Fig. A5.

===
Task: Make me a cup of coffee.
Steps:
1. Go to the long desk against the wall. [desk-15]
2. Choose a cup from those white, plastic cups on the top of the desk. [cups-19]
3. Fill it with coffee at the coffee maker. [coffee maker-16]
4. Walk to the table close to a cabinet. [table-23]
5. Put the cup down. [table-23]
6. Return to the long desk. [desk-15]
7. Fetch a plate from a bunch of steel plates below a picture frame hanging on the wall. [plates-17]
8. Go back to the table. [table-23]
9. Put the cup on the plate on the table. [table-23]
===
Task: Watch tv from the sofa.
Steps:
1. Go to the black table to the left of the fire extinguisher. [table-30]
2. Grab the black remote lying on it. [remote-36]
3. Turn on the tv with the remote. [tv-38]
4. Walk to the table in the middle of the bed and the white cabinet. [table-58]
5. Place the remote here. [table-58]
6. Walk to the black sofa close to the bed. [sofa-14]
7. Sit here to admire tv show. [sofa-14]
===
Task: Clean the mirror.
Steps:
1. Walk to the white cabinet. [cabinet-7]
2. Grab the towel on it. [towel-10]
3. Put the towel into the sink. [sink-37]
4. Turn the faucet on. [faucet-13]
5. Wet the towel in the sink. [sink-37]
6. Turn the faucet off. [faucet-13]
7. Wipe the mirror with the towel. [mirror-11]
8. Put the towel into the sink again. [sink-37]
9. Turn the faucet on. [faucet-13]
10. Wash the towel in the sink. [sink-37]
11. Turn the faucet off. [faucet-13]
12. Wring the towel dry in the sink. [sink-37]
13. Put it back to the cabinet. [cabinet-7]
===
Task: Browse the internet.
Steps:
1. Walk to the desk adorned with papers. [desk-19]
2. Turn on the computer tower behind the desk and the bookshelf. [computer tower-7]
3. Sit down on the nearest chair. [chair-26]
4. Fetch the mouse on the desk. [mouse-8]
5. Look at the screen of the monitor. [monitor-14]
===
Task: Go to sleep.
Steps:
1. Go to the curtain. [curtain-11]
2. Close it. [curtain-11]
3. Walk to the nightstand with the telephone. [nightstand-15]
4. Turn off the lamp on this nightstand. [lamp-19]
5. Go to the other nightstand. [nightstand-14]
6. Set the alarm on it. [alarm clock-28]
7. Lie down on the bed. [bed-20]

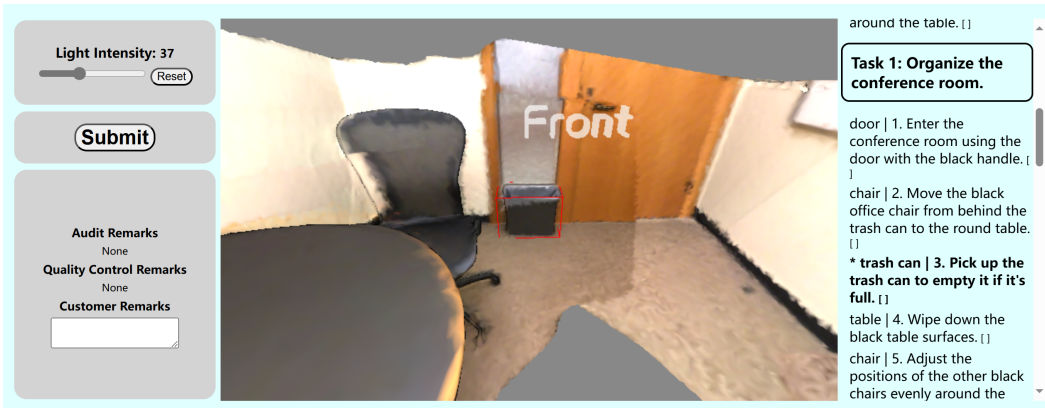Figure A3: <EXAMPLES> in system prompt for GPT-4 task generation.

Figure A4: Screenshot of the interface for human verification.

You are a helpful assistant that can evaluate the quality of task planning given a scene, a task description, a ground truth task planning, and a predicted task planning. To mark a response, you should output a single integer between 1 and 5 (including 1, 5), with format ```Your mark: number```. 5 means that the predicted task planning perfectly solves the problem described in the task and matches the ground truth task planning. 1 means that the predicted task planning is completely irrelevant to the task description and does not match the ground truth task planning.

The scene is represented by a scene graph in the JSON dictionary format. Each entity in the scene graph denotes an object instance, named '<category>-<ID>'. The 'caption' describes the object's attributes, such as 'color', 'material', etc. The 'relations' describes the object's spatial relations with other objects. For example, from the scene graph:
```
'sofa-1': 'relations': ['to the right of armchair-2', 'in front of table-3'], 'caption': 'Grey velvet sofa with a rectangular shape and a back and arms, suitable for use in a living room.', 'armchair-2': 'relations': ['to the left of sofa-1'], 'caption': 'The armchair is made of leather, specifically black leather, and has a spherical shape.', 'table-3': 'relations': [], 'caption': 'The table is a rectangular wooden table with a brown finish, sometimes used as a dining table or coffee table, with a smooth wooden texture and various styles, including a sign or place setting on it, and can have plates or a white cloth on it.'
```

You can know that 'sofa-1' is grey, the 'armchair-2' is made of leather, the 'table-3' is made of wood, the 'armchair-2' is on the left of the 'sofa-1', the 'sofa-1' is in front of the 'table-3'.

Using the provided scene graph, you should decide whether predicted task planning can solve the problem described in task description.
Here are some examples:
```
<example>
```

Your Turn, output with format ```Your mark: number```.
Scene graph: <scene graph>
Task description: <task description>
Ground truth task planning text: <gt plan text>
Ground truth object id: <gt object id>
Predicted task planning text: <pred plan text>

Figure A5: Prompt messages for computing GPT score.