Cook and Clean Together: Teaching Embodied Agents for Parallel Task Execution

Dingkang Liang[1], **Cheng Zhang**[1], **Xiaopeng Xu**[2], **Jianzhong Ju**[3], **Zhenbo Luo**[3], **Xiang Bai**[1]

[1]School of Software Engineering, Huazhong University of Science and Technology
[2]School of Electronic Information and Communications, Huazhong University of Science and Technology
[3]MiLM Plus, Xiaomi Inc.
{dkliang, czhang2024, xbai}@hust.edu.cn, {jujianzhong, luozhenbo}@xiaomi.com

Abstract

Task scheduling is critical for embodied AI, enabling agents to follow natural language instructions and execute actions efficiently in 3D physical worlds. However, existing datasets often simplify task planning by ignoring operations research (OR) knowledge and 3D spatial grounding. In this work, we propose **O**perations **R**esearch knowledge-based 3D Grounded Task **S**cheduling (ORS3D), a new task that requires the synergy of language understanding, 3D grounding, and efficiency optimization. Unlike prior settings, ORS3D demands that agents minimize total completion time by leveraging parallelizable subtasks, e.g., *cleaning the sink while the microwave operates*. To facilitate research on ORS3D, we construct ORS3D-60K, a large-scale dataset comprising 60K composite tasks across 4K real-world scenes. Furthermore, we propose GRANT, an embodied multi-modal large language model equipped with a simple yet effective scheduling token mechanism to generate efficient task schedules and grounded actions. Extensive experiments on ORS3D-60K validate the effectiveness of GRANT across language understanding, 3D grounding, and scheduling efficiency.

Code — https://github.com/H-EmbodVis/GRANT
Dataset — https://huggingface.co/datasets/H-EmbodVis/ORS3D-60K

1 Introduction

Task scheduling is fundamental for embodied agents to efficiently execute human-assigned tasks (Duan et al. 2022; Wang et al. 2024; Huang et al. 2024b; Driess et al. 2023). Achieving this requires the seamless integration of natural language understanding, efficiency optimization, and spatial perception within real-world 3D environments.

Recently, several works (Huang et al. 2024b; Chen et al. 2024; Zhang et al. 2024) have made preliminary attempts on plan generation in 3D environments, allowing models to generate step-by-step plans from human instructions (Fig. 1(a)). Nevertheless, these attempts are oversimplified and exhibit critical limitations that hinder their practical applications. First, they lack consideration of task properties and optimization of efficiency. Under their setting, a model only needs to generate plausible actions in terms of natural language. In contrast, as shown in Fig. 1(b), an embodied agent is assumed to have the capacity to efficiently complete the task by leveraging **O**perations **R**esearch (OR) knowledge. This includes identifying which subtasks can be executed concurrently with other subtasks and maximizing the use of waiting time to achieve optimal efficiency. Second, although their setting assumes an agent operating in 3D environments, it is often reduced to textual question answering, without explicitly grounding each step to the target object's location within the 3D scene. This lack of spatial grounding severely hinders the utility of such plans for downstream embodied executions that require spatial location information (e.g., navigation).

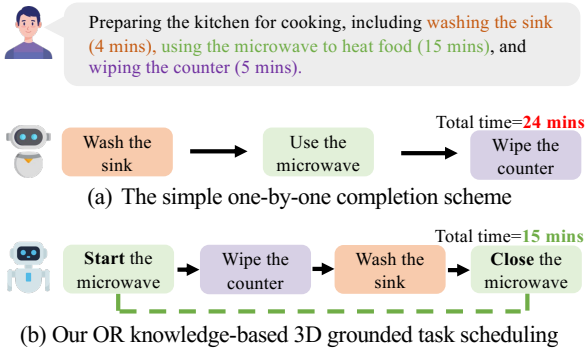


Figure 1: Comparison of different task completion schemes. An embodied agent is expected to use operations research knowledge to efficiently complete tasks through scheduling.

To address these limitations and extend the capability of embodied agents for efficient task scheduling, we propose a new and practical task named **O**perations **R**esearch knowledge-based 3D Grounded Task **S**cheduling (**ORS3D**). In this task, an embodied agent must generate efficient schedules by leveraging OR knowledge and locate the 3D positions of target objects in each action step to complete assigned tasks. As demonstrated in Fig. 2, when we assign a composite task to an embodied agent, we hope it can efficiently complete it by utilizing the waiting periods of subtasks that can be performed concurrently. For example, "Using the microwave" allows the agent to perform other subtasks during its waiting period. To achieve maximum efficiency, the embodied agent must leverage these subtask properties and

Dingkang Liang and Cheng Zhang contributed equally.
Corresponding author: Xiang Bai.

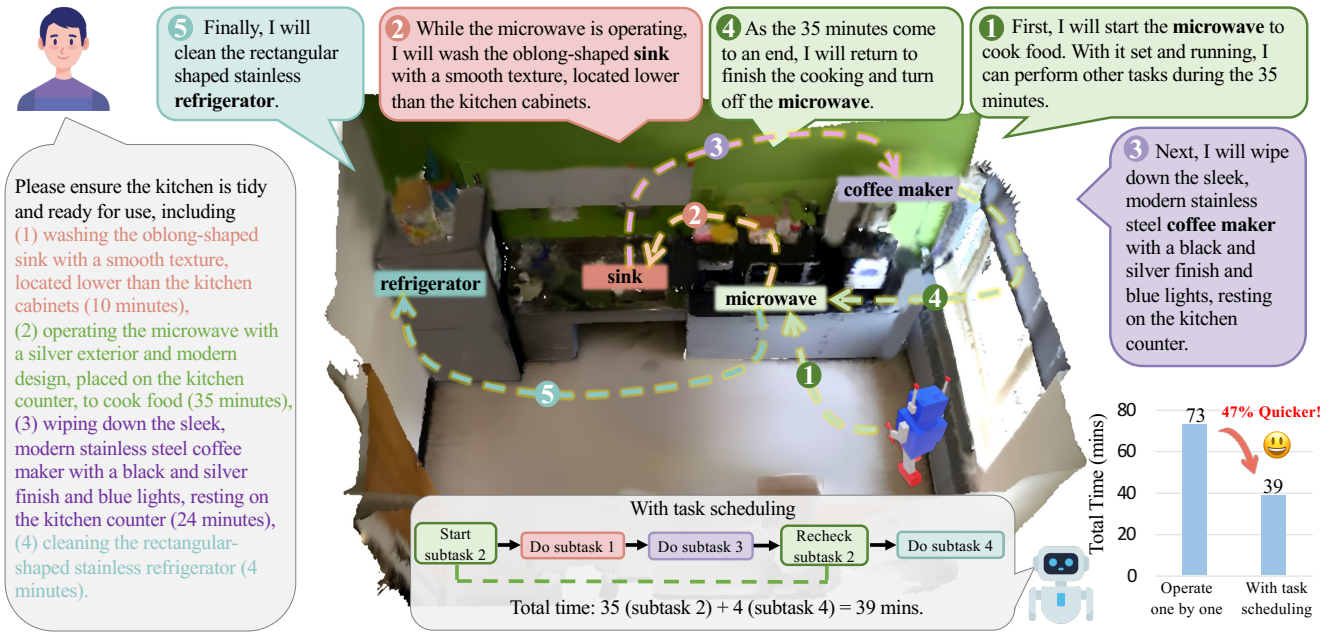


Figure 2: Illustration of the proposed Operations Research knowledge-based 3D Grounded Task Scheduling (ORS3D). When assigned a composite task by a human, the embodied agent needs to complete the subtasks efficiently by carefully scheduling using operations research knowledge and simultaneously locating the target objects in each step for navigation and manipulation.

incorporate OR knowledge to generate an optimal task schedule. Meanwhile, to execute each step in the real world, the agent must accurately localize the target objects within the 3D scene. Therefore, ORS3D poses significant challenges to existing 3D agents (Huang et al. 2024b,a; Deng et al. 2025; Lin et al. 2023; Chen et al. 2024) in two essential aspects: **1)** It requires OR knowledge to identify subtasks that can be performed concurrently and make efficient task schedules. **2)** It entangles language and spatial understanding (i.e., an embodied agent is required to simultaneously generate actions and locate the target objects in the 3D scene).

To facilitate research on this new task, we construct the ORS3D-60K dataset consisting of 60,825 composite tasks across 4,376 real-world indoor scenes. As shown in Tab. 1, compared to existing 3D understanding and task planning-related datasets (Wu et al. 2023; Chen et al. 2024; Zhang et al. 2024; Huang et al. 2024b; Zhu et al. 2024a), ORS3D-60K is the first to incorporate OR knowledge. It also has the largest number of tasks and presents the most significant challenge by requiring models to generate lengthy textual solutions and provide 3D grounding for target objects. To assess the capability of existing methods in addressing this task, we evaluate several baselines (Huang et al. 2024b; Zhu et al. 2024b, 2023; Chen et al. 2024) from language understanding, efficiency optimization, and spatial perception, where the results show they struggle with this challenging task.

To tackle the ORS3D problem, we further propose a **gr**ounded t**a**sk scheduling ag**ent** named **GRANT**, which is empowered by a Multi-modal Large Language Model (MLLM) and equipped with a simple yet effective Scheduling Token Mechanism (STM) to generate efficient task schedules. Specifically, we introduce a learnable scheduling token that links to an external optimization solver to generate task schedules based on task property constraints provided by the MLLM. The solver employs a dynamic programming algorithm to arrange subtasks within the available time intervals of those that can be performed concurrently, producing an optimal execution schedule. During inference, GRANT first predicts subtask properties as constraints, then uses the scheduling token to invoke the solver and generate the optimal schedule, which is subsequently injected back into the model to guide the generation of step-wise action descriptions and target object groundings. Compared to the baseline method (Chen et al. 2024), our approach yields a significant 30.53% improvement in task completion time efficiency, along with notable gains of 1.38% in grounding accuracy and 10.46% in overall performance. As an initial attempt, our method paves the way for further exploration in ORS3D.

In summary, our contributions are as follows: 1) We introduce operations research knowledge-based 3D grounded task scheduling, a new and practical task that meets the common requirement of embodied agents to efficiently complete tasks in the physical world. 2) To support this new task, we construct a large-scale dataset, ORS3D-60K. To the best of our knowledge, we are the first to incorporate operations research knowledge for task scheduling in 3D scenarios. 3) We propose GRANT, an embodied MLLM with a simple yet effective scheduling token mechanism, integrating task scheduling with multimodal understanding to generate efficient, grounded task execution schedules.

2 Related Works

2.1 Task Planning

Task Planning (Ahn et al. 2022; Choi et al. 2024; Zhang et al. 2024; Chen et al. 2023) is crucial, as it enables embod-

Dataset	Reference	#Scenes	#Task	Avg. length	Text output	3D Grounding	Planning	OR knowledge
TaPA (Wu et al. 2023)	arXiv 23	80	15,418	69	✓	✗	✓	✗
Embodied planning (Chen et al. 2024)	arXiv 24	1,319	4,357	37	✓	✓	✓	✗
SG3D (Zhang et al. 2024)	arXiv 24	4,895	22,346	71	✗	✓	✓	✗
ScanReason (Zhu et al. 2024a)	ECCV 24	1,456	12,929	29	✗	✓	✗	✗
LEO (Task planning) (Huang et al. 2024b)	ICML 24	478	13,848	98	✓	✗	✓	✗
Intent3D (Kang et al. 2025)	ICLR 25	1,042	44,990	9	✗	✓	✗	✗
ORS3D-60K (ours)	-	4,376	**60,825**	**311**	✓	✓	✓	✓

Table 1: Comparison with related datasets. "Avg. length" denotes the average word length of each data item. Our dataset is the only one that introduces Operations Research (OR) knowledge for task scheduling.

ied agents to execute human instructions efficiently. Wu et al. (Wu et al. 2023) propose TaPA, a vision-language task planning agent that generates executable textual action steps for robot navigation and manipulation using multi-view images of the 3D scene. Huang et al. (Huang et al. 2024b) construct a task planning dataset that requires embodied agents to generate step-wise plans from instructions. SG3D (Zhang et al. 2024) proposes task-oriented sequential grounding in 3D scenes, where an agent is required to locate each target object in a given sequence of actions. In contrast to previous works, we focus on more complex scheduling scenarios and the integration of multi-modal information processing.

2.2 3D Scene Understanding

3D scene understanding is the foundation of embodied AI, enabling it to act in real-world scenes. 3D scene understanding includes depth estimation (Xu et al. 2023, 2025a), 3D object detection (Kolodiazhnyi et al. 2024b; Liang et al. 2025b; Zhou et al. 2025), segmentation (Takmaz et al. 2023; Xu et al. 2025b; Liang et al. 2024, 2025a), and grounding (Chen, Chang, and Nießner 2020; Huang et al. 2024c; Jiang et al. 2024). Mask3D (Schult et al. 2023) is often used as an off-the-shelf object proposal extractor for downstream tasks or as a 3D scene encoder, as the flexible learned instance queries can be easily assembled to Transformer-based LLMs. OneFormer3D (Kolodiazhnyi et al. 2024a) is an end-to-end method that performs instance and semantic segmentation consistently, utilizing a group of learnable instance queries.

2.3 3D Multi-modal Large Language Models

3D MLLMs (Chen et al. 2024; Wang et al. 2023; Huang et al. 2024a; Kang et al. 2024; Fu et al. 2025; Chen et al. 2025; Zhu et al. 2024b, 2025; Hong et al. 2023) narrow the gap between spatial understanding and natural language processing. Several methods (Huang et al. 2024b; Zhu et al. 2023; Kang et al. 2024; Zhu et al. 2024b) utilize point cloud object proposals from off-the-shelf 3D object detectors to extract 3D scene information. Another line of research (Hong et al. 2023; Zhu et al. 2025) leverages pretrained 2D encoders to reconstruct 3D information for the LLMs. In contrast, other approaches like Grounded 3D LLM (Chen et al. 2024) and 3D-LLaVA (Deng et al. 2025) directly process scene point clouds using 3D scene encoders that are jointly trained with LLMs. However, although existing 3D MLLMs excel at scene understanding, they still lack the ability to leverage OR knowledge for efficient task scheduling and completion.

3 The ORS3D-60K Dataset

In this section, we introduce the definition of **O**perations **R**esearch knowledge-based 3D Grounded Task **S**cheduling (ORS3D) and provide details of the proposed ORS3D-60K dataset.

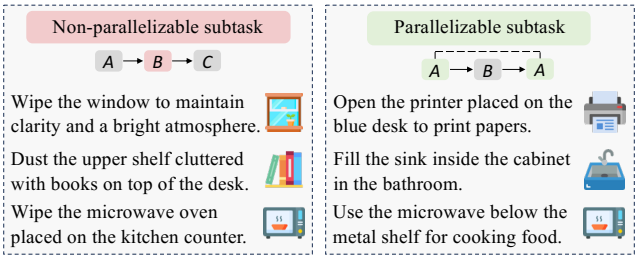


Figure 3: Non-parallelizable subtask & parallelizable subtask.

3.1 Design Principles

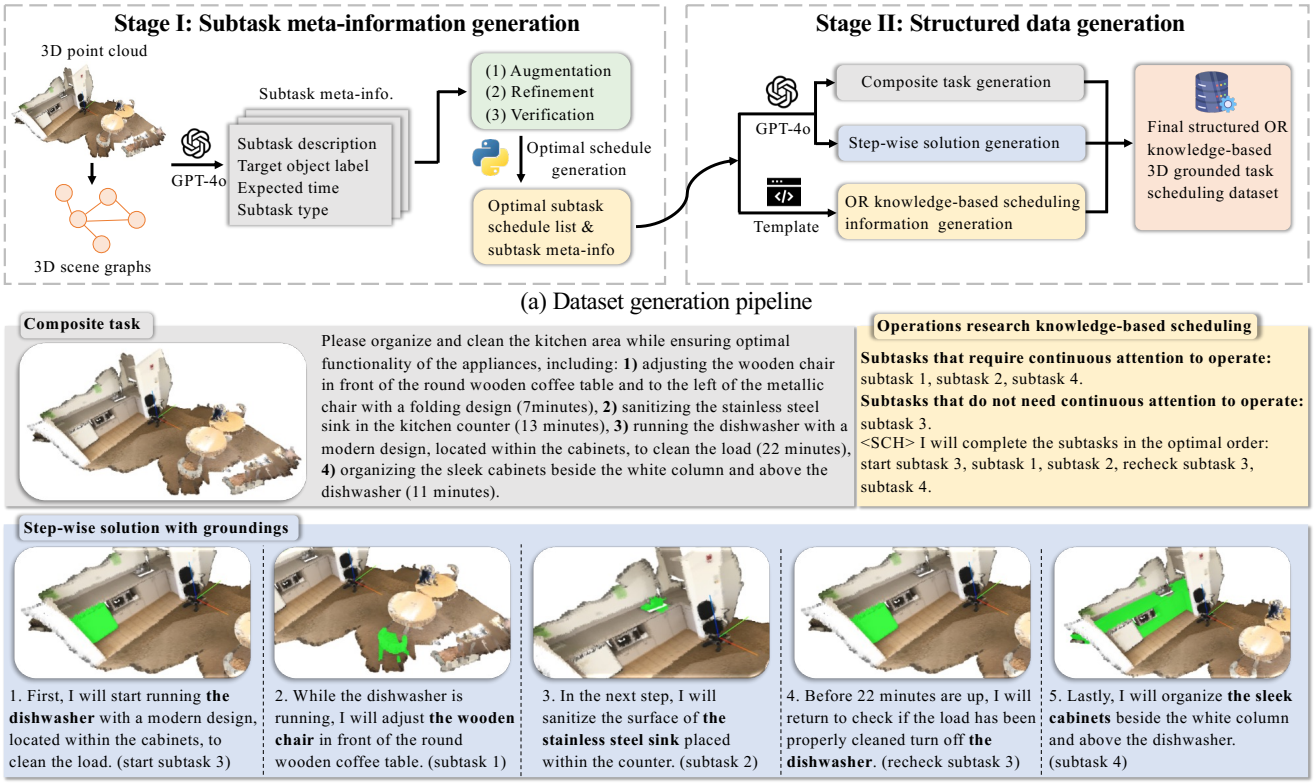
Understanding human instructions, making efficient schedules to complete human-assigned tasks, and interacting with objects are common and frequent requirements in real-world applications for embodied agents.

As illustrated in Fig. 3, tasks assigned to embodied agents can be categorized into two types from an OR perspective: 1) **Non-parallelizable subtask** requires continuous attention of the agent to manipulate the target object, such as wiping the table or dusting the shelf. 2) **Parallelizable subtask** only requires the agent to initiate and recheck the target object upon completion, without continuous attention and manipulation, such as using the microwave to heat food or filling the water sink. The agent needs to exploit the time intervals of parallelizable subtasks to achieve an efficiency objective.

3.2 Problem Formulation

The goal of OR knowledge-based 3D grounded task scheduling is to generate an efficient schedule and accurately locate the target object at each step to complete a composite task.

Specifically, suppose that an embodied agent in a 3D scene is assigned a composite task consisting of n subtasks, denoted as $\mathcal{C} = \{\tau_i\}_{i=1}^{n}$. Each subtask τ_i is an operation involving a target object with an expected time, described by a natural language instruction. To achieve efficient task scheduling, the agent needs to generate a time-efficient schedule $\mathcal{A} = \{a_i|(\tau_i, l_i)\}_{i=1}^{s}$ consisting of s steps to accomplish



(a) Dataset generation pipeline

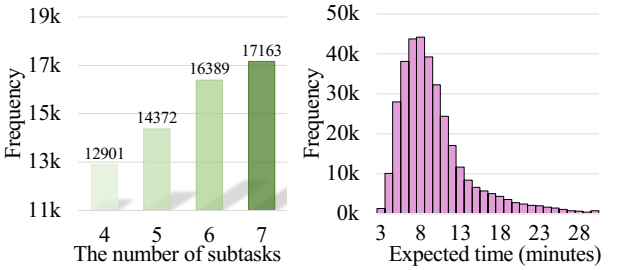
(b) An example of the composite task

Figure 4: (a) The ORS3D-60K dataset generation pipeline, which first generates subtask meta-information from 3D scene graphs, then uses this information to generate the structured dataset. (b) A composite task example from ORS3D-60K dataset. The green color mask indicates the ground-truth target object in the corresponding step.

the composite task. Each step includes a textual action description a_i for subtask τ_i and the 3D location l_i (e.g., 3D bounding box or point mask) of the target object.

3.3 Dataset Construction

The dataset construction pipeline is illustrated in Fig. 4(a). In Stage I, we use 3D point clouds from five real-world datasets: ScanNet (Rozenberszki, Litany, and Dai 2022), HM3D (Ramakrishnan et al. 2021), ARKitScenes (Baruch et al. 2021), 3RScan (Wald et al. 2019), and MultiScan (Mao et al. 2022). They are converted into textual 3D scene graphs (Jia et al. 2024) for subtask meta-information generation via GPT-4o. We refine the outputs for correctness and completeness. For each subtask, we predefine an expected execution time based on its type and complexity, then randomly perturb these times by $\pm 10\%$ to generate diverse optimal schedules. In Stage II, we compute the optimal task schedule using an optimization solver, then convert it into step-wise natural language instructions with phrase-level object grounding via GPT-4o. We also generate OR knowledge-based scheduling explanations using templates. Fig. 4(b) presents a data example from the ORS3D-60K dataset. The composite task comprises a list of subtasks that the embodied agent must complete. At each step, the model is required to simultaneously produce an action description of the operation on a subtask and locate the target object.



(a) #Subtask in each composite task (b) Expected time of each subtask

Figure 5: Distributions of (a) subtask number in each composite task, and (b) the expected time of each subtask.

3.4 Dataset Characteristics

The ORS3D-60K dataset exhibits several distinctive characteristics that make it stand out from existing datasets.

First, our dataset is closely aligned with real-world task-completion scenarios, extending beyond existing 3D visual grounding and question-answering datasets (Chen, Chang, and Nießner 2020; Kang et al. 2025; Zhu et al. 2024a). It is characterized by the inclusion of OR knowledge, which is not considered in existing related 3D understanding datasets.

Second, as shown in Tab. 1, ORS3D-60K has an exceptionally high average text length of 311 words, which poses a

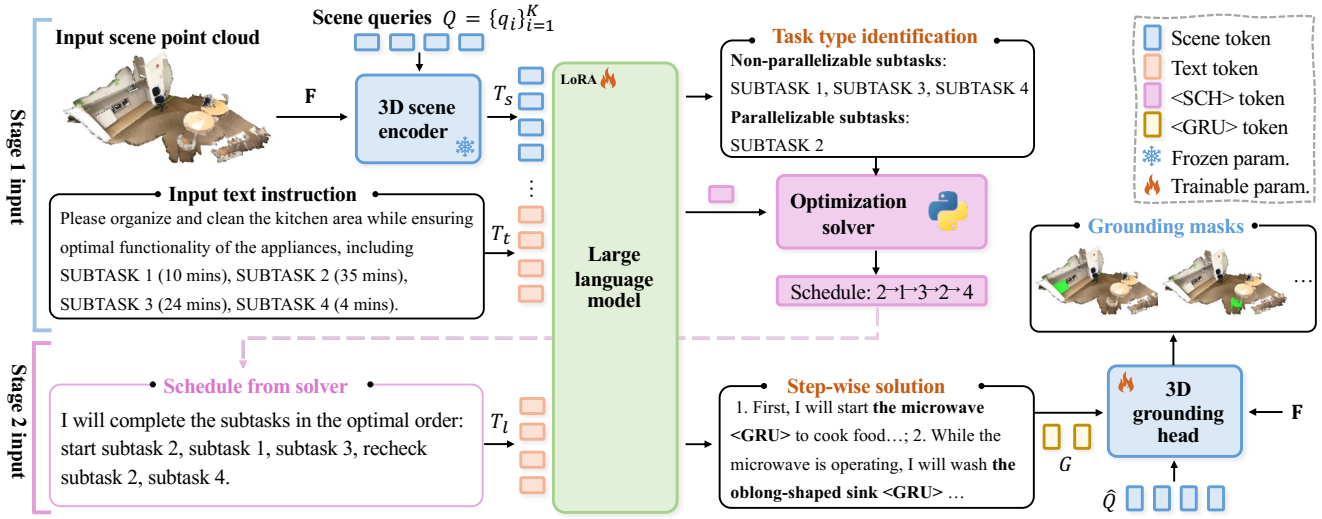


Figure 6: Overview of GRANT. The scene point cloud is processed by a 3D scene encoder into scene tokens. GRANT first infers task properties (stage 1), then uses a scheduling token to generate an optimal schedule (stage 2). The grounding tokens are fed to the 3D grounding head to generate object masks. The input task description is simplified for brevity.

significant challenge for the language processing capabilities of embodied agents. ORS3D-60K dataset contains 60,825 composite tasks across 4,376 scenes, representing the largest scale among existing related datasets. Besides, as shown in Fig. 5(a), ORS3D-60K dataset covers different levels of difficulty, reflected by the varying number of subtasks (4 to 7) in each composite task. The expected time for each subtask (Fig. 5(b)) follows a long-tail distribution, reflecting real-world variability. Furthermore, in our ORS3D setting, 3D grounding is entangled within the text, requiring the model not only to understand what actions to take, but also to accurately identify where each action should occur in the 3D scene. These characteristics make ORS3D-60K dataset large-scale, diverse in task complexity, and highly representative of real-world scenarios.

4 Method

The ORS3D task requires integration of language understanding, efficiency optimization, and spatial perception in 3D environments. Therefore, we propose a **gr**ounded t**a**sk schedul**i**ng age**nt**, termed **GRANT**, where the overall architecture is illustrated in Fig. 6. Specifically, GRANT consists of four components: 1) A 3D scene encoder converting point clouds into scene tokens. 2) An LLM processing multimodal inputs for task understanding. 3) A scheduling token mechanism (STM) connecting the LLM to an optimization solver for efficient scheduling. 4) A grounding head generating point masks for object localization. The scene encoder and grounding head integrate with the LLM through specialized tokens, enabling end-to-end training for simultaneous schedule generation and spatial grounding.

4.1 Multi-modal Input Processing

GRANT takes both the scene point cloud and the textual description as input, which are first tokenized and subsequently fed into the LLM for unified understanding.

Point cloud tokenization. For a scene point cloud $\mathcal{P} \in \mathbb{R}^{N\times 6}$, where each point contains 6-dimensional information $[x, y, z, r, g, b]$ and N is the number of points, a sparse convolutional network is employed to extract point-wise features $\mathbf{F} \in \mathbb{R}^{N\times d}$, where d is the feature dimension. We then use a pre-trained 3D scene encoder to further encode the point cloud features into scene tokens. The 3D scene encoder $\mathcal{E}$ employs a fixed set of K learnable scene queries $Q = \{\boldsymbol{q}_i\}_{i=1}^{K}$, which interact with point cloud features via cross-attention to produce processed scene queries containing rich semantic information. This process can be formulated as:

$$\hat{Q} = \mathcal{E}(Q, \mathbf{F}), \tag{1}$$

where $\hat{Q} = \{\hat{\boldsymbol{q}}_i\}_{i=1}^{K}$ is the processed scene queries. To align with the token embedding dimension D of the LLM, the processed scene queries are projected into scene tokens $T_s = \{\boldsymbol{s}_i\}_{i=1}^{K}$ via a simple linear layer, where each $\boldsymbol{s}_i \in \mathbb{R}^{1\times D}$ represents a scene token.

Text tokenization. We employ a text tokenizer to convert the input composite task description into a sequence of text tokens $T_t = \{\boldsymbol{x}_i\}_{i=1}^{L}$, where each $\boldsymbol{x}_i \in \mathbb{R}^{1\times D}$ and L denotes the length of the input text.

LLM processing. The LLM plays a central role in our model by handling multi-modal inputs and understanding both point clouds and human instructions. It further identifies subtask types, solves complex task scheduling problems, generates descriptive action steps, and provides 3D positions of target objects. As shown in Fig. 6, the scene tokens are prepended to the text tokens and fed to the LLM, which generates output tokens in an auto-regressive manner. The output tokens include specially designed tokens for task scheduling and 3D grounding, which will be elaborated in the following sections.

Algorithm 1: Optimization Solver (single parallelizable subtask)

Input: $\mathcal{I} = \{(\tau_i, c_i, t_i)\}_{i=1}^{n}$
Output: Schedule S^*
Split subtasks into one parallelizable subtask τ_{P} (if any) and non-parallelizable set $S_{\overline{\mathrm{P}}}$
if no parallelizable subtask **then**
    $S^* \leftarrow S_{\overline{\mathrm{P}}}$ ▷ Purely sequential
else
    $T_{\mathrm{P}} \leftarrow$ duration of τ_{P}
    Select $S_{\text{in}} \subseteq S_{\overline{\mathrm{P}}}$ s.t. $\sum_{\tau_i \in S_{\text{in}}} t_i \leq T_{\mathrm{P}}$ and the sum is maximized
    $S_{\text{out}} \leftarrow S_{\overline{\mathrm{P}}} \setminus S_{\text{in}}$
    $S^* \leftarrow S_{\text{out}} + [\tau_{\mathrm{P}}] + S_{\text{in}} + [\tau_{\mathrm{P}}]$
end if
return S^*

4.2 Scheduling Token Mechanism

LLMs exhibit strong capabilities in natural language generation but are generally less effective at solving complex mathematical problems. To address this limitation, we introduce a special <SCH> token that connects to an external solver to obtain an optimal scheduling list. This list is then utilized to guide the LLM for step-wise action generation.

Specifically, for a composite task description, the LLM first identifies the parallelizable and non-parallelizable subtasks (defined in Sec. 3.1). It then constructs the subtask type information as $\mathcal{I} = \{(\tau_i, c_i, t_i)\}_{i=1}^{n}$, where $c_i \in \{\mathrm{P}, \overline{\mathrm{P}}\}$ denotes the subtask type (P: parallelizable, $\overline{\mathrm{P}}$: non-parallelizable) and t_i is the expected time. The information $\mathcal{I}$ is passed to an external optimization solver via the <SCH> token. As defined in Alg. 1, the solver minimizes the total execution time given the subtask types and their expected times. This is formulated as a 0–1 knapsack problem, where the waiting interval of a parallelizable subtask plays the role of the capacity and the durations of non-parallelizable subtasks serve as item weights and values, so that the solver maximizes the utilization of the waiting time of parallelizable subtasks while minimizing the overall completion time.

A simple dynamic programming algorithm is employed to solve this problem. The solver finally returns the optimal schedule of subtask IDs. This scheduling process can be formulated as:

$$S^* = \text{Solver}(\mathcal{I}), \tag{2}$$

where S^* is the optimal subtask completion schedule, represented as a list of subtask IDs. Then, S^* is converted into natural language using predefined templates, then tokenized into T_l by the text tokenizer and concatenated with the preceding tokens to guide the LLM in generating step-wise action descriptions.

4.3 3D Grounding Head

Besides generating action descriptions, the model also needs to simultaneously locate the corresponding target object in order to complete the task in the physical world. To achieve this, we use a special <GRU> token to indicate the target object for grounding in the output of LLM. To align with the dimension of the processed scene queries, all output <GRU> tokens are passed through a simple MLP head into $G = \{\boldsymbol{g}_j\}_{j=1}^{s}$, where each $\boldsymbol{g}_j \in \mathbb{R}^{1\times d}$.

The target scene query is selected through max cosine similarity. Specifically, we compute the cosine similarity between $\boldsymbol{g}_j$ and each $\hat{\boldsymbol{q}}_i$. The scene query with the highest probability is selected as the best match one, denoted as $\boldsymbol{q}^* \in \mathbb{R}^d$. This process can be formulated as:

$$\boldsymbol{q}^* = \arg\max_{\hat{\boldsymbol{q}}_i \in \hat{Q}} \left(\frac{\boldsymbol{g}_j \cdot \hat{\boldsymbol{q}}_i}{\|\boldsymbol{g}_j\| \cdot \|\hat{\boldsymbol{q}}_i\|} \right), \tag{3}$$

The grounding mask is generated by the matched scene query with point cloud features. The mask corresponding to the scene query $\boldsymbol{q}^*$ is computed by taking the dot product between $\boldsymbol{q}^*$ and the point cloud features, followed by a sigmoid activation to obtain a point mask, which is expressed as:

$$\boldsymbol{m} = \sigma\left(\mathbf{F} \cdot \boldsymbol{q}^*\right), \tag{4}$$

where $\boldsymbol{m} \in \mathbb{R}^N$ is the predicted point mask of target object.

Training objectives. For language modeling, we use next-token prediction with cross-entropy loss. For grounding, we align grounding tokens and scene queries via a similarity matrix and supervise it using a binary correspondence matrix with sigmoid focal loss.

5 Experiments

5.1 Implementation Details

The 3D scene encoder of GRANT is initialized with a pre-trained CLASP (Chen et al. 2024), freezing all weights except the projection layer used for alignment. We use Tiny-Vicuna-1B (Chiang et al. 2023) as the LLM and fine-tune it using LoRA (Hu et al. 2021). We use the AdamW optimizer with a cosine learning rate scheduler (initialized as 8×10^{-4}) and a weight decay of 0.1. Models are trained for 10 epochs on the ORS3D-60K training set with a batch size of 1. All experiments are conducted on 8× RTX 4090 GPUs.

5.2 Evaluation Metrics

The model performance is evaluated across three aspects that align with the challenges of the ORS3D task. For output language quality, we use NLP metrics (METEOR & ROUGE). For 3D grounding accuracy on the target object at each step, we adopt the AP@25% detection metric. Considering that the core aspect of the ORS3D task is *scheduling*, we introduce the **Time Efficiency (TE)** metric to measure how well the model utilizes the time intervals of parallelizable subtasks. Rather than using raw completion time, TE normalizes the efficiency of each schedule between a naive sequential baseline and the optimal schedule. Formally, the TE of a predicted task schedule is calculated as:

$$\mathrm{TE} = \frac{\mathcal{T}_{\text{worst}} - \mathcal{T}_{\text{pred}}}{\mathcal{T}_{\text{worst}} - \mathcal{T}_{\text{opt}}} \times 100\%, \tag{5}$$

where $\mathcal{T}_{\text{pred}}$ is the total time of the predicted task schedule, $\mathcal{T}_{\text{opt}}$ is the total time of the ground-truth optimal schedule obtained by the OR solver, and $\mathcal{T}_{\text{worst}}$ is the total time when all subtasks are executed sequentially without any parallelism. Intuitively, the numerator $\mathcal{T}_{\text{worst}} - \mathcal{T}_{\text{pred}}$ measures the time

Method	Venue	3D Obj. Det.	LLM	Language		Scheduling	3D Grounding	Overall ↑
				METEOR ↑	ROUGE ↑	TE ↑	Accuracy ↑	
Commercial LLM/MLLMs (only text input)								
Gemini	-	-	Gemini-2.0-flash	41.67	58.48	24.75		31.22
DeepSeek-R1 (Guo et al. 2025)	-	-	DeepSeek-V3	32.40	41.50	72.63	Unsupported	36.63
GPT-4o	-	-	GPT-4o	49.16	62.19	45.27		39.15
Object-level methods (with detected object proposals)*								
3D-VisTA (Zhu et al. 2023)	ICCV 23	Mask3D	-	Unsupported			54.90‡	13.73
PQ3D (Zhu et al. 2024b)	ECCV 24	Mask3D	-				56.12‡	14.03
LEO† (Huang et al. 2024b)	ICML 24	Mask3D	Vicuna-1B	46.61	60.32	45.63	Unsupported	38.14
Scene-level methods								
Grounded 3D LLM (Chen et al. 2024)	arXiv 24	-	Vicuna-1B	41.96	53.71	42.46	34.00	43.03
GRANT (**ours**)	-	-	Vicuna-1B	**42.82**	**62.78**	**72.99**	**35.38**	**53.49**

Table 2: Experiment results on ORS3D-60K test set. † We adapt LEO by replacing its LLM with Vicuna-1B for a fair comparison. * indicates that these methods require object point clouds from an external 3D detector like Mask3D (Schult et al. 2023). ‡ Results are produced by directly providing step-wise schedules as input. Overall is the average of METEOR, ROUGE, TE, and Grounding Accuracy (treating unsupported metrics as 0).

Method	3D Obj. Det.	Detection		Segmentation
		AP @0.25	AP @0.50	mIoU
3D-VisTA*	Mask3D	54.90	41.88	43.29
PQ3D*	Mask3D	56.12	44.01	46.37
Grounded 3D LLM	-	34.00	23.93	25.56
GRANT (**ours**)	-	**35.38**	**24.79**	**26.71**

(a) 3D grounding performance comparison

Method	Acc.	Para. subtask			Non-para. subtask			Scheduling
		Prec.	Recall	F1	Prec.	Recall	F1	TE
Grounded 3D LLM	77.14	73.80	50.15	59.72	95.17	82.66	88.48	42.46
LEO	79.73	**78.19**	41.57	54.28	95.90	87.37	91.43	45.63
GRANT (**ours**)	**84.65**	73.82	**54.70**	**62.84**	**95.94**	**90.67**	**93.23**	**72.99**

(b) Subtask type recognition and scheduling

Table 3: (a) Comprehensive 3D grounding performance. * indicates that these methods require object point clouds from an external 3D detector like Mask3D (Schult et al. 2023). (b) Impact of subtask type recognition on scheduling efficiency.

saved by the model compared to the naive baseline, while the denominator $\mathcal{T}_{\text{worst}} - \mathcal{T}_{\text{opt}}$ is the maximum possible saving for that task instance. Thus, TE reflects the fraction of the theoretically achievable time savings that the model actually realizes, with $\text{TE} = 0\%$ indicating purely sequential execution and $\text{TE} = 100\%$ matching the optimal schedule.

5.3 Main Results

We conduct a comprehensive comparison between our proposed GRANT and existing methods on ORS3D-60K dataset, as reported in Tab. 2. We evaluate commercial LLM/MLLMs by providing only the text part of the instructions, as they only support images as visual input. Notably, DeepSeek-R1 demonstrates strong performance on task scheduling (TE 72.63%) due to reinforcement learning on mathematical problems. However, these models cannot process point cloud data or directly locate objects in 3D environments, which limits their applicability in embodied scenarios.

Object-level methods (Zhu et al. 2023, 2024b) process on object point clouds obtained from 3D detectors like Mask3D (Schult et al. 2023). While these methods achieve high grounding accuracy, they are limited by their inability to handle long textual inputs and generate multimodal outputs. LEO (Huang et al. 2024b) integrates an LLM for enhanced language performance but lacks target object grounding. These methods focus on object-centric 3D understanding, which is insufficient for task-driven embodied scenarios where an agent requires a comprehensive understanding of the entire 3D environment. For scene-level methods, we use Grounded 3D LLM (Chen et al. 2024) as the baseline. By introducing the STM, our model achieves a substantial gain (30.53%) in task scheduling and further boosts 3D grounding by 1.38%. Overall, our method consistently outperforms baseline methods, validating its effectiveness across language understanding, 3D grounding, and scheduling efficiency.

We also compare the 3D target object grounding performance of different models in Tab. 3(a). Object-level methods achieve higher performance due to the use of additional 3D object detectors. However, their performance heavily depends on the detector's capability, introducing extra complexity in preprocessing 3D point clouds and leading to the loss of full scene information. In contrast, scene-level methods offer a cleaner alternative for 3D grounding by directly processing the entire scene point cloud, making them more suitable for real-world applications.

Accurate recognition of parallelizable subtasks is a prerequisite for effective time scheduling. As shown in Tab. 3(b), our model achieves the highest accuracy in subtask type

Setting	Language		Scheduling	3D Grounding
	METEOR	ROUGE	TE	Accuracy
No scheduling content	35.60	48.89	21.03	15.95
+ Scheduling content	41.29	55.28	47.04	34.74
+ STM (**ours**)	**42.82**	**62.78**	**72.99**	**35.38**
GT scheduling content	53.34	75.06	90.29	38.52

(a) Effect of scheduling token mechanism

Method	Subtask number				Overall
	Four	Five	Six	Seven	
PQ3D (Zhu et al. 2024b)	14.82	14.15	13.40	13.73	14.03
LEO (Huang et al. 2024b)	42.14	40.12	36.42	33.91	38.14
Grounded 3D LLM (Chen et al. 2024)	54.35	45.13	36.59	36.04	43.03
GRANT (**ours**)	**60.23**	**52.98**	**52.03**	**48.70**	**53.49**

(b) Performance across task difficulty levels

# LLM Params.	Language		Scheduling	3D Grounding
	METEOR	ROUGE	TE	Accuracy
1B	42.82	62.78	72.99	35.38
7B	45.19	63.55	73.21	36.25

(c) Effect of scaling LLM

Subtask number	4	5	6	7	10	20	50
Runtime (ms)	1.14	1.28	1.31	1.42	1.49	2.01	3.94

(d) Optimization solver runtime

Table 4: Ablation studies. (a) Effect of scheduling token mechanism. (b) Performance across different task difficulty levels. (c) Effect of scaling LLM. (d) Runtime of the optimization solver.

recognition, with substantial improvements in recall and F1-score for parallelizable subtasks, which in turn leads to significantly higher time efficiency. Therefore, robust subtask type recognition is critical for enhancing scheduling performance.

5.4 Ablation Studies

In this section, we conduct comprehensive ablation studies to validate the effectiveness of our proposed components and analyze the model's behavior under different settings. We specifically investigate the impact of the Scheduling Token Mechanism (STM), the robustness across varying task difficulty levels, the effect of scaling the LLM backbone, and the runtime efficiency of the optimization solver.

Effect of STM. We investigate the impact of the Scheduling Token Mechanism (STM) in Tab. 4(a). The baseline model, lacking scheduling awareness, tends to execute tasks sequentially, resulting in a low Time Efficiency (TE) of 21.03%. Simply adding textual scheduling hints improves TE to 47.04%, yet the model still struggles with optimal planning. In contrast, our STM significantly boosts TE by 25.95% to 72.99%, demonstrating that the external solver provides crucial guidance. The gap between STM and the ground-truth upper bound (90.29%) suggests further gains are possible with better subtask recognition. Notably, STM also benefits 3D grounding and language generation by providing a coherent plan context.

Task difficulty levels. We analyze model robustness across varying task complexities in Tab. 4(b). As the number of subtasks increases from 4 to 7, performance naturally declines due to the heightened challenge. However, GRANT consistently outperforms baselines across all levels. For instance, with 4 subtasks, GRANT achieves a score of 60.23 compared to 54.35 for Grounded 3D LLM. In the more challenging 7-subtask setting, it maintains a strong performance of 48.70 versus 36.04, confirming that GRANT is more resilient to complex long-horizon tasks.

Effect of scaling LLM. We explore the influence of LLM size in Tab. 4(c). Scaling from Vicuna-1B to 7B leads to improvements in language and grounding metrics. However, the gain in scheduling efficiency is relatively marginal (73.21% vs. 72.99%). Given the significant increase in computational resources required for the 7B model, we adopt Vicuna-1B as our default backbone. This choice strikes a favorable balance between performance and efficiency for embodied applications.

Solver runtime. Finally, we evaluate the computational overhead of the optimization solver in Tab. 4(d). The solver demonstrates exceptional efficiency, with runtime scaling linearly. For typical tasks (4-7 subtasks), it takes less than 1.5 ms, and remains below 4 ms even with 50 subtasks. This negligible overhead ensures the STM is highly suitable for real-time deployment in embodied agents.

5.5 Limitation

While this work demonstrates strong performance on the ORS3D-60K benchmark, future work will deploy the framework on physical robots to validate robustness in dynamic environments. Additionally, we will explore integrating the external optimization solver directly within the language model to enable end-to-end differentiable reasoning.

6 Conclusion

In this work, we introduce the ORS3D task that integrates task scheduling from operations research with spatial grounding for embodied agents. We construct the ORS3D-60K dataset, and propose GRANT, an embodied 3D MLLM equipped with the scheduling token mechanism to generate efficient task schedules with grounded actions. Experiments validate the capability of GRANT across language, grounding, and task scheduling. We believe our initial work on the OR knowledge-intensive scenario can inspire future research to further improve the complex planning capabilities and multi-modal integration of embodied agents.

Acknowledgments

This work was supported by the NSFC (Grant No. 62225603 and 623B2038) and in part by the Hubei Provincial Technology Innovation Program (Grant No. 2024BAA007).

References

Ahn, M.; Brohan, A.; Brown, N.; Chebotar, Y.; Cortes, O.; David, B.; Finn, C.; Fu, C.; Gopalakrishnan, K.; Hausman, K.; et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Baruch, G.; Chen, Z.; Dehghan, A.; Dimry, T.; Feigin, Y.; Fu, P.; Gebauer, T.; Joffe, B.; Kurz, D.; Schwartz, A.; et al. 2021. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *Proceedings of the Advances in Neural Information Processing Systems*.

Chen, D. Z.; Chang, A. X.; and Nießner, M. 2020. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Proceedings of European Conference on Computer Vision*, 202–221.

Chen, Y.; Ge, Y.; Ge, Y.; Ding, M.; Li, B.; Wang, R.; Xu, R.; Shan, Y.; and Liu, X. 2023. Egoplan-bench: Benchmarking multimodal large language models for human-level planning. *arXiv preprint arXiv:2312.06722*.

Chen, Y.; Sun, Y.; Chen, X.; Wang, J.; Shen, X.; Li, W.; and Zhang, W. 2025. Integrating Chain-of-Thought for Multimodal Alignment: A Study on 3D Vision-Language Learning. *arXiv preprint arXiv:2503.06232*.

Chen, Y.; Yang, S.; Huang, H.; Wang, T.; Xu, R.; Lyu, R.; Lin, D.; and Pang, J. 2024. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Choi, J.-W.; Yoon, Y.; Ong, H.; Kim, J.; and Jang, M. 2024. LoTa-Bench: Benchmarking Language-oriented Task Planners for Embodied Agents. *Proceedings of the International Conference on Learning Representations*.

Deng, J.; He, T.; Jiang, L.; Wang, T.; Dayoub, F.; and Reid, I. 2025. 3D-LLaVA: Towards Generalist 3D LMMs with Omni Superpoint Transformer. *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Driess, D.; Xia, F.; Sajjadi, M. S. M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; Huang, W.; Chebotar, Y.; Sermanet, P.; Duckworth, D.; Levine, S.; Vanhoucke, V.; Hausman, K.; Toussaint, M.; Greff, K.; Zeng, A.; Mordatch, I.; and Florence, P. 2023. PaLM-E: An Embodied Multimodal Language Model. In *Proceedings of the International Conference on Machine Learning*.

Duan, J.; Yu, S.; Tan, H. L.; Zhu, H.; and Tan, C. 2022. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2): 230–244.

Fu, H.; Zhang, D.; Zhao, Z.; Cui, J.; Liang, D.; Zhang, C.; Zhang, D.; Xie, H.; Wang, B.; and Bai, X. 2025. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of IEEE/CVF International Conference on Computer Vision*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; and Xu, R. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *Nature*.

Hong, Y.; Zhen, H.; Chen, P.; Zheng, S.; Du, Y.; Chen, Z.; and Gan, C. 2023. 3d-llm: Injecting the 3d world into large language models. *Proceedings of the Advances in Neural Information Processing Systems*, 20482–20494.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, H.; Chen, Y.; Wang, Z.; Huang, R.; Xu, R.; Wang, T.; Liu, L.; Cheng, X.; Zhao, Y.; Pang, J.; et al. 2024a. Chatscene: Bridging 3d scene and large language models with object identifiers. In *Proceedings of the Advances in Neural Information Processing Systems*.

Huang, J.; Yong, S.; Ma, X.; Linghu, X.; Li, P.; Wang, Y.; Li, Q.; Zhu, S.-C.; Jia, B.; and Huang, S. 2024b. An Embodied Generalist Agent in 3D World. In *Proceedings of the International Conference on Machine Learning*.

Huang, K.-C.; Li, X.; Qi, L.; Yan, S.; and Yang, M.-H. 2024c. Reason3d: Searching and reasoning 3d segmentation via large language model. *arXiv preprint arXiv:2405.17427*.

Jia, B.; Chen, Y.; Yu, H.; Wang, Y.; Niu, X.; Liu, T.; Li, Q.; and Huang, S. 2024. SceneVerse: Scaling 3D Vision-Language Learning for Grounded Scene Understanding. In *Proceedings of European Conference on Computer Vision*.

Jiang, X.; Lu, L.; Shao, L.; and Lu, S. 2024. Multimodal 3D Reasoning Segmentation with Complex Scenes. *arXiv preprint arXiv:2411.13927*.

Kang, W.; Huang, H.; Shang, Y.; Shah, M.; and Yan, Y. 2024. Robin3D: Improving 3D Large Language Model via Robust Instruction Tuning. *arXiv preprint arXiv:2410.00255*.

Kang, W.; Qu, M.; Kini, J.; Wei, Y.; Shah, M.; and Yan, Y. 2025. Intent3D: 3D Object Detection in RGB-D Scans Based on Human Intention. In *Proceedings of the International Conference on Learning Representations*.

Kolodiazhnyi, M.; Vorontsova, A.; Konushin, A.; and Rukhovich, D. 2024a. Oneformer3d: One transformer for unified point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20943–20953.

Kolodiazhnyi, M.; Vorontsova, A.; Skripkin, M.; Rukhovich, D.; and Konushin, A. 2024b. UniDet3D: Multi-dataset Indoor 3D Object Detection. *arXiv preprint arXiv:2409.04234*.

Liang, D.; Feng, T.; Zhou, X.; Zhang, Y.; Zou, Z.; and Bai, X. 2025a. Parameter-efficient fine-tuning in spectral domain for point cloud learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liang, D.; Hua, W.; Shi, C.; Zou, Z.; Ye, X.; and Bai, X. 2025b. Sood++: Leveraging unlabeled data to boost oriented object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liang, D.; Zhou, X.; Xu, W.; Zhu, X.; Zou, Z.; Ye, X.; Tan, X.; and Bai, X. 2024. Pointmamba: A simple state space model for point cloud analysis. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 37, 32653–32677.

Lin, B. Y.; Huang, C.; Liu, Q.; Gu, W.; Sommerer, S.; and Ren, X. 2023. On grounded planning for embodied tasks with

language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 13192–13200.

Mao, Y.; Zhang, Y.; Jiang, H.; Chang, A.; and Savva, M. 2022. MultiScan: Scalable RGBD scanning for 3D environments with articulated objects. *Proceedings of the Advances in Neural Information Processing Systems*, 35: 9058–9071.

Ramakrishnan, S. K.; Gokaslan, A.; Wijmans, E.; Maksymets, O.; Clegg, A.; Turner, J.; Undersander, E.; Galuba, W.; Westbury, A.; Chang, A. X.; et al. 2021. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *Proceedings of the Advances in Neural Information Processing Systems*.

Rozenberszki, D.; Litany, O.; and Dai, A. 2022. Language-grounded indoor 3d semantic segmentation in the wild. In *Proceedings of European Conference on Computer Vision*, 125–141.

Schult, J.; Engelmann, F.; Hermans, A.; Litany, O.; Tang, S.; and Leibe, B. 2023. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation*, 8216–8223. IEEE.

Takmaz, A.; Fedele, E.; Sumner, R. W.; Pollefeys, M.; Tombari, F.; and Engelmann, F. 2023. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Proceedings of the Advances in Neural Information Processing Systems*.

Wald, J.; Avetisyan, A.; Navab, N.; Tombari, F.; and Nießner, M. 2019. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, 7658–7667.

Wang, T.; Mao, X.; Zhu, C.; Xu, R.; Lyu, R.; Li, P.; Chen, X.; Zhang, W.; Chen, K.; Xue, T.; et al. 2024. EmbodiedScan: A Holistic Multi-Modal 3D Perception Suite Towards Embodied AI. *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Wang, Z.; Huang, H.; Zhao, Y.; Zhang, Z.; and Zhao, Z. 2023. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*.

Wu, Z.; Wang, Z.; Xu, X.; Lu, J.; and Yan, H. 2023. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*.

Xu, G.; Wang, X.; Zhang, Z.; Cheng, J.; Liao, C.; and Yang, X. 2025a. Igev++: Iterative multi-range geometry encoding volumes for stereo matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xu, G.; Wang, Y.; Cheng, J.; Tang, J.; and Yang, X. 2023. Accurate and efficient stereo matching via attention concatenation volume. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4): 2461–2474.

Xu, W.; Shi, C.; Tu, S.; Zhou, X.; Liang, D.; and Bai, X. 2025b. A unified framework for 3d scene understanding. *Advances in Neural Information Processing Systems*, 37: 59468–59490.

Zhang, Z.; Zhu, Z.; Li, P.; Liu, T.; Ma, X.; Chen, Y.; Jia, B.; Huang, S.; and Li, Q. 2024. Task-oriented sequential grounding in 3d scenes. *arXiv preprint arXiv:2408.04034*.

Zhou, X.; Liang, D.; Tu, S.; Chen, X.; Ding, Y.; Zhang, D.; Tan, F.; Zhao, H.; and Bai, X. 2025. Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation. In *Proceedings of IEEE/CVF International Conference on Computer Vision*.

Zhu, C.; Wang, T.; Zhang, W.; Chen, K.; and Liu, X. 2024a. Scanreason: Empowering 3d visual grounding with reasoning capabilities. In *Proceedings of European Conference on Computer Vision*, 151–168.

Zhu, C.; Wang, T.; Zhang, W.; Pang, J.; and Liu, X. 2025. LLaVA-3D: A Simple yet Effective Pathway to Empowering LMMs with 3D-awareness. In *Proceedings of IEEE/CVF International Conference on Computer Vision*.

Zhu, Z.; Ma, X.; Chen, Y.; Deng, Z.; Huang, S.; and Li, Q. 2023. 3D-VisTA: Pre-trained transformer for 3D vision and text alignment. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, 2911–2921.

Zhu, Z.; Zhang, Z.; Ma, X.; Niu, X.; Chen, Y.; Jia, B.; Deng, Z.; Huang, S.; and Li, Q. 2024b. Unifying 3D Vision-Language Understanding via Promptable Queries. In *Proceedings of European Conference on Computer Vision*.