

M3DBench: Let's Instruct Large Models with Multi-modal 3D Prompts

Mingsheng Li¹ Xin Chen^{2,*} Chi Zhang² Sijin Chen¹ Hongyuan Zhu³

Fukun Yin¹ Gang Yu² Tao Chen^{1,†}

¹Fudan University ²Tencent PCG

³Institute for Infocomm Research (I²R) & Centre for Frontier AI Research (CFAR), A^{*}STAR, Singapore

<https://github.com/OpenM3D/M3DBench>

* Project Lead † Corresponding Author

Scene Description

I: Can you describe this room in a detail manner?

R: The room appears to be a living-dining space, featuring a variety of furniture pieces. The dining area houses a square dining table with a light brown top, surrounded by four wooden dining chairs. One chair is positioned at the back of the table, closest to the back wall and to the left of the tan couch. Another chair is against the wall, facing the living room...

Dense Caption

I: Deliver a brief/detailed description of the .

R: A square wooden table./The target object is a square wooden dining table with a light brown top. There are four chairs surrounding the table, one on each side, and they are all tan in color. The placement of the table and chairs suggests a dining area...

Multi-Round Dialogue

Human: Looking around the room, I see a wooden dining chair at the table. It's on the back side, closest to the wall.

AI: Yes, and there's another wooden dining chair at the front of the table. They seem to be part of a set.

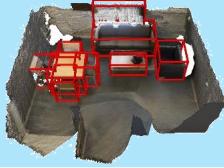
Human: In front of the tan couch, is there a wooden two-tiered coffee table.

AI: That's correct. And the tan couch is facing this coffee table. Behind the couch, there's a wide curtain in front of a window.

Object Detection

I: Identify all objects within the point cloud, specifying both their categories and locations.

R:



Embodied Question Answering

I: Standing around at the armchair situated within the `<box>(2)` and facing the coffee table, what is on my left?

R: The door is on your left.

Embodied Planning

I: If I am at the object close to (1.87,-0.15,0.12) and want to get some fresh air, where should I go?

R: You should move to the door to get some fresh air.

Visual Question Answering

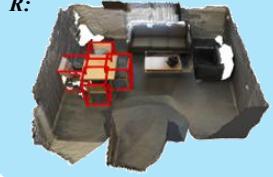
I: What is the size of the object close to the pointed spot `<point>(1)`?

R: It is a coffee table that has a length of 1.1 meters, a width of 0.5 meters, and a height of 0.25 meters.

Visual Grounding

I: Can you locate objects in the scene in the same category as ?

R:



Multi-Region Reasoning

I: Is the `sofa(4)` closer to the `table(3)` or the armchair?

R: The sofa is closer to the armchair.

Figure 1. Examples from M3DBench, which encompasses a variety of 3D-centric tasks. The dataset supports multi-modal instructions that interleave text with visual prompts and covers a variety of fundamental abilities in real-world 3D environments, such as visual perception, scene understanding, spatial reasoning, navigation, and planning.

Abstract

Recently, 3D understanding has become popular to facilitate autonomous agents to perform further decision-making. However, existing 3D datasets and methods are often limited to specific tasks. On the other hand, recent

progress in Large Language Models (LLMs) and Multi-modal Language Models (MLMs) have demonstrated exceptional general language and imagery tasking performance. Therefore, it is interesting to unlock MLM's potential to be 3D generalist for wider tasks. However, current MLMs' research has been less focused on 3D tasks due to

*a lack of large-scale 3D instruction-following datasets. In this work, we introduce a comprehensive 3D instruction-following dataset called M3DBench, which possesses the following characteristics: 1) It supports **general multi-modal instructions** interleaved with text, images, 3D objects, and other visual prompts. 2) It unifies **diverse 3D tasks at both region and scene levels**, covering a variety of **fundamental abilities** in real-world 3D environments. 3) It is a large-scale 3D instruction-following dataset with over 320k **instruction-response pairs**. Furthermore, we establish a new benchmark for assessing the performance of large models in understanding multi-modal 3D prompts. Extensive experiments demonstrate the effectiveness of our dataset and baseline, supporting general 3D-centric tasks, which can inspire future research.*

1. Introduction

The past year has witnessed remarkable success of Large Language Models(LLMs) families[20, 49, 52, 54] in addressing various natural language processing tasks through general instruction tuning [41]. Multi-modal Language Models (MLMs), such as Flamingo [2], BLIP-2 [33], LLaVA [35] have progressed various visual comprehension and reasoning tasks on 2D domain, including visual captioning [6, 50, 58], dialogue [14] and question-answering [23, 25]. To unlock the full potential of these MLMs, it is essential to curate a well-constructed instruction-following dataset [30, 35] that covers diverse vision language (VL) tasks, which empowers the models to handle these tasks without extensive modifications to the architecture. However, current research on MLMs has predominantly overlooked 3D visual and a comprehensive dataset for 3D instruction tuning is missing due to the daunting workload of collecting instructions in ambiguous and cluttered 3D environments.

Previous works have made efforts to construct datasets for specialized 3D task, such as object detection [21, 53], visual grounding [1, 12], dense captioning [1, 12], VQA [4, 62], and navigation [3]. Consequently, most of the models [4, 9, 13, 19, 38, 47] are specialist in only one or two of these tasks, potentially limiting their adaptability across various applications. Works such as LAMM [64], 3D-LLM [24], and Chat-3D [60] have made preliminary attempts in constructing 3D instruction-following datasets, achieving inspiring results. However, the range of visual tasks covered by these datasets is relatively *limited*, which constrains their effectiveness under diverse scenarios. These datasets primarily focus on language-only instructions, posing challenges in identifying specific object within a scene. For example, there might be multiple instances of “wooden chair” in a scene, yet the language prompt pertaining to a specific wooden chair might result in

ambiguity. Furthermore, the lack of a comprehensive evaluation *benchmark* poses challenges in accurately assessing the capability of large models on 3D-centric tasks. Current works, such as LAMM[64], primarily evaluate model’s performance on previous benchmarks that are not designed for assessing MLMs with open-form output [24].

In this paper, we introduce a comprehensive 3D instruction-following dataset called M3DBench, serving as the foundation for developing a versatile and practical general-purpose assistant in the real-world 3D environment. Our dataset comprises a variety of 3D vision-centric tasks at both object and scene levels and over 320K 3D instruction-response pairs, covering fundamental capabilities such as visual perception, scene understanding, spatial reasoning, and embodied planning, VL navigation, as depicted in Tab. 1. Furthermore, to tackle the challenge of ambiguity in language-only instructions, we interleave text instructions with other prompts that provide rich clues about instances in the scene, such as numerical coordinates, pointed region, image, 3D object (as shown in Fig. 1) in M3DBench, to enhance the capabilities in comprehending different granularity, diversity and interactivity concepts (such as “the pointed region” or “find the *⟨image of a whiteboard⟩* in the room”) in the multi-modal instructions.

To evaluate the effectiveness of M3DBench, we develop a simple yet effective baseline model capable of processing interleaved multi-modal instructions, consisting of three components: scene perceiver, multi-modal instruction encoder, and LLM decoder. Furthermore, we develop a comprehensive benchmark aimed at systematically assessing various capabilities of 3D MLMs across multiple dimensions with multi-modal instructions . The evaluation benchmark comprises approximately 1.5K instruction-response pairs, encompassing both region-level and scene-level tasks, such as object localization, scene description, multi-round dialogues, embodied planning, among others. Each instance comprises an instruction, a corresponding 3D scene, and a human-validated response. We will release M3DBench dataset, code, and evaluation strategies to accelerate future research on 3D MLMs.

To summarize, our contributions are listed as following:

- We introduce a large-scale 3D instruction-following dataset that unifies diverse region-level and scene-level 3D-centric tasks, focusing on scene perception, understanding, reasoning, and planning.
- We present a interleaved multi-modal instruction formula designed to enhance the granularity, diversity and interactivity of generated instructions.
- We establish a comprehensive benchmark for evaluating the capabilities of MLMs within 3D scenarios. Extensive experiments demonstrate the effectiveness of both the dataset and the baseline.

Dataset	Statistics		Instruction					Perception		Understanding and Reasoning						Planning	
	#Instruction-response pairs	#Average length of instruction / response	Text	Coord	Point	Box	Image	3D Object	Object Detection	Visual Grounding	Dense Caption	Visual Question Answering	Embody Question Answering	Multi-region Reasoning	Scene Description	Multi-round Dialogue	Embodied Planning
Nr3D [1]	-	-	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗
ScanRefer [12]	-	-	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗
ScanQA [4]	25K	8.77 / 2.42	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
SQA3D [37]	26K	10.49 / 1.10	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗
ScanScribe [73]	-	-	✓	✗	✗	✗	✗	✗	-	-	-	-	-	-	-	-	-
LAMM-3D [64]	10K	13.88 / 119.34	✓	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓	✓	✗
3DLM [24]	202K	43.80 / 8.11	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Chat-3D [60]	57K	9.11 / 48.75	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗
M3DBench	327K	24.79 / 18.48	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1. **Comparison between M3DBench and other 3D VL datasets as well as 3D instruction datasets.** M3DBench has the following characteristics: **1) A comprehensive** instruction-following dataset tailored for 3D scenes. **2) Supporting multi-modal instructions** that interleave text, coordinate, image, 3D object, and so on. **3) Encompassing diverse 3D visual-centric tasks** that span a variety of **fundamental abilities** in real-world 3D environments, such as visual perception, scene understanding, spatial reasoning, navigation, and planning.

2. Related Work

Multi-modal Datasets and 3D Benchmarks. The progress of LLMs [26, 32, 33, 48] has been greatly accelerated by the availability of large-scale image-text data, such as MS COCO Caption [17], Visual Genome [28], LAION-5B [51]. In order to improve models’ comprehension of human instructions in visual tasks, several visual instruction following datasets [22, 30, 35, 64] have been proposed. Additionally, while numerous studies in the field of 3D have presented benchmark datasets for visual grounding [1, 12], dense captioning [1, 12], and visual question answering [4, 37], these datasets are limited to specific tasks. In this paper, we propose a comprehensive dataset that supports interleaved multi-modal instructions and covers various 3D-centric tasks, including visual grounding, dense caption, embodied question answering, multi-region reasoning, scene description, multi-round dialogue, and so on. Refer to Tab. 1 for a detailed comparison between our dataset and other 3D VL datasets [1, 4, 12, 37] as well as exiting 3D visual instruction datasets [24, 60, 64]. Furthermore, rather than providing demonstrations only, we evaluate diverse tasks with quantitative results.

Multi-modal Foundation Models. With the triumph of LLMs [8, 20, 49, 54, 69], recent studies [2, 31, 33, 35] start to explore Vision Language Models (VLMs), extending the capabilities of LLMs in solving diverse visual-related tasks. Early attempts include Flamingo [2], which incorporates visual features through gated cross-attention dense blocks, and BLIP-2 [33], which uses a Q-former as a bridge to reconcile the modality gap between the frozen image encoder and LLMs. In order to enhance the VLMs’ comprehension of human instructions, several visual instruction tuning methods [31, 35] have been proposed. Addressing the adaptation of LLMs to 3D-related tasks, LAMM [64] uses a simple projection layer to connect the 3d encoder and LLM. 3D-LLM [24] utilizes point clouds and text instructions as input, leveraging 2D VLMs as backbones. However, prior works that attempt to integrate the 3D world into

MFM have exhibited limitations in handling interleaved multi-modal instructions and accomplishing various tasks. In this work, we propose to improve the abilities of MFM in addressing diverse 3D-centric tasks and handling interleaved multi-modal instructions with on a comprehensive 3D instruction-following dataset.

3D Vision-language Learning. Recently, there has been growing interest in 3D VL learning. While various 3D representations exist, including voxels, point clouds, and neural fields, previous works have primarily focused on point cloud-text data. Among those, 3D dense captioning [18, 68] aims to generate description of target object within a 3D scene, while 3D visual grounding [63, 65, 71] involves identifying object in a scene based on textual description. In 3D question answering [4, 62], models are required to answer questions based on the visual information. Although these works have achieved impressive results in connecting 3D vision and language, they heavily rely on task-specific model design. In contrast, we develop a unified baseline model capable of decoding multiple 3D-related tasks without the need for specific model designs. Furthermore, we establish a comprehensive benchmark to assess the model’s performance across various tasks.

3. Multi-modal Instruction Dataset

We introduce the strategy for constructing the multi-modal 3D instruction dataset (details in Sec. 3.1), along with the design formula for interleaved multi-modal instructions (details in Sec. 3.2). We then detail the tasks at both the region-level and scene-level covered by the dataset in Sec. 3.3, followed by a statistical and analytical examination of the dataset in Sec. 3.4.

3.1. Dataset Construction

To construct a comprehensive 3D multi-modal instruction-following dataset, we utilize existing datasets [1, 10–12, 21, 27, 29, 67] for 3D-only [47, 66] and 3D-language tasks [4, 18], and collect extensive instruction-response data by prompting LLMs. For 3D-

only tasks, like object detection and visual grounding, we designed instruction and response templates corresponding to specific tasks. Specifically, we collect instructions for 3D-only tasks by providing task descriptions and specifying the desired output format. Responses interleave object coordinates (follow the specified output format in the instructions) and text. For the 3D-language tasks, such as dialogue and question answering, we provided object attributes, textual descriptions, and manually written task construction instructions as well as few-shot in-context learning examples to the GPT-API [40, 52] to generate task-specific instruction data.

Although most responses generated by GPT-API [40, 52] are of high quality, some were irrelevant to the instruction. For instance, certain responses may refer to information derived from the provided textual descriptions. To improve the quality of the 3D instruction-following data, we employ pattern matching with specific keywords to filter out such responses.

3.2. Interleaved Multi-modal Instruction

We design four types of visual prompts, namely, point-level prompt (user click), box-level prompt (pointed region), image prompt, and 3D object prompt. For point-level prompt, we sample points near the specified region of the scene and randomly select one. The box-level prompts are derived from the ground truth bounding boxes in the 3D scene. Image prompts consist of corresponding image regions with 3D scenes, publicly available image data[29], and synthetic images[45]. Regarding 3D objects, we select instances from 3D scenes with ground truth per-point annotations, additionally collecting objects from [11]. Furthermore, we provide specific descriptions, such as “in the pointed region” in instructions to guide the large model to identify visual prompts within the scene. Finally, an interleaved multi-modal instruction I can be defined as an ordered sequence composed of text and visual prompts, represented as $I = [x^1, x^2, \dots, x^M]$, where each element x^i in $\{text, point, box, image, object_{3d}\}$. Additional details can be found in the supplementary materials.

3.3. Task Coverage

Our dataset introduces a unified *instruction-response* format to cover diverse 3D-centric tasks, encompassing essential capabilities ranging from visual perception and understanding to reasoning and planning (detailed in Tab. 1).

3.3.1 Visual Perception

Object Detection(OD) aims at identifying and locating all the objects of interest in a point cloud [39, 42]. Here, we transform the classic OD task into an instruction-following

format by providing task descriptions and specifying the desired output format. Following LAMM [64], we manually design a set of instruction-response templates with placeholders, and each instruction includes the expected output format. The instruction and response templates can be found in the supplementary.

Visual Grounding(VG) involves identifying the target object in the scene based on a natural language referring expression [61, 66]. In M3DBench, we expand the task format of VG. Specifically, our description information for querying extends beyond textual input and includes various visual prompts, such as coordinate, clicked point, image, 3D object, and so on. Moreover, our output is not limited to locating a single target object but can also involve finding objects belonging to the same category.

3.3.2 Scene Understanding and Reasoning

Dense Caption(DC) requires a model to generate natural language descriptions for each object [16, 18]. However, existing DC datasets like ScanRefer [12] and Nr3D [1] provide only short captions. In M3DBench, we reconstruct the DC datasets and introduce terms like *brief* or *detailed* in instruction to generate either concise title or detailed description for the object, which allows for better control over the granularity of the generated caption. The instruction templates can be found in the supplementary.

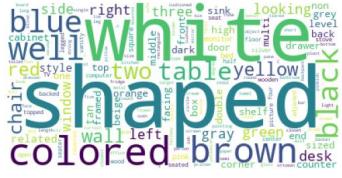
Visual Question Answering(VQA) is a task that requires the model to correctly answer a given question based on the information present in a visual scene [4, 44]. In this work, we curate a collection of free-form, open-ended question-answer pairs using publicly available 3D-language datasets. These VQA pairs cover various aspects at both the object level and scene level, including instance locations and attributes, object counts, room functions, and more.

Embodied Question Answering(EQA). Unlike traditional VQA tasks [4, 44] that primarily focus on answering questions related to global information, EQA requires the agent to first comprehend and analyze the surrounding environment to answer questions under that situation [37]. To collect instruction-following data for EQA, we start by randomly selecting a location within the scene and choosing to face a nearby object for reference direction, and then prompt GPT-4 to generate EQA pairs based on the given situation and text information.

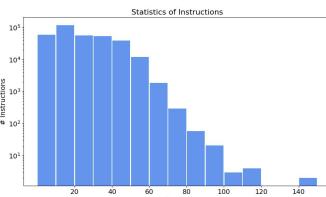
Multi-region Reasoning(MR). Datasets such as DC [1, 12] facilitate understanding and reasoning for individual objects. However, reasoning between distinct regions is often overlooked. For instance, inquiries about the spatial relationship between \langle region 1 \rangle and \langle region 2 \rangle . Here, we introduce MR, which is designed to enhance fine-grained comprehension of multiple regions of interest. Our methodol-



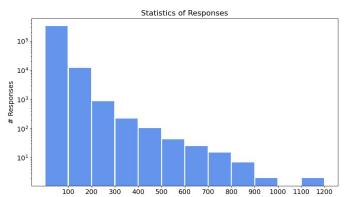
(a) Instruction distribution



(b) Word cloud of responses



(c) # of words in instructions



(d) # of words in responses

Figure 2. The statistics of the M3DBench. (a) The distribution of instructions based on the first word, where the inner circle of the graph represents the frequency of the first word’s occurrence, and the outer circle shows the frequency of verbs and nouns appearing in the instructions corresponding to that first word. (b) The word cloud of responses. (c) The distribution of instruction length. (d) The distribution of response length.

ogy involves feeding object location, descriptions [67], few-shot learning examples, and language instructions to GPT-4 to obtain corresponding responses.

Scene Description(SD). Unlike DC [16, 18], which generates a caption for each object, SD focuses on producing descriptions of the entire scene, extending the descriptive ability of MLMs from the region level to the scene level. To construct the instruction-following data for SD, we extract 3D bounding box annotations from ScanNet [21] and dense captions from the 3D VL datasets [1, 12] as data sources. By prompting the GPT-4, we can generate detailed descriptions for each scene.

Multi-round Dialogue(MD). To construct MDs, we make use of 3D VL datasets and follow a similar approach to that used in LLaVA [35]. During this process, we prompt GPT-4 to generate MDs in a self-questioning and self-answering format, taking advantage of coordinate information and language descriptions from [1, 12].

3.3.3 Planning and Navigation

Embodied Planning(EP). Unlike EQA, which primarily focuses on answering questions, EP requires agents to possess planning and decision-making capabilities. Specifically, the agent needs to perceive the environment, understand user’s intentions, and generate appropriate action instructions to achieve predefined goals [24].

Vision Language Navigation(NLV) require an agent to navigate and move in a real-world 3D environment based on human language instructions. We leverage annotations from existing 3D-language navigation tasks [27] and transform them into an instruction-following format. Instructions are expressed in natural language, while the corresponding response is a trajectory formed by points in space.

3.4. Dataset Statistics and Analysis

Tab. 1 presents the statistics of M3DBench. M3DBench contains over 320K pairs of instruction-following data. Among these pairs, more than 138K instructions include the interleaved multi-modal prompts we proposed.

To assess the diversity of generated instructions, we analyze the distribution of instructions based on the first word, as shown in Fig. 2 (a). Specifically, we extract the first word of each instruction and collected instructions starting with that word. Then we parse the instructions using the Natural Language Toolkit [7], performing processes like tokenization and part-of-speech tagging to extract nouns and verbs from instructions. The findings indicate that instructions in M3DBench are diverse, including various types such as “What” (query), “Can” (request), “Is” (confirmation), “I” (first-person), “Where” (location), and so on. Analyzing the word cloud of responses, as depicted in Fig. 2 (b), we observe answers pertaining to shape, color, count, action, object category, spatial relations, and so on. Furthermore, we demonstrated diversity in the lengths of instructions and responses, as illustrated in Fig. 2 (c) and Fig. 2 (d).

4. Multi-modal Instruction Tuning

We introduce a baseline model that connects scenes with interleaved multi-modal instructions and accomplishes diverse tasks using a unified decoder. As shown in Fig. 3, the framework consists of three parts: scene perceiver, multi-modal instruction encoder, and LLM. First, the 3D scene is processed by the scene perceiver, and the features are then projected into the same feature space as the language embedding using a trainable projection layer (Sec. 4.1). Simultaneously, prompts from different modalities within instructions are encoded using their corresponding prompt encoder (Sec. 4.2). Then the visual and instruction tokens are concatenated and fed into the LLM (Sec. 4.3). Next, we will provide a detailed description of each module.

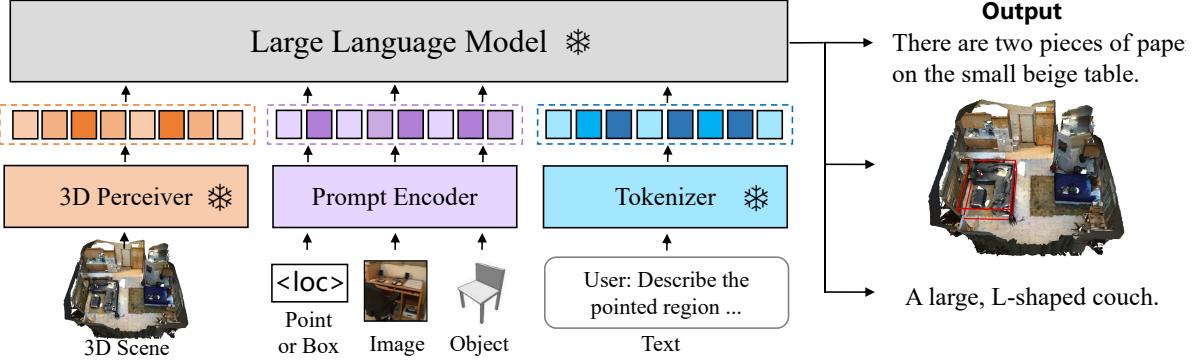


Figure 3. **Overview of our baseline model.** We utilize scene perceiver to extract scene tokens from 3D visual input. Multi-modal instructions are transformed into corresponding instruction tokens via their respective encoders. The scene tokens and multi-modal instruction tokens are then concatenated and fed into a frozen LLM, which generates the corresponding responses subsequently. During the training process, only the projectors are updated.

4.1. 3D Scene Perceiver

Given the point cloud of a scene, denoted as P , we employ a pre-trained 3D encoder to extract 3D feature:

$$f_s = \mathcal{E}^{3D}(P). \quad (1)$$

Similar to LLAVA [35], we also utilize a trainable visual feature projection matrix \mathcal{W}^{3D} to project the visual features into the language embedding space and obtain scene tokens:

$$X_s = \mathcal{W}^{3D} \cdot f_s. \quad (2)$$

The scene embeddings are represented as $X_s = \{x_s^n\}_{n=1}^N$, where $x_s^n \in \mathbb{R}^d$ and N represents the number of visual tokens. d represents the dimension of hidden states in LLM.

4.2. Multi-modal Instruction Encoder

There are a total of six types of prompt formats (Tab. 1) in the interleaved multi-modal instructions: text, numerical coordinate, user click (point), pointed region (box), image, and 3D object. We treat numerical **coordinate** as a specific **text** [15, 72] and use the tokenizer and word embedding from LLM to obtain corresponding tokens. For **user click** and **pointed region**, we utilize two learnable projection matrices to extract point-level and box-level tokens, respectively. In the case of **image** prompt, we employ the frozen CLIP [48] to extract image features, followed by a pre-trained projector from LLava [35] to compute image tokens. For **3D object** input, we downsample them to 1024 points and normalize their coordinates into a unit sphere [73]. Then a pre-trained encoder is used to extract object's features, and a Feed Forward Network (FFN) is inserted between the encoder and LLM to adjust these tokens.

4.3. LLM Decoder

We utilize the pre-trained LLM [20, 55, 69] as a unified decoder for various vision-centric tasks. To accomplish

this, we employ a 3D scene perceiver (Sec. 4.1) to encode the input scene P into discrete scene tokens $X_s = \{x_s^n\}_{n=1}^N$. These tokens are then concatenated with the multi-modal instruction tokens $X_i = \{x_i^n\}_{n=1}^M$. LLM takes both the scene tokens and the multi-modal instruction tokens as input and predicts the probability distribution of the output token $X_o = \{x_o^n\}_{n=1}^L$ in an auto-regressive manner:

$$P_\theta(X_o | X_s, X_i) = \prod_n P_\theta(x_o^l | x_o^{<l}; X_s, X_i). \quad (3)$$

Furthermore, for tasks that rely on coordinates for assessment, such as visual grounding, we decouple them from the output of LLMs (detailed in the supplements). This simple approach enables us to develop a unified framework for a wide range of 3D-only tasks without the need for modifications to the existing LLMs [8, 54, 69].

4.4. Training Strategy

The training objective is to maximize the likelihood of generating this target response sequence $X_o = \{x_o^n\}_{n=1}^L$, given the visual input X_s and multi-modal instruction X_i :

$$\mathcal{L}_\theta = - \sum_{n=1}^L \log P_\theta(x_o^l | x_o^{<l}; X_s, X_i). \quad (4)$$

Here, θ represents the trainable parameters. Note that during training, we freeze the 3D encoder, image encoder, as well as language decoder, and only train all the projection layers to enable rapid iterations. Exploring alternative architecture or refining the training strategy could potentially yield further improvements. We leave this as a direction for future work.

Task	3D Vision Encoder	LLM Decoder	BLEU-1↑	BLEU-2↑	BLEU-3↑	BLEU-4↑	ROUGE↑	METEOR↑	CIDEr↑
Dense Caption	Pointnet++ [46]	OPT-6.7B [69]	3.56	1.43	0.52	0.21	14.18	9.79	17.01
		LLaMA-2-7B [55]	10.60	4.53	1.70	0.73	18.70	13.40	22.05
		Vicuna-7B-v1.5 [20]	2.97	1.04	0.32	0.00	11.78	9.04	13.88
	Transformer [56]	OPT-6.7B [69]	10.72	4.44	1.45	0.0	14.58	10.35	23.76
		LLaMA-2-7B [55]	10.07	3.71	1.38	0.0	17.32	12.03	20.72
		Vicuna-7B-v1.5 [20]	11.96	4.38	1.28	0.0	14.13	9.46	23.72
Visual Question Answering	Pointnet++ [46]	OPT-6.7B [69]	57.45	49.48	43.57	38.78	58.34	30.30	336.96
		LLaMA-2-7B [55]	61.01	53.35	47.63	43.00	61.59	32.05	379.05
		Vicuna-7B-v1.5 [20]	46.30	38.13	32.20	27.56	51.55	27.03	239.98
	Transformer [56]	OPT-6.7B [69]	57.26	50.35	44.97	40.50	59.55	30.64	365.60
		LLaMA-2-7B [55]	60.23	52.41	47.02	42.61	59.24	30.96	356.42
		Vicuna-7B-v1.5 [20]	17.77	14.22	11.86	10.07	22.12	11.32	95.98
Embodied Question Answering	Pointnet++ [46]	OPT-6.7B [69]	47.55	37.69	30.91	24.44	49.17	26.04	212.12
		LLaMA-2-7B [55]	45.85	35.92	29.32	22.79	48.34	24.89	194.09
		Vicuna-7B-v1.5 [20]	21.09	15.61	12.28	9.41	44.06	20.55	169.72
	Transformer [56]	OPT-6.7B [69]	47.37	37.86	31.33	24.76	50.83	25.95	218.01
		LLaMA-2-7B [55]	44.20	33.86	27.49	21.58	45.83	22.74	179.33
		Vicuna-7B-v1.5 [20]	38.24	29.71	24.63	19.64	40.62	21.00	155.12
Multi-region Reasoning	Pointnet++ [46]	OPT-6.7B [69]	57.53	50.03	43.57	38.27	61.23	33.74	363.87
		LLaMA-2-7B [55]	56.24	49.32	43.42	38.46	61.48	34.01	378.17
		Vicuna-7B-v1.5 [20]	47.98	39.18	32.28	26.82	49.87	27.59	212.93
	Transformer [56]	OPT-6.7B [69]	36.92	30.78	25.91	21.60	44.51	24.27	240.89
		LLaMA-2-7B [55]	55.00	47.88	42.31	37.60	59.90	32.56	351.96
		Vicuna-7B-v1.5 [20]	21.96	17.21	13.87	11.06	27.07	12.68	95.40
Embodied Planning	Pointnet++ [46]	OPT-6.7B [69]	49.22	41.11	35.04	29.71	50.90	26.65	133.94
		LLaMA-2-7B [55]	57.66	50.18	44.86	40.76	56.46	29.77	253.09
		Vicuna-7B-v1.5 [20]	21.68	15.27	10.87	8.10	32.73	19.78	83.39
	Transformer [56]	OPT-6.7B [69]	59.47	53.24	48.08	43.46	61.14	33.34	213.15
		LLaMA-2-7B [55]	52.98	45.17	39.05	34.27	49.95	28.70	171.51
		Vicuna-7B-v1.5 [20]	37.50	30.71	25.33	20.54	38.55	21.50	114.91

Table 2. **Benchmark for multiple tasks: Dense Caption (DC), Visual Question Answering (VQA), Embodied Question Answering (EQA), Multi-region Reasoning (MR), Embodied Planning (EP).** We present the performance of baseline methods on our evaluation dataset. ↑ means the higher, the better.

5. Experiments

We first introduce the baseline model, metrics, and implementation details in Sec. 5.1. Additionally, we provide a benchmark on 3D scene understanding, reasoning and description in Sec. 5.2. Finally, we showcase some visualization results in Sec. 5.3. More details, quantitative results, and qualitative examples are provided in supplements.

5.1. Baseline, Metrics, and Implementations

Baseline. Since no prior method that works out of the box with our interleaved multi-modal instruction setup, we develop several variant models as baseline based on LLM [20, 55, 69] to accommodate M3DBench. Specifically, we incorporate two different types of 3D encoders, based on PointNet++ [46] and Transformer [56], into our baseline model. Furthermore, we consider three versions of LLMs as our language decoder: OPT-6.7B [69], LLaMA-2-7B [55], and Vicuna-7B-v1.5 [20]. After end-to-end instruction tuning, we evaluate baseline models on the evaluation dataset to assess their effectiveness.

Evaluation Metrics. The evaluation metrics include both traditional and GPT metrics. Traditional metrics, such as CiDER [57], METEOR [5], Acc@0.25IoU [12], and so on,

are used to measure the model’s performance on specific tasks. For a more comprehensive evaluation of the models’ instruction-following abilities, we employ GPT-4 to assess the quality of the different variants’ responses. Specifically, we provide GPT-4 with the answers generated by different variant models, the reference answers, and evaluation requirements. GPT-4 evaluates these responses and assigns a score ranging from 0 to 100. A higher average score indicates better performance of the model. Furthermore, we request GPT-4 to provide justifications for the scoring results, which helps us better judge the validity of the evaluation.

Implementations. Following previous works in 3D learning [16, 38], we downsample each 3D scene to 40,000 points as our scene input. For the PointNet++-based 3D encoder, we initialize it with the checkpoint obtained from Depth Contrast [70]. As for the Transformer-based encoder, we employ the checkpoint from Vote2Cap-DETR [16]. Additionally, we use the pre-trained encoder ViT-L/14 [48] as our image feature encoder. We train all the baseline models using the Adam optimizer [36] with a cosine annealing scheduler where the learning rate decays from 10^{-5} to 10^{-6} . Our batch size is set to 2 during training, utilizing 4 Nvidia A100 (40G) GPUs, which allows us to complete the training within 2 days.

5.2. Quantitative Evaluation

Understanding, Reasoning, and Planning. To establish a benchmark for scene understanding, reasoning, and planning, we comprehensively evaluated six variant models and reported the quantitative results on our evaluation dataset. Tab. 2 presents the performance of baselines across five tasks: Dense Captioning (DC), Visual Question Answering (VQA), Embodied Question Answering (EQA), Multi-region Reasoning (MR), and Embodied Planning (EP). We employed BLEU 1-4 [43], ROUGE-L [34], METEOR [5], and CiDER [57] as evaluation metrics.

Analyzing the results, one can see that when using the same language decoder, the Pointnet++ [46]-based models underperformed compared to the Transformer [41]-based models in the DC and EP tasks, while outperformed them in the MR task. However, upon switching the language decoder while keeping the 3D encoder constant, Vicuna-7B-v1.5 [20] exhibited lower overall performance compared to other LLMs across almost all tasks. The evaluation of our benchmark dataset suggests a diversity in the performance of MLMs, with each demonstrating unique strengths and weaknesses across diverse tasks. Moreover, the suboptimal performance of current baseline models across various tasks offers potential direction for further development of 3D MLMs. For instance, enhancing the performance of MLMs on benchmark tasks such as scene understanding, perception, and planning is crucial and we leave them for future work to explore.

3D Vision Encoder	LLM Decoder	GPT-4 Score
	OPT-6.7B [69]	9.87
Pointnet++ [46]	LLaMA-2-7B [55]	27.89
	Vicuna-7B-v1.5 [20]	32.37
	OPT-6.7B [69]	16.84
Transformer [56]	LLaMA-2-7B [55]	27.37
	Vicuna-7B-v1.5 [20]	29.08

Table 3. **Benchmark for detailed description.** In practice, we randomly select 39 scenes and provide detailed descriptions generated by GPT-4 for each scene. We then assess the relative scores achieved by different variants. Responses generated by variant models and GPT-4’ descriptions are fed back into GPT-4 for comparative analysis and scoring, along with the provision of relevant explanations for each variant’s answer. Experiments demonstrate that the model based on Vicuna-7B-V1.5 [20] demonstrated superior performance.

Detailed Description. As shown in Tab. 3, regarding detailed description capabilities, utilizing PointNet++ [46] as the scene encoder and Vicuna-7B-V1.5 [20] as the language decoder yielded the best performance, surpassing the sub-optimal variant by a margin of +3.29 points. Another ob-



Figure 4. **Qualitative Results.** We provide visualization results on various 3D-centric tasks in diverse 3D environments. Orange highlights the wrong answer.

servation is that all variants based on OPT [69] demonstrated relatively lower performance. Furthermore, we note that overall, all baseline models demonstrate inferior performance, suggesting that current baseline models possess limited capabilities in handling detailed descriptions. In supplements, we provide a qualitative presentation of the description results and the criteria for GPT-4 scoring.

5.3. Qualitative Results

We showcase some qualitative examples of our baseline model on the evaluation dataset in Fig. 4. One can see that our proposed method, trained on M3DBench, is capable of performing corresponding tasks under a variety of interleaved multi-modal instructions.

6. Conclusion

In this paper, we present M3DBench, a comprehensive multi-modal 3D instruction-following dataset, designed to facilitate the development of MLMs in the 3D domain. M3DBench encompasses a wide range of 3D vision-centric tasks and over 320K pairs of 3D instruction-following pairs, covering fundamental functionalities such as visual perception, scene understanding, spatial reasoning, planning, and navigation. Additionally, M3DBench introduces a novel multi-modal prompting scheme, interweaving language instruction with coordinate, image, pointed region, and other visual prompts. We also develop a simple yet efficient baseline model to validate the effectiveness of M3DBench, providing benchmarks for multiple tasks. Comprehensive quantitative and qualitative results demonstrate that models trained with M3DBench can successfully follow human instructions and complete 3D visual-related tasks. We hope that our proposed multi-modal 3D instruction dataset, baseline model, and benchmarks will inspire and fuel future explorations in the field of 3D MLMs.

References

- [1] 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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 422–440. Springer, 2020. [2](#), [3](#), [4](#), [5](#), [16](#)
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. [2](#), [3](#)
- [3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018. [2](#)
- [4] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139, 2022. [2](#), [3](#), [4](#)
- [5] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. [7](#), [8](#), [25](#)
- [6] Simone Bianco, Luigi Celona, Marco Donzella, and Paolo Napoletano. Improving image captioning descriptive-ness by ranking and llm-based fusion. *arXiv preprint arXiv:2306.11593*, 2023. [2](#)
- [7] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, 2006. [5](#)
- [8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. [3](#), [6](#), [16](#)
- [9] Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16464–16473, 2022. [2](#)
- [10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. [3](#), [16](#)
- [11] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. [4](#)
- [12] 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*, pages 202–221. Springer, 2020. [2](#), [3](#), [4](#), [5](#), [7](#), [16](#)
- [13] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X Chang. D 3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding. In *European Conference on Computer Vision*, pages 487–505. Springer, 2022. [2](#)
- [14] Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages. *arXiv preprint arXiv:2305.04160*, 2023. [2](#)
- [15] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multi-modal llm’s referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. [6](#)
- [16] Sijin Chen, Hongyuan Zhu, Xin Chen, Yinjie Lei, Gang Yu, and Tao Chen. End-to-end 3d dense captioning with vote2cap-detr. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11124–11133, 2023. [4](#), [5](#), [7](#), [25](#)
- [17] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. [3](#)
- [18] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3193–3203, 2021. [3](#), [4](#), [5](#)
- [19] Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. In

- Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18109–18119, 2023. 2
- [20] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna.lmsys.org (accessed 14 April 2023)*, 2023. 2, 3, 6, 7, 8, 24, 25
- [21] 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. 2, 3, 5, 16
- [22] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructclip: Towards general-purpose vision-language models with instruction tuning, 2023. 3
- [23] Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10867–10877, 2023. 2
- [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. *arXiv preprint arXiv:2307.12981*, 2023. 2, 3, 5
- [25] Wenbo Hu, Yifan Xu, Y Li, W Li, Z Chen, and Z Tu. Bliva: A simple multimodal llm for better handling of text-rich visual questions. *arXiv preprint arXiv:2308.09936*, 2023. 2
- [26] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. 3
- [27] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 104–120. Springer, 2020. 3, 5, 16
- [28] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 3
- [29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 3, 4
- [30] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023. 2, 3
- [31] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 3
- [32] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 3
- [33] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 2, 3
- [34] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 8, 25
- [35] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 2, 3, 5, 6, 25
- [36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7
- [37] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqad3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022. 3, 4
- [38] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 2, 7
- [39] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 4
- [40] R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2, 2023. 4, 16, 24, 26
- [41] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022. 2, 8
- [42] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7463–7472, 2021. 4
- [43] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 8, 25
- [44] Maria Parelli, Alexandros Delitzas, Nikolas Hars, Georgios Vlassis, Sotirios Anagnostidis, Gregor Bachmann, and Thomas Hofmann. Clip-guided vision-language pre-training for question answering in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5606–5611, 2023. 4
- [45] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and

- Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 4
- [46] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 7, 8, 24, 25
- [47] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 2, 3
- [48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 6, 7, 25
- [49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 2, 3
- [50] Noam Rotstein, David Bensaid, Shaked Brody, Roy Ganz, and Ron Kimmel. Fuscap: Leveraging large language models to fuse visual data into enriched image captions. *arXiv preprint arXiv:2305.17718*, 2023. 2
- [51] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 3
- [52] John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, et al. Chatgpt: Optimizing language models for dialogue. *OpenAI blog*, 2022. 2, 4
- [53] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 2
- [54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 3, 6
- [55] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaee, Nikolay Bashlykov, Soumya Batra, Prajwala Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 6, 7, 8, 24, 25, 26
- [56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 7, 8, 24, 25, 26
- [57] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 7, 8, 25
- [58] Teng Wang, Jinrui Zhang, Junjie Fei, Yixiao Ge, Hao Zheng, Yunlong Tang, Zhe Li, Mingqi Gao, Shanshan Zhao, Ying Shan, et al. Caption anything: Interactive image description with diverse multimodal controls. *arXiv preprint arXiv:2305.02677*, 2023. 2
- [59] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022. 16
- [60] 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. 2, 3
- [61] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19231–19242, 2023. 4
- [62] Xu Yan, Zhihao Yuan, Yuhao Du, Yinghong Liao, Yao Guo, Zhen Li, and Shuguang Cui. Clevr3d: Compositional language and elementary visual reasoning for question answering in 3d real-world scenes. *arXiv preprint arXiv:2112.11691*, 2021. 2, 3
- [63] Zhengyuan Yang, Songyang Zhang, Liwei Wang, and Jiebo Luo. Sat: 2d semantics assisted training for 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1856–1866, 2021. 3
- [64] Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Lu Sheng, Lei Bai, Xiaoshui Huang, Zhiyong Wang, et al. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *arXiv preprint arXiv:2306.06687*, 2023. 2, 3, 4, 17
- [65] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1791–1800, 2021. 3
- [66] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1791–1800, 2021. 3, 4
- [67] Zhihao Yuan, Xu Yan, Zhuo Li, Xuhao Li, Yao Guo, Shuguang Cui, and Zhen Li. Toward explainable and fine-grained 3d grounding through referring textual phrases. *arXiv preprint arXiv:2207.01821*, 2022. 3, 5, 16
- [68] Zhihao Yuan, Xu Yan, Yinghong Liao, Yao Guo, Guanbin Li, Shuguang Cui, and Zhen Li. X-trans2cap: Cross-modal knowledge transfer using transformer for 3d dense captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8563–8573, 2022. 3
- [69] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuhui Chen, Christopher Dewan, Mona Diab,

- Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 3, 6, 7, 8, 24, 25
- [70] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021. 7, 25
- [71] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2928–2937, 2021. 3
- [72] Liang Zhao, En Yu, Zheng Ge, Jinrong Yang, Haoran Wei, Hongyu Zhou, Jianjian Sun, Yuang Peng, Runpei Dong, Chunrui Han, et al. Chatspot: Bootstrapping multimodal llms via precise referring instruction tuning. *arXiv preprint arXiv:2307.09474*, 2023. 6
- [73] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2921, 2023. 3, 6

Supplementary Material

This supplementary material provides further details about M3DBench (Sec. A), quantitative experiments (Sec. B) on multi-round dialogue and 3D localization, additional experiments for held-out evaluation (Sec. C), implementation details (Sec. D) of baseline model and prompt for GPT-4 evaluation (Sec. E).

A. Dataset

In Sec. A.1, we provide more examples in M3Dbench for each task. Following that, we will introduce the dataset construction in Sec. A.2 and provide statistics for the evaluation dataset in Sec. A.3.

A.1. More Examples

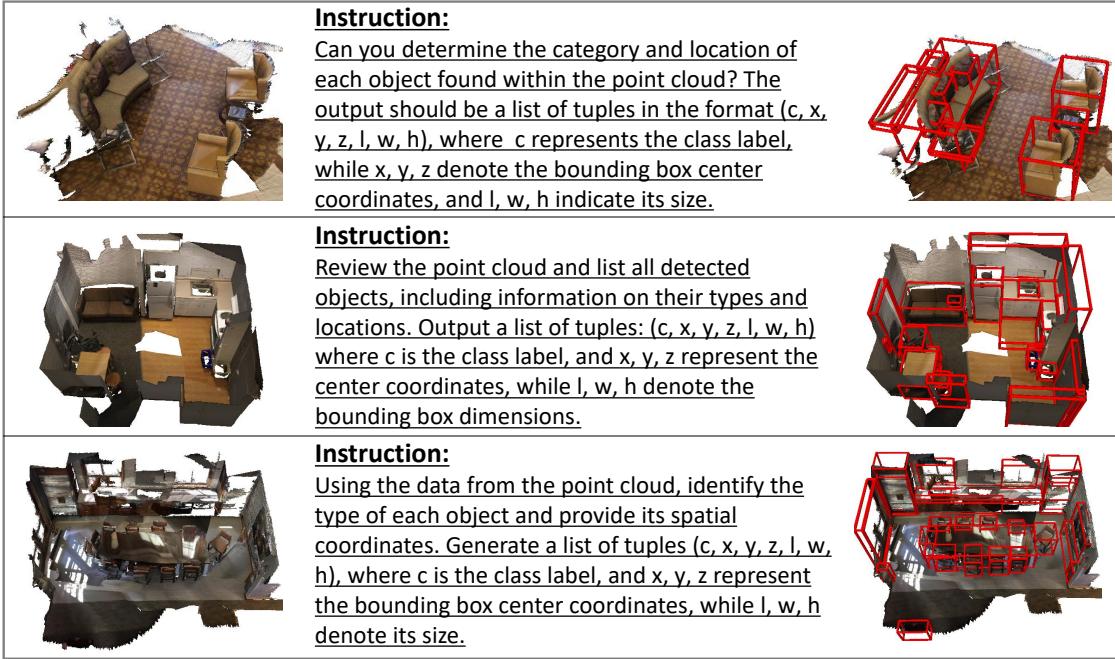


Figure 5. Examples of 3D object detection. The left column represents the 3D scene, the middle column displays the instructions, and the right column shows the annotations for the object detection task. We save annotations in textual format and for visualization purposes here, we extract the bounding boxes from the text.

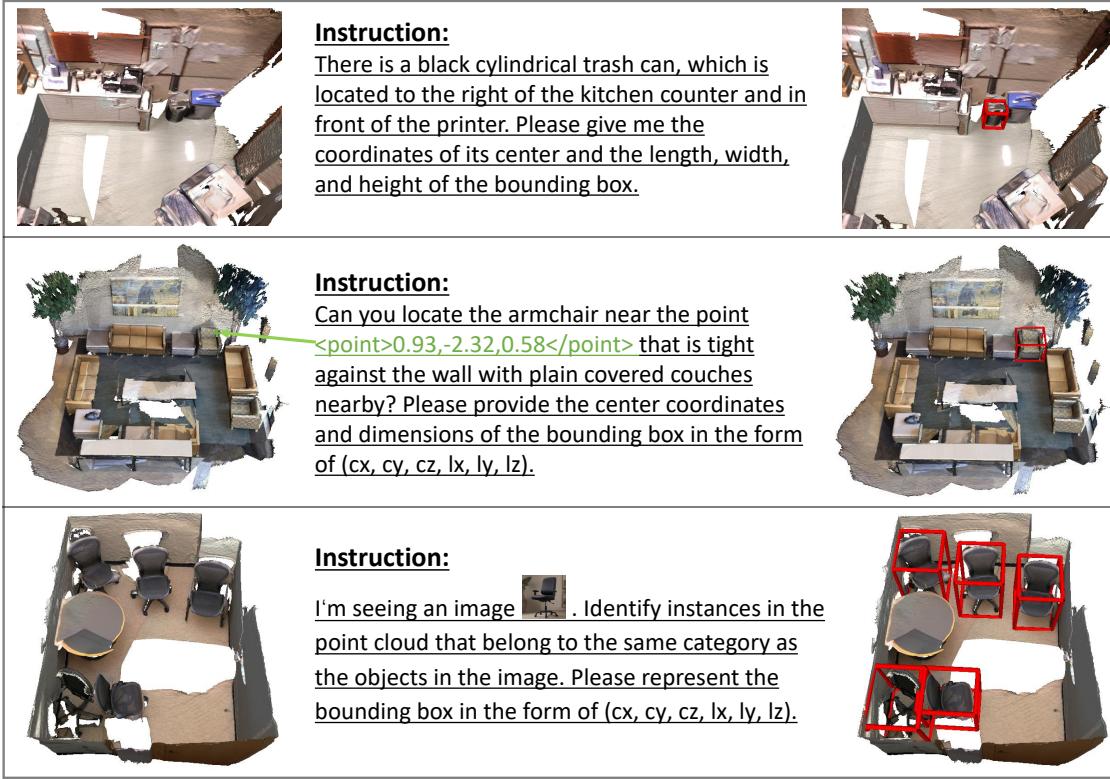


Figure 6. **Examples of 3D visual grounding.** The left column represents the 3D scene, the middle column displays the instructions, and the right column shows the annotations for the visual grounding. M3DBench includes interleaved multi-modal instructions, and the annotations extend beyond annotating a single target object, encompassing the identification of multiple objects.

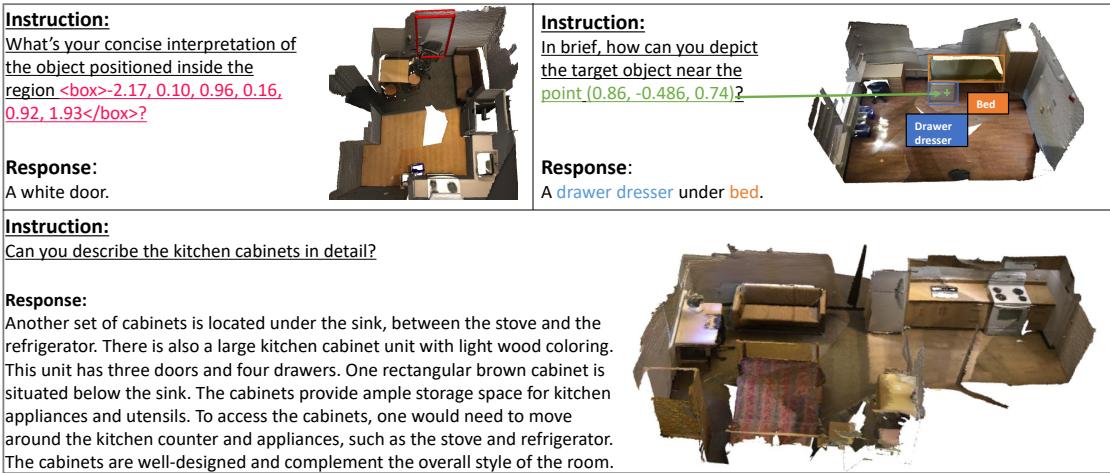


Figure 7. **Examples of 3D dense caption.** We design diverse multi-modal instructions for dense captions for M3DBench. Additionally, we introduce terms such as *brief* or *detailed* within instructions to generate either concise titles or detailed descriptions for objects.



Figure 8. Examples of 3D visual question answering. M3DBench comprises open-ended, free-form questions involving instance location, shape and size, object count, scene type, object and room functionality, and more. For instance, when asked about the functionality of a *rectangular silver object* in the upper-right scene, the answer begins by identifying the object and then describing its functionality. Furthermore, the two examples below illustrate instances where there might be multiple objects of *the same category* in the scene.

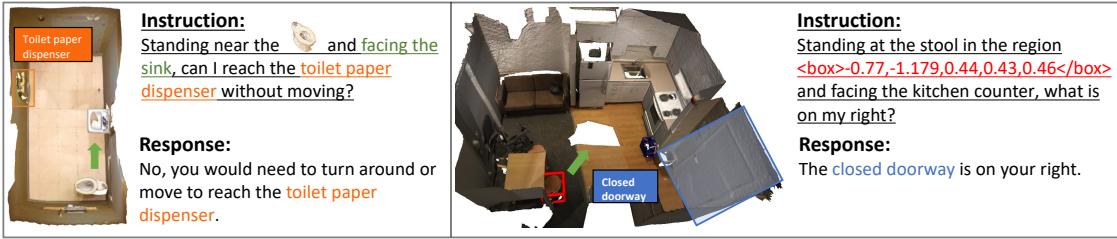


Figure 9. Examples of embodied question answering. Embodied question answering requires the agent to understand the surrounding environment in order to answer questions under that situation.

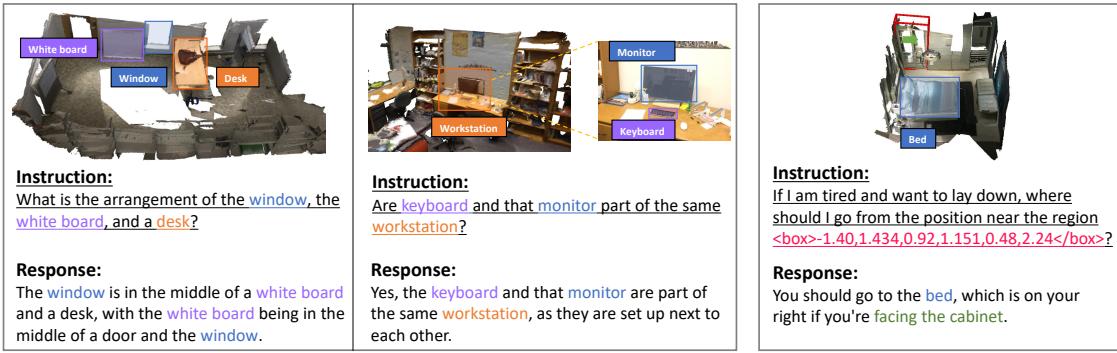


Figure 10. Examples of multi-region reasoning (left) and embodied planning (right). In multi-region reasoning tasks (left), at least two objects are involved, querying their relative relationships and sizes, which enables a detailed comprehension of the scene. On the other hand, embodied planning (right) requires an agent to perceive the environment, understand the user’s intentions, and then generate appropriate responses or actions to achieve predetermined goals.

<p>Instruction: <u>Can you describe the objects and their locations in the room?</u></p> <p>Response: The room is filled with various objects, including bags hanging on the wall, a black chair in front of a computer monitor, a white desk with a trash can underneath, a bookshelf in between doors, and a large round table with two white objects on top of it. The bags on the wall come in different colors and are located in various positions, such as the black bag with white writing hanging close to the door, the red-ish bag closest to the floor, and the black bag to the far right of the two sets of bags hanging on the wall to the right of the doorway. The bookshelf in between doors is the second one from the top and contains a group of books closest to the windows and another group closest to the couch. The trash can underneath the white desk is only visible when facing the wall with two white doors and a bookshelf in between. The large round table is located behind a red table next to the couch and has two white objects on top of it. The white desk has a computer monitor on it, with a lamp next to it and a trash can underneath. A black chair is in front of the computer monitor, and a tall lamp is located close to the door of the room near bookshelves.</p>	<p>Instruction: <u>Can you describe the scene in detail, including the room layout, object types, and object locations?</u></p> <p>Response: The scene appears to be an office space with cubicle desks and windows. The layout of the room is not clear, but there are multiple sets of doors and windows. The desks are brown and appear to be organized in rows. To locate a specific desk, one must stand at a certain point and face a certain direction. For example, standing at the table and looking at the cubicle desks, the middle option is the front left. The middle of the two cubicles in the front right is also an option. If standing in front of the wall with smaller windows, the rightmost desk directly next to the wall of windows should be selected. Another option is the desk that is second from the left and directly next to the window on the same wall. The desk closest to the set of doors that are not parallel to each other and nearest to the same door should also be selected. The largest highlighted box that is not against a wall and closest to the cubicles is another option. Additionally, the clear square located in the corner of the room closer to the doors with the exit sign above it should be selected.</p>
--	--

Figure 11. Examples of scene description. Scene description requires integrating the visual information of the entire scene to generate a detailed and comprehensive introduction. This description will encompass aspects such as objects within the scene, the type of room, its functionality, and so on.

	<p>Human: Looking around the room, I see a wooden dining chair at the table. It's on the back side, closest to the wall. AI: Yes, and there's another wooden dining chair at the front of the table. They seem to be part of a set. Human: Right. And in front of the tan couch, there's a wooden two-tiered coffee table. AI: That's correct. And the tan couch is facing this coffee table. Behind the couch, there's a wide curtain in front of a window. Human: I see. And the dining table, it's square with a light brown top, isn't it? AI: And it's off to the side of the living room. Around the table, there are 4 wooden chairs.</p>
	<p>Human: Let's start with the door. It's white and has a gold handle on the left side. AI: Yes, the door is located on the far right of the room, next to a blue chair. It's a simple yet functional design. Human: I see a black suitcase under the table. It's the second one from the left. AI: That's correct. The suitcase is conveniently placed under the table, perhaps for easy access or to save space. Human: There's also a small blue trash can with a fish design on it. It's right next to a white space heater. AI: Yes, the trash can's unique design adds a touch of personality to the room. Its location next to the heater suggests it's a frequently used area. Human: Moving on, there's a dark brown curtain with long panels that touch the floor. It's right next to the bed. AI: Yes, the curtain is touching the headboard of the bed on the right. It probably provides privacy for the bed area and adds an elegant touch to the room.</p>

Figure 12. Examples of multi-round dialogue. Multi-round dialogue necessitates the agent's ability to engage in natural and coherent communication with humans. This capability involves not only understanding and generating language but also ensuring accuracy and coherence in context.

A.2. Dataset Construction

In this work, we introduce a comprehensive 3D instruction tuning dataset, M3DBench, which serves as the foundation for developing versatile and practical general-purpose assistants in the real-world 3D environment. M3DBench comprises 3D data from publicly available datasets [1, 10–12, 21, 27, 67], along with interleaved multi-modal instructions and responses generated using self-instruct methods[59] and GPTs [8, 40]. From Tabs. 4 to 10, we provide detailed description of the

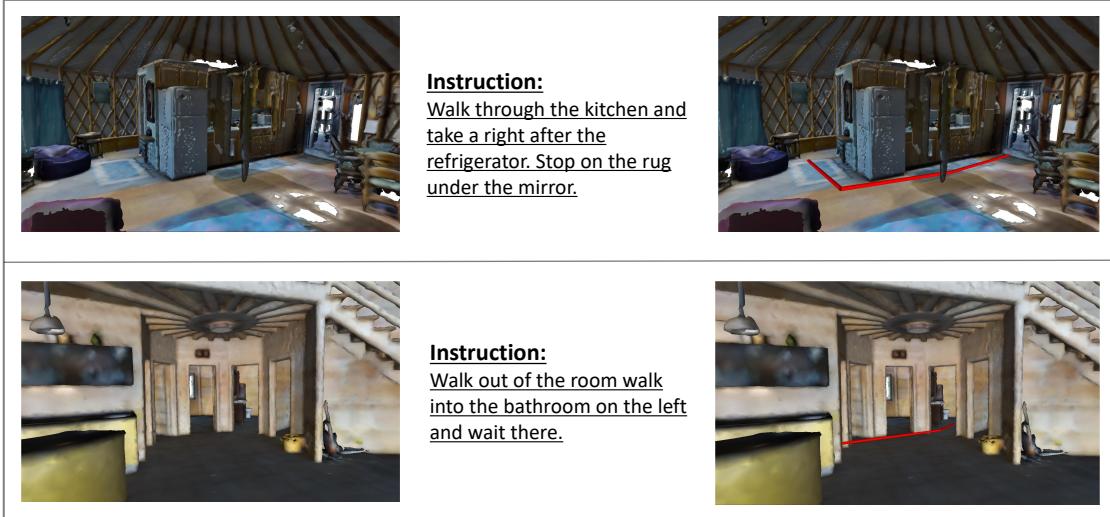


Figure 13. **Examples of 3D vision language navigation.** The 3D scene is depicted in the left column, instructions are presented in the middle column, and annotations for the vision language navigation task are shown in the right column. Annotations are stored in textual format, and for visual representation here, we extract the pathway from the text.

prompts designed for various 3D tasks, each comprising system messages and manually crafted context examples. For tasks such as object detection, we manually design instruction and response templates, then replace the template’s keywords with annotations to construct instruction-response data [64], as illustrated in Tab. 11 and Tab. 12. Furthermore, we have developed an interleaved multi-modal instruction formula by substituting corresponding templates for the $\langle target \rangle$ in the instructions, as shown in Tab. 13.

A.3. Evaluation Dataset

To quantitatively evaluate the effectiveness of instruction-tuned MLMs, we constructed an evaluation dataset to assess the models’ performance across various dimensions such as visual perception, scene understanding, spatial reasoning, and embodied planning. Our evaluation dataset consists of over 1.5K data samples, distributed as shown in Fig. 14.

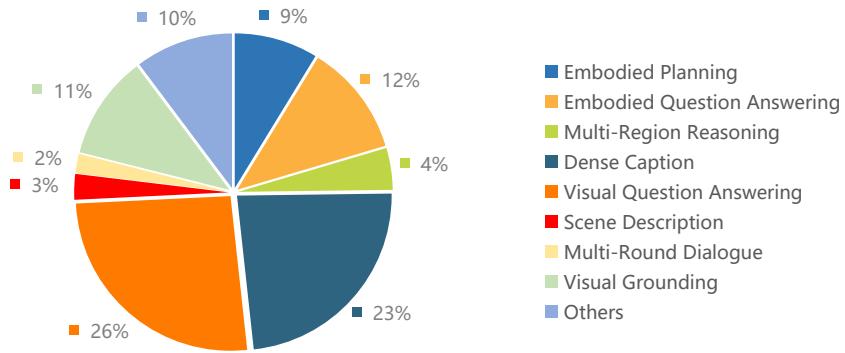


Figure 14. **The evaluation dataset covers a range of fundamental abilities within real-world 3D environments**, such as visual perception, scene comprehension, spatial reasoning, and embodied planning.

```

messages = [ {"role":"system", "content":f"""You are an AI visual assistant, and you are seeing an object in a 3D scene. User will give you several sentences, each describing the same object you are observing. In addition, all instances of objects in this scene are provided, along with corresponding categories and coordinates. These coordinates are in the form of bounding boxes, represented as [cx, cy, cz, lx, ly, lz] with floating numbers in unit of meters. These values correspond to the x, y, z coordinates of bounding box center and length of bounding box along x, y, z axis."}]

```

Summary and describe the target object in a detail manner, including details like the object placements, object attributes, object functions, relative position with the surrounding objects, and so on. Here are the requirements: 1) Describe using the tone of seeing the target object and surroundings. Don't generate descriptions that cannot be reasoned based on the given information confidently. 2) Descriptions should be concise, effective, diverse and logical. 3) Do not mention any specific spatial coordinate values and do not mention the source of information. The description should be more than 100 words and less than 150 words.""}]

```

]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']})
    messages.append({"role":"user", "content": '\n'.join(query)})

```

```

messages = [ {"role":"system", "content":f"""You are an AI visual assistant, and you are seeing an object in a 3D scene. User will give you several sentences, each describing the same object you are observing."}]

```

Summary the target object in a brief manner, only containing the object attribute. Here are the requirements: 1) Describe using the tone of seeing the target object. Don't generate descriptions that cannot be reasoned based on the given information confidently. 2) Descriptions should be concise, effective, and logical. 3) Do not mention any specific spatial coordinate values and do not mention the source of information. The description should be less than 5 words.""}]

```

]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']})
    messages.append({"role":"user", "content": '\n'.join(query)})

```

Table 4. System message used to generate detailed (top) and brief (bottom) dense caption data in M3DBench.

```

messages = [ {"role":"system", "content":f"""\nYou are an AI visual assistant, and you are seeing an object in a\n3D scene. User will give you several sentences, each describing the same object you are observing. In\naddition, all instances of objects in this scene are provided, along with corresponding categories and\ncategories. These coordinates are in the form of bounding boxes, represented as [cx, cy, cz, lx, ly, lz] with\nfloating numbers in unit of meters. These values correspond to the x, y, z coordinates of bounding box center\nand length of bounding box along x, y, z axis.\n"""
]

```

Generate some questions and give corresponding answer of the target object, including details like the object placements, object attributes, object functions, relative position with the surrounding objects, and so on. Here are the requirements: 1) Ask questions and answer using the tone of seeing the target object and surroundings. Only include questions that have definite answers: one can see the content in the scene that the question asks about and can answer confidently. Do not ask any question that cannot be answered confidently. Do not ask about uncertain details. 2) Replace the specific target object's name in the question with the placeholder '<target>'. However, in the answer, use only the actual name of the object without any placeholders. 3) Ask diverse questions (e.g., 'What...', 'How...', 'Which...', 'Where...', 'If...', 'Is...', 'Are...', etc.) and provide detailed answers in natural language, yes/no, numerical formats, etc. 4) Ensure questions and answers are concise, logical and effective. 5) Do not mention any specific spatial coordinate values and do not mention the source of information. Keep each question or answer under 50 words.""""}

```

]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']})
    messages.append({"role":"user", "content": '\n'.join(query)})

```

Table 5. System message used to generate instruction-response pairs for visual question answering in M3DBench.

```

messages = [ {"role":"system", "content":f"""\nYou are an AI visual assistant, and you are seeing an object in a\n3D scene. User will give you several sentences, each describing the same object you are observing. In addition,\nall instances of objects in this scene are provided, along with corresponding categories and coordinates.\nThese coordinates are in the form of bounding boxes, represented as [cx, cy, cz, lx, ly, lz] with floating\nnumbers in unit of meters. These values correspond to the x, y, z coordinates of bounding box center and\nlength of bounding box along x, y, z axis.\n"""
]

```

You need to generate embodied questions (e.g. Standing in front of the <target> and facing the towels. Can I see myself in the mirror?) and give a corresponding answer. Assuming you are positioned at the target object and facing a nearby object, please begin each question by describing the situation (position, orientation, etc.). Then provide the corresponding answer for the question. Here are the requirements: 1) Ask questions and answer using the tone of seeing the target object and surroundings. Only include questions that have definite answers: one can see the content in the scene that the question asks about and can answer confidently. Do not ask any question that cannot be answered confidently. Do not ask about uncertain details. 2) Replace the specific target object's name in the question with the placeholder '<target>'. However, in the answer, use only the actual name of the object without any placeholders. 3) Ask diverse questions (e.g., 'What...', 'How...', 'Which...', 'Where...', 'If...', 'Is...', 'Are...', etc.) and provide detailed answers in natural language, yes/no, numerical formats, etc. 4) Ensure questions and answers are concise, logical and effective. 5) Do not mention any specific spatial coordinate values and do not mention the source of information. Keep each question or answer under 50 words.""""}

```

]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']})
    messages.append({"role":"user", "content": '\n'.join(query)})

```

Table 6. System message used to generate instruction-response pairs for embodied question answering in M3DBench.

```

messages = [ {"role":"system", "content": f"""You are an AI visual assistant, and you are seeing an object in a 3D scene. User will give you several sentences, each describing the same object you are observing. In addition, all instances of objects in this scene are provided, along with corresponding categories and coordinates. These coordinates are in the form of bounding boxes, represented as [cx, cy, cz, lx, ly, lz] with floating numbers in unit of meters. These values correspond to the x, y, z coordinates of bounding box center and length of bounding box along x, y, z axis."}]

```

Generate some questions and give corresponding answer based on the relationship between these objects.

Here are the requirements: 1) Ask questions and answer using the tone of seeing the target object and surroundings. Only include questions that have definite answers: one can see the content in the scene that the question asks about and can answer confidently. Do not ask any question that cannot be answered confidently. Do not ask about uncertain details. 2) Replace the names of related objects in the question with placeholders like '<region 1>', '<region 2>', '<region 3>', etc. However, in the answer, use only the actual names of the objects without any placeholders. 3) Ask diverse questions (e.g., 'What...', 'How...', 'Which...', 'Where...', 'If...', 'Is...', 'Are...', etc.) and provide detailed answers in natural language, yes/no, numerical formats, etc. 4) Ensure questions and answers are concise, logical and effective. 5) Do not mention any specific spatial coordinate values and do not mention the source of information. Keep each question or answer under 50 words.""""}]

```

]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']})
    messages.append({"role":"user", "content": '\n'.join(query)})

```

Table 7. System message used to generate instruction-response pairs for multi-region reasoning in M3DBench.

```

messages = [ {"role":"system", "content": f"""You are an AI visual assistant that can analyze a 3D scene. User will give you several sentences, each describing the same scene you are observing. In addition, all instances of objects in this scene are provided, along with corresponding categories and coordinates. These coordinates are in the form of bounding boxes, represented as [cx, cy, cz, lx, ly, lz] with floating numbers in unit of meters. These values correspond to the x, y, z coordinates of bounding box center and length of bounding box along x, y, z axis."}]

```

Summary and describe the scene in a detail manner, including details like the scenario, scene types, room functions, object types, object counts, object locations, object attributes, relative relationships between the objects, and so on. Here are the requirements: 1) You should design a question before describing the scene. The question should be 1 to 2 sentences long. The type of question should be diverse. Either an imperative sentence or a question is permitted. For example, describe the scene in detail. 2) Describe using the tone of seeing the whole scene. Don't generate descriptions that cannot be reasoned based on the given information confidently. 3) Descriptions should be concise, effective, diverse and logical. 4) Do not mention any specific spatial coordinate values and do not mention the source of information. The description should be more than 200 words and less than 250 words.""""}]

```

]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']})
    messages.append({"role":"user", "content": '\n'.join(query)})

```

Table 8. System message used to generate instruction-response pairs for scene description in M3DBench.

```

messages = [ {"role":"system", "content": f"""You are an AI visual assistant that can analyze a 3D scene. User will give you several sentences, each describing the same scene you are observing. In addition, all instances of objects in this 3D are provided, along with corresponding categories and coordinates. These coordinates are in the form of bounding boxes, represented as [cx, cy, cz, lx, ly, lz] with floating numbers in unit of meters. These values correspond to the x, y, z coordinates of bounding box center and length of bounding box along x, y, z axis.

Design a conversation between a human and you discussing various aspects related to the scene. Include topics such as the given scenario, room functionality, type of scene, object categories, object counts, their respective locations, attributes, relationships between objects, and so on. Here are the requirements: 1) Both the human and you should discuss the scene in the tone of seeing the whole scene. Use a variety of sentence structures in the conversation. Avoid discussing details that cannot be confidently answered or are uncertain. 2) Initiate the conversation by choosing a specific topic. Ensure the conversation flows naturally and covers a wide range of details while maintaining coherence. 3) Do not mention any specific spatial coordinate values and do not mention the source of information. Each conversation should take at least 5 rounds."""}
]

for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']})
    messages.append({"role":"user", "content": '\n'.join(query)})

```

Table 9. System message used to generate instruction-response pairs for multi-round dialogue in M3DBench.

```

messages = [ {"role":"system", "content": f"""You are an AI visual assistant, and you are seeing an object in a 3D scene. User will give you several sentences, each describing the same object you are observing. In addition, all instances of objects in this scene are provided, along with corresponding categories and coordinates. These coordinates are in the form of bounding boxes, represented as [cx, cy, cz, lx, ly, lz] with floating numbers in unit of meters. These values correspond to the x, y, z coordinates of bounding box center and length of bounding box along x, y, z axis.

Generate embodied questions, including planning(e.g. I feel tired/I want to study and where should I go next?), navigation (e.g. how to go from the <target> to the position of bed?). Assuming you are positioned at the target object and facing a nearby object, please begin each question by describing the situation (position, orientation, etc.). Then provide the corresponding answer for the question. Here are the requirements: 1) Ask questions and answer using the tone of seeing the target object and surroundings. Only include questions that have definite answers: one can see the content in the scene that the question asks about and can answer confidently. Do not ask any question that cannot be answered confidently. Do not ask about uncertain details. 2) Replace the specific target object's name in the question with the placeholder '<target>'. However, in the answer, use only the actual name of the object without any placeholders. 3) Ask diverse questions (e.g., 'What...', 'How...', 'Which...', 'Where...', 'If...', 'Is...', 'Are...', etc.) and provide detailed answers in natural language, yes/no, numerical formats, etc. 4) Ensure questions and answers are concise, logical and effective. 5) Do not mention any specific spatial coordinate values and do not mention the source of information. Keep each question or answer under 50 words."""}
]

for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']})
    messages.append({"role":"user", "content": '\n'.join(query)})

```

Table 10. System message used to generate instruction-response pairs for embodied planning in M3DBench.

- Can you determine the category and location of each object found within the point cloud? The output should be a list of tuples in the format (c, x, y, z, l, w, h) , where c represents the class label, while x, y, z denote the bounding box center coordinates, and l, w, h indicate its size.
- Identify the object types within the point cloud and deliver tuples (c, x, y, z, l, w, h) based on spatial data. The format of the result should comprise tuples (c, x, y, z, l, w, h) , where c denotes the class label, and x, y, z represent the center coordinates of the bounding box, while l, w, h indicate its dimensions.
- Review the point cloud and list all detected objects, including information on their types and locations. Output a list of tuples: (c, x, y, z, l, w, h) where c is the class label, and x, y, z represent the center coordinates, while l, w, h denote the.
- Can you classify and locate objects within the point cloud? Provide (c, x, y, z, l, w, h) tuples for each. Generate a result in the format of tuples (c, x, y, z, l, w, h) , where c signifies the class label, and x, y, z denote the center coordinates of the bounding box, while l, w, h represent its size.
- Using the data from the point cloud, identify the type of each object and provide its spatial coordinates. Generate a list of tuples (c, x, y, z, l, w, h) , where c is the class label, and x, y, z represent the bounding box center coordinates, while l, w, h denote its size.
- From the point cloud data, extract object types and spatial coordinates as (c, x, y, z, l, w, h) tuples. The result should be structured as tuples (c, x, y, z, l, w, h) , where c represents the class label, and x, y, z indicate the bounding box's center coordinates, while l, w, h specify its dimensions.

- Positioned at the $\langle bbox \rangle$ location within the point cloud, an object within the $\langle class \rangle$ category can be observed.
- The point cloud includes an object at the $\langle bbox \rangle$ position, which can be classified under the category of $\langle class \rangle$.
- At the $\langle bbox \rangle$ position in the point cloud, there is an item categorized as $\langle class \rangle$.
- The $\langle bbox \rangle$ position of the point cloud allows for the identification of an object that belongs to the $\langle class \rangle$ category.
- Within the point cloud, an object classified as $\langle class \rangle$ is situated at the $\langle bbox \rangle$ position.
- An object that can be classified as $\langle class \rangle$ is located at the $\langle bbox \rangle$ position within the point cloud.
- The $\langle bbox \rangle$ of the point cloud reveals the presence of an object categorized as $\langle class \rangle$.
- At the $\langle bbox \rangle$ position within the point cloud, there exists an object that falls under the $\langle class \rangle$ category.
- The point cloud contains an object at the $\langle bbox \rangle$ position, which can be identified as $\langle class \rangle$.

Table 11. Some examples of question (top) and answer (bottom) templates for 3D object detection in M3DBench.

- Describe the <target> concisely.
- Can you provide a brief overview of the <target>?
- Provide a brief description of the given <target>.
- Can you relay a brief and clear account of the <target>?
- Offer a clear and concise depiction of the <target>.
- Sum up the main aspects of the <target> succinctly.
- In brief, how can you depict the <target>?
- Could you share a short summary of the <target>'s features?
- Summarize the visual content of the <target>.
- What's your concise interpretation of the <target>?
- In a short description, what is the <target>?
- Convey a brief description of the essential features of the <target>.
- How would you describe the <target> in brief?

- Describe the following <target> in detail.
- Provide a detailed description of the given <target>.
- Offer a thorough analysis of the <target>.
- Clarify the contents of the displayed <target> with great detail.
- Analyze the <target> in a comprehensive and detailed manner.
- I would appreciate a full and detailed explanation of the <target>.
- I'm interested in a detailed exploration of the <target>; could you provide that?
- Can you dissect the <target>, giving us a comprehensive understanding?
- Share a rich and detailed narrative of the <target>.
- Offer a profound and comprehensive insight into the <target>.

Table 12. Some examples of instructions for brief (top) and detailed (bottom) dense caption in M3DBench.

```

Replace <target> with point prompt:
f"{{object_name}} close to the pointed spot <point>{x},{y},{z}</point>"  

f"{{object_name}} situated near the pointed point <point>{x},{y},{z}</point>"  

f"{{object_name}} positioned close to the pointed location <point>{x},{y},{z}</point>"  

f"{{object_name}} near the pointed point <point>{x},{y},{z}</point>"  

f"{{object_name}} close to the pointed location <point>{x},{y},{z}</point>"  

f"{{object_name}} situated near the given point <point>{x},{y},{z}</point>"  

f"{{object_name}} situated close to the pointed location <point>{x},{y},{z}</point>"  

f"{{object_name}} positioned near the pointed point <point>{x},{y},{z}</point>"  
  

Replace <target> with box prompt:
f"{{object_name}} in the region <box>{x},{y},{z},{l},{w},{h}</box>"  

f"{{object_name}} situated in the region <box>{x},{y},{z},{l},{w},{h}</box>"  

f"{{object_name}} inside the area <box>{x},{y},{z},{l},{w},{h}</box>"  

f"{{object_name}} placed in the region <box>{x},{y},{z},{l},{w},{h}</box>"  

f"{{object_name}} with center at [{cx}, {cy}, {cz}] and dimensions [{lx}, {ly}, {lz}]  

f"{{object_name}} positioned at center [{cx}, {cy}, {cz}] with size [{lx}, {ly}, {lz}]  

f"{{object_name}} centered at [{cx}, {cy}, {cz}] with measurements [{lx}, {ly}, {lz}]  

f"{{object_name}} having central coordinates [{cx}, {cy}, {cz}] and measurements of [{lx}, {ly}, {lz}]  
  

Replace <target> with image prompt:
f"<image>{image_path}</image>"  
  

Replace <target> with 3d object prompt:
f"<obj_3d>{3d_object_path}</obj_3d>"
```

Table 13. The formula for interleaved multi-modal instruction generation in M3DBench.

B. Experiments on Dialogue and Localization

Quantitative Evaluation on Multi-round Dialogue. We score the conversational abilities of the baseline models using GPT-4 [40]. For multi-round dialogue, the baseline model employing PointNet++ [46] as the scene encoder and Vicuna-7B-V1.5 [20] as the language decoder demonstrates the optimal performance, surpassing the next best variant by +0.71 points. Similar to the conclusions derived from the results of detailed description (detailed in the Sec. 5.2), all OPT-based variants [69] exhibit relatively lower performance. In Sec. E, we provide prompts used for scoring conversations with GPT-4 [40], along with qualitative results for multi-turn dialogues and the GPT-4 [40] scoring criteria.

3D Vision Encoder	LLM Decoder	Relative Score
Pointnet++ [46]	OPT-6.7B [69]	40.97
	LLaMA-2-7B [55]	44.74
	Vicuna-7B-v1.5 [20]	46.06
Transformer [56]	OPT-6.7B [69]	29.52
	LLaMA-2-7B [55]	38.61
	Vicuna-7B-v1.5 [20]	45.35

Table 14. **Benchmark for multi-round dialogue.** *Relative Score* is generated by the GPT-4 [40], based on the evaluation of the model’s response.

3D Vision Encoder	LLM Decoder	Acc@0.25IoU
Pointnet++ [46]	OPT-6.7B [69]	3.09
	LLaMA-2-7B [55]	1.60
Transformer [56]	OPT-6.7B [69]	1.22
	LLaMA-2-7B [55]	3.57

Table 15. **Benchmark for object localization.** We assess the baseline model’s ability to identify and localize objects in the 3D scene. Specifically, the baseline model is tasked with outputting the location of the target object a given specific instruction. The metric utilized is Acc@0.25IoU.

Quantitative Evaluation on 3D object Localization. For the 3D object localization task (i.e., finding the object in a scene that best matches a given instruction), we propose using a unified output format to represent object position. To acquire localization data, we derive 3D bounding boxes from the “[cx, cy, cz, l, w, h]” provided in the generated text. Here, cx, cy,

cz correspond to the x, y, z coordinates of the bounding box center, while l, w, h represent the size of the bounding box along the x, y, z axes. For each value defining the 3D bounding box, we retain one decimal place. In Tab. 15, we present baseline performances regarding 3D localization. Results indicate that our proposed baseline model exhibits suboptimal performance on localizing. We leave the improvement of MLMs' abilities in 3D scene perception and localization for future work.

C. Held-out Evaluation

Training and Evaluation Protocols. In order to assess the zero-shot performance of the baseline model fine-tuned on the multi-modal instruction data for unseen tasks, we partition our dataset into two types: held-in and held-out datasets. Specifically, we consider embodied question answering (EQA) and embodied planning (EP) from M3DBench as unseen tasks, with their corresponding dataset (held-out dataset) excluded during the training process. We train the baseline model on the training dataset for the remaining tasks (held-in dataset) and evaluate the model's performance using the validation set from the held-out dataset.

Baselines and Metrics. We utilize a pre-trained masked transformer encoder[16] as the scene encoder and employ two large language models, OPT-6.7B [69] and LLaMA-2-7B [55], as the decoder in our baseline model. Furthermore, we employ BLEU 1-4 [43], ROUGE-L [34], METEOR [5], and CiDER [57] as evaluation metrics.

Task	LLM Decoder	BLEU-1↑	BLEU-2↑	BLEU-3↑	BLEU-4↑	ROUGE↑	METEOR↑	CIDEr↑
Embodied Question Answering	OPT-6.7B[69]	28.76	21.67	17.51	13.96	30.78	17.64	139.06
	LLaMA-2-7B [55]	35.76	27.42	21.89	16.83	40.04	20.47	163.71
Embodied Planning	OPT-6.7B[69]	21.13	16.07	12.36	8.99	28.96	16.28	47.62
	LLaMA-2-7B [55]	33.80	25.15	19.23	14.71	33.30	19.65	58.21

Table 16. **Zero-shot results on Embodied Question Answering (EQA) and Embodied Planning (EP).** For held-out evaluation, we demonstrate the performance of baseline methods on two tasks. The upward arrow (\uparrow) indicates that higher values represent better performance. Notably, we find that leveraging LLaMA-2 [55] as the language decoder exhibits superior zero-shot generalization compared to the OPT-based [69] model.

Result Analysis. In Tab. 16, we present the performance of the baseline model for held-out evaluation. Additionally, we compare baselines using different LLMs as language decoders. All baselines follow the same training and evaluation protocols described above. In summary, we draw three insights: 1) through instruction tuning and multi-task learning on the held-in dataset of M3DBench, the baseline model exhibits reasoning ability when dealing with tasks that it hasn't encountered before. 2) LLaMA-based [55] model outperforms the baseline model based on OPT [69] in zero-shot generalization. 3) There remain gaps in zero-shot results compared to results from full supervised instruction fine-tuning (detailed in the Sec. 5.2). These findings indicate that through instruction tuning and multi-task learning on M3DBench, our model demonstrates reasoning abilities on tasks that haven't encountered before. This emphasizes the significance of instruction tuning for achieving zero-shot generalization.

D. Implementation

Scene Encoder. As introduced in Sec. 5.1, we employ two commonly used types of 3D pre-trained feature extractors as scene encoders: one based on PointNet++ [46] and the other based on Transformer [56]. The PointNet++-based scene encoder comprises four layers for feature extraction and down-sampling, coupled with two layers for feature aggregation and up-sampling [70]. The final layer generates features for sampled points, from which we derive scene-level 256-dimensional features via global max-pooling. In addition, the Transformer-based encoder initially tokenizes input point clouds into 2048 point tokens through a set abstraction layer [46], followed by three cascaded Transformer encoder blocks with masking radii of 0.16, 0.64, and 1.44 [16]. Between the first two Transformer blocks, there is an additional set abstraction layer that downsamples the encoded tokens, with each token represented by 256 dimensions.

Multi-modal Prompt Encoder. We utilize the tokenizer and word embedding from pre-trained LLM [20, 55, 69] to process text and coordinate instructions. For image inputs, we employed the pre-trained ViT-L/14 [48] as the image encoder, adopting a trainable projector based on the LLaVA [35] to collect image tokens. Regarding 3D object inputs, we utilized a pre-trained 3D encoder [16] to extract object features, obtaining object-level tokens via another projector. For point-level and box-level

prompts, we directly employ linear layers to project the corresponding prompt features into the LLM embedding space. We leave the design of more optimal models for exploration in future work.

Trainable Parameters. The model comprises roughly 52 million trainable parameters, accounting for less than 1% of the frozen LLM backbone’s (LLaMA-2-7B [55]) parameter count.

E. GPT-4 Evaluation

We employ the template in Tab. 17 to prompt GPT-4 [40] and obtain corresponding evaluation results. Specifically, we prompt GPT-4 [40] with four inputs: system message, question, reference answer, and models’ responses, which comprise answers from various baseline models. We prompt GPT-4 [40] to assess responses for accuracy, relevance, descriptive details, etc., and assign scores within a range of 0 to 100. Higher scores indicate better quality of the responses. Moreover, we request GPT-4 [40] to provide explanations for the scoring results, assisting in our evaluation of the scoring results’ validity. In Fig. 15 and Fig. 16, we present model’s (utilizing transformer [56] as the scene encoder and LLaMA [55] as the language decoder) responses, GPT-4 scores, and GPT-4 justifications on detailed description and multi-round dialogue.

[System Message] You are an expert in scene details description. We will provide you a question, a reference answer for this question, and responses from six AI assistants. We would like your feedback on the performance of the AI assistants on their responses to the question. You need to evaluate each assistant’s response based on the question and reference answer. Please rate them on accuracy, relevance, naturalness, level of detail of their answers, and correspondence score from 0 to 100. Higher scores indicate better overall quality of the model’s response. Please first output the first line indicating the scores for six AI assistants, respectively. In the next line, provide a comprehensive explanation of your evaluation, avoiding any potential bias, and ensuring that the order in which the responses were presented does not affect your judgment.

Table 17. The template employed to prompt GPT-4 for assessing the quality of responses across different models.

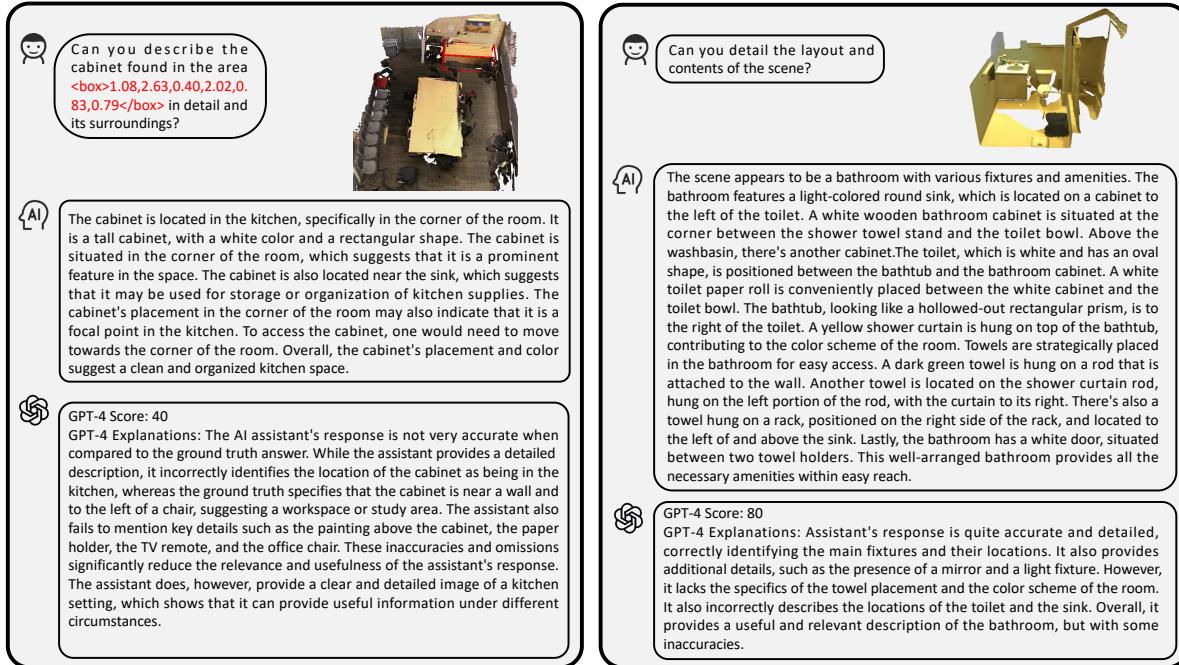


Figure 15. **Qualitative results on scene description.** We present visualization results for scene description, along with the scores provided by GPT-4 and the criteria used by GPT-4 for scoring.

 Let's start with the towels. I see a couple of rectangular towels on a rack at the left side of the room.

 Yes, one towel is the leftmost on the rack, and another is in the middle of the rack. To the left of the towels, there's a white door.

 And to the right of the door, there's a gray trash can. It's next to the vanity sink.

 That's correct. The trash can is conveniently located for easy disposal of waste. The sink it's next to is part of a bathroom vanity that appears to be made of marble or composite.

 And there's also a rectangular brown bench in the room. It's placed adjacent to the door.

 Yes, the bench is positioned near the door. It could be a convenient spot for placing items or sitting down.

 So, this is a well-equipped bathroom.

 The towels are in a convenient location, and the trash can is nearby for easy disposal of waste. The bench could be used for sitting or placing items.

 **GPT-4 Score:** 60
GPT-4 Explanations: The AI assistant's response is partially relevant and accurate, but it does not fully address the statement made by Agent 1. The assistant repeated information about the towels, trash can, and bench, which is accurate but not necessary in this context. Instead, the assistant should have agreed with or added to Agent 1's conclusion about the bathroom being well-equipped, as the ground truth answer did. The assistant's response lacks the necessary level of detail and correspondence in this specific context.

 Let's start with the furniture. I see a brown wooden chair and a nightstand. What could they be used for?

 The chair and nightstand could form a comfortable reading corner or a place to relax. The nightstand could hold a lamp, books, or other personal items.

 I see. Now, there are two doors in the room. Are they entry points to this room?

 Yes, they could be entry points to this room or they could lead to other rooms or spaces within the house. They play a key role in the room's layout and function.

 Now, let's talk about the trash cans. I see a few of them around the room.

 Yes, there are several trash cans in the room. This suggests a focus on cleanliness and organization. They are conveniently placed near the door and the table.

 Lastly, I see two windows above the table. What do they add to the room?

 The windows add natural light and a sense of openness to the room. They also provide a view of the outside world, which could be a source of inspiration or relaxation.

 **GPT-4 Score:** 95
GPT-4 Explanations: The AI assistant's response is highly relevant and accurate, providing a thoughtful interpretation of the function of the windows in the room. It correctly identifies the benefits of natural light and the view, which aligns with the ground truth answer. The assistant also adds a unique perspective by suggesting the view could be a source of inspiration or relaxation. However, it lacks the detail about blinds and control over privacy and light, which was mentioned in the ground truth answer. Therefore, the response is not completely detailed, but it is still very useful and corresponds well overall.

Figure 16. **Qualitative results on multi-round dialogue.** We present visualization results for multi-round dialogue, along with the scores provided by GPT-4 and the criteria used by GPT-4 for scoring.