Chat-Scene: Bridging 3D Scene and Large Language Models with Object Identifiers

Haifeng Huang[1,2†] **Yilun Chen**[2†] **Zehan Wang**[1†] **Rongjie Huang**[1] **Runsen Xu**[2] **Tai Wang**[2]
Luping Liu[1] **Xize Cheng**[1] **Yang Zhao**[3] **Jiangmiao Pang**[2] **Zhou Zhao**[1,2*]

[1]Zhejiang University [2]Shanghai AI Laboratory [3]Bytedance Inc.
{huanghaifeng}@zju.edu.cn

Abstract

Recent advancements in 3D Large Language Models (LLMs) have demonstrated promising capabilities for 3D scene understanding. However, previous methods exhibit deficiencies in general referencing and grounding capabilities for intricate scene comprehension. In this paper, we introduce the use of object identifiers and object-centric representations to interact with scenes at the object level. Specifically, we decompose the input 3D scene into a set of object proposals, each assigned a unique identifier token, which enables efficient object referencing and grounding during user-assistant interactions. Given the scarcity of scene-language data, we model the scene embeddings as a sequence of explicit object-level embeddings, derived from semantic-rich 2D or 3D representations. By employing object identifiers, we transform diverse 3D scene-language tasks into a unified question-answering format, facilitating joint training without the need for additional task-specific heads. With minimal fine-tuning on all downstream tasks, our model significantly outperforms existing methods on benchmarks including ScanRefer, Multi3DRefer, Scan2Cap, ScanQA, and SQA3D. Code has been released at https://github.com/ZzZZCHS/Chat-Scene.

1 Introduction

Recent advancements in Large Language Models (LLMs) [14, 39, 47, 15, 65, 32] have established language as a universal interface for creating general-purpose assistants. This breakthrough has been instrumental in the development of Multi-modal LLMs (MLLMs), which effectively tackle a broad spectrum of multi-modal tasks. While significant strides have been made in 2D MLLMs [33, 35, 77, 79, 34, 31, 67], current 3D MLLMs still face significant challenges that must be overcome to achieve a general-purpose assistant for 3D scene understanding.

Object referencing and grounding are essential for advanced scene understanding. Object referencing involves a model's precise comprehension of the semantics associated with a user-specified object, while object grounding requires the model's ability to localize a target object within the scene. These capabilities are vital for various 3D scene-language tasks such as dense captioning [12] and visual grounding [4, 75, 1]. However, current 3D MLLMs lack general referencing and grounding capabilities, often failing in tasks that necessitate precise object referencing or grounding—contrary to the objectives of addressing general-purpose tasks.

Regarding the object referencing capability, several 3D MLLMs [6, 23, 54] employ additional prompt encoders to comprehend user-specified objects, but they still lack grounding capabilities. The 3D-

† Equal contribution.
* Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3620

LLM [21] incorporates location tokens to enable object grounding, a technique validated in the 2D domain [67]. However, this approach underperforms on the 3D grounding benchmark, ScanRefer [4], compared to traditional expert models. The ineffectiveness of location tokens in the 3D domain primarily arises from the significant data scarcity in the scene-language area. Current 3D scene-language datasets [4, 75, 1] contain only tens of thousands of grounding instances, a scale much smaller than the million-level datasets used for training 2D MLLMs [67, 72]. Given the exponentially greater complexity of 3D spaces compared to 2D spaces, robust training of location tokens for 3D MLLMs may require substantially more data than is currently used for 2D MLLMs. Therefore, our objective is to explore more efficient methods for object referencing and grounding and to mitigate the impact of data scarcity.

We observe that most existing 3D MLLMs convert the 3D scene into hidden 3D scene embeddings, employing either a Q-Former-based module [21, 6] or direct projection methods [23]. Such architectures inherently lack the capability to efficiently interpret individual object instances. To address this limitation, we propose a novel approach for **representing and interacting with 3D scenes at the object level** within the language model. This method incorporates two principal designs: (i) referencing 3D scene using object identifiers, and (ii) representing 3D scene using well-trained object-centric representations. The first component offers a unified format for object referencing and grounding, while the second alleviates the requirement for extensive scene-language datasets.

Reference 3D scene using object identifiers. Objects play a crucial role in defining and interpreting a scene, as their organization shapes the entire 3D landscape. This intuition is evident in most 3D scene understanding benchmarks, including 3D grounding, VQA, and dense captioning, all of which annotate at the object level. To effectively model scene embeddings at the object level, the entire 3D scene can be decomposed into a set of object proposals via reliable 3D detectors [45, 27, 71, 50, 30]. Importantly, we assign objects with object identifiers—a set of learnable identifier tokens $\{\texttt{<OBJ}k\texttt{>}\}_{k=1...n}$—to distinguish them during language modeling. This design allows the LLM to reference respective objects using discrete identifier tokens. As the example shown in Figure 1, the chair and the two trash cans are labeled as "<OBJ013>", "<OBJ023>", and "<OBJ032>", respectively. This avoids the text ambiguity that arises from subjective viewing words like "rightmost". Besides, the lengthy description like "the chair located at the southwest corner of the rightmost table" often complicates user-assistant interaction. These identifiers enable efficient object referencing and grounding during user-assistant interactions. By using these identifiers, we convert diverse 3D scene-language tasks into a unified format of question-answering pairs, facilitating joint training without any additional task-specific heads.

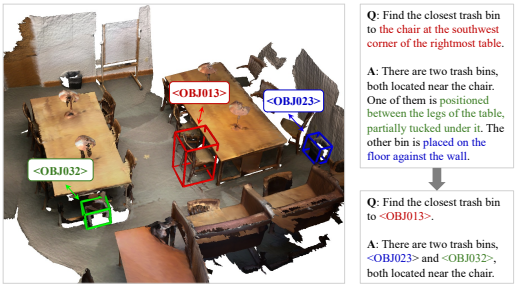


Figure 1: An example of using object identifiers during the conversation.

Represent 3D scene using well-trained object-centric representations. The scene-level representation requires a large amount of paired scene-language data for training, which is generally unaffordable and labor-intensive due to its complex real-world scenarios. To address this challenge, our model represents the scenes using a set of object-level embeddings, which obtain object-centric representations from well-trained 2D and 3D models. Specifically, after obtaining object proposals from prior detectors (either 2D or 3D), we extract the object features using well-trained 3D object-centric representations [78] or 2D representations [40]. Due to the million-level pre-training, these representations contain abundant semantic and visual cues. Through simple linear layers, we project them into the embedding space of the language model. Combined with the object identifier, the sequences of object-level embeddings are thus constructed into scene embeddings and fed into the LLM. With merely *two* epochs of fine-tuning on all downstream tasks, extensive experiments on either 3D, 3D+2D, or 2D-only settings demonstrate the effectiveness of our model in various downstream 3D scene understanding tasks.

We perform comprehensive experiments across five representative 3D scene-language datasets, including ScanRefer [4], Multi3DRefer [75], Scan2Cap [12], ScanQA [2], and SQA3D [38]. Our model consistantly enhances state-of-the-art performance across all these datasets without fine-tuning

https://doi.org/10.52202/079017-3620

on specific tasks. Notably, it surpasses previous methods by 3.7% (Acc@0.5) on ScanRefer, 14.0% (F1@0.5) on Multi3DRefer, 8.7% (CIDEr@0.5) on Scan2Cap, and 7.7% (CIDEr) on ScanQA.

Our contributions are summarized as follows:

- We propose an enhanced 3D MLLM which models and interacts with 3D scenes at the object level.
- We introduce object identifiers to enable efficient referencing and grounding within 3D scenes. By leveraging these identifiers, we convert diverse 3D scene-language tasks into a unified question-answering format, facilitating joint training without necessitating additional task-specific heads.
- We effectively represent the 3D scene through a sequence of multi-modal object-centric representations derived from well-trained foundation models, which alleviate the impact of scene-language data scarcity.
- Our model significantly enhances state-of-the-art performance across various 3D scene-language datasets without fine-tuning for specific tasks. Extensive experiments on either 3D, 3D+2D, or 2D-only settings demonstrate the effectiveness of our model for 3D indoor scene understanding.

2 Related Work

3D Scene-language Understanding In the rapidly evolving field of 3D scene understanding, there is an increasing focus on using language to provide both contextual knowledge and query conditions, thus enabling precise interpretation of user intentions. This process, known as "3D scene-language understanding", leverages language to more effectively grasp the intricacies of 3D environments in a manner consistent with human cognition. The primary tasks in this domain include: 1) 3D Visual Grounding [4, 1, 25, 75, 7, 52, 76, 53, 48], which involves locating specified objects within a 3D scene based on textual queries; 2) 3D Dense Captioning [12, 69, 28, 8, 9], which demands proficiency in both localizing and captioning objects densely in the scene; 3) 3D Visual Question Answering [2, 43, 38], which focuses on general scene question answering. Initial efforts concentrated on specialized tasks, resulting in limited generalizability across different 3D scene understanding tasks. Recent initiatives such as 3DJCG [3] and D3Net [**?**] have aimed to unify tasks like 3D visual grounding and dense captioning, leveraging their synergistic benefits to enhance overall model performance. Advances like 3D-VisTA [80] and 3D-VLP [29] are working to develop a more general 3D visual-language framework through pre-training techniques for better scene-language alignment. However, despite these models' adeptness at handling various 3D scene tasks, their reliance on task-specific heads limits their adaptability for broader user-assistant interactions.

Multi-modal Large Language Models. Recent advancements in large language models (LLMs) have exhibited impressive capabilities in intricate reasoning and interactive dialogues with humans [14, 39, 47, 15]. There is a growing interest in enhancing the scope of LLMs to encompass additional modalities [31, 33, 35, 77, 79, 20, 19, 21, 54, 65, 32, 61, 23, 10]. In the 3D realm, PointLLM [61] directly maps point clouds into the embedding space of the LLM. Both Imagebind-LLM [20] and Point-LLM [19] integrate the 3D modality into LLMs by establishing a joint embedding space among 3D point clouds, images, audio, and text. These models perform well at the object level but encounter difficulties when interpreting complex spatial relationships in 3D scenes. To improve scene understanding, 3D-LLM [21] incorporates positional embedding and learns location tokens. Nevertheless, it projects 3D features into the input space of pre-trained 2D Vision-Language Models (VLMs). Involving 2D encoders make it difficult to grasp the 3D spatial structure and intricate relationships among objects. Chat-3D [54] tackles this limitation by directly utilizing 3D scene-text data to align the 3D scene with the LLM, overcoming the challenge of limited data availability through a pre-alignment phase. However, the architectural design of this model limits its focus on specific target objects during interactions. Current 3D MLLMs face challenges in precise object referencing and grounding, limiting their functionality to straightforward tasks. By incorporating object identifiers into the LLM, we significantly enhance the object referencing and grounding capabilities of 3D MLLMs, thereby showing potential for complex real-world applications.

3D Representation Learning Recently, numerous efforts have been made to learn discriminative and robust representations for 3D point clouds, which serve as a fundamental visual modality. Approaches such as PointBERT [68], Point-MAE [41], Transformer-OcCo [51], and Point-m2ae [73] employ self-supervised learning techniques to extract meaningful representations of 3D objects from unlabeled point cloud data. Another set of works [62, 36, 74, 17, 58, 56, 55, 57] seeks to extend

 https://doi.org/10.52202/079017-3620

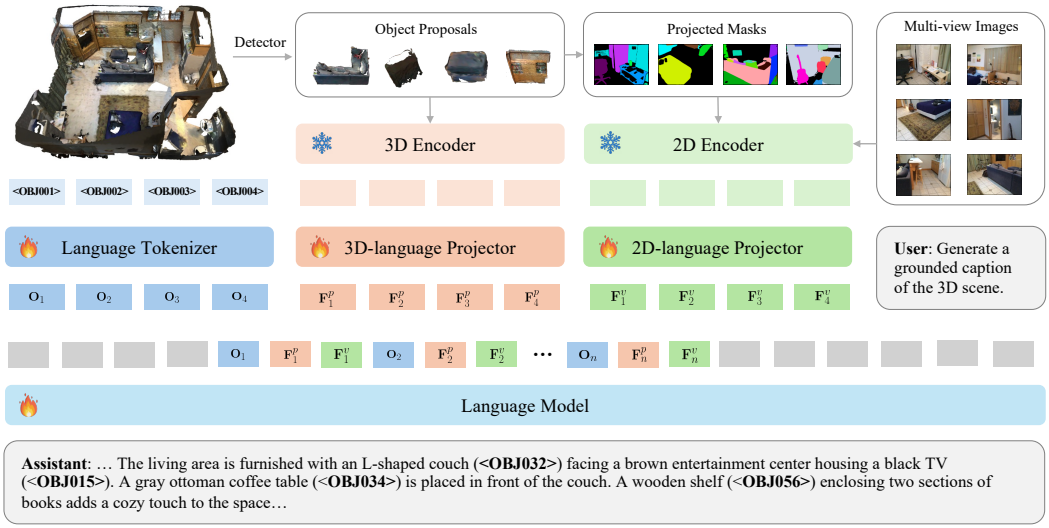


Figure 2: **Overall model architecture.** The model processes a 3D scene's point cloud input by decomposing it into object proposals via a pre-trained detector. Subsequently, the 3D and 2D encoders are employed to extract object-centric representations. After projection layers, they are combined with object identifiers to form the scene embeddings as a sequence of object-level embeddings, which are then fed into the LLM. The assigned unique identifiers enable efficient object referencing in subsequent interactions.

representation from other modalities to the 3D domain. For example, ULIP [62] and OpenShape [36] construct 3D-image-text triplets to align point clouds within the CLIP [44] representation space. C-MCR [58] and Ex-MCR [56] learn contrastive representations between various modalities, including 3D point clouds. They leverage knowledge from existing MCR spaces to tackle to challenge of lacking paired data. These robust 3D representations effectively capture detailed information about 3D objects. Our approach involves segmenting the 3D scene at the instance level and extracting a set of object features to represent the entire scene.

3 Method

3.1 Overview

Our motivation is to facilitate object referencing and grounding for 3D MLLMs while simultaneously addressing the scarcity of scene-language data. We propose representing 3D scenes at the object level by using object identifiers for referencing and employing well-trained, object-centric representations for scene depiction. Section 3.2 delineates the model architecture, which primarily consists of generating a sequence of object-level embeddings to represent the entire scene. Section 3.3 provides illustrations of the prompt template through examples. Lastly, Section 3.4 details the training methodology of our model.

3.2 Model Architecture

As illustrated in Figure 2, our method processes a 3D scene's point cloud by decomposing it into object proposals using a pre-trained detector. We then employ pre-trained 3D and 2D encoders to derive object-centric representations from point clouds and multi-view images, respectively. These representations are subsequently mapped into the token embedding space of a language model. By incorporating n object identifiers into the language model's vocabulary, we link these identifiers to the corresponding object proposals, thereby facilitating efficient object referencing and grounding during user-assistant interactions. Finally, the scene embeddings, composed of a sequence of object-level embeddings, are input into the LLM.

https://doi.org/10.52202/079017-3620

Object Detector. Given a point cloud from a 3D scene, we decompose it into n objects using the pre-trained detector Mask3D [45]. Compared to object detection models [11, 46], the instance segmentation model is preferred in this work due to due to its capability to generate accurate masks, which are essential for subsequent projection into 2D masks on multi-view images. The point cloud of the i-th object is denoted as $\mathbf{P}_i \in \mathbb{R}^{m_i \times 6}$, where m_i represents the number of points in the i-th object, and the 6D information for each point comprises its 3D coordinates and RGB colors.

Object Identifiers. To achieve a localized understanding of 3D scenes, we introduce a set of learnable identifier tokens $\{\texttt{<OBJ}i\texttt{>}\}_{i=1...n}$, designated as object identifiers, into the token vocabulary of the language model. These identifiers are processed by the tokenizer to produce their respective token embeddings $\{\mathbf{O}_i\}_{i=1...n}$. The identifier tokens are then integrated with the object tokens to establish one-to-one correspondences, thereby enabling object referencing and grounding using identifiers in subsequent interactions.

Object-level Embeddings. After extracting object proposals from the 3D scene, we derive object features using well-trained 3D and 2D object representations. Owing to the million-level scale of pre-training, these representations are rich in semantic and visual cues. By employing simple linear layers, we project them into the embedding space of the language model. Together with identifier token embeddings, this process yields object-level embeddings for each object.

3D Encoder. The 3D encoder excels in extracting spatial and shape attributes from point clouds. We employ a pre-trained 3D encoder Uni3D [78] to derive object-centric 3D representations. This embedding processes each object's point cloud $\mathbf{P}_i$, outputting the feature $\mathbf{Z}_i^p$ for each object.

2D Encoder. The 2D encoder adeptly extracts semantically rich features from 2D images. We project the point clouds for each object onto multi-view images, creating a sequence of 2D masks. Utilizing a pre-trained DINOv2 [40], we extract and aggregate local features from all masked regions across the multi-view images of each object, taking into account both mask areas and multi-view information. We opted for DINOv2 over the more common CLIP [43] due to its superior handling of local features within images. The 2D encoder processes the multi-view images and their corresponding projected masks from each object's point cloud $\mathbf{P}_i$, generating the visual feature $\mathbf{Z}_i^v$ for each object. Details are provided in Appendix A.

Visual-Language Projectors. To align the extracted object representations with the language model, we employ a 3D-language projector $f_p(\cdot)$ and a 2D-language projector $f_v(\cdot)$ to map the 3D point cloud features and 2D visual features into the token embedding space of the language model. For the i-th object, these features are represented as token embeddings $\mathbf{F}_i^p$ and $\mathbf{F}_i^v$.

$$\mathbf{F}_i^p = f_p(\mathbf{Z}_i^p); \quad \mathbf{F}_i^v = f_v(\mathbf{Z}_i^v). \tag{1}$$

Scene Embeddings. Following the process described above, we obtain an object identifier token embedding $\mathbf{O}_i$, a 3D object token embedding $\mathbf{F}_i^p$, and a 2D object token embedding $\mathbf{F}_i^v$ for each object. We combine the identifier token embeddings and object token embeddings in an one-to-one correspondence manner to formulate a sequence of object-level embeddings, which represents the constructed scene embeddings and then be fed into the LLM to represent the whole scene.

3.3 Prompt Template

Despite variations in task formulations, both referencing and grounding are unified using object identifiers. As illustrated in Table 1, the system message encodes object information in the scene as a sequence of "<OBJi> <object>", where <OBJi> denotes the identifier token for the i-th proposal, and <object> serves as the placeholder for object tokens. The language tokenizer converts <OBJi> into its token embedding $\mathbf{O}_i$ and <object> into the combined object token features $\mathbf{F}_i^p$ and $\mathbf{F}_i^v$. As illustrated by the following interaction, the user can directly employ identifier tokens to reference specific objects, while the assistant uses these tokens in responses to precisely ground target objects.

3.4 Training Strategy

Most existing MLLMs [18, 35, 23] adopt a two-stage training approach, comprising an initial alignment phase to train the projector exclusively, followed by a fine-tuning phase for both the projector and the language model. This method not only demands extra data and extended training duration for alignment but also complicates determining the optimal duration for the initial phase. Consequently, we opt for a single-stage process, concurrently training both the projectors and the

 https://doi.org/10.52202/079017-3620

Table 1: **Prompt template for the language model.**

System: A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions. The conversation centers around an indoor scene: [<OBJ001> <object> <OBJ002> <object> ... <OBJn> <object>].
User: Find the closest trash bin to <OBJ013>.
Assistant: There are two trash bins, <OBJ023> and <OBJ032>, both located near the chair.

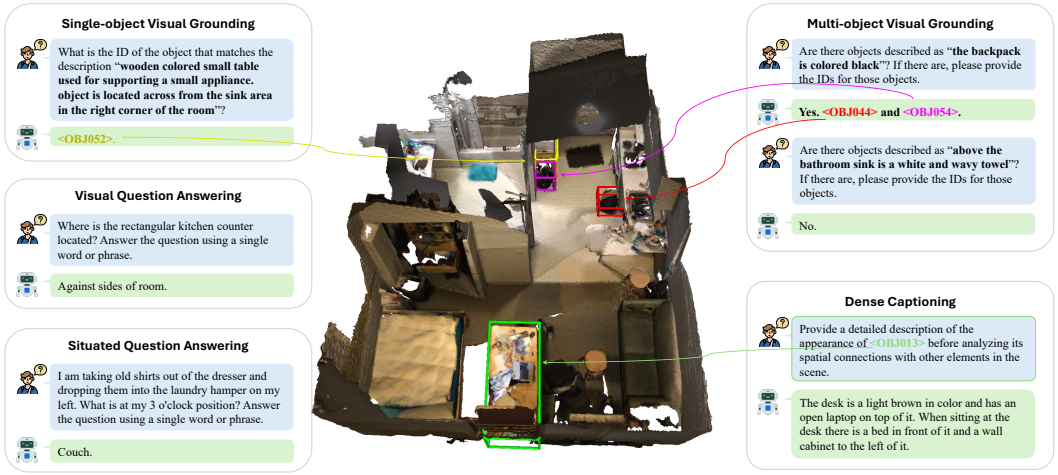


Figure 3: **Examples of various 3D scene-language understanding tasks**. All the tasks are unified to single-turn question-answering pairs without extra task heads. Object identifiers are used to reference and ground the object during the conversation.

language model. In our experiments, we observe that this jointly trained model already exhibits superior performance without the necessity of fine-tuning for specific downstream tasks.

Training Data We aggregate essential training data for downstream tasks and standardize it into uniform instruction formats. The downstream tasks encompass 3D visual grounding (ScanRefer & Multi3DRef), 3D dense captioning (Scan2Cap), and 3D visual question answering (ScanQA & SQA3D). We incorporate the training sets from these datasets into our training corpus. Each task is adaptable to a single-turn user-assistant interaction, as illustrated in Figure 3.

Training Objective We have unified all tasks into a consistent user-assistant interaction format, and as a result, the sole training loss in the joint-training phase is the Cross-Entropy loss of the language model. The training objective is to optimize the trainable parameters, denoted by θ, aiming to minimize the negative log-likelihood of the target response sequence s^{res} generated by the assistant. Specifically, given the input prefix sequence s^{prefix}, which encompasses both system messages and user instructions, the loss function is expressed as follows:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{k} \log P(s_i^{\text{res}} | s_{[1,\ldots,i-1]}^{\text{res}}, s^{\text{prefix}}), \tag{2}$$

where k is the number of tokens in the response sequence, and $s_{[1,\ldots,i-1]}^{\text{res}}$ denotes the sequence of the previous $i-1$ tokens in the response. The set of trainable parameters θ includes two vision-language projectors, newly added n token embeddings for object identifiers, and the language model itself.

Table 2: **Performance comparison.** "Expert models" are tailored for specific tasks using task-oriented heads, while "LLM-based models" are designed for general instructions and responses.

	Method	ScanRefer		Multi3DRefer		Scan2Cap		ScanQA		SQA3D	
		Acc@0.25	Acc@0.5	F1@0.25	F1@0.5	C@0.5	B-4@0.5	C	B-4	EM	EM-R
Expert Models	ScanRefer [4]	37.3	24.3	-	-	-	-	-	-	-	-
	ScanQA [2]	-	-	-	-	-	-	64.9	10.1	-	-
	3DJCG [3]	49.6	37.3	-	-	49.5	31.0	-	-	-	-
	3D-VLP [29]	51.4	39.5	-	-	54.9	32.3	67.0	11.1	-	-
	M3DRef-CLIP [75]	51.9	44.7	42.8	38.4	-	-	-	-	-	-
	3D-VisTA [80]	50.6	45.5	-	-	66.9	34.0	72.9	13.1	48.5	-
	ConcreteNet [48]	50.6	46.5	-	-	-	-	-	-	-	-
	Vote2Cap-DETR++ [9]	-	-	-	-	67.6	**37.1**	-	-	-	-
LLM-based Models	LAMM [66]	-	3.4	-	-	-	-	42.4	5.8	-	-
	Chat-3D [54]	-	-	-	-	-	-	53.2	6.4	-	-
	3D-LLM [21]	30.3	-	-	-	-	-	69.4	12.0	-	-
	LL3DA [6]	-	-	-	-	65.2	36.8	76.8	13.5	-	-
	LEO [23]	-	-	-	-	68.4	36.9	80.0	11.5	-	53.7
	Scene-LLM [18]	-	-	-	-	-	-	80.0	12.0	54.2	-
	Chat-Scene (Ours)	**55.5**	**50.2**	**57.1**	**52.4**	**77.1**	36.3	**87.7**	**14.3**	**54.6**	**57.5**

4 Experiments

4.1 Datasets and Metrics

Datasets. We conducted experiments on five benchmarks: ScanRefer [4] for single-object visual grounding, Multi3DRefer [75] for multi-object visual grounding, Scan2Cap [12] for dense captioning, and both ScanQA [2] and SQA3D [38] for visual question answering. These benchmarks are based on the ScanNet dataset [16], which comprises richly annotated RGB-D scans of real-world indoor scenes, including both 2D and 3D data across 1,513 scans. All benchmarks adhere to the same train/validation/test splits, facilitating joint training and evaluation.

Metrics. We adhere to the commonly used metrics in these benchmarks. For ScanRefer [4], we assess thresholded accuracy with Acc@0.25 and Acc@0.5, where predictions are deemed positive if they exhibit higher Intersection over Union (IoU) with the ground truths than the thresholds of 0.25 and 0.5, respectively. In evaluating grounding for a flexible number of target objects in Multi3DRefer [75], we employ the F1 score at IoU thresholds of 0.25 and 0.5. For Scan2Cap [12], we utilize CIDEr@0.5 and BLEU-4@0.5 (abbreviated as C@0.5 and B-4@0.5), integrating image captioning metrics with the IoU scores between predicted and target bounding boxes. For ScanQA [2], the metrics CIDEr [49] and BLEU-4 [42], abbreviated as C and B-4, are used. SQA3D [38] is evaluated using exact match accuracy (EM) and its refined version, EM-R, as proposed by LEO [23].

4.2 Comparison with State-of-the-art Methods

Compared Baselines. The baseline models can be classified into two principal categories: traditional expert models and general multi-modal LLMs. Traditional expert models generate outputs in a fixed format tailored to specific tasks. Conversely, LLM-based models yield open-ended outputs suitable for a broader range of applications.

- **Expert Models:** Models such as ScanRefer [4] and ScanQA [2] establish initial benchmarks for the ScanRefer and ScanQA datasets, respectively. 3DJCG [3] integrates multiple tasks within a single architecture, unifying visual grounding and dense captioning tasks due to their synergistic nature. Both 3D-VLP [29] and 3D-VisTA [80] aim to develop versatile 3D visual-language frameworks by focusing on pre-training strategies for 3D scene-language alignment. M3DRef-CLIP [75] introduces multi-object grounding, enhancing single-object grounding performance. ConcreteNet [48], the state-of-the-art (SOTA) model on the ScanRefer benchmark, innovates three methods to augment verbo-visual fusion for dense 3D visual grounding. Vote2Cap-DETR++ [9] decouples the processes of caption generation and object localization through parallel decoding, making it the SOTA model on the Scan2Cap benchmark.
- **LLM-based Models:** LAMM [66] extends research on 2D MLLM to point clouds but lacks a design tailored for 3D scene tasks. Chat-3D [54] introduces an object-centric method yet fails to address general 3D scene tasks comprehensively. 3D-LLM [21] utilizes location tokens for object grounding but is constrained by data scarcity. LL3DA [6] develops an assistant that processes

 https://doi.org/10.52202/079017-3620

Table 3: **Ablation studies on object identifiers.** "Plain Text" employs plain text for object numbers, "Gaussian" uses fixed Gaussian embeddings, and "Learnable" learns new identifier tokens. "Token Cost" denotes the total tokens for N objects, including object identifiers.

Identifier Token Type	Token Cost	ScanRefer Acc@0.5	Multi3DRef F1@0.5	Scan2Cap C@0.5	ScanQA CIDEr	SQA3D EM
Plain Text	$6N$	47.2	49.6	73.1	84.9	53.7
Gaussian	$3N$	46.1	49.4	71.7	82.5	53.4
Learnable	$3N$	**50.2**	**52.4**	**77.1**	**87.7**	**54.6**

Table 4: **Ablation studies on multi-modal object-centric representations.** "Early Fusion" merges object features before language model input, whereas "Separate Token" keeps them distinct. "Single" denotes using a single image to extract 2D feature of an object, while "Multi" uses multi-view images.

Fusion Method	Token Cost	3D	2D	ScanRefer Acc@0.5	Multi3DRef F1@0.5	Scan2Cap C@0.5	ScanQA CIDEr	SQA3D EM
–	$2N$	✓	–	41.2	43.8	64.9	80.3	53.4
		×	Single	32.9	37.2	65.9	83.7	53.2
		×	Multi	45.8	49.1	75.7	**88.2**	54.4
Separate Token	$3N$	✓	Single	45.7	49.1	73.8	86.5	53.2
		✓	Multi	**50.2**	**52.4**	**77.1**	87.7	**54.6**
Early Fusion	$2N$	✓	Single	42.4	46.8	70.9	85.9	52.9
		✓	Multi	46.9	50.2	74.4	88.0	53.9

point cloud data directly, responding to textual instructions and visual prompts. LEO [23] pioneers an embodied generalist approach by incorporating action tokens. Scene-LLM [18] merges scene-level and egocentric 3D information, enhancing understanding and reasoning in 3D environments. However, LL3DA, LEO, and Scene-LLM lack grounding capabilities.

Analysis. As shown in Table 2, our model surpasses previous methods across almost all metrics without task-specific fine-tuning, suggesting a promising unified framework for 3D scene understanding. For visual grounding tasks, our model boosts the state-of-the-art performance by 3.7% (Acc@0.5) on ScanRefer and 14.0% (F1@0.5) on Multi3DRefer, demonstrating excellent grounding capabilities. For the dense captioning task, we improve the SOTA performance by 8.7% (CIDEr@0.5) on Scan2Cap, highlighting strong object referring and captioning ability. For VQA tasks on ScanQA and SQA3D, which do not require object referencing and grounding, we still achieve consistent performance enhancement, demonstrating improved overall 3D scene understanding and reasoning.

4.3 Ablation Study

Object Identifiers. Table 3 shows that the format of object identifiers affects both performance and token cost. For comparing token costs, we consider scenes with hundreds of objects. A straightforward approach is to use plain text for object identifiers such as "Obj001", which is tokenized into four tokens ("Obj", "0", "0", "1"). Including two object tokens (3D & 2D), representing a single object requires six tokens in total. This high token cost makes the approach impractical for real-world scenarios. Thus, we explored using a single token per identifier by adding new tokens to the language's vocabulary. We assess two strategies: employing fixed random Gaussian embeddings ("Gaussian") and using learnable tokens ("Learnable"). The results show that learnable tokens enhance performance and reduce token costs simultaneously. Lowering token costs from $6N$ to $3N$ significantly reduces memory usage and accelerates training/inference when handling 3D scenes with numerous objects.

Multi-modal Object-centric Representations. We evaluate various methods for retrieving object features and combining features from different sources (3D and 2D), as shown in Table 4. As described in Section 3.2, the 3D and 2D features are derived from the 3D encoder and 2D encoder, respectively. We assess two methods of extracting 2D features: one from a single-view image ("Single") and another from multi-view images ("Multi").

First, we evaluate the performance of using either a single 3D feature or a single 2D feature for the object token. The results show that using a 2D feature derived from multi-view images yields

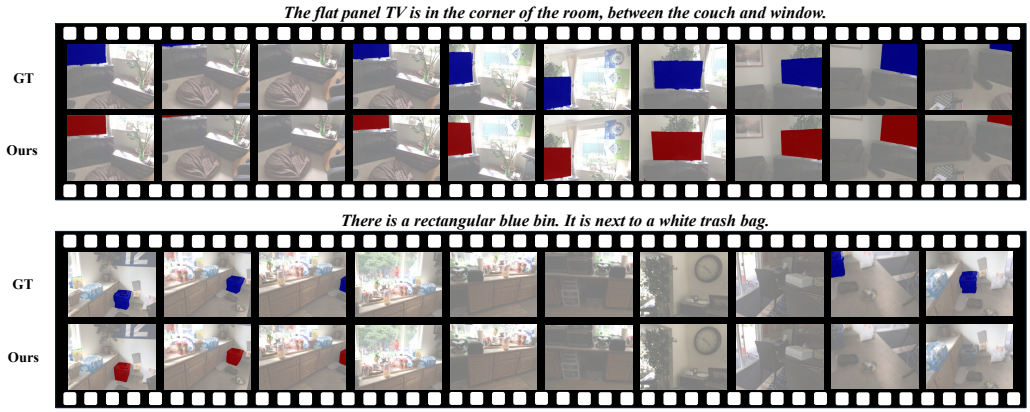


Figure 4: **Visualization results of video grounding for video input.** "GT" denotes the projected 2D masks derived from the ground-truth 3D point cloud mask.

better performance than using a 3D feature. This suggests that semantic information from 2D visual contexts is more crucial than spatial information from 3D point clouds. It may also indicate that the pre-trained 2D encoder is more reliable than the pre-trained 3D encoder due to the abundance of 2D image-text data compared to 3D-text data for pre-training.

Next, we assess the combination of both 3D and 2D features using two fusion methods. The first method, named "Separate Token" in the table, involves using two separate object tokens (3D and 2D object tokens) to represent object information. The second method, termed "Early Fusion", combines the 3D and 2D features into a single token for each object. The results indicate that combining 3D and 2D features consistently improves performance compared to using a single 3D/2D feature, highlighting the importance of utilizing both modalities. Fusing 3D and 2D tokens reduces the token cost, while it results in a slight performance drop. This provides an option for situations where the token limit is tight, suggesting that combining multi-modal features into one token is acceptable.

4.4 Experiments with 2D Video Input

In practical applications of 3D scene understanding, acquiring indoor RGB (video) scan is simpler than obtaining a processed 3D point cloud from RGB-D images. We examine our model's ability to adapt to video input (without depth) for 3D indoor scenes based on the ScanNet [16] dataset. For video input, we use a tracking-based video detector DEVA [13] to extract object proposals. This process involves detecting objects in each frame and merging these proposals across frames via the tracking module. After extracting objects from the video, we perform the same operations as for the 3D tasks. The grounding results can then be evaluated on video frames with 2D masks.

Tasks and Metrics. We assess video grounding and VQA tasks for video input. For video grounding, we use descriptions annotated in ScanRefer and project the ground-truth object's point cloud to 2D masks in video frames, allowing us to compute the IoU between the predicted masks and GT masks in 2D images. Given a video with a frame length of L, the predicted masks are a series of 2D masks denoted as $\{\mathbf{M}_i^{\mathrm{p}} \in \mathbb{R}^{H\times W}\}_{i=1...L}$ and the GT masks denoted as $\{\mathbf{M}_i^{\mathrm{g}} \in \mathbb{R}^{H\times W}\}_{i=1...L}$, where H and W are the height and width of an image, respectively. We concatenate these masks along the temporal axis to obtain a predicted spatial-temporal mask $\tilde{\mathbf{M}}^{\mathrm{p}} \in \mathbb{R}^{H\times W\times L}$ and a GT spatial-temporal mask $\tilde{\mathbf{M}}^{\mathrm{g}} \in \mathbb{R}^{H\times W\times L}$. We propose calculating the Spatial-Temporal IoU (ST-IoU) between the predicted mask $\tilde{\mathbf{M}}^{\mathrm{p}}$ and the GT mask $\tilde{\mathbf{M}}^{\mathrm{g}}$. Thus, similar to the metrics for the grounding task on ScanRefer, we use Acc@0.25 and Acc@0.5 to measure accuracy based on the ST-IoU threshold. For VQA tasks, we use the annotations of ScanQA and SQA3D along with their respective metrics for evaluation.

Table 5: **Evaluation results for video input.**

Method	Video Grounding Acc@0.25	Video Grounding Acc@0.5	ScanQA CIDEr	SQA3D EM
Random	1.4	0.5	–	–
Ours	**22.8**	**10.8**	**85.6**	**52.9**
Upperbound	54.9	22.2	–	–

https://doi.org/10.52202/079017-3620

Performance Analysis. Table 5 presents the evaluation results of video grounding and VQA tasks. For video grounding, we compute upper bound results to assess the quality of object masks extracted by the video detector. Compared to the upper bound and random results, our method demonstrates strong grounding ability. Visualization results are provided in Figure 4. The second example shows that the tracking-based video detector might lose track of an object after it has been out of sight for a prolonged period. This is a primary reason for the low quality of the extracted object masks. Missing parts of the frames that contain the target object leads to the low Acc@0.5 result of the upper bound. For VQA tasks, we achieve comparable results to those using objects extracted from 3D inputs. This indicates that despite the lower quality of extracted objects from video input, our approach of constructing scene embeddings from sequences of object-level embeddings efficiently enhances overall scene comprehension, thereby improving QA performance on ScanQA and SQA3D.

5 Conclusion

To enable efficient object referencing and grounding abilities in 3D MLLMs, this paper proposes modeling and interacting with 3D scenes at the object level. It decomposes the input 3D scene into a set of object proposals assigned with object identifiers. Given the scarcity of scene-language data, we model the scene embeddings as a sequence of explicit object-level embeddings, derived from semantic-rich 2D and 3D representations. By using object identifiers, we transform diverse 3D scene-language tasks into a unified question-answering format. With minimal fine-tuning on all downstream tasks, our model significantly outperforms existing methods across various benchmarks.

Acknowledgments and Disclosure of Funding

This work was supported by National Key R&D Program of China (2022ZD0162000).

References

[1] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. 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] D. Azuma, T. Miyanishi, S. Kurita, and M. 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.

[3] D. Cai, L. Zhao, J. Zhang, L. Sheng, and D. 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.

[4] D. Z. Chen, A. X. Chang, and M. Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020.

[5] J. Chen, W. Luo, X. Wei, L. Ma, and W. Zhang. Ham: Hierarchical attention model with high performance for 3d visual grounding. *arXiv preprint arXiv:2210.12513*, 2, 2022.

[6] S. Chen, X. Chen, C. Zhang, M. Li, G. Yu, H. Fei, H. Zhu, J. Fan, and T. Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding, reasoning, and planning. *arXiv preprint arXiv:2311.18651*, 2023.

[7] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev. Language conditioned spatial relation reasoning for 3d object grounding. *Advances in Neural Information Processing Systems*, 35:20522–20535, 2022.

[8] S. Chen, H. Zhu, X. Chen, Y. Lei, G. Yu, and T. 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.

[9] S. Chen, H. Zhu, M. Li, X. Chen, P. Guo, Y. Lei, Y. Gang, T. Li, and T. Chen. Vote2cap-detr++: Decoupling localization and describing for end-to-end 3d dense captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

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

[11] Y. Chen, Z. Yu, Y. Chen, S. Lan, A. Anandkumar, J. Jia, and J. M. Alvarez. Focalformer3d: focusing on hard instance for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8394–8405, 2023.

[12] Z. Chen, A. Gholami, M. Nießner, and A. 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.

[13] H. K. Cheng, S. W. Oh, B. Price, A. Schwing, and J.-Y. Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1316–1326, 2023.

[14] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. 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.

[15] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[16] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. 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.

[17] R. Dong, Z. Qi, L. Zhang, J. Zhang, J. Sun, Z. Ge, L. Yi, and K. Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv preprint arXiv:2212.08320*, 2022.

[18] R. Fu, J. Liu, X. Chen, Y. Nie, and W. Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024.

[19] Z. Guo, R. Zhang, X. Zhu, Y. Tang, X. Ma, J. Han, K. Chen, P. Gao, X. Li, H. Li, et al. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023.

[20] J. Han, R. Zhang, W. Shao, P. Gao, P. Xu, H. Xiao, K. Zhang, C. Liu, S. Wen, Z. Guo, et al. Imagebind-llm: Multi-modality instruction tuning. *arXiv preprint arXiv:2309.03905*, 2023.

 https://doi.org/10.52202/079017-3620

[21] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan. 3d-llm: Injecting the 3d world into large language models. *arXiv preprint arXiv:2307.12981*, 2023.

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

[23] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

[24] P.-H. Huang, H.-H. Lee, H.-T. Chen, and T.-L. Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1610–1618, 2021.

[25] S. Huang, Y. Chen, J. Jia, and L. Wang. Multi-view transformer for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15524–15533, 2022.

[26] A. Jain, N. Gkanatsios, I. Mediratta, and K. Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *European Conference on Computer Vision*, pages 417–433. Springer, 2022.

[27] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020.

[28] Y. Jiao, S. Chen, Z. Jie, J. Chen, L. Ma, and Y.-G. Jiang. More: Multi-order relation mining for dense captioning in 3d scenes. In *European Conference on Computer Vision*, pages 528–545. Springer, 2022.

[29] Z. Jin, M. Hayat, Y. Yang, Y. Guo, and Y. Lei. Context-aware alignment and mutual masking for 3d-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10984–10994, 2023.

[30] M. Kolodiazhnyi, A. Vorontsova, A. Konushin, and D. Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. *arXiv preprint arXiv:2311.14405*, 2023.

[31] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023.

[32] B. Li, Y. Zhang, L. Chen, J. Wang, J. Yang, and Z. Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023.

[33] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[34] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[35] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[36] M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han, H. Cai, F. Porikli, and H. Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *arXiv preprint arXiv:2305.10764*, 2023.

[37] J. Luo, J. Fu, X. Kong, C. Gao, H. Ren, H. Shen, H. Xia, and S. Liu. 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16454–16463, 2022.

[38] X. Ma, S. Yong, Z. Zheng, Q. Li, Y. Liang, S.-C. Zhu, and S. Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

[39] R. OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2, 2023.

[40] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[41] Y. Pang, W. Wang, F. E. Tay, W. Liu, Y. Tian, and L. Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022.

[42] K. Papineni, S. Roukos, T. Ward, and W.-J. 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.

[43] M. Parelli, A. Delitzas, N. Hars, G. Vlassis, S. Anagnostidis, G. Bachmann, and T. 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.

[44] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

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

[46] Y. Shen, Z. Geng, Y. Yuan, Y. Lin, Z. Liu, C. Wang, H. Hu, N. Zheng, and B. Guo. V-detr: Detr with vertex relative position encoding for 3d object detection. *arXiv preprint arXiv:2308.04409*, 2023.

[47] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[48] O. Unal, C. Sakaridis, S. Saha, F. Yu, and L. Van Gool. Three ways to improve verbo-visual fusion for dense 3d visual grounding. *arXiv preprint arXiv:2309.04561*, 2023.

[49] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[50] T. Vu, K. Kim, T. M. Luu, T. Nguyen, and C. D. Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022.

[51] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9782–9792, 2021.

[52] Z. Wang, H. Huang, Y. Zhao, L. Li, X. Cheng, Y. Zhu, A. Yin, and Z. Zhao. 3drp-net: 3d relative position-aware network for 3d visual grounding. *arXiv preprint arXiv:2307.13363*, 2023.

[53] Z. Wang, H. Huang, Y. Zhao, L. Li, X. Cheng, Y. Zhu, A. Yin, and Z. Zhao. Distilling coarse-to-fine semantic matching knowledge for weakly supervised 3d visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2662–2671, 2023.

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

[55] Z. Wang, Z. Zhang, X. Cheng, R. Huang, L. Liu, Z. Ye, H. Huang, Y. Zhao, T. Jin, P. Gao, et al. Freebind: Free lunch in unified multimodal space via knowledge fusion. In *Forty-first International Conference on Machine Learning*.

[56] Z. Wang, Z. Zhang, L. Liu, Y. Zhao, H. Huang, T. Jin, and Z. Zhao. Extending multi-modal contrastive representations. *arXiv preprint arXiv:2310.08884*, 2023.

[57] Z. Wang, Z. Zhang, H. Zhang, L. Liu, R. Huang, X. Cheng, H. Zhao, and Z. Zhao. Omnibind: Large-scale omni multimodal representation via binding spaces. *arXiv preprint arXiv:2407.11895*, 2024.

[58] Z. Wang, Y. Zhao, X. Cheng, H. Huang, J. Liu, L. Tang, L. Li, Y. Wang, A. Yin, Z. Zhang, et al. Connecting multi-modal contrastive representations. *arXiv preprint arXiv:2305.14381*, 2023.

[59] T.-Y. Wu, S.-Y. Huang, and Y.-C. F. Wang. Dora: 3d visual grounding with order-aware referring. *arXiv preprint arXiv:2403.16539*, 2024.

[60] Y. Wu, X. Cheng, R. Zhang, Z. Cheng, and J. 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.

[61] R. Xu, X. Wang, T. Wang, Y. Chen, J. Pang, and D. Lin. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023.

[62] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1179–1189, 2023.

[63] J. Yang, X. Chen, S. Qian, N. Madaan, M. Iyengar, D. F. Fouhey, and J. Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. *arXiv preprint arXiv:2309.12311*, 2023.

[64] Z. Yang, S. Zhang, L. Wang, and J. 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.

[65] Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023.

[66] Z. Yin, J. Wang, J. Cao, Z. Shi, D. Liu, M. Li, X. Huang, Z. Wang, L. Sheng, L. Bai, et al. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *Advances in Neural Information Processing Systems*, 36, 2024.

 https://doi.org/10.52202/079017-3620

[67] H. You, H. Zhang, Z. Gan, X. Du, B. Zhang, Z. Wang, L. Cao, S.-F. Chang, and Y. Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023.

[68] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022.

[69] Z. Yuan, X. Yan, Y. Liao, Y. Guo, G. Li, S. Cui, and Z. 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.

[70] Z. Yuan, X. Yan, Y. Liao, R. Zhang, S. Wang, Z. Li, and S. 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.

[71] D. Zhang, D. Liang, H. Yang, Z. Zou, X. Ye, Z. Liu, and X. Bai. Sam3d: Zero-shot 3d object detection via segment anything model. *arXiv preprint arXiv:2306.02245*, 2023.

[72] H. Zhang, H. You, P. Dufter, B. Zhang, C. Chen, H.-Y. Chen, T.-J. Fu, W. Y. Wang, S.-F. Chang, Z. Gan, et al. Ferret-v2: An improved baseline for referring and grounding with large language models. *arXiv preprint arXiv:2404.07973*, 2024.

[73] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, and H. Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35:27061–27074, 2022.

[74] R. Zhang, L. Wang, Y. Qiao, P. Gao, and H. Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21769–21780, 2023.

[75] Y. Zhang, Z. Gong, and A. X. Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15225–15236, 2023.

[76] L. Zhao, D. Cai, L. Sheng, and D. 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.

[77] Y. Zhao, Z. Lin, D. Zhou, Z. Huang, J. Feng, and B. Kang. Bubogpt: Enabling visual grounding in multi-modal llms. *arXiv preprint arXiv:2307.08581*, 2023.

[78] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[79] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[80] Z. Zhu, X. Ma, Y. Chen, Z. Deng, S. Huang, and Q. 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.

Table 6: **Performance comparison on the validation set of ScanRefer [4].**

Method	Venue	Unique		Multiple		Overall	
		Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
ScanRefer [4]	ECCV20	76.33	53.51	32.73	21.11	41.19	27.40
TGNN [24]	AAAI21	68.61	56.80	29.84	23.18	37.37	29.70
SAT [64]	ICCV21	73.21	50.83	37.64	25.16	44.54	30.14
InstanceRefer [70]	ICCV21	75.72	64.66	29.41	22.99	38.40	31.08
3DVG-Transformer [76]	ICCV21	81.93	60.64	39.30	28.42	47.57	34.67
MVT [25]	CVPR22	77.67	66.45	31.92	25.26	40.80	33.26
3D-SPS [37]	CVPR22	84.12	66.72	40.32	29.82	48.82	36.98
ViL3DRel [7]	NeurIPS22	81.58	68.62	40.30	30.71	47.94	37.73
3DJCG [3]	CVPR22	83.47	64.34	41.39	30.82	49.56	37.33
D3Net [**?**]	ECCV22	–	72.04	–	30.05	–	37.87
BUTD-DETR [26]	ECCV22	84.2	66.3	46.6	35.1	52.2	39.8
HAM [5]	ArXiv22	79.24	67.86	41.46	34.03	48.79	40.60
3DRP-Net [52]	EMNLP23	83.13	67.74	42.14	31.95	50.10	38.90
3D-VLP [29]	CVPR23	84.23	64.61	43.51	33.41	51.41	39.46
EDA [60]	CVPR23	85.76	68.57	**49.13**	37.64	54.59	42.26
M3DRef-CLIP [75]	ICCV23	85.3	77.2	43.8	36.8	51.9	44.7
3D-VisTA [80]	ICCV23	81.6	75.1	43.7	39.1	50.6	45.8
ConcreteNet [48]	ECCV24	86.40	82.05	42.41	38.39	50.61	46.53
DOrA [59]	ArXiv24	–	–	–	–	52.80	44.80
Ours	–	**89.59**	**82.49**	47.78	**42.90**	**55.52**	**50.23**

A Implementation Details

For the 3D point cloud input, we utilize the 3D instance segmentation model Mask3D [45] to extract 100 object proposals. We then employ the pre-trained encoder Uni3D [78] to obtain 3D features and DINOv2 [40] for 2D features. Both the 3D-language projector $f_p(\cdot)$ and the 2D-language projector $f_v(\cdot)$ are three-layer MLPs. For the video input, we use the open-vocabulary video segmentation model DEVA [13] to extract object proposals, with an average object number of 48. We choose the Vicuna-7B-v1.5 model [14] as the language model, which is based on LLaMA 2 [47]. We fine-tune the language model using LoRA [22], with the rank set to 16. The base learning rate is set to 5e-6 with a cosine annealing schedule, and the batch size is 32. The training takes 2 epochs and the total training step is 3200. The entire training process takes approximately 8 hours on 4 NVIDIA A100 GPUs.

2D Encoder. We describe the method for deriving 2D representations from the projected input masks and multi-view images. The 2D model DINOv2 [40] computes a 257×1024 feature vector for each image, comprising a 1×1024 CLS feature and 256×1024 patch features. These patch features represent local areas in the images divided into 16×16 patches. For each object mask within an image, we extract patch features only where the patch intersects with the mask. We then average these extracted patch features along the patch axis to obtain a 1×1024 feature vector per image. For patch features sourced from multi-view images, we average them weighted by the mask size on each image. We apply this procedure to extract DINOv2 features from multi-view images for each object, thus generating the final object representations. In our ablation study, described in Section 4.3, we introduce a "Single" approach where we select the image with the largest mask for the current object and extract features solely from this image.

B Quantitative Results

For the five datasets, we employ evaluation metrics as proposed in their respective original publications. We compare our model against a comprehensive array of state-of-the-art methods, as detailed in Table 6 for ScanRefer [4], Table 7 for Multi3DRefer [75], Table 8 for Scan2Cap [12], Table 9 for ScanQA [2], and Table 10 for SQA3D [38].

C Qualitative Analysis

In this section, we perform a comprehensive qualitative analysis of ScanQA [2] and ScanRefer [4]. Additional analysis of other datasets will be included in the final version.

 https://doi.org/10.52202/079017-3620

Table 7: **Performance comparison on the validation set of Multi3DRefer [75].**

Method	Venue	ZT w/o D	ZT w/ D	ST w/o D		ST w/ D		MT		ALL	
		F1	F1	F1@0.25	F1@0.5	F1@0.25	F1@0.5	F1@0.25	F1@0.5	F1@0.25	F1@0.5
3DVG-Trans+ [76]	ICCV21	87.1	45.8	–	27.5	–	16.7	–	26.5	–	25.5
D3Net (Grounding) [**?**]	ECCV22	81.6	32.5	–	38.6	–	23.3	–	35.0	–	32.2
3DJCG (Grounding) [3]	CVPR22	**94.1**	**66.9**	–	26.0	–	16.7	–	26.2	–	26.6
M3DRef-CLIP [75]	ICCV23	81.8	39.4	53.5	47.8	34.6	30.6	43.6	37.9	42.8	38.4
Ours	–	90.3	62.6	**82.9**	**75.9**	**49.1**	**44.5**	**45.7**	**41.1**	**57.1**	**52.4**

Table 8: **Performance comparison on the validation set of Scan2Cap [12].**

Method	Venue	@0.25				@0.5			
		C	B-4	M	R	C	B-4	M	R
Scan2Cap [12]	CVPR21	56.82	34.18	26.29	55.27	39.08	23.32	21.97	44.48
3DJCG [3]	CVPR22	64.70	40.17	27.66	59.23	49.48	31.03	24.22	50.80
X-Trans2Cap [69]	CVPR22	61.83	35.65	26.61	54.70	43.87	25.05	22.46	45.28
D3Net [**?**]	ECCV22	–	–	–	–	62.64	35.68	25.72	53.90
3D-VLP [29]	CVPR23	70.73	41.03	28.14	59.72	54.94	32.31	24.83	51.51
Vote2Cap-DETR [8]	CVPR23	71.45	39.34	28.25	59.33	61.81	34.46	26.22	54.40
3D-VisTA	ICCV23	71.0	36.5	28.4	57.6	66.9	34.0	27.1	54.3
LL3DA	CVPR24	74.17	**41.41**	27.76	59.53	65.19	36.79	25.97	55.06
LEO	ICML24	–	–	–	–	68.4	36.9	27.7	57.8
Vote2Cap-DETR++ [9]	T-PAMI24	76.36	41.37	28.70	60.00	67.58	**37.05**	26.89	55.64
Ours	–	**81.94**	38.23	**29.01**	**60.57**	**77.19**	36.34	**28.01**	**58.12**

C.1 3D Question Answering

We present four evaluation results for 3D question answering on ScanQA [2] dataset, as shown in Figure 5. In this dataset, both the input and the output do not contain object referencing. This dataset does not include object referencing in either the input or output. Example (a) necessitates the model's ability to perceive an object's appearance, specifically its color. Example (b) demands that the model identify a target object based on a descriptive prompt. Example (c) involves the model describing the position of a target object, while example (d) tests the model's capability to count objects. Our model demonstrates relatively high performance on the first three types of tasks but often struggles with the fourth, particularly when the count of target objects is large. Accurately perceiving and localizing each object is essential for the counting task; failure to do so results in incorrect answers. This challenge persists in both our method and previous methods. Moreover, the inferior annotation quality within the ScanQA dataset exacerbates this issue. For instance, the question in example (d), "How many black chairs are on the right?" lacks a precise definition of "right", leading to potential confusion for the model.

C.2 3D Visual Grounding

We present six evaluation results for 3D visual grounding on ScanRefer [4] dataset, as illustrated in Figure 6. This task challenges the model to localize a target object based on a descriptive prompt. For simpler scenarios, such as those in examples (a) and (b), our model performs adequately. However, it struggles with the remaining four examples for various reasons.

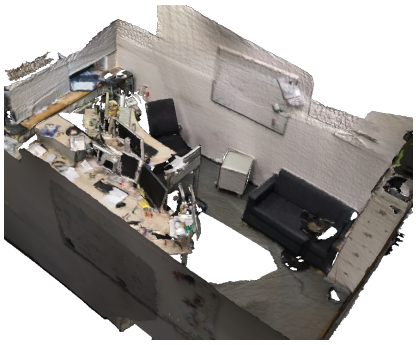
In example (c), the model is tasked with identifying a chair "against the wall" but erroneously selects a chair that is not positioned as described. This highlights a deficiency in our model's understanding of interior structural elements like walls, ceilings, and floors. Despite the presence of segmented annotations for these surfaces, they are typically not utilized in training because they are not considered objects per se. This limitation is likely shared by many current methods. Future work is necessary to enhance the model's recognition of these elements, given their significance in comprehending the entirety of a 3D scene. In example (d), the challenge involves identifying two pillows "placed on the armchair', one black and the other white. The model correctly locates a pillow on the armchair but fails to distinguish it by color. Example (e) presents a scenario where the model confuses a window for a door, likely due to their similar appearances and the often incomplete nature of the input point cloud. In example (f), the model's selection meets the description, illustrating a flaw in the ScanRefer dataset annotations: some descriptions may correspond to multiple objects, rendering them ineffective for evaluating the visual grounding of a single object.

Table 9: **Performance comparison on the validation set of ScanQA [2].**

Method	Venue	EM@1	B-1	B-2	B-3	B-4	ROUGE-L	METEOR	CIDEr	SPICE
ScanQA [2]	CVPR22	21.05	30.24	20.40	15.11	10.08	33.33	13.14	64.86	13.43
3D-VLP [29]	CVPR23	21.65	30.53	21.33	16.67	11.15	34.51	13.53	66.97	14.18
3D-LLM [21]	NeurIPS23	20.5	39.3	25.2	18.4	12.0	35.7	14.5	69.4	–
LL3DA [6]	CVPR24	–	–	–	–	13.53	37.31	15.88	76.79	–
LEO [23]	ICML24	–	–	–	–	11.5	39.3	16.2	80.0	–
Scene-LLM [18]	ArXiv24	**27.2**	**43.6**	26.8	19.1	12.0	40.0	16.6	80.0	–
Ours	–	21.62	43.20	**29.06**	**20.57**	**14.31**	**41.56**	**18.00**	**87.70**	**20.44**

Table 10: **Performance comparison on the test set of SQA3D [38].**

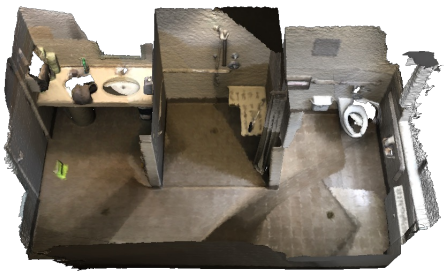
Method	Venue	Test set						Avg.
		What	Is	How	Can	Which	Others	
SQA3D [38]	ICLR23	31.64	63.80	46.02	69.53	43.87	45.34	46.58
3D-VisTA [80]	ICCV23	34.8	63.3	45.4	69.8	47.2	48.1	48.5
Scene-LLM [18]	ArXiv24	40.9	**69.1**	45.0	**70.8**	47.2	52.3	54.2
Ours	–	**45.38**	67.02	**52.04**	69.52	**49.85**	**54.95**	**54.57**



Q: What color is the filing cabinet on the floor?

A: White. ✔

(a)



Q: What is attached to the wall on the left, just above the sink counter?

A: Paper towel dispenser. ✔

(b)

Q: In what part of the room is the brown piano located?

A: Left corner. ✔

(c)

Q: How many black chairs are on the right?

A: Eleven. ✖ (**GT:** Twelve.)

(d)

Figure 5: **Visualization results of 3D question answering on ScanQA [2].**

https://doi.org/10.52202/079017-3620

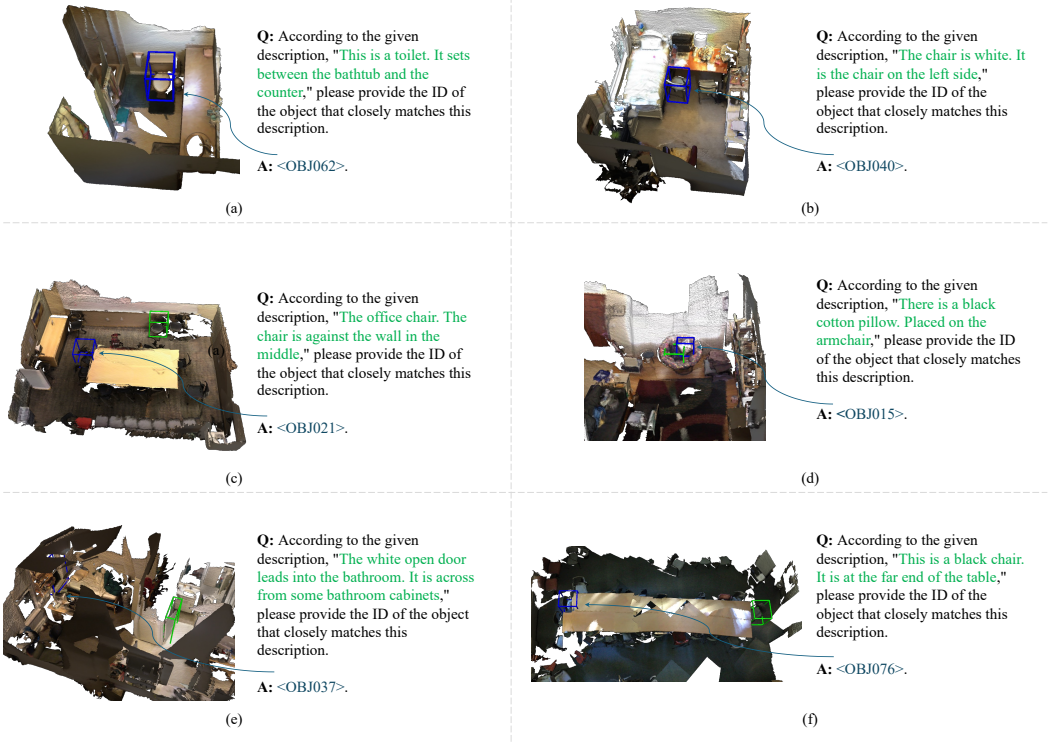


Figure 6: **Visualization results of 3D visual grounding on ScanRefer [4].** The predicted blue box is transformed from the segmented point cloud of the predicted object (ID). The green ground truth box is provided when the prediction is wrong.

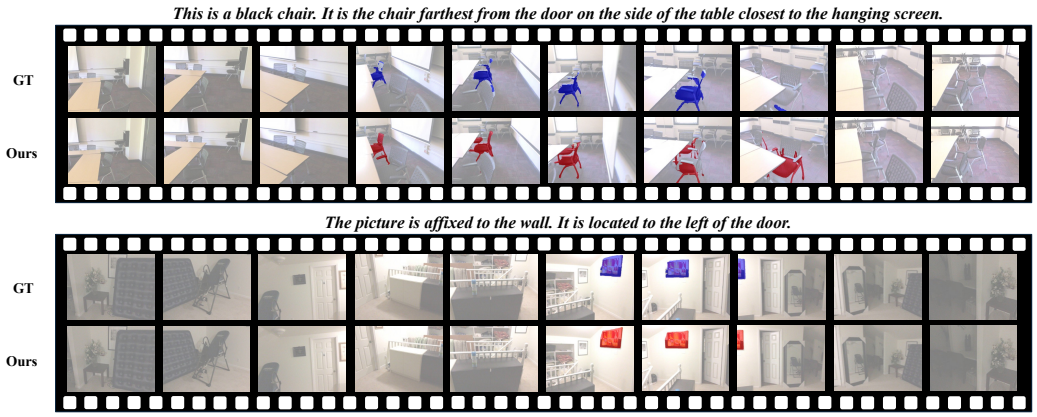


Figure 7: **Additional visualization results of video grounding for video input.**

Table 11: **Comparison of input/output formats.** The use of object identifiers make it possible to solve tasks with single or even multiple object reference in the input/output.

Query	Answer	Task	Expert Models	3D MLLMs	Ours (Example)
Normal text	Normal text	VQA	✓ 3D-VisTA, 3D-VLP ✗ 3DJCG, M3DRef, MVT	✓ 3D-LLM, Chat-3D, LL3DA, LEO ✗ LLM-Grounder	Q: How many trash cans are there in the scene? A: Three.
Normal text	Referring to single object	Visual Grounding	✓ 3D-VisTA, 3D-VLP, 3DJCG, M3DRef, MVT	✓ 3D-LLM, LEO, LLM-Grounder ✗ Chat-3D, LL3DA	Q: Find the trash can that is next to the whiteboard. A: <OBJ034>.
Referring to single object	Normal text	Object Captioning	✓ 3D-VisTA, 3D-VLP, 3DJCG ✗ M3DRef, MVT	✓ 3D-LLM, Chat-3D, LL3DA, LEO ✗ LLM-Grounder	Q: Describe <OBJ004> in the scene. A: This is a yellow rectangle table with four legs. It is the closest table to the door.
Normal text	Referring to multiple objects	Multi-object Visual Grounding	✓ M3DRef ✗ 3D-VisTA, 3D-VLP, 3DJCG, MVT	✗	Q: Find all the objects that satisfy the description "This is a lamp. It is bronze in color and sits on a table". A: They are <OBJ006>, <OBJ007>, and <OBJ009>.
		Grounded Scene Captioning	✗	✗	Q: Describe the scene. A: ... There is a table (<OBJ004>) with two lamps (<OBJ007> and <OBJ008>) on it. …
Referring to multiple objects	Normal text	Multi-object Summary	✗	✗	Q: What are the potential uses for <OBJ003> and <OBJ004> in the room? A: They are two long brown tables, which are likely intended for use in a collaborative workspace.
Referring to multiple objects	Referring to single object	Multiple Choice QA	✗	✗	Q: Among <OBJ011>, <OBJ014>, and <OBJ012>, which one is the closest to the door? A: <OBJ011>.
Referring to multiple objects	Referring to multiple objects	Complex Planning / …	✗	✗	Q: I want to remove the tables (<OBJ004> and <OBJ022>) from the room. What should I do? A: First, remove the two lamps (<OBJ007> and <OBJ008>) from table (<OBJ004>) and the backpack (<OBJ009>) from table (<OBJ022>). …

D Input/Output Format Comparison

We present comparative examples of input/output formats in Figure 11. We juxtapose our model against leading **expert models** (3D-VisTA [80], 3D-VLP [29], 3DJCG [3], M3DRef [75], and MVT [25]) as well as **3D MLLMs** (3D-LLM [21], Chat-3D [54], LLM-Grounder [63], LL3DA [6], and LEO [23]). We enumerate several common tasks for varied combinations of input and output formats and provide examples of interactions using object identifiers with our model. It is necessary to acknowledge that some responses shown here are not directly produced by our model due to a lack of adequate data for fine-tuning on these tasks, such as multi-object summaries, multiple-choice QA, and complex planning. The comparison underscores the potential for employing object identifiers to address complex tasks involving single or multiple object references in the input/output, representing a significant enhancement over previous methods restricted to simple tasks with basic formats.

E Limitation and Societal Impact

Limitation. The primary limitation of our method stems from its reliance on pre-trained foundation models, including 2D/3D detectors and 2D/3D encoders. In our experiments, we froze these models under the presumption of their robust performance. Yet, they occasionally produce incorrect results, as evidenced by the failure cases detailed in the supplementary material. To establish an end-to-end pipeline and enhance model performance further, future work will involve integrating these foundation models into the entire training process.

Another significant challenge is the scarcity of data. While the development of 2D vision-language models has benefited from the availability of millions of image-text pairs for pre-training, the 3D-language domain grapples with a dearth of corresponding data, adversely affecting the alignment between 3D and language spaces. This issue is particularly acute in 3D scene understanding, where the limited availability of scene-language pairs restricts training by failing to provide adequate spatial relationship data. Despite our model's impressive performance in various evaluations, it occasionally misclassifies objects, notably in underrepresented classes such as "hair dryer" and "soap dish." Future initiatives should aim to enhance data volume to bolster the 3D MLLM's scene understanding capabilities.

Societal Impact. Our model has achieved consistent performance improvements in multiple tasks related to 3D indoor scene understanding, which is beneficial and potentially applicable to downstream

https://doi.org/10.52202/079017-3620

Table 12: **Prompt templates for different tasks.** <Description> is replaced by an object's description in ScanRefer/Multi3DRefer/Scan2Cap. <Question> and <Answer> denotes the question and answer in ScanQA/SQA3D. <Situation> is the situation in SQA3D.

Single-object Visual Grounding (ScanRefer): **User**: What is the ID of the object that matches the description "<Description>"? **Assistant**: <OBJXXX>.
Multi-object Visual Grounding (ScanRefer): **User**: Are there objects described as "<Description>"? If there are, please provide the IDs for those objects. **Assistant**: Yes. <OBJXXX>, <OBJXXX>, and <OBJXXX>. (for matched cases) **Assistant**: No. (for unmatched cases)
Dense Captioning (Scan2Cap): **User**: Provide a detailed description of the appearance of <OBJXXX> before analyzing its spatial connections with other elements in the scene. **Assistant**: <Description>
Visual Question Answering (ScanQA): **User**: <Question> The answer should be a phrase or a single word. **Assistant**: <Answer>
Situated Question Answering (SQA3D): **User**: <Situation> <Question> The answer should be a phrase or a single word. **Assistant**: <Answer>

applications. However, due to the training data not covering all possible scenarios, the model may experience hallucinations during prediction, thereby posing some risks in system applications.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contribution of building a 3D MLLM with object identifiers is clearly claimed in both abstract and introduction (Section 1).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix E.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/079017-3620

Justification: The paper focuses on practical aspects of training neural networks, and no new theoretical results are included.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The implementation details in given in Appendix A. The model design, training data, input/output format is illustrated in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code in the supplementary material with a README which includes instructions for preparing data, training, and evaluating. Also, we will release the code to public after careful organization.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the implementation details in Appendix A. The training and test details are included in the code and instructions in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We keep the same basic training settings (learning rate, batch size, training steps, random seed) across all the experiments reported in Section 4 to produce a meaningful ablation study. However, we do not report error bars due to the limited time and resources available for training the LLM multiple times.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

 https://doi.org/10.52202/079017-3620

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the required hardware resources (GPU type, GPU number, and execution time) in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We've reviewed the NeurIPS Code of Ethics and we make sure the research conforms with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impact in Appendix E.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The model in our paper is based on an open-source large language model, and our fine-tuning procedure does not introduce additional risks. Users are encouraged to adhere to the safeguards provided by the open-source LLM.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators of the used datasets and models in our paper are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

https://doi.org/10.52202/079017-3620

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code/model provided in the supplementary material is documented with detailed instructions.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-3620