

PARIS3D: Reasoning-based 3D Part Segmentation Using Large Multimodal Model

Amrin Kareem¹, Jean Lahoud¹, and Hisham Cholakkal¹

Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE

Abstract. Recent advancements in 3D perception systems have significantly improved their ability to perform visual recognition tasks such as segmentation. However, these systems still heavily rely on explicit human instruction to identify target objects or categories, lacking the capability to actively reason and comprehend implicit user intentions. We introduce a novel segmentation task known as reasoning part segmentation for 3D objects, aiming to output a segmentation mask based on complex and implicit textual queries about specific parts of a 3D object. To facilitate evaluation and benchmarking, we present a large 3D dataset comprising over 60k instructions paired with corresponding ground-truth part segmentation annotations specifically curated for reasoning-based 3D part segmentation. We propose a model that is capable of segmenting parts of 3D objects based on implicit textual queries and generating natural language explanations corresponding to 3D object segmentation requests. Experiments show that our method achieves competitive performance to models that use explicit queries, with the additional abilities to identify part concepts, reason about them, and complement them with world knowledge. Our source code, dataset, and trained models are accessible at this url.

Keywords: 3D · Vision-Language Models · Reasoning

1 Introduction

The rapid advancements in 3D data capture technologies, including LIDARs and RGB-D cameras, have led to a growing demand for automated analysis of 3D point clouds. 3D semantic segmentation, the process of automatically assigning predefined semantic labels to each point in a cloud, is crucial for enabling complex tasks such as scene understanding. Similarly, 3D part segmentation involves further segmenting object instances into their components, such as identifying the handle of a pot or the lid of a bottle. These tasks find applications in various fields including autonomous vehicles, mobile robotics, industrial automation, augmented reality, and medical imagery analysis. While recent advancements in pre-trained 3D representations and the introduction of various 3D datasets have significantly improved 3D perception, the capacity for nuanced reasoning in 3D contexts remains limited. This limitation is primarily due to the lack of comprehensive datasets for reasoning and describing 3D scenes and objects.

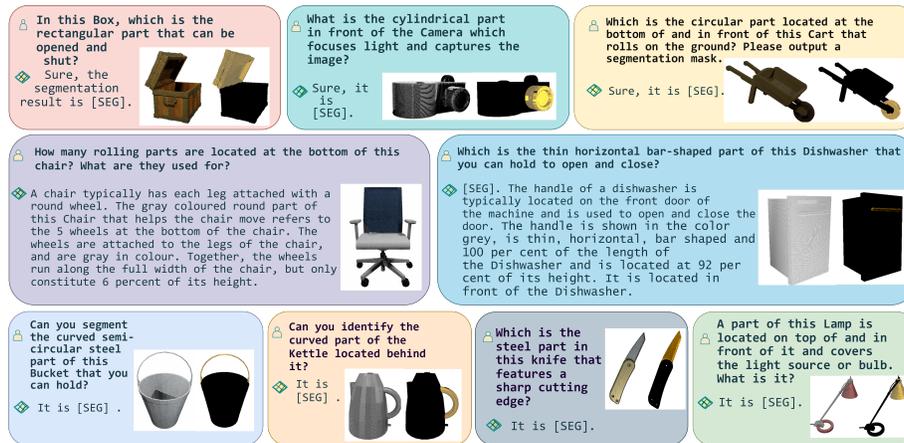


Fig. 1: Capabilities of PARIS3D. Parts of 3D objects are segmented based on reasoning, shape, location, material, colour, and concept instructions. Additionally, for the segmentations, PARIS3D can explain why it chose that region, or describe 3D objects with respect to their parts. The original point clouds are on the left. The segmented parts are shown to the right, highlighted in golden colour.

In contrast, 2D images accompanied by textual descriptions have contributed to significant progress in large-scale image language models. Recent methods have emerged, performing segmentation based on 2D images projected from 3D scenes. However, they lack the ability to reason about object concepts and their 3D properties, functions, or parts in a conversational manner.

Here, we define a model as having reasoning capabilities when (i) it can understand implicit instructions, such as referencing properties of an object or its parts without explicit articulation, and (ii) explain or justify its responses, whether generated text or predicted segmentation masks. Such reasoning ability is one of the fundamental cognitive skills possessed by humans, essential for daily activities ranging from locating items to manipulating tools. Previous 3D vision methods are capable of exploring the semantics of a given environment based on instructions containing pre-defined category names. These explicit categories are limiting and do not address cases where instructions can be implicit and may contain complex queries. For example, an automated system might need to find the part of the kettle that can be opened to pour water, without knowing the explicit category name "lid". To this end, we introduce the 3D reasoning part segmentation task, which aims at segmenting object parts based on general queries, including color, 3D information, and implicit functionalities.

In this work, we introduce a novel task termed reasoning part segmentation in 3D. This task involves generating a part segmentation mask for a 3D object based on implicit textual queries requiring complex reasoning. These queries go beyond simple references and encompass intricate descriptions, demanding sophisticated reasoning or worldly knowledge. For example, knowing where to hold a kettle or

which part of a bottle to open in a fine-grained manner. To address this challenging task, we propose PARIS3D (**R**eason**I**ng-based **3D P**Art **S**egmentation), a multi-modal Large Language Model (LLM) capable of reasoning on user input text, predicting 3D part segmentation masks, and providing explanations for the model’s response. As depicted in Figure 1, PARIS3D adeptly navigates diverse scenarios, encompassing complex 3D reasoning, material, color, shape, and location-based knowledge, providing explanatory responses and detailed descriptions. Additionally, to validate the effectiveness of PARIS3D and support future research, we establish an evaluation benchmark and a dataset named RPSeg3D for reasoning 3D part segmentation. Our RPSeg3D dataset comprises 2624 3D objects and over 60k instructions, providing persuasive evaluation metrics for the task.

In summary, our key contributions are the following:

- We introduce the reasoning part segmentation task for 3D objects, emphasizing the necessity of reasoning capabilities which is crucial for the development of intelligent perception systems.
- We provide a comprehensive dataset named RPSeg3D for reasoning-based 3D part segmentation, comprising over 2624 3D objects and 60k instructions, serving as a useful resource for future research.
- We present PARIS3D, a novel approach for 3D part segmentation, and further improve its capabilities by fine-tuning on our RPSeg3D dataset.

2 Related Work

3D Semantic Segmentation. The challenge of understanding and reasoning within 3D environments has been an ongoing research focus. The goal of 3D semantic segmentation is to acquire semantic predictions for each point in a cloud. Notable contributions include point-based approaches [19, 38], methods which incorporate intricately crafted point convolution methods [46, 50], voxel-based strategies [5, 10], including those that employ 3D sparse convolutions [11] for generating point-wise segmentation outcomes, as well as transformer-based techniques [23]. Multi-view semantic segmentation methods such as DeepViewAgg [41], Diffuser [22, 34], 3D-CG [15], 3D-CLR [16] in 3D vision concentrate on improving representation learning by generating 2D renderings from 3D under multiple view points. These works have shown the effectiveness of multi-view representations in enhancing the performance and robustness of various 3D tasks. Nonetheless, these methods rely on a predefined set of semantic labels, whereas we focus in our proposed method on responding to complex reasoning-based queries and explaining them.

Large Multimodal Models. Extensive research on Large Language Models (LLMs) has demonstrated reasoning capabilities, prompting an exploration into extending these skills into the visual domain through Large Multimodal Models. LMMs are a highly adaptable and versatile means to perform tasks requiring language and vision capabilities. Prominent models such as BLIP-2 [26], LLaVA [29] and MiniGPT-4 [55] generally employ a dual-phase training process, aligning

Method	Input		Task		
	Point Cloud	Reasoning Query	Conversation	Segmentation	Explanation
SQA3D [33], 3D-VisTA [56]	✓	✓	✗	✗	✗
ViewRefer [13], Point-Bind [12]	✓	✗	✗	✗	✗
3D-OVS [30]	✗	✗	✗	✓	✗
OpenMask3D [45]	✓	✗	✗	✓	✗
PLA [6], OpenScene [37]	✓	✗	✗	✓	✗
Chat-3D [49]	✓	✓	✓	✗	✓
M3DBench [27]	✓	✗	✓	✗	✗
LLM-Grounder [52]	✓	✗	✓	✗	✓
3D-LLM [17]	✓	✓	✓	✗	✓
LL3DA [4], PointLLM [51]	✓	✗	✓	✗	✓
PARIS3D	✓	✓	✓	✓	✓

Table 1: Comparison of recent 3D segmentation models and Large Multi-modal Models (LMMs) emphasizing their capabilities for 3D reasoning and conversations. Reasoning query means the model is asked to self-reason a task and either output text or perform an action. Segmentation highlights models that can respond with 3D segmentation masks, and Conversation represents models that can provide a conversation-style answer to the user. Among these, our proposed PARIS3D stands out with comprehensive 3D understanding and reasoning, segmentation in response to natural language queries, and conversational capabilities.

visual representations with the linguistic embeddings of LLMs through extensive image-text and video-text datasets [2, 3, 28, 35, 42–44]. Recent efforts have focused on the convergence of multi-modal LLMs with vision tasks, where VisionLLM [48] provides a versatile interaction interface for a spectrum of vision-centric tasks through instruction tuning. Yet, it does not fully leverage the complex reasoning potential of LLMs. GPT4RoI [53] integrates spatial boxes as inputs and trains models on region-text pairs, showcasing a novel approach. Our method aims to benefit from these advances in the LMM space by merging the vision-language abilities of LMMs and the reasoning of LLMs in a novel 3D perception task.

Language Instructed 3D Tasks. The integration of point clouds with natural language processing has widespread implications, drawing considerable interest in the realms of 3D scene understanding. This fast-growing field promises enhancements in human-robot interaction, metaverse, robotics, and embodied intelligence. Central to the dialogue systems designed for 3D environments are two critical capabilities: perception within three-dimensional spaces and reasoning. Recently, there has been a rise in the number of tasks uniting 3D scenes and language, such as 3D captioning, 3D question answering, 3D situated question answering, embodied Q and A, planning, navigation, 3D assisted multi-turn dialogue, 3D object detection, and scene description. We divide the 3D perception task models into 3 categories (see Table 1. Separated by dotted lines.). The first encompasses models that perform tasks like 3D captioning, situated question answering, and visual grounding [33, 56], visual grounding [14]. These models are capable of providing a word or phrase as its text output. The second category has 3D semantic segmentation models that output 3D segmentation masks. 3D-

OVS [30], Openmask3D [45], OpenScene [37], PLA [7], perform open-vocabulary semantic segmentation for 3D scenes. These methods, on the other hand, cannot provide a conversational output to a user query or explain the reasoning for their tasks. The third comprises models that employ an LLM and perform visual perception tasks such as captioning, scene understanding, and visual grounding, providing conversational outputs [4, 12, 17, 27, 49, 51, 52]. However, they do not perform fine-grained semantic segmentation or reasoning-based 3D vision tasks. As such, we identify a gap in performing 3D segmentation in response to complex natural language prompts instead of a single phrase or word. Distinct from the existing works in this domain, our research is committed to: (1) streamlining the integration of 3D segmentation capabilities into multi-modal LLMs, and (2) enhancing current perception systems with 3D reasoning-based segmentation and explanation abilities, thereby pushing the boundaries of what is achievable in the intersection of language and visual understanding.

3 Reasoning 3D Part Segmentation

Semantic segmentation involves assigning a semantic label to each geometric primitive, such as points [38], voxels [9], or superpoints [25]. In part segmentation, object instances are decomposed into their components. Given a coloured point cloud P , the goal of a 3D segmentation model is to predict its label for each point. However, in our reasoning segmentation task, we go further to output a 3D segmentation mask M , given an input point cloud P and an implicit query text instruction x_{txt} . The task shares a similar formulation with the referring segmentation task [20], with an additional challenge for the model to reason about the fine-grained parts in response to implicit queries and output the corresponding segmentation mask. The complexity of the query text in reasoning part segmentation is a key differentiator. Instead of providing the names of the parts, the query text may include more intricate expressions that involve an understanding of structures, geometries, and semantics of 3D objects. By introducing this task, we aim to bridge the gap between user intent and system response, enabling more intuitive and dynamic interactions in 3D object perception.

3.1 Our RPSeg3D Dataset

Considering the unavailability of established datasets and evaluation benchmarks in the literature, we introduce a dataset, named RPSeg3D, specifically designed for the reasoning 3D part segmentation task. Our dataset comprises 2624 3D objects and over 60k instructions. We use 718 objects and their corresponding instructions as the train set, and the remaining 1906 objects along with their instructions are used for testing. For reliable and fair assessment, we have aligned the 3D objects with those from PartNet-Ensemble [31], annotating them with implicit text instructions and using ground truth labels to generate high-quality target masks. To generate the text instructions corresponding to each 3D object part, we prepare a set of templates, for example, "Which part of this object

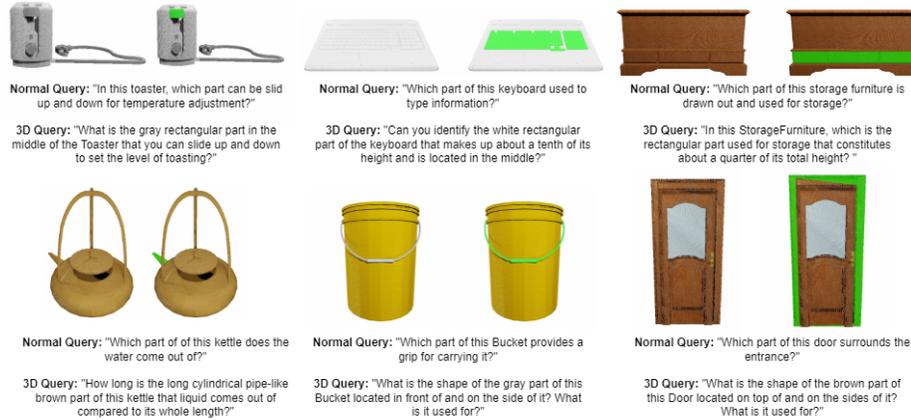


Fig. 2: Examples of the annotated object-instruction pairs for training with two types of queries. On the left is one view of the rendered image from the original point cloud. On the right is the corresponding ground truth segmentation mask, shown in green.

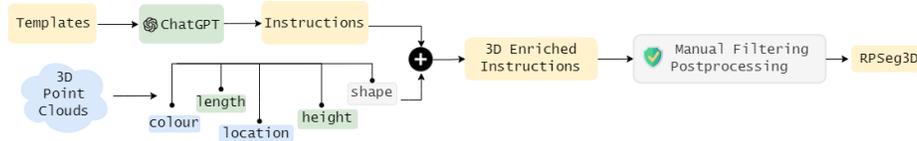


Fig. 3: Preparing the instructions of RPSeg3D. Simple templates are provided to GPT-3.5, which populates them with part information-related segmentation instructions. In parallel, colour, shape, location, and dimension-related data is extracted from 3D point clouds. Enriching the instructions with this information and manually checking them for inaccuracies, we obtain the RPSeg3D dataset for part segmentation.

<does this function/looks like this>?". We leverage GPT-3.5 [36] for building instructions using these templates by supplying information about the part and rephrasing. We also extract 3D information from the point cloud, thus designing the instructions to cover relations, dimensions (length, height), comparisons, colour, texture features, object concepts, and functions. This was further verified manually to avoid inaccurate prompts at test time. The steps are illustrated in Figure 3. To cover 3D object segmentation tasks effectively, our text instructions consist of two types: 1) normal queries; 2) 3D queries; as illustrated in Figure 2. The dataset is partitioned into train and test splits, containing 718 and 1906 3D objects and over 16k and 47k instructions, respectively. As the primary purpose of the dataset is evaluation, the testing set includes a larger number of instructions.

3.2 Reasoning 3D Part Segmentation Architecture

Our method takes as input a dense and coloured 3D point cloud of an object. One of the common methods of 3D analysis is predominantly using point clouds

to represent 3D data. However, this contrasts with human spatial reasoning processes. Humans typically engage with their surroundings through active exploration, synthesizing perspectives from multiple vantages to form an integrated 3D understanding, rather than processing a 3D environment at one go. Our approach advocates for 3D reasoning derived from multi-view imagery. This approach also benefits from the large-scale 2D pretraining available in vision-language models, similar to previous methods that have taken advantage of pre-trained vision-language models for 3D vision tasks. Thus, we render multiple images x_{img} from K predefined camera poses by rasterization. The camera poses cover all parts of the object since they are uniformly distributed around the input point cloud. Given a complex text instruction x_{txt} along with the images, we feed them into a multimodal LLM, denoted by F , and the visual backbone, F_{enc} . F outputs a text response \hat{y}_{txt} . In parallel, F_{enc} extracts the visual embeddings, f , from each of the input images x_{img} . The formulation is as follows:

$$\hat{y}_{txt} = F(x_{img}, x_{txt}), f = F_{enc}(x_{img}) \quad (1)$$

Whenever the LLM is expected to yield a binary segmentation mask, the resultant text output, denoted as \hat{y}_{txt} is required to an extra token, which sends a request for the segmentation output. Following this, the embedding at the last layer, \hat{h}_{seg} , corresponding to the additional segmentation token is extracted. This embedding is subsequently processed through an MLP (Multilayer Perceptron) projection layer, represented as γ , to derive h_{seg} and then input to the mask decoder F_{dec} as its prompt embedding, thereby tightly coupling the text output from F and the mask predicted by the decoder. Hence, the mask decoder takes the visual features f along with $\langle \text{SEG} \rangle$ based prompt embeddings to perform segmentation, generating the final segmentation mask, \hat{M} similar to [24]. The process is as follows:

$$h_{seg} = \gamma(\hat{h}_{seg}) \quad (2)$$

The architectural specifics of the decoder, F_{dec} , are in accordance with [21].

$$\hat{M} = F_{dec}(h_{seg}, f) \quad (3)$$

Once the K segmentations are obtained from each camera angle, a 3D semantic voting module computes scores and assigns the semantic labels for each part. We have K semantic segmentation masks M_k , where k is the view from which the image was rendered from the point cloud. We aggregate the masks from multiple views and lift them to obtain a three-dimensional semantic segmentation of the original point cloud. To achieve this, we segment the input point cloud P into a set of superpoints SP_i , similar to [25]. Superpoints refer to a way of representing large 3D point clouds as a collection of interconnected geometrically simple shapes. This representation is advantageous because (i) it considers complete object parts as a whole, making them easier to identify, (ii) this method can provide a detailed description of the relationship between adjacent objects, and (iii) the size of superpoints is determined by the number of simple structures in a

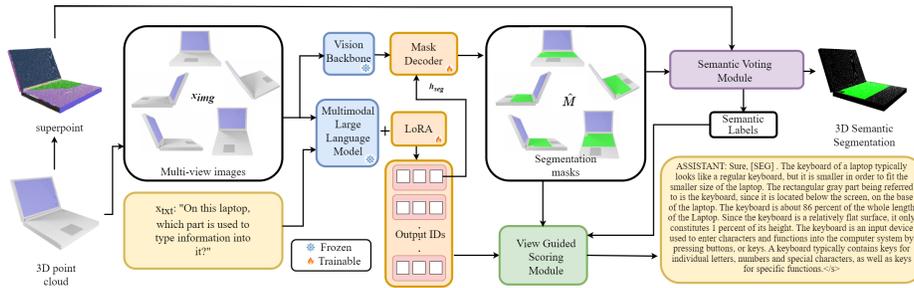


Fig. 4: Overview of the proposed reasoning-based 3D part segmentation approach named PARIS3D. It comprises four subsequent steps: **(i)** The 3D point cloud is rendered into K multi-view images x_{img} using a renderer. **(ii)** These images are passed through a frozen vision backbone (F_{enc}) and multi-modal large language model (F) of the reasoning module. F also accepts the text query x_{txt} , and produces text outputs corresponding to each view. **(iii)** The decoder decodes the final layer embedding which contains the extra token, thus producing K segmentation masks. **(iv)** Finally, a mask to 3D segmentation algorithm lifts the projections back into 3D and a view-guided scoring module is used to obtain the final text response.

scene, rather than the total number of points, making the representation several orders of magnitude smaller, thus more efficient to work with. Each superpoint contains points with similar normals and colours, suggesting they belong to the same instance. This superpoint-based part labelling not only conserves computational resources but also potentially improves performance by leveraging 3D priors. In the 3D semantic voting step, we assign semantic labels for each superpoint through a voting mechanism similar to [31]. This technique leverages information from multiple views and the superpoints, so that even if one mask covers irrelevant points, the aggregation of masks from all views counters the effect of such errors. For a superpoint SP_i and a part category j , a score $s_{i,j}$ is computed reflecting the proportion of visible points present in the predicted segmentation masks of category j in each view:

$$s_{i,j} = \frac{\sum_k \sum_{p \in SP_i} [V_k(p)] [\exists b \in M_k^j : I_m(p)]}{\sum_k \sum_{p \in SP_i} [V_k(p)]} \quad (4)$$

Here, $[\cdot]$ is the Iverson bracket, which turns true predicates into 1 and false into 0. M_k^j , represents the predicted mask of category j in view k . $V_k(p)$ indicates the visibility of 3D point p in the view k . $I_m(p)$ checks if the projection of point p in view k falls within the mask m . The superpoint is then assigned the semantic label of the part category with the highest score.

To provide the most relevant text explanation corresponding to the 3D masks, we propose a view-guided scoring module. It assigns a score to each view based on its correspondence with the final semantic labels. The best text explanation is chosen based on the highest score. Specifically, for each view, we compute the intersection over union (IoU) of the mask with the projections of the final 3D semantic labels output by the model, and the corresponding text explanation is

saved. This is called the view-guided score of the explanation. At the end of the score computation for all the masks, we choose the text explanation with the highest score, and in turn, the most correspondence with the output labels.

3.3 Training the PARIS3D Architecture

Training Data Formulation The training data consists of 718 3D objects rendered into multiview images, resulting in over 16k image-instruction pairs. Out of the 718, 360 objects with their instructions are used for training the model and 358 objects with their instructions are used for validation. Each image is provided with an annotation file, which has instructions corresponding to the 3D point cloud that the image came from, the name of the image, and its ground truth mask. Generation of instructions follows the same steps as Section 3.1 illustrated in Figure 3. Images in the training set may have more than one instruction in its "instruction" field. This is helpful to introduce diversity as users may randomly select one as the reasoning query during training, thus obtaining a better model.

Distillation-Based Explanation Refining Training the multiview model requires explanatory data as well as training image-instruction pairs. To build the explanatory data, we built an annotation pipeline using a distillation approach. Using the multi-view images as input to the teacher model [24], we generate explanations for each part of the object. These explanations serve as our pseudolabels which we use as ground truth explanations of the student model. We augment these annotations with 3D features extracted from the 3D point clouds from which the images were rendered. The 3D feature-augmented explanations contain appropriate responses and critical elements differentiating object parts from each other such as location, size, shape, material, and colour.

Objective We leverage LoRA [18] to perform efficient fine-tuning of the pre-trained multi-modal LLM F [29] to retain its generalization capability. We completely freeze the vision backbone F_{enc} . The decoder F_{dec} is fully fine-tuned. The word embeddings of the LLM and the projection layer of γ are also trainable, allowing the model to learn the specific meanings and semantic concepts of parts. The model is trained end-to-end using the text generation loss L_{txt} and the segmentation mask loss L_{mask} . The overall objective is given as:

$$L = \lambda_{txt}L_{txt} + \lambda_{mask}L_{mask} \quad (5)$$

Here, L_{mask} encourages the model to produce high-quality segmentation results. It is a combination of per-pixel binary cross-entropy (BCE) loss and DICE loss, with corresponding loss weights λ_{bce} and λ_{dice} , given by:

$$L_{mask} = \lambda_{bce}BCE(\hat{M}, M) + \lambda_{dice}DICE(\hat{M}, M) \quad (6)$$

where y_{txt} and M are the ground-truth targets. L_{txt} is the auto-regressive cross-entropy loss for text generation. It is computed as:

$$L_{txt} = CE(\hat{y}_{txt}, y_{txt}) \quad (7)$$

4 Experiments

We perform quantitative and qualitative evaluation of PARIS3D on our dataset for reasoning-based semantic segmentation.

Implementation Details and Metrics For the experiments, we follow [24] where the multi-modal LLM F is LLaVA-13B-v1-1 [29] and the vision backbone F_{enc} is the ViT-H SAM. The projection layer of γ is an MLP with channels of [256, 4096, 4096]. We use our dataset RPSeg3D which contains coloured point clouds and rendered 2D images of them. Using Pytorch3D [40], each input point cloud is rendered into $K = 10$ colour images. The fine-tuning scripts for the LLaVA and SAM architecture are based on DeepSpeed [1] engine. We adopt the settings of [24] for the optimizer (AdamW [32]) and its learning rate (0.0003). Similar steps are followed for the learning rate scheduler (WarmupDecayLR), text generation loss $\lambda_{textgen}$ weight (1.0) and the mask loss λ_{mask} weight (1.0), the BCE loss λ_{bce} (2.0), dice loss λ_{dice} (0.5), batch size per device (2), and the gradient accumulation step (10). The semantic segmentation metric used is category mIoU, following [31]. It is calculated as follows: first, mIoU for each part category is calculated for all test objects. Then, part mIoUs that belong to each object category are averaged to compute the object category mIoU.

4.1 Reasoning Part Segmentation Results

We establish the reasoning part segmentation task on our dataset RPSeg3D. Table 2 shows the results of reasoning part segmentation. We observe that without any fine-tuning, the model’s performance is low compared to its fine-tuned counterparts. The general observation is that 3D queries help the model to output better masks. We compare our method to two baselines. The first baseline is LISA [24] applied to the multiple views without any finetuning. The second baseline consists of LISA finetuned on few-shot part segmentation data. When fine-tuned with 3D information, our model performs better than the baselines for normal and 3D prompts.

4.2 3D Semantic Segmentation Comparison with Existing Models

Table 3 shows the results of semantic segmentation compared to existing methods. Our method has better performance than all the fully supervised baselines and achieves competitive results with [31]. For this baseline, we observe that the model with prompt tuning done separately on each category achieves impressive few-shot performance but when unified into a single model for all categories, the performance is only slightly better than its zero-shot performance (27.2%). This is attributed to the redundant part names across object categories, hindering the model’s learning of the semantic meanings of part names. This results in performance to drop significantly from when only one category is learnt per model. To prompt for segmentation, the baseline models [38, 39, 54] are provided with the

Method	Val			Test		
	Normal Query	3D Query	Overall	Normal Query	3D Query	Overall
LISA-MV	16.60	20.16	18.38	17.60	20.57	19.08
LISA-MV (ft)	50.43	50.28	50.35	50.75	50.81	50.78
PARIS3D	55.33	55.50	55.42	55.94	57.60	56.77

Table 2: Results of reasoning part segmentation. LISA-MV [24] is LISA in multi-view setting without fine-tuning. LISA-MV (ft) is the experiment in which it has been fine-tuned on few-shot part segmentation data. Our proposed PARIS3D method has been fine-tuned with 3D queries and explanations. When fine-tuned with 3D information, our model performs better than the baselines for normal and 3D prompts. Here *Test* is the test set of 1906 shapes and *Val* is the validation set of 358 3D shapes with their instructions.

3D Data	Method	Overlapping Categories							Non-overlapping Categories							Overall (45)				
		Bottle	Chair	Display	Door	Knife	Lamp	Storage-Furniture	Table	Overall (17)	Camera	Cart	Dis-Penzer	Kettle	Kitchen Pot		Oven	Suit-case	Toaster	Overall (28)
Extra data (45x8+28k)	PointNet++ [38]	48.8	84.7	78.4	45.7	35.4	68.0	46.9	63.7	55.6	6.5	6.4	12.1	20.9	15.8	34.3	40.6	14.7	25.4	36.8
	PointNeXt [39]	68.4	91.8	89.4	43.8	58.7	64.9	68.5	52.1	58.5	33.2	36.3	26.0	45.1	57.0	37.8	13.5	8.3	45.1	50.2
	SoftGroup [47]	41.4	88.3	62.1	53.1	31.3	82.2	60.2	54.8	50.2	23.6	23.9	18.9	57.4	45.5	13.6	18.3	26.4	30.7	38.1
Few-shot (45x8)	PartSLIP* [31]	83.4	85.3	84.8	40.8	65.2	66.0	53.6	42.4	56.3	58.3	88.1	73.7	77.0	69.6	73.5	70.4	60.0	61.3	59.4
	PointNet++ [38]	27.0	42.2	30.2	20.5	22.2	10.5	8.4	7.3	18.1	9.7	11.6	7.0	28.6	31.7	19.4	3.3	0.0	21.8	20.4
	PointNeXt [39]	67.6	65.1	53.7	46.3	59.7	55.4	20.6	22.1	39.2	26.0	47.7	22.6	60.5	66.0	36.8	14.5	0.0	41.5	40.6
	SoftGroup [47]	20.8	80.5	39.7	16.3	38.3	38.3	18.9	24.9	32.8	28.6	40.8	42.9	60.7	54.8	35.6	29.8	14.8	41.1	38.0
	ACD [8]	22.4	39.0	29.2	18.9	39.6	13.7	7.6	13.5	19.2	10.1	31.5	19.4	40.2	51.8	8.9	13.2	0.0	25.6	23.2
	Prototype [54]	60.1	70.8	67.3	33.4	50.4	38.2	30.2	25.7	41.1	32.0	36.8	53.4	62.7	63.3	36.5	35.5	10.1	46.3	44.3
	PartSLIP+	64.8	69.5	59.5	24.5	34.5	37.1	32.0	40.1	35.3	25.5	75.7	15.6	30.5	58.4	31.1	49.4	6.6	26.7	29.9
Ours	84.0	81.0	70.1	68.4	47.2	61.2	39.4	45.1	55.1	29.3	71.7	40.1	59.3	78.8	59.1	61.6	24.9	59.1	57.6	

Table 3: Comparison to previous 3D part segmentation methods. Object category mIoU(%) is shown. In the 45x8+28k setting, baseline models use an additional 28k training shapes for 17 overlapping object categories. These are categories present in common with PartNet dataset. For the remaining 28 non-overlapping object categories, there are only 8 shapes per object category during training. PartSLIP* indicates that **one model has been trained for each category**. + shows our implementation of PartSLIP where one model is trained for all the categories together.

class IDs of the parts to be segmented (e.g: 0 to 103 including the background). In [31], the model is provided with the names of the parts to be segmented (e.g. "seat", "arm", "back" of a chair). Our method, PARIS3D, uses language instructions to prompt segmentation. For a fair comparison, we use hand-crafted prompts with short instructions containing generic concept and location-based clues about the part to be segmented.

4.3 Ablation Study

Instruction Rephrasing. We perform additional experiments with varying instruction types to study the merits of using rephrased prompts in training the model (Table 4 here) compared to using only category names and manually generated prompts for each category. These experimental results on the test set show that GPT-generated prompts provide superior performance, both without and with color and 3D information. In both cases, performance improves from using category names to manual prompts to GPT prompts, indicating the strong

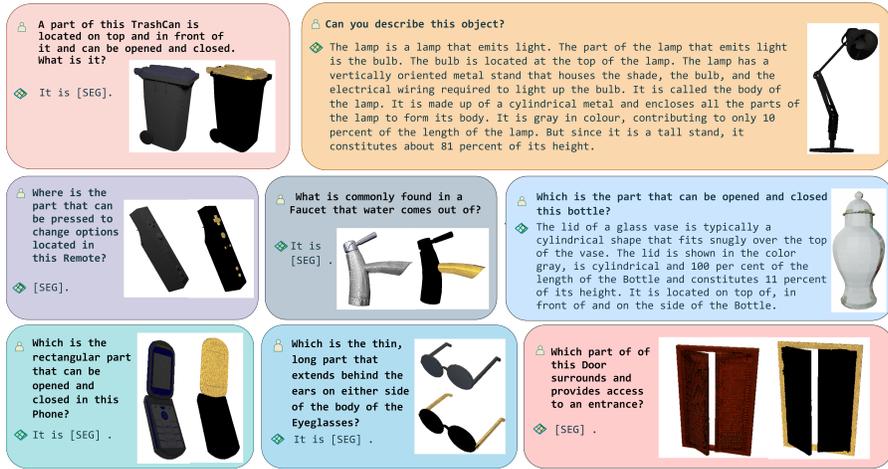


Fig. 5: Qualitative results of PARIS3D’s performance. We showcase examples from three tasks: reasoning 3D object part segmentation, object description, and reasoning question-answering, demonstrating its capabilities in offering in-depth reasoning, 3D understanding, part segmentation, and conversational abilities.

Instruction Type	Category Name	Manually Generated	GPT-generated	Category Name	Manually generated	GPT-generated
Colour + 3D	✗	✗	✗	✓	✓	✓
Overall mIoU	38.0	41.5	46.7	43.0	52.4	57.6

Table 4: A study on the effectiveness of rephrased prompts on the overall model performance. Here, the ability of the model to accurately segment an object part based on an implicit (reasoning-based) description is evaluated for different types of training prompts. The results show the superior performance of GPT-generated prompts over manual prompts, both with and without colour and 3D information.

generalization capabilities introduced by GPT-based prompts. For instance, using color and 3D information prompts along with ‘manual prompts’ resulted in an overall mIoU of 52.4%. However, using GPT to expand simple templates into varied prompts for describing object part categories resulted in about a 5.2% improvement over the manually curated training templates (last vs second-last column).

One Model vs Multiple Models. One of the baseline models [31] trained a model for each of the 45 categories, loading one at a time to evaluate each category. When we replicate this setting for PARIS3D, there are significant jumps in performance as shown in Table 5. However, one-model-for-one-category does not offer a generalizable solution in a real-world problem setting, where multiple object categories and their parts need to be analysed.

Number of Rendered Views. The effect of incrementing the number of views rendered from the input point cloud is shown in Table 6. With only 1 view in-

Method	Chair	Dispenser	Keyboard	Eyeglasses	Bucket
PartSLIP [31]	85.3	73.7	53.6	88.3	36.5
PARIS3D	86.5	75.6	88.2	92.2	83.9

Table 5: Results of PARIS3D on one model trained for each of 5 categories. In [31], 45 models were trained and each point cloud was tested by loading its corresponding model to perform evaluation. Repeating this exact setting by training a model on each category for PARIS3D, we easily gain +1.2%, +1.9%, +34.6%, +3.9%, and +47.4% improvements on the tested categories.

formation provided, the segmentations are not meaningful. With 5 views, the segmentation quality improves, and improves further with an increase in the number of views. For each point cloud, we use 10 rendered views in our main experiments.

Single prompt vs Multiple prompts. In Table 7, we show ablation experiments regarding the number of training prompts. Single prompt refers to one concept-based reasoning training instruction that follows a template. In further experiments, we use the template to generate multiple (about 5-6) rephrased instructions for each part category using [36]. Training with these multiple training prompts per part category results in a better-performing model than with a single prompt.

No. of views	Bottle	Bucket	Camera	No. of prompts	Bottle	Knife	Clock	Chair
1	1.1	1.1	1.2	1	73.8	44.3	45.5	69.5
5	23.7	40.1	12.1	5-6	83.9	52.7	51.3	78.7
10	84.8	84.9	29.3					

Table 6: Ablation experiments on the number of rendered views.

Table 7: Ablation experiments on the number of prompts provided in training data.

4.4 Qualitative Results

In Figure 5, we provide examples of PARIS3D output for 3D object part segmentation tasks. PARIS3D is capable of segmenting fine-grained details in categories like buttons of a remote or legs of an eyeglass from implicit queries that expect it to do self-reasoning to generate the right answer, and then segment the part. Moreover, it can perform tasks like object description and question-answering with regard to object parts and give detailed answers. This demonstrates its conversational, reasoning, and part segmentation capabilities.



Fig. 6: Qualitative results on real-world examples of point clouds. The input point cloud is shown on the left and the segmentation mask is in yellow on the right. Our model generalizes to point clouds with a significant domain gap from those it was trained on and displays impressive results with segmenting small object parts such as the furniture’s knob in this example.

4.5 Generalizability to Real-World Examples

Most existing 3D tasks and their corresponding models can usually only handle cases within the same distribution of the training sets without generalization, since they are sensitive to the format of the input and significant domain gap between synthetic experiments and real-world examples. Thus, it can be difficult to use them in use cases involving real point clouds from an everyday setting. To demonstrate the generalizability of PARIS3D to data derived from the real world, we perform our 3D segmentation on real point clouds shot using a smartphone’s LiDAR sensor, as suggested by [31]. In Figure 6, we show qualitative examples of passing the fused point clouds through the PARIS3D architecture to obtain part segmentation labels as in the previous experiments without much drop in performance.

5 Conclusion

In this work, we introduce a novel challenge within 3D segmentation, reasoning-based part segmentation. This task requires models to infer, reason, and explain based on implicit user instructions, making it considerably more complex than the regular 3D referring segmentation task. We introduce a dataset for this task, RPSeg3D, to enable effective evaluation. We believe this dataset will play a crucial role in fostering the growth of technologies in this area. Additionally, we outline a pipeline that integrates 3D segmentation capabilities into multimodal Large Language Models (LLMs), showcasing our model, PARIS3D, which exhibits competitive performance. It additionally demonstrates the ability to identify part concepts, reason about them, and complement them with world knowledge. However, we identify limitations: Although the model has improved generalizability, there is a possibility that the model may occasionally result in inaccurate predictions while being prompted with inappropriate, incorrect, or biased text inputs. Also, the model in its current form cannot perform instance segmentation. This is a direction for future research as we expand the dataset to accommodate such tasks.

References

1. Aminabadi, R.Y., Rajbhandari, S., Zhang, M., Awan, A.A., Li, C., Li, D., Zheng, E., Rasley, J., Smith, S., Ruwase, O., He, Y.: Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale (2022)
2. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval (2022)
3. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts (2021)
4. Chen, S., Chen, X., Zhang, C., Li, M., Yu, G., Fei, H., Zhu, H., Fan, J., Chen, T.: Ll3da: Visual interactive instruction tuning for omni-3d understanding, reasoning, and planning (2023)
5. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3070–3079. IEEE Computer Society, Los Alamitos, CA, USA (jun 2019). <https://doi.org/10.1109/CVPR.2019.00319>, <https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00319>
6. Ding, R., Yang, J., Xue, C., Zhang, W., Bai, S., Qi, X.: Pla: Language-driven open-vocabulary 3d scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)
7. Ding, R., Yang, J., Xue, C., Zhang, W., Bai, S., Qi, X.: Pla: Language-driven open-vocabulary 3d scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7010–7019 (June 2023)
8. Gadelha, M., RoyChowdhury, A., Sharma, G., Kalogerakis, E., Cao, L., Learned-Miller, E., Wang, R., Maji, S.: Label-efficient learning on point clouds using approximate convex decompositions. In: European Conference on Computer Vision (ECCV) (2020)
9. Graham, B., Engelcke, M., van der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks (2017)
10. Graham, B., Engelcke, M., Maaten, L.v.d.: 3d semantic segmentation with submanifold sparse convolutional networks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9224–9232 (2018). <https://doi.org/10.1109/CVPR.2018.00961>
11. Graham, B., van der Maaten, L.: Submanifold sparse convolutional networks (2017)
12. Guo, Z., Zhang, R., Zhu, X., Tang, Y., Ma, X., Han, J., Chen, K., Gao, P., Li, X., Li, H., Heng, P.A.: Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following (2023)
13. Guo, Z., Tang, Y., Zhang, R., Wang, D., Wang, Z., Zhao, B., Li, X.: Viewrefer: Grasp the multi-view knowledge for 3d visual grounding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15372–15383 (October 2023)
14. Guo, Z., Tang, Y., Zhang, R., Wang, D., Wang, Z., Zhao, B., Li, X.: Viewrefer: Grasp the multi-view knowledge for 3d visual grounding with gpt and prototype guidance (2023)
15. Hong, Y., Du, Y., Lin, C., Tenenbaum, J.B., Gan, C.: 3d concept grounding on neural fields. NeurIPS (2022)
16. Hong, Y., Lin, C., Du, Y., Chen, Z., Tenenbaum, J.B., Gan, C.: 3d concept learning and reasoning from multi-view images. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)

17. Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., Gan, C.: 3d-llm: Injecting the 3d world into large language models. *NeurIPS* (2023)
18. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: *International Conference on Learning Representations* (2022), <https://openreview.net/forum?id=nZvKeeFYf9>
19. Huang, Q., Wang, W., Neumann, U.: Recurrent slice networks for 3d segmentation of point clouds. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2626–2635 (2018). <https://doi.org/10.1109/CVPR.2018.00278>
20. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: ReferItGame: Referring to objects in photographs of natural scenes. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 787–798. Association for Computational Linguistics, Doha, Qatar (Oct 2014). <https://doi.org/10.3115/v1/D14-1086>, <https://aclanthology.org/D14-1086>
21. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023)
22. Kundu, A., Yin, X., Fathi, A., Ross, D., Brewington, B., Funkhouser, T., Pantofaru, C.: Virtual multi-view fusion for 3d semantic segmentation. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*. pp. 518–535. Springer (2020)
23. Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J.: Stratified transformer for 3d point cloud segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8500–8509 (2022)
24. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692* (2023)
25. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs (2018)
26. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models (2023)
27. Li, M., Chen, X., Zhang, C., Chen, S., Zhu, H., Yin, F., Yu, G., Chen, T.: M3dbench: Let’s instruct large models with multi-modal 3d prompts (2023)
28. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context (2015)
29. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning (2023)
30. Liu, K., Zhan, F., Zhang, J., XU, M., Yu, Y., El Saddik, A., Theobalt, C., Xing, E., Lu, S.: Weakly supervised 3d open-vocabulary segmentation. In: Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) *Advances in Neural Information Processing Systems*. vol. 36, pp. 53433–53456. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/a76b693f36916a5ed84d6e5b39a0dc03-Paper-Conference.pdf
31. Liu, M., Zhu, Y., Cai, H., Han, S., Ling, Z., Porikli, F., Su, H.: Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 21736–21746 (June 2023)
32. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019)
33. Ma, X., Yong, S., Zheng, Z., Li, Q., Liang, Y., Zhu, S.C., Huang, S.: Sqa3d: Situated question answering in 3d scenes. In: *International Conference on Learning Representations* (2023), <https://openreview.net/forum?id=IDJx97BC38>

34. Mascaro, R., Teixeira, L., Chli, M.: Diffuser: Multi-view 2d-to-3d label diffusion for semantic scene segmentation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 13589–13595 (2021). <https://doi.org/10.1109/ICRA48506.2021.9561801>
35. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips (2019)
36. OpenAI: Gpt-4 technical report (2023)
37. Peng, S., Genova, K., Jiang, C.M., Tagliasacchi, A., Pollefeys, M., Funkhouser, T.: Openscene: 3d scene understanding with open vocabularies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
38. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation (2017)
39. Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
40. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d (2020)
41. Robert, D., Vallet, B., Landrieu, L.: Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5575–5584 (2022)
42. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J.: Laion-5b: An open large-scale dataset for training next generation image-text models (2022)
43. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs (2021)
44. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018)
45. Takmaz, A., Fedele, E., Sumner, R.W., Pollefeys, M., Tombari, F., Engelmann, F.: OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In: Advances in Neural Information Processing Systems (NeurIPS) (2023)
46. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. Proceedings of the IEEE International Conference on Computer Vision (2019)
47. Vu, T., Kim, K., Luu, T.M., Nguyen, X.T., Yoo, C.D.: Softgroup for 3d instance segmentation on 3d point clouds. In: CVPR (2022)
48. Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., Dai, J.: Visionllm: Large language model is also an open-ended decoder for vision-centric tasks (2023)
49. Wang, Z., Huang, H., Zhao, Y., Zhang, Z., Zhao, Z.: Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. arXiv preprint arXiv:2308.08769 (2023)
50. Xu, M., Ding, R., Zhao, H., Qi, X.: Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In: CVPR (2021)
51. Xu, R., Wang, X., Wang, T., Chen, Y., Pang, J., Lin, D.: Pointllm: Empowering large language models to understand point clouds (2023)

52. Yang, J., Chen, X., Qian, S., Madaan, N., Iyengar, M., Fouhey, D.F., Chai, J.: Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent (2023)
53. Zhang, S., Sun, P., Chen, S., Xiao, M., Shao, W., Zhang, W., Liu, Y., Chen, K., Luo, P.: Gpt4roi: Instruction tuning large language model on region-of-interest (2023)
54. Zhao, N., Chua, T.S., Lee, G.H.: Few-shot 3d point cloud semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
55. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models (2023)
56. Ziyu, Z., Xiaojian, M., Yixin, C., Zhidong, D., Siyuan, H., Qing, L.: 3d-vista: Pre-trained transformer for 3d vision and text alignment. In: ICCV (2023)