PointLLM: Empowering Large Language Models to Understand Point Clouds

Runsen Xu[1,2], Xiaolong Wang[3], Tai Wang[2†], Yilun Chen[2], Jiangmiao Pang[2†], and Dahua Lin[1,2,4]

[1] The Chinese University of Hong Kong
[2] Shanghai AI Laboratory
[3] Zhejiang University
[4] Centre for Perceptual and Interactive Intelligence

Abstract. The unprecedented advancements in Large Language Models (LLMs) have shown a profound impact on natural language processing but are yet to fully embrace the realm of 3D understanding. This paper introduces PointLLM, a preliminary effort to fill this gap, empowering LLMs to understand point clouds and offering a new avenue beyond 2D data. PointLLM understands colored object point clouds with human instructions and generates contextually appropriate responses, illustrating its grasp of point clouds and common sense. Specifically, it leverages a point cloud encoder with a powerful LLM to effectively fuse geometric, appearance, and linguistic information. To overcome the scarcity of point-text instruction following data, we developed an automated data generation pipeline, collecting a large-scale dataset of more than 730K samples with 660K different objects, which facilitates the adoption of the two-stage training strategy prevalent in MLLM development. Additionally, we address the absence of appropriate benchmarks and the limitations of current evaluation metrics by proposing two novel benchmarks: Generative 3D Object Classification and 3D Object Captioning, which are supported by new, comprehensive evaluation metrics derived from human and GPT analyses. Through exploring various training strategies, we develop PointLLM, significantly surpassing 2D and 3D baselines, with a notable achievement in human-evaluated object captioning tasks where it surpasses human annotators in over 50% of the samples. Codes, datasets, and benchmarks are available at https://github.com/OpenRobotLab/PointLLM.

Keywords: Multi-Modal LLM · 3D Understanding · Point Cloud

1 Introduction

Recent developments in large language models (LLMs) [4, 6, 34–36, 42, 47, 48] have showcased their remarkable capabilities in natural language processing,

† Corresponding authors: {wangtai, pangjiangmiao}@pjlab.org.cn

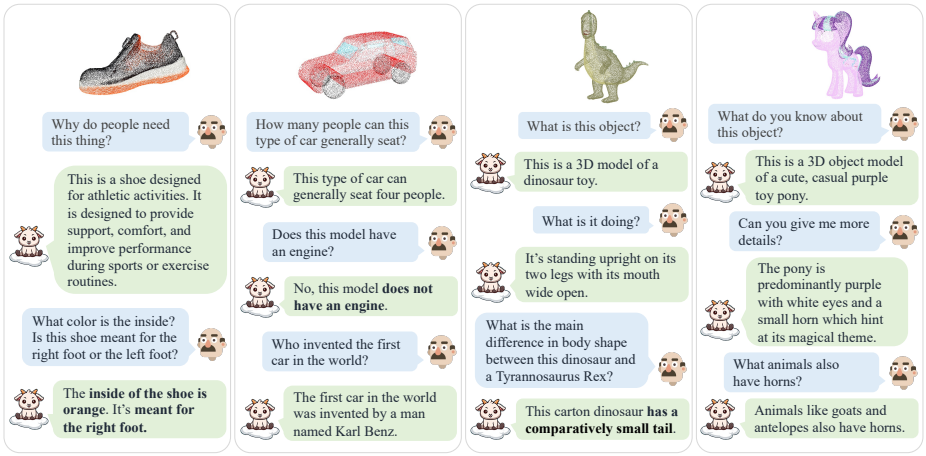


Fig. 1: We introduce PointLLM, a multi-modal large language model capable of understanding colored point clouds of objects. It perceives object types, geometric, and appearance without concerns for ambiguous depth, occlusion, or viewpoint dependency.

acting as generalized interfaces [18] for a broad range of tasks [4, 42]. Beyond text, the exploration of multi-modal LLMs (MLLMs) now extends to processing audio [22], images [1, 23, 28, 31, 35, 59, 61], and more.

The next step in this evolution lies in understanding 3D structures, and particularly point clouds. Suppose we want to embed LLMs in 3D design software for interactive 3D content creation/editing via text commands (*i.e.* 3D copilot), this requires LLMs to understand 3D content states, which can be represented as point clouds. In robotics, LLMs as control centers need to understand the environments for perception and planning, where point clouds captured through depth sensors or LiDAR are important observations.

While existing efforts to integrate LLMs with 2D images [8, 10, 31, 61] also provide 3D comprehension, they face difficulties like depth ambiguity, occlusion, and viewpoint dependency. In contrast, point clouds, as an efficient and universal 3D representation, offer direct geometric and appearance information. Despite these benefits, the integration of point clouds with LLMs is still underexplored.

Recently, connecting pre-trained encoders with LLMs using projection layers and employing a two-stage training of alignment and instruction tuning has been proven effective for developing MLLMs across various domains [8,10,23,25,31,61]. We pose a question: Can this established framework be successfully adapted to the realm of point clouds? In this work, we affirmatively answer this question by introducing PointLLM (Fig. 1), our preliminary effort to empower LLMs to understand point clouds, with a focus on 3D objects.

The first difficulty to be address is the lack of training data, the point-text instruction following datasets essential for teaching models to interpret point clouds and follow user commands. While manual compilation is costly and labor-intensive, we devised an automated data collection pipeline using GPT-4 to gen-

erate diverse instructions from Objaverse's [9] captions. This produced a large-scale dataset comprising 660K brief-description for different objects and 70K complex instructions, enabling the model's two-stage training for this domain.

Evaluating model performance with appropriate tasks and metrics presents another challenge. We aim to assess point cloud comprehension in MLLMs and existing discriminative-based 3D perception benchmarks fall short for this purpose due to the generative nature of MLLMs. We introduce two novel benchmarks: Generative 3D Object Classification and 3D Object Captioning, based on a hypothesis that LLMs' understanding of point clouds is reflected by their ability to identify the object's category and the accuracy and details of captions, which elaborate the information they perceive. We also observe the limitations of some widely used NLP metrics like BLEU-1 [37], ROUGE-L [30], and METEOR [3] for their short caption bias and inability to reflect linguistic diversity. To counter these shortcomings, we devise new evaluation metrics that combine human and GPT-4/ChatGPT evaluation with data-driven approaches, establishing a comprehensive evaluation framework. To our knowledge, we are the first to introduce generative object classification in this field.

In our study, we evaluated various training recipes and observed that an optimal number of projection layers enhances feature clustering, aligning point and text features effectively. We also found that employing max pooling to aggregate point tokens reduces the token number and greatly enhances training speed, albeit with a slight performance trade-off. Further analysis of data variability revealed that the model performance peaks with about 600K samples for alignment and diverse instruction data notably enhances fine-tuning, emphasizing the value of data quantity and diversity. These insights led to the development of PointLLM, which markedly surpasses 2D and 3D baselines, impressively scoring higher than human annotators in over 50% of the object captioning samples.

2 Related Work

Multi-modal large language models. Multi-modal Large Language Models (MLLMs) are designed to comprehend and interpret a wide range of information that extends beyond mere text-based data [56], including but not limited to images [14,23,31,51,61], audio [22], motion [25], etc. Broadly, the models can be classified into two categories. The first category includes models that employ a large language model to interface with individual, modality-specific models or APIs [16, 22, 38, 46, 52]. This approach circumvents the need for model training but is heavily dependent on the availability and capabilities of pre-existing models or APIs. The second category pertains to models that employ an end-to-end training strategy. There are two prominent paradigms within this category. The first involves training the model from scratch, similar to text-only LLMs, using large-scale multi-modal corpora and datasets [23, 39]. The second paradigm builds on pre-trained LLMs and unimodal encoders. [1, 2, 8, 10, 11, 14, 27, 28, 31, 44, 59–61]. This strategy typically involves a two-stage process: alignment of the unimodal encoder with the LLM's feature space, followed by instruction-based

fine-tuning. In our work, we adhere to the alignment and tuning strategy to construct an MLLM capable of understanding 3D object point clouds.

Object point cloud understanding with language. Inspired by models like CLIP [41], which bridges visual and textual modalities, similar advancements have emerged in the 3D object domain [19, 24, 32, 50, 54, 55, 58, 62]. PointCLIP [58], PointCLIPv2 [62], and CLIP2Point [24] utilize depth image projections of point clouds for 3D recognition with 2D CLIP models. Others, such as ULIP [54], JM3D [50], OpenShape [32], and CG3D [19], train point cloud encoders to align with CLIP representations using triplets of point clouds, images, and texts. ULIP-2 [55] and OpenShape [32] have expanded this by employing image-captioning models for automatic data generation, enhancing training triplet scalability. Cap3D [33] and UniG3D [45] adopt similar approaches for point-text dataset generation. In our work, we leverage Cap3D's captions on Objaverse for automatic instruction-data generation in training PointLLM. The recently introduced 3D-LLM [21] also seeks to enable LLMs to comprehend 3D, by rendering objects into multi-view images, using 2D foundational models like CLIP [41] and SAM [26] for feature extraction, and 2D MLLMs such as BLIP [28] for output generation. Concurrently, Point-Bind LLM [15] aligns point cloud features with ImageBind [13] and uses 2D MLLMs like Imagebind-LLM [17] for generation. Though simple, it faces challenges like hallucination due to its retrieval nature. Distinctively, PointLLM directly aligns point clouds with LLM by end-to-end training, avoiding complicated data pre-processing and enabling accurate, open-ended, and free-form interactions.

3 Methodology

This section elucidates our strategy for the automatic generation of point-text instruction-following data. We then delve into the architecture of PointLLM, which takes as input an object point cloud and user instruction and outputs responses. Lastly, we detail our loss function and two-stage training strategy.

3.1 Point-Text Instruction Following Data

The daunting challenge in the development of an end-to-end multi-modal LLM is procuring large-scale multi-modal instruction-following data, vital for representation learning, aligning latent spaces, and orienting the model to adhere to human intentions [1, 8, 29, 31, 61]. However, manual labeling of such data is cost-prohibitive and labor-intensive. To overcome this, we follow [31] and propose an automated data generation technique utilizing the large-scale point cloud captioning dataset, Cap3D [33], with the assistance of GPT-4 [35]. The generated dataset adheres to a uniform instruction following template, shown in Tab. 1, and consists of brief-description instructions and complex instructions, which aid in latent space alignment and instruction tuning, respectively.

Brief-description instructions. The Cap3D [33] dataset provides two variations of captions for the 3D objects in Objaverse [9]: those generated by image-captioning models and those annotated by humans. While there are 660K objects

Table 1: Instruction following template. {System Prompt} is the system prompt used by the pre-trained LLM, {p_tokens} are point tokens, and {Instruction} and {Response} denote user instructions and model responses. Losses are computed only for model responses and the end-of-sentence token </s>.

{System Prompt}	
USER: ASSISTANT:	<p_start>{p_tokens}<p_end>{Instruction 1} {Response 1}</s>
USER: ASSISTANT:	{Instruction 2} {Response 2}</s>
USER: ASSISTANT:	{Instruction 3} {Response 3}</s>

accompanied by generated captions, only 40K samples have human-annotated captions. For brief-description instruction, we utilize the model-generated split due to the need for a larger data volume for aligning the latent spaces of point cloud and text modalities [31]. We created a list of 30 instructions to instruct the model to provide a succinct description of a given 3D object point cloud. A random instruction from this list is chosen as the user instruction, and the caption from Cap3D is used directly as the model response, forming a single-round instruction following sample. This results in 660K brief-description instruction data, each corresponding to a unique object point cloud.

Complex instructions. Beyond brief descriptions, it's crucial that the model learns to understand objects from a variety of angles, responding accurately to diverse human instructions. To facilitate this, we employ GPT-4 to produce complex instruction-following data. Specifically, a caption from Cap3D is used to stimulate GPT-4 to craft a more comprehensive description that identifies the object's type, appearance, functionalities, and any other inferable information. Similar to the process for generating brief-description instructions, we also curate a set of 30 distinct prompts, each pushing the model to describe the 3D object in depth. One of these prompts is randomly coupled with the newly crafted description, forming a training sample. GPT-4 is further used to generate conversations (*i.e.*, Q&A pairs) that delve into diverse aspects of the object based on the captions, such as the object's functionality or materials, and the corresponding answers should be informative and comprehensive. For each object, GPT-4 generates 3 single-round conversations and 1 multi-round conversation with 3 Q&A pairs, all ensuring logical relevance.

With a focus on data quality, we selected 15K captions from the Cap3D human-annotated split for data generation, each comprising more than five words. After filtering incorrect GPT-4 outputs, we collected 70K complex instructions, including 15K detailed descriptions, 40K single-round conversations, and 15K multi-round conversations. The instruction lists, GPT-4 prompts, data examples, and distribution analysis can be found in the supplementary material.

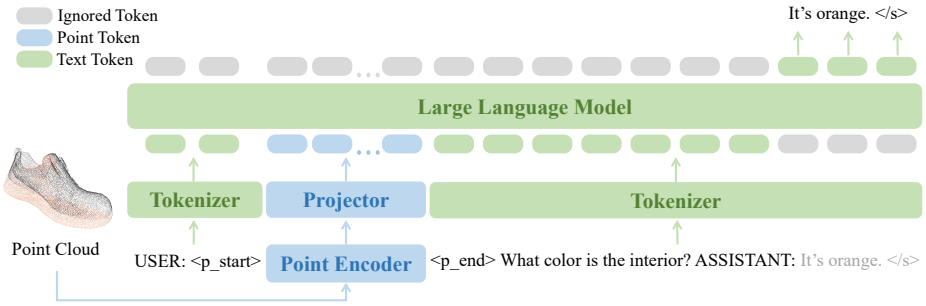


Fig. 2: An overview of PointLLM. The point encoder extracts features from the input point cloud and the projector projects them to the latent space of the LLM backbone. The LLM backbone processes sequences of point and text tokens and generates the predicted tokens as the output.

3.2 Model Architecture

As shown in Fig. 2, our PointLLM is a generative model that aims to complete multi-modal sentences containing both point clouds and texts. The model consists of three main components: a pre-trained point cloud encoder f_{pe}, a projector f_{proj}, and a pre-trained large language model (LLM) backbone f_{llm}.

The point cloud encoder f_{pe} takes as input a point cloud $P \in \mathbb{R}^{n \times d}$, where n is the number of points and d is the feature dimension of each point. The output of the encoder is a sequence of point features $X = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^{m \times c}$, where m is the number of point features and c is the feature dimension. The projector f_{proj} is a MLP that maps the point features X to point tokens $Y = (y_1, y_2, ..., y_m) \in \mathbb{R}^{m \times c'}$, where c' is the dimension of the point tokens, which is the same as the text tokens.

The LLM backbone f_{llm} is a decoder-only Transformers [49], which accepts a sequence of tokens, composed of both text and point tokens. This mixed sequence of tokens is denoted as $Z = (z_1, z_2, ..., z_k) \in \mathbb{R}^{k \times c'}$, where k is the total number of tokens. Utilizing a self-attention mechanism, the LLM backbone is capable of understanding the contextual relationships between different types of tokens, enabling it to generate responses based on both text and point cloud inputs. Formally, the output of the LLM backbone f_{llm} is a sequence of predicted tokens $\hat{Z} = (\hat{z}_1, \hat{z}_2, ..., \hat{z}_k) \in \mathbb{R}^{k \times c'}$. The prediction of the i-th token, $\hat{z}_i$, is conditioned on all previous tokens, $Z_{<i} = (z_1, ..., z_{i-1})$, expressed mathematically as

$$\hat{z}_i = f_{llm}(Z_{<i}). \tag{1}$$

Each $\hat{z}_i$ is passed through a final linear layer followed by a softmax operation, mapping the hidden states into a probability distribution over the vocabulary. This additional layer is denoted as $f_{vocab} : \mathbb{R}^{c'} \rightarrow \mathbb{R}^{V}$, where V is the size of the vocabulary. The final prediction $\tilde{z}_i$ for the i-th token is the word in the vocabulary with the highest probability:

$$\tilde{z}_i = \arg \max_{w \in \text{vocab}} f_{vocab}(\hat{z}_i)[w]. \tag{2}$$

3.3 Training

Loss function. We train PointLLM by minimizing the negative log-likelihood of the text token at each position. Our loss function is only computed on text tokens that constitute the model's responses, including the end-of-sentence token </s>. We exclude the tokens from human instructions, ensuring that the model focuses on learning to generate accurate and coherent responses. The end-to-end nature of this training approach enables PointLLM to effectively integrate point cloud and text modalities.

Two-stage training. Our training procedure comprises two stages, each focusing on different aspects of the model.

During the first stage, termed the **feature alignment stage**, we freeze the parameters of the point cloud encoder and the LLM, and train only the MLP projector. At this stage, the training process uses brief-description instructions, aiming to align point features with the text token space effectively. This stage also includes the adjustment of token embeddings for the two newly added special tokens <p_start> and <p_end>.

In the second stage, referred to as the **instruction tuning stage**, we freeze the point cloud encoder while jointly training the projector and the LLM. This second stage uses complex instructions and helps the model build its ability to understand and respond to complex instructions including point cloud data.

4 Benchmarks and Evaluation

Evaluating multi-modal LLMs presents a challenge due to the lack of a unified metric for their diverse outputs. Current 3D point cloud benchmarks primarily focus on discriminative tasks, such as classification or retrieval, missing the generative and open-vocabulary aspects of LLMs. To address this, we introduce two novel benchmarks: Generative 3D Object Classification and 3D Object Captioning, designed to assess model generalization and understanding of point clouds. Our underlying hypothesis is that models' comprehension of point clouds, at the very least, manifests in the accurate classification of objects. Furthermore, this comprehension is proportional to the accuracy and details of the captions, which elaborate the information they perceive. To support these benchmarks, we've developed a novel and comprehensive evaluation framework that incorporates human, GPT-4/ChatGPT, and traditional metrics. The supplementary material provides the prompts and the human verification of the GPT evaluation.

4.1 Generative 3D Object Classification

The task of generative 3D object classification involves prompting the model to freely answer the object type from its point cloud, distinguishing it from discriminative models that classify objects based on probability comparisons. We consider two settings: close-set zero-shot and open-vocabulary setting.

Close-set zero-shot classification. In this scenario, the object type belongs to a fixed set of categories, and the model never sees any samples of this dataset

during training. This tests the model's ability to generalize to unseen domains. We use the test split of the ModelNet40 [53] dataset as our source of data, which contains point clouds of 40 different object categories. We use ChatGPT as a post-processor to select one of the ModelNet40 categories based on the model's answer. If ChatGPT selects the correct option, then we consider the model's classification correct; otherwise, we consider it incorrect. Please refer to the supplementary material for more discussions about this task's setting.

Open-vocabulary classification. In this scenario, the object type is not limited to a predefined set of categories, but can be any word or phrase that identifies the object. This reflects the real-world setting where new objects can appear at any time, and the model needs to be able to recognize them without retraining. We randomly select 200 objects from the Objaverse [9] dataset, incorporating human-annotated captions from Cap3D [33] as ground truth labels for the task. We employ GPT-4 to verify if the model's response matches the intended object type as described in the human caption, allowing for varied expressions that correctly identify the object. For instance, responses like "a cup" or "a coffee mug" are considered correct classification for a human caption of "a blue mug." GPT-4 is preferred in this setting for its precision in recognizing synonymous object descriptions, in contrast to ChatGPT, which is more prone to false negatives by not acknowledging similar terms for the same object. ChatGPT, however, is used in the close-set setting as it performs accurately but at 95% less cost.

4.2 3D Object Captioning

3D object captioning evaluates the model's detailed understanding of point clouds. We utilize the same 200 objects as for the open-vocabulary classification and prompt our model to caption them in detail. Human-annotated captions serve as reference ground truths for automatic evaluation.

For a comprehensive and robust evaluation, we employ three distinct methods to assess performance in this task:

1. **Human evaluation.** Human evaluators conduct a binary classification and counting task, reviewing randomly shuffled captions from various models and human-annotated captions for the objects, without knowing their sources. Using the official Objaverse [9] explorer, evaluators visually inspect each object and systematically assess each object attribute (such as type, color, material, *etc.*) mentioned in the model's response, assigning one correct or hallucination point for each attribute based on its accuracy. The aggregate of these points forms the correctness and hallucination scores. Precision, the ratio of accurate information in the model's response, is then calculated. For detailed scoring criteria, please refer to the supplementary material.
2. **GPT-4 evaluation.** Given a model-generated caption and its corresponding human reference, GPT-4 identifies the attributes mentioned in the human caption and calculates the percentage of these attributes that are either correctly or partially matched in the model's caption, scoring from 0 to 100 with an explanation. As one of the most advanced language models, GPT-4 is

well-equipped to perform such tasks. Our method of calculating the correct percentage and assigning scores offers an advantage over approaches like those in [31], which directly generate an overall score without transparency.
3. **Traditional metric evaluation.** We report traditional metrics results including BLEU-1 [37], ROUGE-L [30], and METEOR [3] following [21]. Though widely used, these metrics have limitations as detailed in Sec. 5.3. Therefore, we incorporate and rely more on two additional data-driven metrics, Sentence-BERT [43] and SimCSE [12] similarity, which compute the similarity of sentence embeddings between model-generated and human captions.

5 Experimental Results

5.1 Experimental Settings

Implementation details. We use the LLaMA model [48] as our LLM backbone, with the 7B and 13B Vicuna [5] checkpoint. Point-BERT [57], pre-trained with ULIP-2 [55] on the Objaverse [9] dataset, serves as our point encoder. The 200 objects from Objaverse utilized for our benchmarks are not seen during any stage of the training. We utilize $n = 8192$ points and $d = 6$ dimensions for each point cloud. We assign a black color to point clouds from ModelNet40, as they lack color information. The point encoder outputs $m = 513$ point features, each with $c = 384$ dimensions. The projector contains three linear layers with the GeLU [20] activation, which maps point features to tokens with $c' = 5120$ (7B model) or $c' = 5120$ (13B model) dimensions. As we add two additional special tokens, the vocabulary size of PointLLM is $V = 32003$. All experiments are conducted on 8 $\times$ 80G A100 GPUs. GPT-4 and ChatGPT in this paper all refer to OpenAI's "gpt-4-0613" and "gpt-3.5-turbo-0613" models respectively. More implementation and training details are provided in the supplementary material.
Baselines. Our analysis includes comparisons with baselines capable of performing the same generative classification and captioning tasks. We focus on 3D-LLM [21] and Point-Bind LLM [15]; 3D-LLM is assessed solely on the Objaverse dataset due to its current lack of support for pure point clouds, while Point-Bind LLM, not supporting colored point clouds, is excluded from captioning. We also compare with two popular 2D MLLMs, InstructBLIP [8] and LLaVA [31], to explore the performance gap between image-based and point-based MLLMs and to highlight the advantages of point clouds over single-view images.

5.2 Generative 3D Object Classification

Tab. 2 shows the classification accuracy of various models on our proposed tasks. For 2D MLLMs' image inputs, we randomly sample rendered images of ModelNet40 point clouds and Objaverse objects. We prompt all the models with the same prompts of two types: the **I**nstruction-typed (I) prompt "What is this?" and the **C**ompletion-type (C) prompt "This is an object of ".

Experimental results demonstrate PointLLM's superiority over both 2D and 3D MLLMs on ModelNet40 and Objaverse datasets for various prompt types.

Table 2: Generative 3D object classification results on the ModelNet40 (M40.) test split and Objaverse (Obj.). The results show the classification accuracy under the **I**nstruction-typed (I) prompt "What is this?" and the **C**ompletion-typed (C) prompt "This is an object of " as well as the average accuracy.

Model	Input	M40.(I)	M40.(C)	Obj.(I)	Obj.(C)	Avg.
InstructBLIP-7B [8]	Single-V. Img.	19.53	31.48	45.00	42.00	34.50
InstructBLIP-13B [8]	Single-V. Img.	25.97	31.40	37.00	31.50	31.47
LLaVA-7B [31]	Single-V. Img.	39.75	39.67	49.50	50.50	44.86
LLaVA-13B [31]	Single-V. Img.	37.12	36.06	53.00	50.50	44.17
3D-LLM [21]	3D Obj. + Mul.-V. Img.	-	-	49.00	41.50	45.25
Point-Bind LLM [15]	3D Point Cloud	51.90	39.71	6.00	4.50	25.53
PointLLM-7B	3D Point Cloud	**53.44**	51.82	55.00	51.00	52.82
PointLLM-13B	3D Point Cloud	53.00	**52.55**	**56.50**	**51.50**	**53.39**

Compared with 2D models, PointLLM offers direct point cloud engagement, showcasing enhanced 3D object comprehension over single-view images. This method effectively addresses challenges like occlusion and viewpoint variation, leveraging rich 3D geometry and appearance data from colored point clouds. PointLLM shows more consistent classification accuracy across different prompts than other 3D models, underlining its prompt robustness. Utilizing a pre-trained point encoder and an LLM backbone, PointLLM efficiently translates point cloud data into descriptive natural language, capturing the object's identity.

The zero-shot performance on ModelNet40 further illustrates our model's aptitude for generalization. Even though ModelNet40 comprises point clouds unseen during training, PointLLM recognizes them using its pre-existing knowledge and perception abilities honed during our two-stage training. This adaptability to unseen domains and novel objects, without necessitating retraining, is crucial for real-world deployment as a foundation model.

5.3 3D Object Captioning

Tab. 3 displays the results of the captioning benchmark, averaged across objects. Each model was prompted with "Caption this 3D model in detail."

In Tab. 3 our models significantly outperform all baselines in key evaluation metrics for 3D object captioning, especially in human correctness score and GPT-4 evaluations. These scores reflect a model's ability to capture and articulate the intricate details of objects. Notably, PointLLM achieves the highest correctness scores, producing more accurate and detailed captions than other models, even rivaling human annotations. In addressing hallucination, a common challenge, our PointLLM exhibits the lowest hallucination scores and highest precision scores, indicating its effectiveness in generating detailed, accurate captions with less false information. The Sentence-BERT and SimCSE results further confirm our model's capability in producing captions more semantically aligned with the ground truth. Interestingly, all 13B models, regardless of being 2D or 3D MLLMs,

Table 3: 3D object captioning results on Objaverse. Evaluation encompasses human (correctness, hallucination, precision) and GPT-4 assessments, supplemented by Sentence-BERT, SimCSE, BLEU-1, ROUGE-L, and METEOR metrics. A primary focus is placed on human and GPT-4 evaluation, along with data-driven metrics. "*" indicates PointLLM was prompted for shorter captions with no more than 20 words.

Model	Corr.	Hallu.↓	Prec.	GPT-4	S.-BERT	SimCSE	B-1.	R-L.	MET.
InstructBLIP-7B [8]	2.56	0.77	76.99	45.34	47.41	48.48	4.27	8.28	12.99
InstructBLIP-13B [8]	2.58	1.13	69.56	44.97	45.90	48.86	4.65	8.85	13.23
LLaVA-7B [31]	2.76	0.86	76.30	46.71	45.61	47.10	3.64	7.70	12.14
LLaVA-13B [31]	2.43	0.86	73.97	38.28	46.37	45.90	4.02	8.15	12.58
3D-LLM [21]	1.77	1.16	60.39	33.42	44.48	43.68	16.91	19.48	**19.73**
PointLLM-7B	3.04	**0.66**	**82.14**	44.85	47.47	48.55	3.87	7.30	11.92
PointLLM-13B	**3.10**	0.84	78.75	**48.15**	**47.91**	**49.12**	3.83	7.23	12.26
PointLLM-13B*	2.12	0.39	84.39	44.27	50.15	50.83	**17.09**	**20.99**	16.45
Human	2.67	0.22	92.46	100.00	100.00	100.00	100.00	100.00	100.00

have higher hallucination scores than their 7B counterparts. This suggests larger MLLMs may be more challenging to fine-tune for precision. The investigation of this trend and its causes is an intriguing research direction.

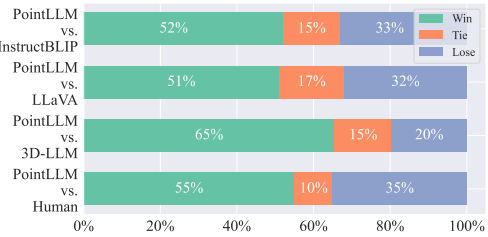


Fig. 3: Win rate comparison.

We analyzed the human evaluation data to compare our models with baselines and human annotations. Win rates, calculated based on the correctness score for the 13B variants, are presented in Fig. 3. PointLLM demonstrates notable performance, outperforming counterparts in over half of the test samples, including against human annotations (55% vs. 35%). This underscores PointLLM's ability to effectively capture and convey 3D object details, hinting at its potential for scalable, human-like captioning of 3D objects. More win rate comparisons are detailed in the supplementary material.

Limitations of traditional metrics. Our evaluation also highlights the limitations of conventional NLP metrics like BLEU-1, ROUGE-L, and METEOR. These metrics are biased toward shorter captions. For instance, 3D-LLM achieves higher scores on these metrics by producing shorter captions (averaging 20 words compared to others' 69+) that do not necessarily indicate superior quality, as confirmed by human evaluation. To further verify this, we prompt PointLLM-13B to generate captions with no more than 20 words, which improved these metric scores significantly, as shown in Tab. 3. However, this preference for short captions contradicts our benchmark, which necessitates that models produce detailed captions to demonstrate a comprehensive understanding of point clouds. Also, these metrics often fail to capture the semantic similarity or diversity of

Table 4: Biased Traditional metrics for different captions. The biased scores demonstrate the limitations of these metrics.

Caption	B-1.	R-L.	MET.
Private jet	**100.00**	**100.00**	**100.00**
there is a black jet engine in a dark background	10.00	18.18	17.86
This is a 3D model of a cartoon-style commercial airplane.	0.00	0.00	0.00

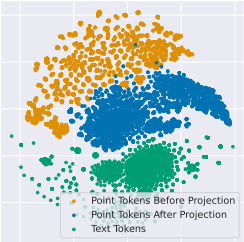


Fig. 4: T-SNE visualization of tokens.

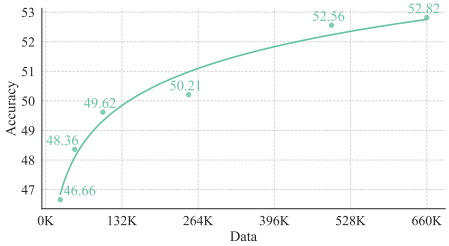


Fig. 5: Ablation on data for alignment.

the captions as they primarily measure the overlap of n-grams or their varieties. An example in Tab. 4 highlights this: inaccurately describing a "Private jet" as a "jet engine" scores higher compared to accurately identifying it as an "airplane". It's worth noting that these metrics are proposed for machine translation, not captioning. Therefore, we prioritize more comprehensive and reliable measures like human and GPT-4 evaluation along with Sentence-BERT and SimCSE.

5.4 Ablation Studies

In this section, we explore various model design choices. Additionally, we examine the impact of different data variations on the training process. The average accuracy of PointLLM on our generative classification benchmark is reported.

Projection layers. While the alignment of tokens from different modalities to the text space using projection layers is effective and widely used in various domains [31,61], the optimal number of layers required remains an open question. Our experiments, ranging from 1 to 4 projection layers with different hidden dimensions, are detailed in Tab. 5. Results from both the 7B and 13B models indicate that 3 projection layers yield the best performance. This suggests that both an insufficient and an excessive number of layers can detrimentally affect performance. A balance in the number of layers is thus crucial for optimal model functionality. We also investigate the projection layers' impact by visualizing point tokens' features pre- and post-projection, and text tokens using T-SNE as shown in Fig. 4. Post-projection, point tokens cluster more and align closer with text tokens, verifying the effect of aligning feature spaces. The non-complete overlap may result from the generative, not contrastive, alignment.

Max pooling. Unlike sequential or grid-based text and image tokens, point cloud tokens are permutation invariant. Concatenating these tokens with text

Table 5: Projection layers.

Hidden Dims.	7B-Acc.	13B-Acc.
N.A.	50.63	52.62
1024	51.05	49.00
1024, 2048	**52.82**	**53.39**
1024, 2048, 4096	52.15	51.40

Table 6: Max pooling.

Pooling	Acc.	A100 GPU-Hours
7B w/	48.72	**34**
7B w/o	**52.82**	126
13B w/	51.10	**56**
13B w/o	**53.39**	213

Table 7: Fine-tuning data.

Single	Multi.	Detailed	Accuracy
✓			40.14
✓	✓		45.79
✓	✓	✓	**52.82**

introduces unnecessary causal dependence and may not be optimal for feature fusion. Inspired by max pooling's symmetric properties [40], we experimented with aggregating point token information through max pooling before the projection layer. While this method didn't enhance performance as shown in Tab. 6, it greatly improved efficiency. Training time measured by 80G-A100 GPU-Hour reduced by about 75%. This underscores the necessity in developing efficient point cloud fusion mechanisms for MLLMs.

Training data. To determine the optimal quantity of data for feature alignment, we experimented with varying data volumes on our 7B PointLLM, maintaining constant iteration times by duplicating training epochs. Results in Fig. 5 suggest that increasing the data volume improves downstream performance, plateauing at around 600K samples. Further, as shown in Tab. 7, incorporating more types of data during fine-tuning consistently yields performance improvements, underscoring the importance of our diverse instruction-following dataset.

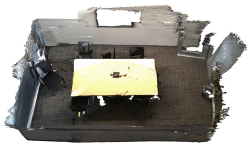
5.5 Qualitative Results

Fig. 1 demonstrates PointLLM's ability to accurately perceive interior details of shoes and cars, overcoming occlusion and viewpoint challenges. More qualitative comparisons of different 13B models are shown in Tab. 8. Sample 1 from ModelNet40 shows a typical 2D MLLM failure: mistaking a laptop for letters due to depth perception issues inherent in single-view images. While multiple views could potentially alleviate this, they pose challenges in terms of optimal view selection and increased model complexity. Point clouds, however, directly provide object geometry, avoiding issues with depth, occlusion, or viewpoint. Sample 2 highlights PointLLM's capability to generate detailed, accurate captions, outperforming other models and even human annotations, while avoiding severe hallucinations. Notably, despite being trained exclusively on Objaverse, PointLLM can handle scene-level point clouds from the unseen ScanNet [7] with reasonable outputs, indicating its potential for broader applications. Effectively handling scene-level point clouds necessitates more high-quality data, a resource currently unavailable. We leave it as future work. Additional qualitative results in the supplementary material further illustrate the advantages of using point clouds for 3D understanding and PointLLM's superiority.

6 Conclusions and Future Directions

In this study, we present PointLLM, a novel and powerful MLLM designed for understanding 3D object point clouds, alongside an automated data genera-

Table 8: Qualitative comparisons. We show the qualitative results of models on ModelNet40 [53], Objaverse [9], and ScanNet [7]. Our PointLLM produces more accurate and detailed results than baselines and even human-annotated ground truths.

Samples 1, 2		
Ground Truth	Laptop	A cartoon black monster like a dragon
Prompt	What is this?	Briefly caption this 3D model.
InstructBLIP [8]	symbol letter l	a black lizard with a sharp tooth in a dark room
LLaVA [31]	A small, grainy, black and white letter j.	A 3D model of a dark, menacing dragon.
3D-LLM [21]	-	A black and white tiger with long legs, standing on its hind leg.
Point-Bind LLM [15]	This is a laptop computer.	The 3D model features a large, ornate gargoyle with a horned helmet, sitting on top of a building.
PointLLM	**The 3D model represents a notebook computer, typically a laptop.**	**The 3D model depicts a menacing black dragon, with its mouth opened wide revealing a row of sharp teeth.**
PointLLM	(The outputs for ScanNet-Scene0611_00 are shown below.)	
	**This is a 3D model of an interior space in a building, featuring a table and chairs. The room is designed in a minimalistic style, with light-colored walls and dark-colored chairs. The table, presumably made of wood, is the focal point of the room, providing a space for various activities such as dining, study or work. The chairs, looking comfortable and sturdy, suggest a space designed for prolonged sitting.**	

tion pipeline and a large-scale dataset. We unveil two innovative benchmarks equipped with a comprehensive evaluation framework, also highlighting the current benchmarks and metrics' limitations. All resources will be open-source. Looking ahead, we aim to refine PointLLM's comprehension of scene-level point clouds and extend its capabilities to include point cloud generation for interactive 3D content creation. Another exciting direction is leveraging PointLLM to generate high-quality 3D object captions at scale, benefiting text-to-3D applications, for which we provide preliminary results in the supplementary material.

Acknowledgements. We would like to acknowledge Xiangyu Yue for providing feedback about this paper, and thank Lihe Ding, Shaocong Dong, and Jiaming Han for their assistance with the experiments. This research was partially supported by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd. under the Innovation and Technology Commission (ITC)'s InnoHK and Shanghai Artificial Intelligence Laboratory.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning (2022)
2. Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., et al.: Openflamingo: An open-source framework for training large autoregressive vision-language models. arXiv:2308.01390 (2023)
3. Banerjee, S., Lavie, A.: METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: ACL Workshop (2005)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners (2020)
5. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (2023), `https://lmsys.org/blog/2023-03-30-vicuna/`
6. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv:2204.02311 (2022)
7. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017)
8. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P.N., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. NeurIPS (20243)
9. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: CVPR (2023)
10. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023)
11. Gao, P., Han, J., Zhang, R., Lin, Z., Geng, S., Zhou, A., Zhang, W., Lu, P., He, C., Yue, X., Li, H., Qiao, Y.: Llama-adapter v2: Parameter-efficient visual instruction model. arXiv:2304.15010 (2023)
12. Gao, T., Yao, X., Chen, D.: Simcse: Simple contrastive learning of sentence embeddings. arXiv:2104.08821 (2021)
13. Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K.V., Joulin, A., Misra, I.: Imagebind: One embedding space to bind them all. In: CVPR (2023)
14. Gong, T., Lyu, C., Zhang, S., Wang, Y., Zheng, M., Zhao, Q., Liu, K., Zhang, W., Luo, P., Chen, K.: Multimodal-gpt: A vision and language model for dialogue with humans. arXiv:2305.04790 (2023)
15. Guo, Z., Zhang, R., Zhu, X., Tang, Y., Ma, X., Han, J., Chen, K., Gao, P., Li, X., Li, H., 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)
16. Gupta, T., Kembhavi, A.: Visual programming: Compositional visual reasoning without training. In: CVPR (2023)
17. Han, J., Zhang, R., Shao, W., Gao, P., Xu, P., Xiao, H., Zhang, K., Liu, C., Wen, S., Guo, Z., et al.: Imagebind-llm: Multi-modality instruction tuning. arXiv:2309.03905 (2023)

18. Hao, Y., Song, H., Dong, L., Huang, S., Chi, Z., Wang, W., Ma, S., Wei, F.: Language models are general-purpose interfaces. arXiv:2206.06336 (2022)
19. Hegde, D., Valanarasu, J.M.J., Patel, V.: Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition. In: ICCV (2023)
20. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv:1606.08415 (2016)
21. Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., Gan, C.: 3d-llm: Injecting the 3d world into large language models (2023)
22. Huang, R., Li, M., Yang, D., Shi, J., Chang, X., Ye, Z., Wu, Y., Hong, Z., Huang, J., Liu, J., et al.: Audiogpt: Understanding and generating speech, music, sound, and talking head. arXiv:2304.12995 (2023)
23. Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O.K., Liu, Q., et al.: Language is not all you need: Aligning perception with language models. arXiv:2302.14045 (2023)
24. Huang, T., Dong, B., Yang, Y., Huang, X., Lau, R.W., Ouyang, W., Zuo, W.: Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In: ICCV (2023)
25. Jiang, B., Chen, X., Liu, W., Yu, J., Yu, G., Chen, T.: Motiongpt: Human motion as a foreign language. arXiv:2306.14795 (2023)
26. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. ICCV (2023)
27. Li, B., Zhang, Y., Chen, L., Wang, J., Pu, F., Yang, J., Li, C., Liu, Z.: Mimic-it: Multi-modal in-context instruction tuning. arXiv:2306.05425 (2023)
28. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2023)
29. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022)
30. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out (2004)
31. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023)
32. Liu, M., Shi, R., Kuang, K., Zhu, Y., Li, X., Han, S., Cai, H., Porikli, F., Su, H.: Openshape: Scaling up 3d shape representation towards open-world understanding. arXiv preprint arXiv:2305.10764 (2023)
33. Luo, T., Rockwell, C., Lee, H., Johnson, J.: Scalable 3d captioning with pretrained models. arXiv:2306.07279 (2023)
34. OpenAI: Chatgpt. `https://openai.com/blog/chatgpt` (2022)
35. OpenAI: Gpt-4 technical report. arXiv:2303.08774 (2023)
36. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback (2022)
37. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL (2002)
38. Patil, S.G., Zhang, T., Wang, X., Gonzalez, J.E.: Gorilla: Large language model connected with massive apis. arXiv preprint arXiv:2305.15334 (2023)
39. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Wei, F.: Kosmos-2: Grounding multimodal large language models to the world. arXiv:2306.14824 (2023)
40. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017)

41. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
42. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. In: JMLR (2020)
43. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv:1908.10084 (2019)
44. Su, Y., Lan, T., Li, H., Xu, J., Wang, Y., Cai, D.: Pandagpt: One model to instruction-follow them all. arXiv:2305.16355 (2023)
45. Sun, Q., Li, Y., Liu, Z., Huang, X., Liu, F., Liu, X., Ouyang, W., Shao, J.: Unig3d: A unified 3d object generation dataset. arXiv:2306.10730 (2023)
46. Surís, D., Menon, S., Vondrick, C.: Vipergpt: Visual inference via python execution for reasoning. arXiv:2303.08128 (2023)
47. Team, I.: Internlm: A multilingual language model with progressively enhanced capabilities. `https://github.com/InternLM/InternLM` (2023)
48. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv:2302.13971 (2023)
49. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need (2017)
50. Wang, H., Tang, J., Ji, J., Sun, X., Zhang, R., Ma, Y., Zhao, M., Li, L., Zhao, Z., Lv, T., et al.: Beyond first impressions: Integrating joint multi-modal cues for comprehensive 3d representation. In: ACM MM (2023)
51. Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., et al.: Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. arXiv:2305.11175 (2023)
52. Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., Duan, N.: Visual chatgpt: Talking, drawing and editing with visual foundation models. arXiv:2303.04671 (2023)
53. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: CVPR (2015)
54. Xue, L., Gao, M., Xing, C., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S.: Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In: CVPR (2023)
55. Xue, L., Yu, N., Zhang, S., Li, J., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S.: Ulip-2: Towards scalable multimodal pre-training for 3d understanding. arXiv:2305.08275 (2023)
56. Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., Chen, E.: A survey on multimodal large language models. arXiv:2306.13549 (2023)
57. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J.: Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In: CVPR (2022)
58. Zhang, R., Guo, Z., Zhang, W., Li, K., Miao, X., Cui, B., Qiao, Y., Gao, P., Li, H.: Pointclip: Point cloud understanding by clip. In: CVPR (2022)
59. Zhang, R., Han, J., Liu, C., Gao, P., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Qiao, Y.: Llama-adapter: Efficient fine-tuning of language models with zero-init attention. arXiv:2303.16199 (2023)
60. Zhang, S., Sun, P., Chen, S., Xiao, M., Shao, W., Zhang, W., Chen, K., Luo, P.: Gpt4roi: Instruction tuning large language model on region-of-interest. arXiv:2307.03601 (2023)

61. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv:2304.10592 (2023)
62. Zhu, X., Zhang, R., He, B., Guo, Z., Zeng, Z., Qin, Z., Zhang, S., Gao, P.: Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In: ICCV (2023)