More Text, Less Point: Towards 3D Data-Efficient Point-Language Understanding

Yuan Tang[1,*], **Xu Han**[1,*], **Xianzhi Li**[1,†], **Qiao Yu**[1], **Jinfeng Xu**[1],
Yixue Hao[1], **Long Hu**[1], **Min Chen**[2]

[1]Huazhong University of Science and Technology,
[2]South China University of Technology,
{yuan_tang, xhanxu, xzli, qiaoyu_epic, yixuehao, hulong}@hust.edu.cn, jinfengxu.edu@gmail.com, minchen@ieee.org

Abstract

Enabling Large Language Models (LLMs) to comprehend the 3D physical world remains a significant challenge. Due to the lack of large-scale 3D-text pair datasets, the success of LLMs has yet to be replicated in 3D understanding. In this paper, we rethink this issue and propose a new task: 3D Data-Efficient Point-Language Understanding. The goal is to enable LLMs to achieve robust 3D object understanding with minimal 3D point cloud and text data pairs. To address this task, we introduce GreenPLM, which leverages more text data to compensate for the lack of 3D data. First, inspired by using CLIP to align images and text, we utilize a pre-trained point cloud-text encoder to map the 3D point cloud space to the text space. This mapping leaves us to seamlessly connect the text space with LLMs. Once the point-text-LLM connection is established, we further enhance text-LLM alignment by expanding the intermediate text space, thereby reducing the reliance on 3D point cloud data. Specifically, we generate 6M free-text descriptions of 3D objects, and design a three-stage training strategy to help LLMs better explore the intrinsic connections between different modalities. To achieve efficient modality alignment, we design a zero-parameter cross-attention module for token pooling. Extensive experimental results show that GreenPLM requires only 12% of the 3D training data used by existing state-of-the-art models to achieve superior 3D understanding. Remarkably, GreenPLM also achieves competitive performance using text-only data.

Code — https://github.com/TangYuan96/GreenPLM

Introduction

Recent advancements in large language models (LLMs) have revolutionized natural language processing, demonstrating emergent intelligence and exceptional capabilities in language understanding and generation (OpenAI 2023; Yang et al. 2024; Dubey et al. 2024). However, LLMs are *blind* to the 3D physical world because they lack the ability to capture and understand 3D objects. Solving this challenging multimodal 3D-language understanding task could benefit many applications, such as autonomous driving, robotics and embodied AI (Driess et al. 2023; Fu et al. 2024).

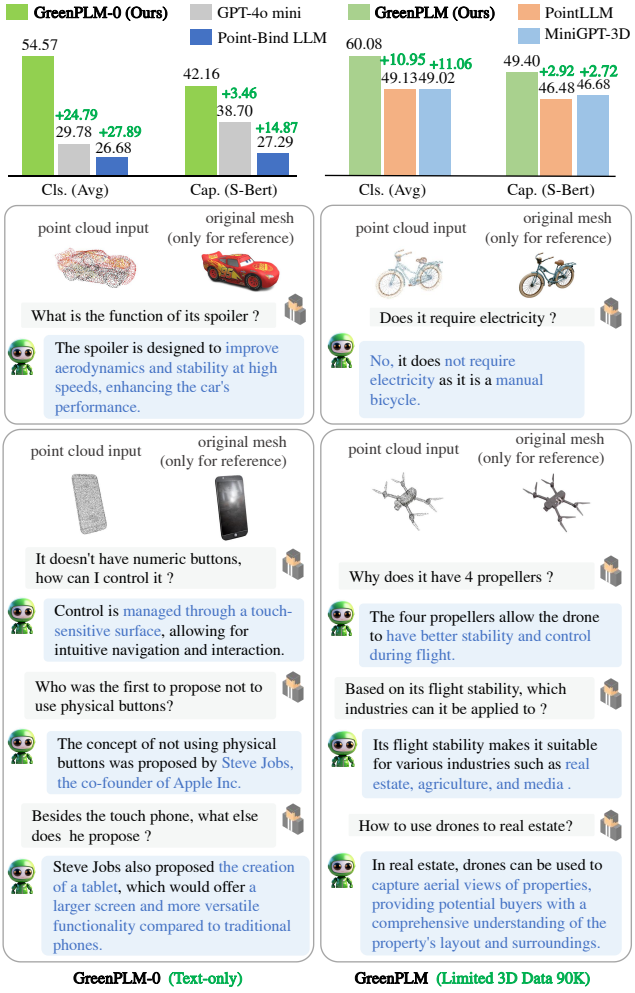


Figure 1: We propose GreenPLM, which expands the text space to reduce the need for 3D data. GreenPLM achieves strong 3D understanding using just 12% of the 3D data or even with text-only data.

Inspired by CLIP (Radford et al. 2021), multimodal large language models (MLLMs) can map different modality inputs to a text space closer to LLMs using pre-trained multimodal encoders, enabling LLMs to understand data beyond

[*]Equal contribution.
[†]Corresponding author.

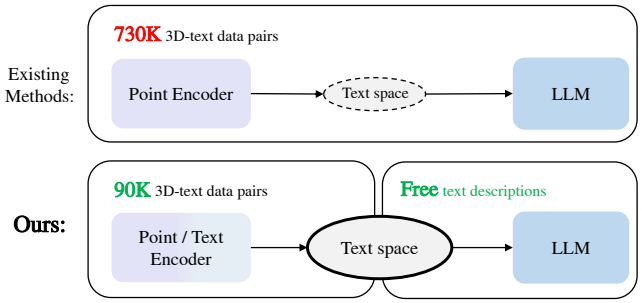


Figure 2: Existing methods like PointLLM use massive 3D-text data (∼730K) to enhance the point-text mapping, therefore realize point-language understanding, while we can also achieve this with only a small number of 3D data (∼90K) and free-text descriptions for better point-LLM alignment.

just language. Existing 3D point-language models follow a similar approach, applying LLMs to 3D understanding by learning from 3D point-text data pairs (Luo et al. 2024; Qi et al. 2024b). For example, PointLLM (Xu et al. 2023) and ShapeLLM (Qi et al. 2024a) employ pre-trained multimodal point cloud encoders (Xue et al. 2024; Qi et al. 2024a), mapping the point cloud space to the text space. This leaves the alignment of point cloud with LLMs to only align the text space with LLMs, which is relatively easier for LLMs. Finally, they propose to train the 3D-LLMs with large amount of 3D-text data pairs, thus enhancing the LLMs' 3D understanding capabilities. However, this field remains underexplored. The primary reason is that training LLMs requires billions of datas, while 3D-text pair data is scarce because 3D data itself is hard to acquire and requires expensive annotations. Consequently, the scaling law that drives LLMs success are difficult to achieve in the 3D domain, directly limiting the development of 3D foundation models.

In this paper, we revisit the 3D data bottleneck and pose a question: ***Can we achieve robust 3D understanding with minimal 3D data?*** To answer this question, we propose a new task: 3D Data-Efficient Point-Language Understanding (3DEPL). The goal is to enable LLMs to achieve robust 3D understanding using as little 3D point cloud-text data pairs as possible. This requires the model to explore the intrinsic connections between different modalities, and effectively leverage the powerful language comprehension capabilities of LLMs to achieve data-efficient 3D understanding.

To address this data-limited multimodal alignment problem, we propose GreenPLM. Intuitively, as shown in Fig. 2, we observe that after establishing the *point-text-LLM* connection, instead of increasing point-text data pairs to optimize the *point-text* mapping like in existing methods (Xu et al. 2023; Qi et al. 2024a), we can also enhance the *text-LLM* alignment by simply adding more text data. This approach can also improve the point-LLM alignment and, more importantly, reduce the reliance on point-text data pairs, shifting the data bottleneck from expensive and scarce 3D-text data to abundant and cheap text data. That is, the text-LLM alignment method fits perfectly with the goal of 3D data-efficient point-language understanding, also offers an alternative solution for aligning point clouds with LLMs, enabling GreenPLM to achieve robust 3D understanding even with limited 3D data.

In detail, GreenPLM solves the 3DEPL task with key techniques across three perspectives: data, training strategy, and model architecture. (1) We bring T3D dataset, a 6M text dataset of 3D object descriptions and conversations for free, the largest to our knowledge, to expand the text space for better *text-LLM* alignment and compensate for the scarcity of expensive 3D data. (2) We propose a 3-stage training strategy designed to help LLMs better uncover the intrinsic connections between different modalities. Specifically, we propose a coarse-to-fine training approach, progressing from data to model. The first two stages fine-tune the LLMs with text-only data, while the final stage uses minimal 3D data for further point-LLMs alignment. (3) From the architecture's perspective, we design a parameter-free cross-attention module for token pooling, namely 0M-Pooling, which better utilizes the encoder's output tokens, thereby aligning point clouds with LLMs more effectively. This, we can achieve excellent performance with only an efficient LLM (Abdin et al. 2024). Together, we can complete training in just 26.6 hours using a single 3090 GPU (24GB), leaving opportunities for efficient end-side deployment.

To fairly and reasonably evaluate the models, we introduce a new metric to measure the efficiency of 3D data usage, and establish a new evaluation benchmark based on open-source LLMs. Experimental results show that our GreenPLM outperforms previous models using only 12% of the 3D data. It even surpasses GPT4Point (660K) (Qi et al. 2024b) without any 3D data, maintaining extremely 3D data-efficient point-language understanding, which demonstrates the effectiveness of our approach. The contributions of this paper are as follows:

- We introduce a new task of 3D data-efficient point-language understanding, aiming to enable LLMs to achieve robust 3D understanding with minimal 3D data.
- We propose GreenPLM to tackle this 3D data-limited task from a novel perspective, enhancing point-LLM alignment with more free-text data. Specifically, we introduce a 6M T3D dataset, design a 3-stage training strategy, and present a 0M-Pooling module for token pooling.
- We introduce the Accuracy-to-3D-Data Ratio (A3DR) to measure the efficiency of 3D data usage and establish an evaluation benchmark based on open-source LLMs.
- GreenPLM outperforms previous models using only 12% of 3D data and even surpasses GPT4Point (660K 3D data) using only text, demonstrating superior 3D data efficiency.

Related Work

3D Point-Language Understanding

To enable LLMs to understand the 3D physical world, early attempt (Hong et al. 2023) projects 3D point clouds into 2D images, relying on 2D-LLMs for comprehension. However, 2D-based method lose crucial 3D information, leading to issues like occlusion, ambiguity, and hallucination. Point-Bind LLM (Guo et al. 2023) attempts to establish a 3D-2D-LLM connection, but this non-robust link leads to unstable

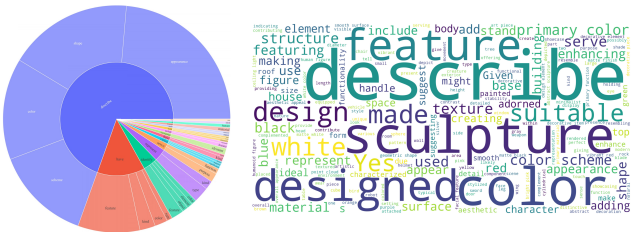


Figure 3: T3D dataset distribution.

Data Type	Size	Sample
Caption	1M	A 3D model of a friendly white dog...
Brief Desc.	1M	Q: Summarize the 3D point cloud object briefly. A: The 3D object is a white dog model...
Detail Desc.	1M	Q: Offer a detailed description of this point cloud. A: This 3D object depicts a white dog sitting...
Single Conv.	3M	Q: What is the posture of the dog? A: The dog is sitting upright. ...
Multi Conv.	1M	Q1: What type of animal does the 3D model represent? A1: The 3D model represents a dog. Q2: Can you describe the dog's posture? A2: The dog is sitting upright. ... ...

Table 1: 3D object description and conversations of T3D.

performance. Recently, with the availability of large-scale 3D-text data (Luo et al. 2024; Qi et al. 2024b) and multi-modal encoders, methods like PointLLM (Xu et al. 2023) and ShapeLLM (Qi et al. 2024a) connect point encoders with LLMs and fine-tune the 3D Point Cloud-LLMs (3D-LLMs) using vast amounts of 3D-text data. Unfortunately, compared to images, 3D-text data remains extremely scarce (LAION-5B vs. Objaverse-1M) (Schuhmann et al. 2022; Deitke et al. 2023) and expensive, let alone the near infinite and free text data, making it challenging to build powerful 3D foundation models according to the scaling law. Also, training 3D-LLMs is resource-intensive, often requiring 8xA100 GPUs for hundreds of hours. Although MiniGPT-3D (Tang et al. 2024) reduces training time to 26.8h on a single GPU, the 3D data bottleneck persists. Our GreenPLM proposes to solve this 3D data bottleneck by leveraging extensive text data to compensate for the lack of 3D data, and introducing a 3-stage training strategy for effective and efficient alignment.

Multimodal Encoders in 3D-LLM

The encoder maps raw data into a more compact embedding space, which can then be aligned with LLMs. To reduce the training cost, one can intuitively employ a multimodal pre-trained encoder, such as CLIP (Radford et al. 2021), which has been trained on text-image pairs, for aligning 2D images with LLMs. This makes it easier to align data from different modalities with LLMs. Similarly, some existing 3D-LLMs use multimodal pre-trained encoders (Huang et al. 2023; Xue et al. 2023; Qi et al. 2023; Gao et al. 2024; Chen et al. 2024a) to map point clouds into embedding space, followed by fine-tuning the 3D-LLM. However, even without training the encoder, constructing the 3D-LLM still requires a vast amount of point-text data (Xu et al. 2023; Zhou et al. 2023; Qi et al. 2024a; Tang et al. 2024). We observe that existing methods underutilize the potential of the text encoder, only focusing on aligning point encoder with LLM. In contrast, we propose leveraging the cost-efficient text space and the text encoder to reduce the dependency on 3D data.

Method

To enable LLMs to achieve robust 3D understanding with minimal 3D data, we propose using more text data to reduce reliance on 3D data. First, we generate a 6M text dataset of 3D object descriptions and conversations. Then, to better uncover connections between different modalities, we design a 3-stage training strategy. Finally, we introduce a parameter-free token pooling module to efficiently utilize information from the encoder's output token sequence. The details of these three parts are as follows.

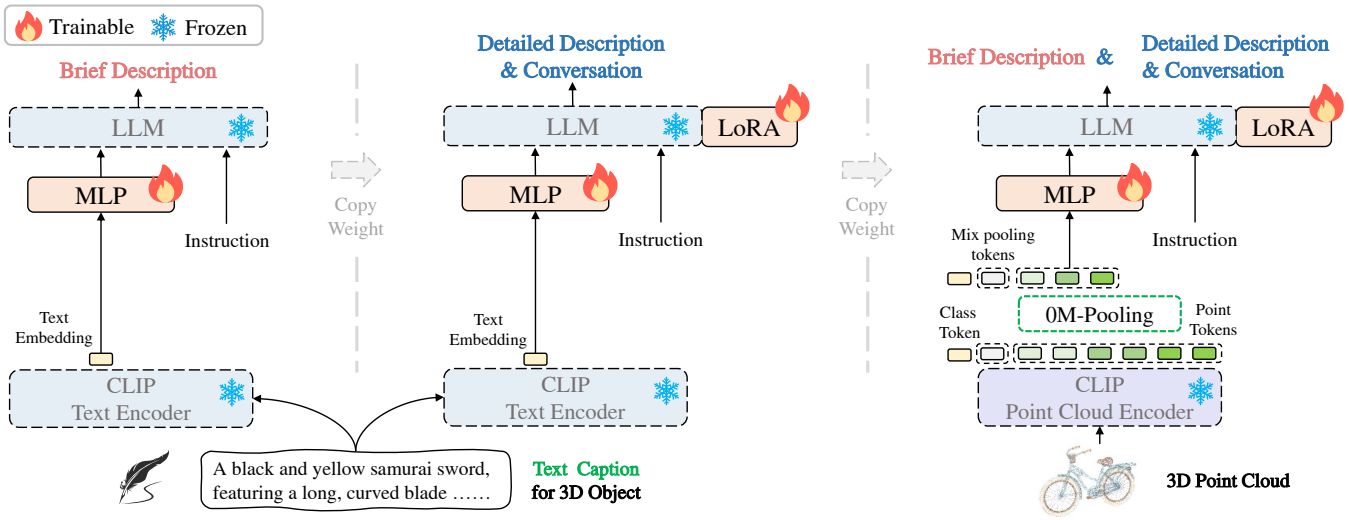
3D Object Description and Conversation Dataset

Leveraging multimodal pre-trained encoders, we propose using large amounts of text data to compensate for the lack of 3D data pairs. Specifically, we first align the text encoder with the LLM using extensive text data. Since the text encoder is already aligned with the point encoder, we then only need a small amount of 3D data for point encoder-LLM alignment.

To achieve this, we bring T3D, a 6M text dataset of 3D object descriptions and conversations. Fig. 3 shows the verb-noun distribution and a visualized word cloud. Instead of using the closed-source GPT-4 (OpenAI 2023), we use the equally powerful open-source model Qwen2-72B-Instruct (Yang et al. 2024) to construct this dataset. We select object categories from Cap3D (Luo et al. 2024) and DiffuRank (Luo, Johnson, and Lee 2024), and we design prompts to generate 5 types of data: 1M captions, 1M brief descriptions, 1M detailed descriptions, 3M single-round conversations, and 1M multi-round conversations. The object descriptions help the LLMs learn rich semantic knowledge, while the conversations enable the LLMs to extract useful information from the context to improve 3D understanding. Notably, this dataset is constructed without any manual annotation or post-processing, requiring only minimal model inference cost. Five types of data, totaling 6M samples in the Caption-Question-Answer format, are shown in Table 1. During training, we input the Caption into the text encoder, pass the encoded tokens through a projector, and then input them along with the Question into the LLM, which outputs a response to calculate the loss against the Answer. More detailed prompts and distributions are in Appendix.

3-Stage Training Strategy

For better multimodal encoder-LLM alignment and minimizing the use of 3D point-text data pairs, we propose a 3-stage training strategy, as shown in Fig. 4. Our design principle is to first use a large amount of text data to align the text encoder with the LLM via a MLP projector (Stage I and II). Then, using only a small amount of 3D point-text datas, we align the point cloud encoder with the LLM via the same projector (Stage III). Specifically, for each stage, we will introduce the pipeline, trainable layers, and data aspects as follows.



(a) Stage I (b) Stage II (c) Stage III

Figure 4: Illustration of 3-Stage Training Strategy. We expand the text space by feeding more text data in Stage I & II, thus reduce the demand of 3D data in Stage III. We input the text/point cloud to the encoders, then align with LLM via a MLP projector. Additionally, we design a 0M-Pooling module to efficiently compress the token sequence output by point encoder.

Stage I is shown in Fig. 4(a). First, we input a text caption $\boldsymbol{D}$ of a 3D object into the pre-trained text encoder f_{text}, obtaining the global text embedding $\boldsymbol{C}_t$ as the encoder output. $\boldsymbol{C}_t$ is then passed through a learnable MLP projector f_{proj} to connect with the LLM f_{LLM}. The LLM input consists of the projector output $f_{proj}(\boldsymbol{C}_t)$, and the text tokens of an instruction prompt $\boldsymbol{I}$, such as "What is this?". Finally, the LLM outputs a brief description $\boldsymbol{R}_{brief}$ of the 3D object, which can be used to calculate the loss with the ground-truth description. The formulas are as follows:

$$\boldsymbol{C}_t = f_{text}(\boldsymbol{D}), \tag{1}$$

$$\boldsymbol{R}_{brief} = f_{LLM}(f_{proj}(\boldsymbol{C}_t), \mathrm{h}(\boldsymbol{I})), \tag{2}$$

where h is the LLM's tokenizer.

***Trainable Layers & Data*:** Note that, only the projector f_{proj} is a trainable MLP, while the rest, including the text encoder f_{text} and LLM f_{LLM}, have frozen weights. We train the model using a large dataset of brief descriptions (1M) from our T3D dataset, as shown in Tab. 1.

Stage II is shown in Fig. 4(b), Stage II is similar to Stage I. We also first input a caption of a 3D object into the text encoder f_{text}, then extract the global text embedding and pass it to the projector f_{proj}. The projector output, along with a complex instruction, is then fed to the LLM f_{LLM}. Finally, the LLM outputs detailed description and conversation results, which are then used to calculate the loss.

***Trainable Layers & Data*:** The differences from Stage I are as follows: (1) The weights of the projector f_{proj} are copied from Stage I for initialization and remain trainable. (2) We use LoRA (Hu et al. 2021) to train the LLM f_{LLM} in this stage to achieve better multimodal alignment. The text encoder f_{text} remains frozen. We use only 210K detailed descriptions and conversation data for 3D objects from our T3D dataset, such as describing an object in $\sim$50 words and engaging in multi-turn conversations, as shown in Tab. 1.

Notably, to enhance the perception robustness of the LLM, we add Gaussian noise to the encoder's output features to simulate the semantic discrepancies between different modalities, inspired by Chen et al. (2024b). After two stages of pure text training, our GreenPLM acquires the ability to comprehend raw 3D point clouds by directly replacing the text encoder f_{text} with a point encoder f_{pc} without weight tuning.

Stage III is shown in Fig. 4(c), we use 3D point cloud as input. The 3D point cloud $\boldsymbol{P}$ is fed into the point cloud encoder f_{pc} to output a token sequence. Unlike previous stages that use only the global text embedding (corresponding to the class token in the point encoder) for the projector, in this stage, we extract representations from all tokens $\boldsymbol{T}_{pc}$ to more effectively leverage information from the point encoder. To reduce the token sequence length for efficiency, we introduce a parameter-free token pooling module based on cross-attention, namely 0M-Pooling, which compresses the token length from 512 to 32. The pooled point tokens $\boldsymbol{T}^p_{pc}$, along with three tokens from Mix-pooling and the class token $\boldsymbol{C}_{pc}$, are input to the projector. Thus, the projector f_{proj} receives 32+3+1=36 tokens. We then feed the projector's output, along with the instruction $\boldsymbol{I}$, into f_{LLM} to generate the predict responses $\boldsymbol{R}_{pred}$ of descriptions or conversations. The responses will be used to compute loss with the ground truth. This stage can be formulated as:

$$[\boldsymbol{C}_{pc}, \boldsymbol{T}_{pc}] = f_{pc}(\boldsymbol{P}), \quad \boldsymbol{T}^p_{pc} = \text{0M-Pooling}(\boldsymbol{T}_{pc}), \tag{3}$$

$$\boldsymbol{R}_{pred} = f_{LLM}\left(f_{proj}\left(\boldsymbol{C}_{pc}, \mathrm{Mix}\left(\boldsymbol{T}_{pc}\right), \boldsymbol{T}^p_{pc}\right), \mathrm{h}(\boldsymbol{I})\right), \tag{4}$$

where Mix represents Mix-pooling of max, mean, and sum.

***Trainable Layers & Data*:** Similar to Stage II, the weights of projector f_{proj} here are copied from the previous stage

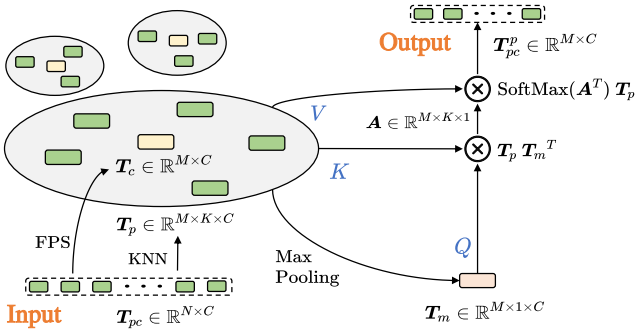


Figure 5: Illustration of 0M-Pooling, which compresses N tokens to M tokens ($M << N$).

and then still kept trainable. We continue using LoRA (Hu et al. 2021) to train f_{LLM} for efficient point-LLM alignment. The point cloud encoder f_{pc} remain frozen. In this stage, we train using only a small amount of 3D-text pairs (90K).

Loss Function For all training stages, given a pair of LLM output $\boldsymbol{R}$ and text ground truth $\boldsymbol{y}$, GreenPLM is optimized under a causal language modeling objective (Liu et al. 2018):

$$\mathcal{L} = \text{CrossEntropyLoss}\left(\boldsymbol{R}, \text{h}\left(\boldsymbol{y}\right)\right), \tag{5}$$

where CrossEntropyLoss is the cross-entropy loss, and h denotes the LLM's tokenizer.

0M-Pooling

To fully leverage the output of the point cloud encoder, we extract information from all output tokens $\boldsymbol{T}_{pc}$, not just the class token, while reducing computational load. As shown in Fig. 5, we design a zero-parameter token pooling module based on cross-attention, namely 0M-Pooling, which compresses the 512 output tokens down to 32 tokens, without introducing any learnable parameters, defined as:

$$\begin{aligned} \boldsymbol{T}_c &= \text{FPS}(\boldsymbol{T}_{pc}), & \boldsymbol{T}_p &= \text{KNN}(\boldsymbol{T}_c, \boldsymbol{T}_{pc}), \\ \boldsymbol{T}_m &= \text{MaxPool}(\boldsymbol{T}_p), & \boldsymbol{T}_{pc}^{p} &= \text{SoftMax}((\boldsymbol{T}_p \boldsymbol{T}_m^T)^T)\boldsymbol{T}_p, \end{aligned} \tag{6}$$

where $\boldsymbol{T}_{pc} \in \mathbb{R}^{N\times C}$ ($N = 512$) is the output point token sequence of the point cloud encoder, $\boldsymbol{T}_c \in \mathbb{R}^{M\times C}$ ($M = 32$) is the central token gained via farthest point sampling (FPS) from T_{pc}, and $\boldsymbol{T}_p \in \mathbb{R}^{M\times K\times C}$ ($K = 8$) represents the K-Nearest Neighborhood (KNN) tokens of $\boldsymbol{T}_c$ within $\boldsymbol{T}_{pc}$. Then, we pass $\boldsymbol{T}_p$ to Max Pooling on the K dimension to get $\boldsymbol{T}_m \in \mathbb{R}^{M\times 1\times C}$. Finally, we use cross-attention in Equ.(6) to aggregate information from $\boldsymbol{T}_{pc} \in \mathbb{R}^{512\times C} \rightarrow \boldsymbol{T}_{pc}^{p} \in \mathbb{R}^{32\times C}$. We obtain the compressed token $\boldsymbol{T}_{pc}^{p}$ using zero trainable parameters. Notely, the $\boldsymbol{T}_{pc}$ input to 0M-Pooling is from the point encoder's second-to-last layer.

Experiment

Implementation details. We use Phi-3 (Abdin et al. 2024) as the LLM backbone, with EVA-CLIP-E (Sun et al. 2023) and ViT (Dosovitskiy et al. 2020) both trained by Uni3D (Zhou et al. 2023) as the text encoder and point encoder, respectively. The point encoder outputs 512+1 tokens, each with $C = 1024$. The MLP projector consists of two linear layers and a GeLU activation, mapping the encoder's output tokens to tokens with 3072 dimensions of Phi-3. Our GreenPLM has 63.3M trainable parameters and requires only 26.6 hours of training on a single 3090 GPU. Besides the standard 3-stage training of GreenPLM, we also train GreenPLM-0 with text-only data, utilizing only Stages I and II. During inference, we simply replace the text encoder in GreenPLM-0 with the point encoder from Uni3D without weight tuning. More detailed training settings are included in Appendix.

Baselines. To validate our 3D data-free capability, we compared GreenPLM-0 with the SoTA 2D-LLMs, InstructBLIP and LLaVA, as well as the 3D-2D-LLM model Point-Bind LLM (Guo et al. 2023). To evaluate GreenPLM with limited 3D data, we choose the SoTA 3D-LLMs PointLLM (Xu et al. 2023) and MiniGPT-3D (Tang et al. 2024). For fairness, we train both using the same 90K limited 3D point-text datas.

Evaluation Settings. An efficient and accurate model evaluation method is a shared goal in the MLLM community. We observe that existing evaluation approaches often rely on GPT-4 and GPT-3.5 to assess the similarity between generated results and ground truth sentences. While this method provides accurate evaluations, it has two major drawbacks: inconsistent API versions and high evaluation costs. For instance, the *GPT-3.5-turbo-0613* model used in PointLLM and MiniGPT-3D is no longer maintained, making it difficult to replicate the results. To address these issues, we propose a new benchmark based on open-source models and introduce a new metric to evaluate data efficiency. Specifically, we use two prompts for the classification task: an Instruction-type (I) prompt, "What is this?", and a Completion-type (C) prompt, "This is an object of.". For the captioning task, we use a single prompt: "Caption this 3D model in detail.". We then replace GPT-4 and GPT-3.5 with the open-source Qwen2-72B-Instruct (Yang et al. 2024) (Qwen2 for short) to evaluate the model's output. We also introduce the Accuracy-to-3D-Data Ratio (A3DR) metric to assess a model's efficiency in utilizing 3D data, defined as follows:

$$\text{A3DR(Acc)} = \frac{2}{1+\exp(-\frac{\lambda\times\text{Acc}}{\text{Size}+\epsilon})} - 1, \tag{7}$$

where Size is the size of 3D data (K), Acc is the accuracy, $\epsilon = 1e-5$ prevents zero division, $\lambda = 3$ adjusts discrimination.

Generative 3D Object Classification

We validate the model's recognition ability by performing the generative 3D object classification task on the ModelNet40 dataset (Wu et al. 2015) and Objaverse dataset (Deitke et al. 2023), using I-type and C-type prompts, with results shown in Tab. 2. For close-set zero-shot classification on ModelNet40, we let Qwen2 select the closest matching category in the 40 classes as the model's output. For open-vocabulary classification on Objaverse, we use Qwen2 to evaluate if the model's output describes the category of ground truth sentence.

As shown in Tab. 2, our GreenPLM-0 achieves an average classification accuracy (AvgAcc) of 54.57% without

Model	Reference	LLM Size	3D Data Size	Input	ModelNet40		Objaverse		Average	A3DR (Avg)
					(I)	(C)	(I)	(C)		
Text-only Data in Training										
InstructBLIP-7B (Dai et al. 2024)	NIPS23	7B	0K	Single-Img.	17.67	22.81	21.50	26.00	22.00	1.000
InstructBLIP-13B (Dai et al. 2024)	NIPS23	13B	0K	Single-Img.	21.56	21.92	21.50	21.50	21.62	1.000
LLaVA-1.5-7B (Liu et al. 2024)	CVPR24	7B	0K	Single-Img.	27.11	21.68	37.50	30.00	29.07	1.000
LLaVA-1.5-13B (Liu et al. 2024)	CVPR24	13B	0K	Single-Img.	27.71	27.76	39.50	35.50	32.62	1.000
GPT-4o mini (Jacob et al. 2024)	OpenAI	-	0K	Single-Img.	22.00	23.10	39.00	35.00	29.78	1.000
Point-Bind LLM (Guo et al. 2023)	arXiv23	7B	0K	Point Cloud	46.60	45.02	7.50	7.58	26.68	1.000
GreenPLM-0 (Ours)	-	3.8B	0K	Point Cloud	**62.60** (+16.00)	**62.68** (+17.66)	**48.00** (+40.50)	**45.00** (+37.42)	**54.57** (+27.89)	**1.000**
Limited 3D Data in Training										
PointLLM-7B (Xu et al. 2023)	ECCV24	7B	90K	Point Cloud	45.22	39.30	59.00	53.00	49.13	0.674
MiniGPT-3D (Tang et al. 2024)	MM24	2.7B	90K	Point Cloud	43.56	43.03	54.50	55.00	49.02	0.673
GreenPLM (Ours)	-	3.8B	90K	Point Cloud	**58.95** (+13.73)	**62.36** (+19.33)	**60.50** (+1.50)	**58.50** (+3.50)	**60.08** (+10.95)	**0.762**
Extensive 3D Data in Training										
GPT4Point (Qi et al. 2024b)	CVPR24	2.7B	660K	Point Cloud	21.39	21.07	49.00	46.50	34.49	0.078
PointLLM-7B (Xu et al. 2023)	ECCV24	7B	730K	Point Cloud	51.34	50.36	62.00	63.00	56.68	0.116
PointLLM-13B (Xu et al. 2023)	ECCV24	13B	730K	Point Cloud	51.70	52.67	61.50	63.00	57.22	0.117
MiniGPT-3D (Tang et al. 2024)	MM24	2.7B	730K	Point Cloud	61.99	60.49	65.00	68.50	64.00	0.131

Table 2: Generative 3D object classification results on the ModelNet40 test split and Objaverse. The accuracy (%) under the **I**nstruction-typed (I) prompt "What is this?" and the **C**ompletion-type (C) prompt "This is an object of" are reported.

Model	Reference	LLM Size	3D Data Size	Input	Qwen2	Sentence-BERT	SimCSE
Text-only Data in Training							
InstructBLIP-7B (Dai et al. 2024)	NIPS23	7B	0K	Single-Img.	16.10	35.79	36.67
InstructBLIP-13B (Dai et al. 2024)	NIPS23	13B	0K	Single-Img.	13.79	33.52	35.60
LLaVA-1.5-7B (Liu et al. 2024)	CVPR24	7B	0K	Single-Img.	17.80	39.32	41.08
LLaVA-1.5-13B (Liu et al. 2024)	CVPR24	13B	0K	Single-Img.	16.00	39.64	40.90
GPT-4o mini (Jacob et al. 2024)	OpenAI	-	0K	Single-Img.	26.00	38.70	39.13
Point-Bind LLM (Guo et al. 2023)	arXiv23	7B	0K	Point Cloud	1.93	27.29	25.35
GreenPLM-0 (Ours)	-	3.8B	0K	Point Cloud	**15.93** (+14.00)	**42.16** (+14.87)	**40.90** (+15.55)
Limited 3D Data in Training							
PointLLM-7B (Xu et al. 2023)	ECCV24	7B	90K	Point Cloud	35.77	46.48	47.01
MiniGPT-3D (Tang et al. 2024)	MM24	2.7B	90K	Point Cloud	35.05	46.68	47.75
GreenPLM (Ours)	-	3.8B	90K	Point Cloud	**42.55** (+6.78)	**49.40** (+2.72)	**49.36** (+1.61)
Extensive 3D Data in Training							
GPT4Point (Qi et al. 2024b)	CVPR24	2.7B	660K	Point Cloud	21.75	41.10	41.24
PointLLM-7B (Xu et al. 2023)	ECCV24	7B	730K	Point Cloud	42.20	48.50	48.92
PointLLM-13B (Xu et al. 2023)	ECCV24	13B	730K	Point Cloud	40.40	49.07	48.41
MiniGPT-3D (Tang et al. 2024)	MM24	2.7B	730K	Point Cloud	48.17	49.54	51.39

Table 3: 3D object captioning results on Objaverse. The results are from Qwen2 evaluation and traditional metrics.

using any 3D data, outperforming all 2D-based models. It surpasses LLaVA-1.5-13B by +21.95 and Point-Bind LLM by +27.89 in AvgAcc. Remarkably, our model also exceeds GPT4Point (660K), which is trained with 660K 3D data, by +20.08 and performs on par with PointLLM-7B (730K). With only a small amount of 3D data (90K), GreenPLM achieves an average accuracy of 60.08%, surpassing PointLLM and MiniGPT-3D by +10.95 and +11.06 in AvgAcc, respectively. GreenPLM even outperforms PointLLM-13B (730K) while using a smaller LLM, and obtains results comparable to SoTA model MiniGPT-3D (730K). Additionally, GreenPLM (90K) outperforms MiniGPT-3D (90K) and MiniGPT-3D (730K) on the A3DR (average accuracy) by +0.089 and +0.631, respectively. These results demonstrate the high 3D data-efficiency of our model.

3D Object Captioning

We evaluate the ability to understand 3D context through a 3D object captioning task, as shown in Tab. 3. Following previous works (Xu et al. 2023; Tang et al. 2024), we assess the similarity between the model's response and the ground truth caption using an LLM, and also evaluate embedding similarity using Sentence-BERT (Reimers and Gurevych 2019) (S-BERT) and SimCSE (Gao, Yao, and Chen 2021).

It is evident that all models without 3D data underperform compared to those trained with 3D data, as they lose significant 3D information. However, our GreenPLM-0 can still outperforms Point-Bind LLM and achieves comparable results to powerful 2D-LLMs by a large margin. When using a small amount of 3D data (90K), our Qwen2 score surpasses MiniGPT-3D (90K) by +7.50, with S-BERT and SimCSE scores also exceeding by +2.72 and +1.61, respectively. Similarly, GreenPLM (90K) achieves a Qwen2 score higher than PointLLM-13B (730K) by +2.15, with S-BERT and SimCSE scores comparable to MiniGPT-3D (730K) while using only 12% of 3D data. These results again demonstrate GreenPLM's ability to efficiently extract 3D information from even small amounts of 3D data or purely text data.

Sample	
Prompt Label	What is this? Guitar
Ins-BLIP	telescope
GPT-4o mini	The image appears to be a simple black outline of a tool or object, possibly resembling the shape of a knife or a similar implement.
P-Bind	This is a bird flying in the sky.
Ours (Text-only)	The object is a Gibson Les Paul electric guitar, predominantly black with a glossy finish. It has a maple fretboard, a single-coil pickup, and a vibrato bridge. The guitar's body is adorned with the Gibson logo, and it includes a strap button for easy carrying.
PointLLM	This 3D model represents a black electric guitar equipped with a distinctive headstock.
Ours (Limited 3D data)	This is a 3D model of a cartoon-style electric guitar. The guitar is predominantly black, giving it a sleek and modern appearance. The design is simplified and stylized, typical of cartoon aesthetics, making it suitable for use in animated films, video games, or other digital media. Despite its cartoonish appearance, it retains the recognizable features of an electric guitar, such as the fretboard and strings.

Table 4: Qualitative comparisons. Conversation example of a guitar in ModelNet40. Our GreenPLM generates more detailed and insightful responses compared to others.

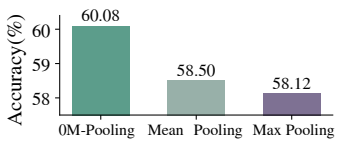


Figure 6: Ablation on 0M-Pooling.

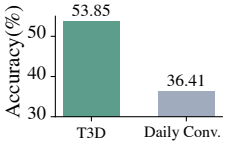


Figure 7: Ablation on T3D caption.

Qualitative Results

Fig. 1 and Tab. 4 present the qualitative results. As shown in Fig. 1, whether trained on text-only or with minimal 3D data, GreenPLM provides accurate, context-aware responses in multi-turn conversations. Tab. 4 shows that our GreenPLM-0 effectively identifies objects and understands details like color and components with text-only data. 2D-based methods like Instruct-BLIP (Ins-BLIP) and GPT-4o mini lose 3D information, suffering from occlusion, ambiguity and severe hallucinations. Point-Bind LLM (P-B LLM) lacks accurate 3D perception due to its non-robust 3D-2D-LLM connection. While using few 3D data (90K), GreenPLM offers significantly more detailed descriptions and better captures local details in point clouds, such as guitar strings and octopus suction cups, compared to PointLLM.

#No.	Stage I	Stage II	Stage III	Acc.
1	✓			53.85
2		✓		47.03
3			✓	45.29
4	✓		✓	58.25
5		✓	✓	42.78
6	✓	✓		**54.57**
7	✓	✓	✓	**60.08**
#No.	Class Token	Global Tokens	Pooled Point Tokens	Acc.
8	✓			38.36
9	✓	✓		45.42
10	✓	✓	✓	**60.08**

Table 5: Ablation on 3-Stage Training and Token Fusion.

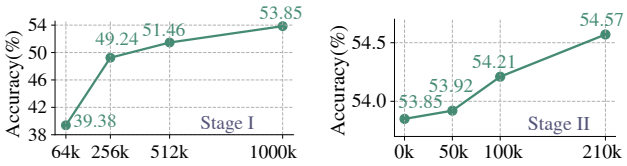


Figure 8: Ablation on Text data size in Stage I & II.

Ablation Study

We conduct ablation experiments on the generative 3D object classification task and report the average accuracy.

Training stages. As shown in Tab. 5, removing any stage reduces performance, with the biggest drop when Stage I is removed. This is because Stage I trains the MLP projector to align the encoder with the LLM. Comparing rows #4 and #7, we observe that Stage II helps the LLM better align with the semantic space. The results of rows #6 and #7 indicate that Stage III injects 3D information into the LLM, significantly enhancing the model's 3D understanding.

0M-Pooling. As shown in Fig. 6, when we replace 0M-Pooling with Max Pooling or Mean Pooling, the accuracy drops by 1.96 and 1.58, respectively, even though the learnable parameters remain zero. This demonstrates that our 0M-Pooling module effectively and efficiently captures point cloud information from the token sequence, enhancing GreenPLM's 3D understanding ability.

T3D dataset. To test the impact of captions in our T3D dataset, which serve as input to the text encoder, we replace captions with low-information sentences in Stage I, and generate a 1M daily conversation dataset (example in Fig. 9). Using daily conversation data causes a significant performance drop in Fig. 7, indicating that captions provide more effective semantic information for the model. Moreover, we assess the impact of text data size in Stages I and II. As shown in Fig. 8, with more text data, the model learns from a larger text space, leading to a stronger point-text-LLM connection. This confirms the effectiveness of the text space, reducing the need for 3D data and addressing the 3DEPL task.

Token Fusion before MLP projector. In Stage III, the tokens input into the MLP projector consist of three parts: the Class token, the Mix-Pooled token, and the 0M-Pooled

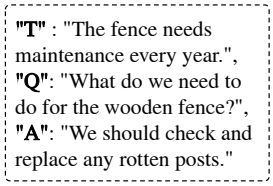


Figure 9: Daily text data.

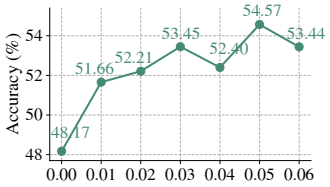


Figure 10: Noise Std.

token. We conduct ablation experiments on these three tokens, as shown in Tab. 5. The results demonstrate that both Mix-Pooling and 0M-Pooling enhance the model's ability to extract information from the token sequence.

Noise level in Stage I & II. Adding Gaussian noise to the token sequence output by the text encoder forces the LLM to learn useful information from noisy data, thereby improving the model's robustness. As shown in Fig. 10, we experiment with different noise levels. As the standard deviation (std) of the noise increases from 0 to 0.06, GreenPLM's accuracy initially increases and then decreases, reaching its peak at 0.05. The results demonstrate that appropriately adding noise can enhance the model's ability to extract cross-modal information, therefore improving its 3D understanding.

Conclusion

To enable LLMs to achieve strong 3D understanding with minimal 3D data, we introduce a new task: 3D Data-Efficient Point-Language Understanding. We propose GreenPLM, which employs a 3-stage training strategy that increases text data in Stage I & II to reduce the need for 3D data in Stage III. We create a 6M T3D dataset and an unified benchmark. Results show that GreenPLM achieves performance comparable to state-of-the-arts using only 12% of the 3D data. Remarkably, our model performs well even without 3D data.

Limitations. Our approach has limitations. Due to time and resource constraints, we couldn't explore all text and 3D data combinations. We believe scaling up either could further improve performance. Additionally, we only test feasibility on small objects, and will explore GreenPLM's potential for larger scenes in future work.

Acknowledgments

This work was supported by the China National Natural Science Foundation No. 62202182, No. 62176101, No. 62276109, and also supported by Guangdong Basic and Applied Basic Research Foundation 2024A1515010224, 2024A1515030017 and 2024A1515011153.

Xianzhi Li is with the school of computer science and technology, Huazhong University of Science and Technology, and also with Guangdong Intelligent Robotics Institute. Long Hu and Yixue Hao are with School of Computer Science and Technology, Huazhong University of Science and Technology, and also with Guangdong HUST Industrial Technology Research Institute. Min Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong 510640, China, and also with Pazhou Laboratory, Guangzhou, Guangdong 510640, China.

References

Abdin, M.; Jacobs, S. A.; Awan, A. A.; Aneja, J.; Awadallah, A.; Awadalla, H.; Bach, N.; Bahree, A.; Bakhtiari, A.; Behl, H.; et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Chen, S.; Chen, X.; Zhang, C.; Li, M.; Yu, G.; Fei, H.; Zhu, H.; Fan, J.; and Chen, T. 2024a. LL3DA: Visual Interactive Instruction Tuning for Omni-3D Understanding Reasoning and Planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26428–26438.

Chen, Y.; Wang, Q.; Wu, S.; Gao, Y.; Xu, T.; and Hu, Y. 2024b. Tomgpt: Reliable text-only training approach for cost-effective multi-modal large language model. *ACM Transactions on Knowledge Discovery from Data*.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P. N.; and Hoi, S. 2024. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36.

Deitke, M.; Schwenk, D.; Salvador, J.; Weihs, L.; Michel, O.; VanderBilt, E.; Schmidt, L.; Ehsani, K.; Kembhavi, A.; and Farhadi, A. 2023. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13142–13153.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Driess, D.; Xia, F.; Sajjadi, M. S.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fu, D.; Li, X.; Wen, L.; Dou, M.; Cai, P.; Shi, B.; and Qiao, Y. 2024. Drive like a human: Rethinking autonomous driving with large language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 910–919.

Gao, T.; Yao, X.; and Chen, D. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Gao, Y.; Wang, Z.; Zheng, W.-S.; Xie, C.; and Zhou, Y. 2024. Sculpting Holistic 3D Representation in Contrastive Language-Image-3D Pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22998–23008.

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

Hong, Y.; Zhen, H.; Chen, P.; Zheng, S.; Du, Y.; Chen, Z.; and Gan, C. 2023. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36: 20482–20494.

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

Huang, T.; Dong, B.; Yang, Y.; Huang, X.; Lau, R. W.; Ouyang, W.; and Zuo, W. 2023. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22157–22167.

Jacob, M.; Kevin, L.; Shengjia, Z.; Eric, W.; Hongyu, R.; Haitang, H.; Nick, S.; and Felipe, P. S. 2024. GPT-4o mini: advancing cost-efficient intelligence. [Online; accessed 16-August-2024].

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26296–26306.

Liu, P. J.; Saleh, M.; Pot, E.; Goodrich, B.; Sepassi, R.; Kaiser, L.; and Shazeer, N. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Luo, T.; Johnson, J.; and Lee, H. 2024. View selection for 3d captioning via diffusion ranking. *arXiv preprint arXiv:2404.07984*.

Luo, T.; Rockwell, C.; Lee, H.; and Johnson, J. 2024. Scalable 3d captioning with pretrained models. *Advances in Neural Information Processing Systems*, 36.

OpenAI, R. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5).

Qi, Z.; Dong, R.; Fan, G.; Ge, Z.; Zhang, X.; Ma, K.; and Yi, L. 2023. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning*, 28223–28243. PMLR.

Qi, Z.; Dong, R.; Zhang, S.; Geng, H.; Han, C.; Ge, Z.; Yi, L.; and Ma, K. 2024a. Shapellm: Universal 3d object understanding for embodied interaction. *arXiv preprint arXiv:2402.17766*.

Qi, Z.; Fang, Y.; Sun, Z.; Wu, X.; Wu, T.; Wang, J.; Lin, D.; and Zhao, H. 2024b. Gpt4point: A unified framework for point-language understanding and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26417–26427.

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

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Schuhmann, C.; Beaumont, R.; Vencu, R.; Gordon, C.; Wightman, R.; Cherti, M.; Coombes, T.; Katta, A.; Mullis, C.; Wortsman, M.; et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294.

Sun, Q.; Fang, Y.; Wu, L.; Wang, X.; and Cao, Y. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Tang, Y.; Han, X.; Li, X.; Yu, Q.; Hao, Y.; Hu, L.; and Chen, M. 2024. MiniGPT-3D: Efficiently Aligning 3D Point Clouds with Large Language Models using 2D Priors. *arXiv preprint arXiv:2405.01413*.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

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

Xue, L.; Gao, M.; Xing, C.; Martín-Martín, R.; Wu, J.; Xiong, C.; Xu, R.; Niebles, J. C.; and Savarese, S. 2023. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1179–1189.

Xue, L.; Yu, N.; Zhang, S.; Panagopoulou, A.; Li, J.; Martín-Martín, R.; Wu, J.; Xiong, C.; Xu, R.; Niebles, J. C.; et al. 2024. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 27091–27101.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Fan, Z. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*.

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