SegPoint: Segment Any Point Cloud via Large Language Model

Shuting He[1], Henghui Ding[2], Xudong Jiang[1], and Bihan Wen[1] ✉

[1] Nanyang Technological University
[2] Institute of Big Data, Fudan University
{heshuting555, henghui.ding}@gmail.com, {exdjiang, bihan.wen}@ntu.edu.sg
https://heshuting555.github.io/SegPoint

Abstract. Despite significant progress in 3D point cloud segmentation, existing methods primarily address specific tasks and depend on explicit instructions to identify targets, lacking the capability to infer and understand implicit user intentions in a unified framework. In this work, we propose a model, called SegPoint, that leverages the reasoning capabilities of a multi-modal Large Language Model (LLM) to produce point-wise segmentation masks across a diverse range of tasks: 1) 3D instruction segmentation, 2) 3D referring segmentation, 3) 3D semantic segmentation, and 4) 3D open-vocabulary semantic segmentation. To advance 3D instruction research, we introduce a new benchmark, *Instruct3D*, designed to evaluate segmentation performance from complex and implicit instructional texts, featuring 2,565 point cloud-instruction pairs. Our experimental results demonstrate that SegPoint achieves competitive performance on established benchmarks such as ScanRefer for referring segmentation and ScanNet for semantic segmentation, while delivering outstanding outcomes on the *Instruct3D* dataset. To our knowledge, SegPoint is the first model to address these varied segmentation tasks within a single framework, achieving satisfactory performance.

Keywords: *Instruct3D* dataset · Unified framework · 3D point cloud segmentation · Large language model

1 Introduction

3D point cloud segmentation, a critical challenge in the 3D vision community, aims to interpret and classify each point in a point cloud to understand its semantic properties [27, 45, 46, 54, 56, 80]. This longstanding issue has spurred significant advancements across various fields, including robotics, autonomous driving, virtual reality, *etc*. This challenge has evolved into a series of specialized tasks, each targeting a specific segmentation aspect. Overall, tasks cover basic semantic and instance segmentation [3, 7, 53, 74], as well as more practical tasks

✉ Corresponding author

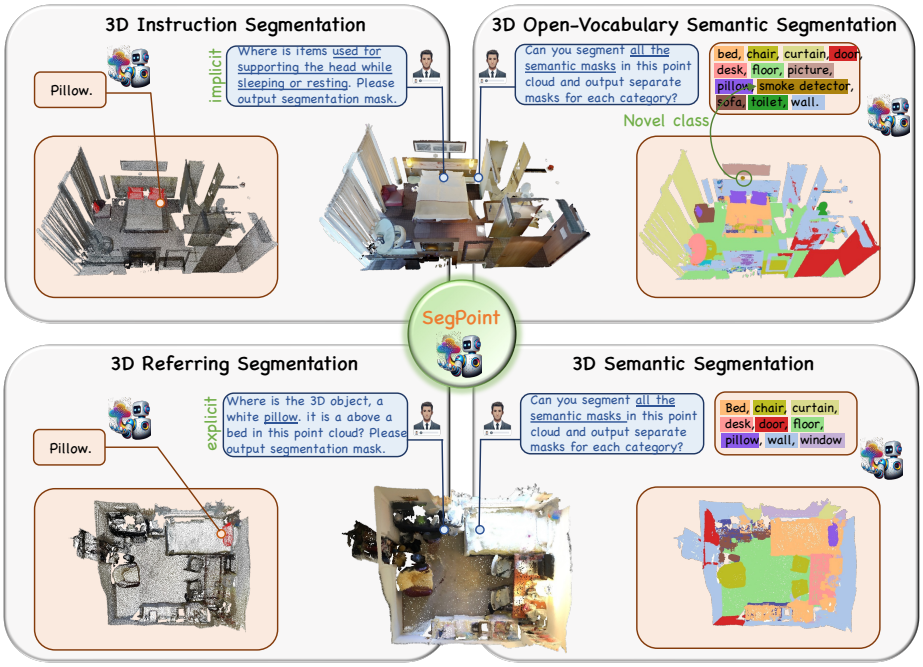

Fig. 1: Example of functionality in SegPoint. SegPoint can complete various point cloud tasks in a unified framework by leveraging task-specific prompts, including 1) 3D instruction segmentation, 2) 3D referring segmentation, 3) 3D semantic segmentation, and 4) 3D open-vocabulary semantic segmentation.

such as referring segmentation [1, 4, 24, 33, 79, 84], which segments points based on explicit textual descriptions, and open-vocabulary segmentation [12, 40, 42, 57, 65, 77] designed for the dynamic and complex nature of real-world.

Despite significant progress achieved within the 3D community toward accurately segmenting objects through specifically designed models, each model is typically developed to tackle one specific segmentation task, leading to inefficiencies and a lack of versatility for real-world application. Furthermore, previous perception approaches heavily depend on predefined categories or explicit expressions for language understanding. Such approaches fall short in interpreting and acting on implicit instructions often found in human language, a critical gap that hinders the development of truly intelligent next-generation perception systems. This brings a pivotal question: *Is it possible to design a unified model capable of comprehensively addressing all aforementioned 3D tasks with human-like instructions?* The exploration of this question not only challenges the current paradigms of 3D point cloud segmentation but also opens the door to groundbreaking advancements in robotic perception and interaction.

In this work, we propose a model called SegPoint, leveraging the Large Language Model's (LLM) advanced ability to reason and comprehend user instructions. To enhance 3D scene comprehension, we integrate a Geometric Enhancer

Module that extracts local semantics from point clouds, seamlessly incorporating this geometric insight into the feature extraction process. Furthermore, a Geometric-guided Feature Propagation is designed to utilize semantic flows derived from geometric priors and the LLM's hidden embeddings, facilitating the generation of fine-grained, high-quality features for accurate dense prediction tasks. Unlike previous attempts at 2D field [8, 28, 50], we do not depend on additional costly pre-trained segmentation models like SAM [26].

Moreover, we introduce a benchmark named *Instruct3D*, designed to advance research in the field of segmentation driven by implicit and complex instructions. Understanding these nuanced instructions necessitates reasoning abilities and extensive knowledge of the world. It includes a total of 2,565 diverse pairs of instructions and point clouds for tuning and evaluation. Our comprehensive experiments demonstrate the benchmark's utility in evaluating the model's capability of segmentation based on human-like instructions.

Taking advantage of a multi-modal LLM and task-specific prompts, SegPoint is capable of generating segmentation masks for a wide range of tasks in a unified model: 1) 3D instruction segmentation, 2) 3D referring segmentation, 3) 3D semantic segmentation, and 4) 3D open-vocabulary semantic segmentation, as depicted in Fig. 1. SegPoint achieves competitive results on established benchmarks like ScanRefer [4] for referring segmentation and ScanNet [7] for semantic segmentation while showing remarkable performance on the *Instruct3D* dataset.

In summary, our main contributions are as follows:

- We propose SegPoint, the first 3D segmentation model that can comprehend human intentions and address multiple segmentation tasks within one framework by harnessing the Large Language Model's reasoning capabilities.
- We present a Geometric Enhancer Module that integrates comprehensive scene information into the process of 3D scene understanding. Besides, A Geometric-guided Feature Propagation is designed to achieve accurate and fine-grained segmentation. These two modules supplement the missing local information and grasp fine-grained features for dense prediction tasks.
- We introduce a new task called 3D instruction segmentation and construct a new dataset *Instruct3D*, which necessitates a model's self-reasoning to interpret implicit instructions for segmenting the target object.
- Our experimental findings reveal that SegPoint not only competes strongly in 3D semantic, referring, and open-vocabulary semantic segmentation but also excels in 3D instruction segmentation, showcasing its versatility and effectiveness across a spectrum of segmentation challenges.

2 Related Work

2.1 Multi-modal Large Language Model

Inspired by the exceptional reasoning abilities of Large Language Models, researchers are delving into transferring these capabilities into the vision realm [9,

28, 36, 55], developing multi-modal LLMs. Flamingo [2], BLIP-2 [31], mPLUG-OWL [73], Otter [30], LLaVA [37] and MiniGPT-4 [83] first construct image-text feature alignment followed by instruction tuning and achieve superior performance. Recent studies have increasingly concentrated on integrating foundational models with tasks that demand a refined understanding at the region or pixel level. VisionLLM [61] introduces a versatile interaction interface for various vision-centric tasks via instruction tuning, albeit without fully leveraging LLMs for intricate reasoning tasks. Kosmos-2 [43] has built substantial data of grounded image-text pairs, thereby embedding grounding capabilities into LLMs. DetGPT [44] links the fixed multi-modal LLM with an open-vocabulary detector, facilitating user instruction-based detection tasks. This growing interest has spurred further innovations, including GPT4RoI [78], LLaVA-grounding [76], Ferret [75], LISA [28], GlaMM [50], PixelLM [52], Sphinx [32], each contributing to the evolving landscape of instruction-based, multi-modal understanding.

Building on advancements of multi-modal large language models in 2D image domain, the field is witnessing a seamless transition into 3D spaces. PointLLM [70] harnesses the prowess of large language models and trains with 3D point clouds following the paradigm of LLaVA [37]. 3D-LLM [20] utilizes 2D foundation models to encode multi-view images of 3D point clouds. Point-Bind [15] aligns point clouds with Image-Bind [14] and leverages ImageBind-LLM to reason multi-modality input without 3D-instruction data training. GPT4Point [47] pioneers in facilitating a unified approach towards 3D object understanding and generation, setting a new standard for versatility. However, these models primarily concentrate on scene-level insights, often overlooking the intricate details at the region, or point level. Contrasting with existing models, our research focuses on two pivotal goals: 1) efficiently inject segmentation capabilities into multi-modal LLMs to conduct point-level understanding and 2) design a unified framework for 3D point cloud segmentation via the reasoning ability of LLMs.

2.2 3D Point Cloud Segmentation

3D point cloud segmentation, a crucial task in computer vision, can be categorized into 3D semantic, instance, and panoptic segmentation. 3D semantic segmentation [3, 7, 19, 45, 53, 74, 80] assigns each point in a 3D space to specific, predefined classes. In contrast, 3D instance segmentation [25, 27, 54] goes a step further by classifying each point and differentiating between distinct objects of the same class. 3D panoptic segmentation [68, 82] aims to group 3D points according to their semantics and identities. Lately, more practical tasks have emerged, such as 3D referring segmentation [1, 4, 17, 23, 67, 79], which extends referring expression segmentation [8, 10, 11, 33–35] to 3D and segments a target instance based on explicit linguistic descriptions, and 3D open-vocabulary segmentation [12,40,42,57], designed to identify and segment unseen objects beyond a fixed set of known categories.

Despite the significant progress made by transformer-based models: Mask3D [54], SPFormer [56], MAFT [29], and OneFormer3D [27] in basic 3D segmentation tasks, their application in real-world scenarios requiring human language

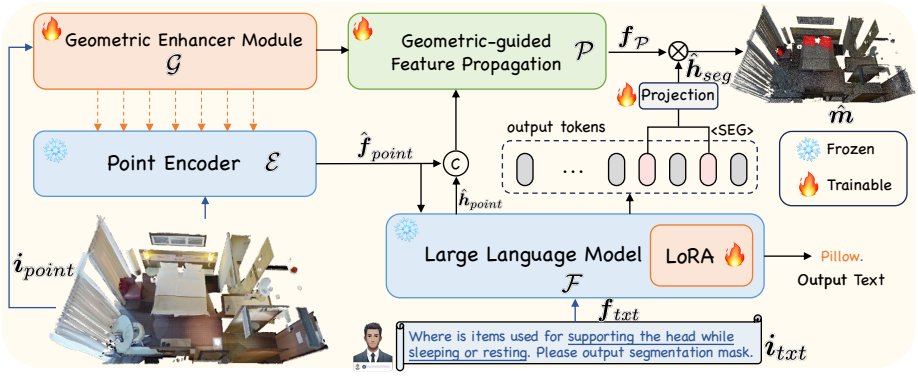

Fig. 2: The pipeline of SegPoint. Given input point cloud and text query, the multimodal LLM $\mathcal{F}$ generates text output. Geometric Enhancer Module $\mathcal{G}$ injects geometric information into Point Encoder $\mathcal{E}$ and obtains point features $\hat{\boldsymbol{f}}_{point}$. Per-point embeddings $\boldsymbol{f}_{\mathcal{P}}$ derived from Geometric-guided Feature Propagation $\mathcal{P}$ multiplied with the embedding associated with the `<SEG>` token yield the final segmentation masks.

interaction remains constrained. Besides, Seal [38] aims to segment any point cloud through distilling vision foundation models, while it doesn't use language as cues. TGNN [22] is the first work to tackle referring segmentation problem that proposes aggregating textual features by considering the neighboring local structure of each instance but it heavily depends on explicit expressions or predefined categories for language understanding. Furthermore, the development of models tailored to specific segmentation tasks restricts their versatility and applicability in diverse real-world scenarios. Therefore, it is imperative to develop more intelligent interaction ways and a unified model for 3D point segmentation.

3 Approach

3.1 Architecture Overview

The overall architecture of SegPoint is presented in Fig. 2. SegPoint mainly comprises four parts: *i*) a pre-trained point encoder $\mathcal{E}$ tailored for aligning with textual data; *ii*) a large language model $\mathcal{F}$ endowed with advanced reasoning capabilities; *iii*) a Geometric Enhancer Module $\mathcal{G}$ responsible for extracting geometric representation from input point clouds and infusing these priors into the point encoder; and *iv*) a Geometric-guided Feature Propagation $\mathcal{P}$ which is key to achieving precise mask generation. The collaboration between the Geometric Enhancer Module and Geometric-guided Feature Propagation is crucial, as it equips LLMs with the ability to generate masks effectively in various scenarios.

3.2 Vanilla Baseline

The input of the framework is the text instructions $\boldsymbol{i}_{txt}$ and point cloud $\boldsymbol{i}_{point} \in \mathbb{R}^{N\times(3+F)}$. Specifically, a point cloud scene, comprising N points, each includes

3D coordinates $\in \mathbb{R}^3$ and an auxiliary feature vector $\in \mathbb{R}^F$ (e.g., color). The point cloud $\boldsymbol{i}_{point}$ is fed into the point encoder $\mathcal{E}$, which extracts point features $\boldsymbol{f}_{point} \in \mathbb{R}^{N_1 \times D}$, where $N_1 \ll N$, D is the feature dimension. Concurrently, the text instruction $\boldsymbol{i}_{txt}$ undergoes tokenization via $\mathcal{F}_{tokenize}$. These prepared inputs are then fed into the Large Language Model $\mathcal{F}$, resulting in a textual response $\boldsymbol{y}$. The above process can be formulated as:

$$\boldsymbol{f}_{point} = \mathcal{E}(\boldsymbol{i}_{point}), \quad \boldsymbol{f}_{txt} = \mathcal{F}_{tokenize}(\boldsymbol{i}_{txt}), \quad \boldsymbol{y} = \mathcal{F}(\boldsymbol{f}_{point}, \boldsymbol{f}_{txt}). \tag{1}$$

Building on the approach introduced by LISA [28], SegPoint enhances the segmentation capabilities of Large Language Models (LLMs) by expanding their vocabulary with a new special token, `<SEG>`. This modification enables the model to recognize and predict the `<SEG>` token within the output sequence as a signal to identify segmentation targets. Upon detecting a `<SEG>` token, the corresponding output sequence belonging to `<SEG>` token is extracted and processed through an MLP layer γ, generating mask embeddings $\boldsymbol{h}_{seg}$. The final step involves computing each binary mask prediction $\boldsymbol{m} \in \mathbb{R}^N$ by performing a dot product between the mask embeddings $\boldsymbol{h}_{seg}$ and the upsampled per-point embeddings derived from the point features $\boldsymbol{f}_{point}$. The formulation of the aforementioned process is given by:

$$\boldsymbol{h}_{seg} = \gamma(\boldsymbol{y}_{[seg]}), \quad \boldsymbol{m} = \boldsymbol{h}_{seg} \otimes UpS.(\boldsymbol{f}_{point}), \tag{2}$$

where $UpS.$ denotes the upsampling operation following PointNet++ [46] on $\boldsymbol{f}_{point}$. The vanilla baseline represents an initial attempt to bridge the gap between LLMs' text comprehension and point cloud segmentation tasks. It encounters two primary issues. Firstly, the point encoder is trained on a scene-level dataset for classification achieving alignment between text and point clouds, not specifically trained for dense prediction tasks. Besides, the point encoder's first layer employs Farthest Point Sampling (FPS) [46] to reduce the point cloud to N_1 points, risking the loss of details vital for accurate dense predictions. Secondly, the operation of directly upsampling from N_1 to N points to obtain per-point embeddings is prone to losing structural information and introducing a notable degree of noise, undermining the model's efficacy in segmentation tasks.

3.3 Geometric Enhancer Module

To adapt the pre-trained point encoder for dense prediction tasks while maintaining its superior scene recognition capability, our objective is to harness the geometric information across the entire scene to guide the further feature learning process. Drawing inspiration from recent advancements in 2D computer vision, where studies [5,41,62,64] demonstrate that convolutions enhance transformers' ability to capture local spatial information, we introduce the Geometric Enhancer Module (GEM). This module is specifically designed to grasp the local geometric contexts within point clouds while enabling the preservation of the point encoder's foundational architecture and information integrity.

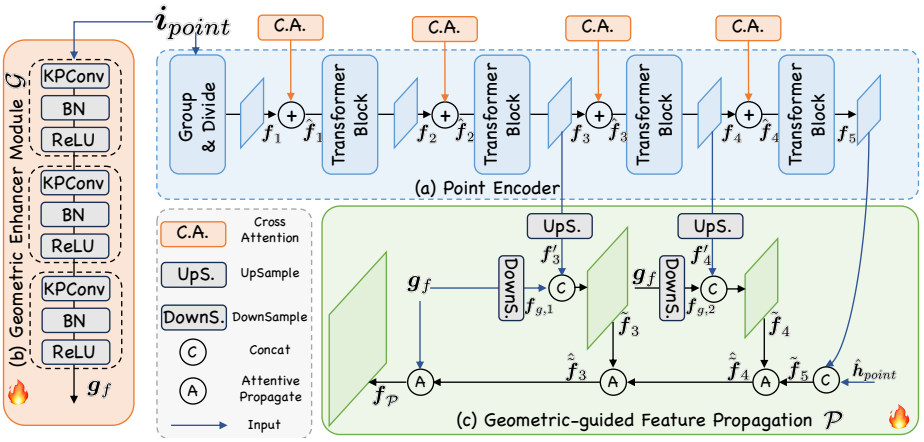

Fig. 3: Architecture of the proposed (b) Geometric Enhance Module (GEM) and (c) Geometric-guided Feature Propagation (GFP) interaction with (a) Point Encoder.

As shown in Fig. 3, Geometric Enhancer Module $\mathcal{G}$ is composed of three blocks, each featuring a KPConv [58] layer followed by BN and ReLU activation. The architecture is similar to the 2D convolutional stem [16]. We utilize KPConv instead of vanilla convolution or linear layer here to facilitate grasping the local geometric information effectively. The resultant geometric feature, represented by $\boldsymbol{g}_f \in \mathbb{R}^{N \times D}$, contains the features across all points, thereby supplementing the missing local information. This $\boldsymbol{g}_f$ is then leveraged to infuse geometric insights into the point encoder's features via a cross-attention mechanism, the above process can be expressed as:

$$\boldsymbol{g}_f = \mathcal{G}(\boldsymbol{i}_{point}), \quad \hat{\boldsymbol{f}}_i = \boldsymbol{f}_i + g_i \cdot \text{softmax}\left(\frac{\boldsymbol{f}_i \boldsymbol{g}_f^T}{\sqrt{D}}\right)\boldsymbol{g}_f, \tag{3}$$

where $\boldsymbol{f}_i$ represents the feature from the i-th block of the point encoder and l consecutive transformer layers are regarded as one block for the convenience of explanation. To fine-tune the integration of geometric information, we introduce a learnable gating factor g_i that modulates the balance between the attention layer's output and the input feature $\boldsymbol{f}_i$. This gating factor is initially set to zero, to ensure that the incorporation of geometric data does not abruptly alter $\boldsymbol{f}$ feature distribution. Such an approach allows for the preservation and effective utilization of the point encoder's pre-trained weights. Upon processing through the Geometric Enhancer Module (GEM), the modified output of the point encoder, LLM are formulated as:

$$\hat{\boldsymbol{f}}_{point} = \mathcal{E}(\boldsymbol{i}_{point}, \boldsymbol{g}_f), \quad \hat{\boldsymbol{y}} = \mathcal{F}(\hat{\boldsymbol{f}}_{point}, \boldsymbol{f}_{txt}), \quad \hat{\boldsymbol{h}}_{seg} = \gamma(\hat{\boldsymbol{y}}_{[seg]}). \tag{4}$$

3.4 Geometric-guided Feature Propagation

Addressing the challenge of upsampling point clouds from a sparse set of N_1 points to a denser set of N points is crucial, as direct upsampling inevitably intro-

duces noise and results in information loss, leading to sub-optimal performance in segmentation tasks. To mitigate these issues, we introduce Geometric-guided Feature Propagation designed to generate high-quality per-point embeddings. Geometric features $\boldsymbol{g}_f$, which carry comprehensive point information, serving as a "gold message" for enhancing the upsampling process. By integrating these geometric features, we aim to significantly improve the quality and accuracy of the generated dense per-point embeddings.

As illustrated in Fig. 3, we begin by upsampling the higher-layer features $\boldsymbol{f}_3, \boldsymbol{f}_4$ from a smaller set of points N_1 to larger sets N_3, N_2, employing PointNet++'s [46] propagation techniques. This step yields features $\boldsymbol{f}'_3 \in \mathbb{R}^{N_3 \times D}$, and $\boldsymbol{f}'_4 \in \mathbb{R}^{N_2 \times D}$. Subsequently, we perform downsampling on the geometric features $\boldsymbol{g}_f$ from the original number of points N to reduced counts N_2, N_3, respectively, utilizing the Farthest Point Sampling (FPS) technique. In this process, we directly obtain the features of the sampled points without performing additional k-nearest neighbor (k-NN) and pooling operations to simplify the computation and produce features $\boldsymbol{f}_{g,1} \in \mathbb{R}^{N_3 \times D}$, and $\boldsymbol{f}_{g,2} \in \mathbb{R}^{N_2 \times D}$.

In the next phase, we integrate the up- and downsampled features, processing them through fully connected layers and ReLU activation to update the feature vectors $\tilde{\boldsymbol{f}}_3 \in \mathbb{R}^{N_3 \times D}$, and $\tilde{\boldsymbol{f}}_4 \in \mathbb{R}^{N_2 \times D}$. Note that the last layer feature $\boldsymbol{f}_5$ bypasses this step. Instead, we concatenate it with $\hat{\boldsymbol{h}}_{point}$ from the LLM output to form $\tilde{\boldsymbol{f}}_5$ to perceive multi-modal information from LLM.

Finally, to enable information exchange across different point densities, we propose attentive propagation. Take the propagation from $\tilde{\boldsymbol{f}}_5$ to $\tilde{\boldsymbol{f}}_4$ as example. Here, $\tilde{\boldsymbol{f}}_4 \in \mathbb{R}^{N_2 \times D}$ acts as a set of local centers. For each local center within $\tilde{\boldsymbol{f}}_4$, we identify its neighboring points from $\tilde{\boldsymbol{f}}_5$ using the k-NN algorithm, resulting in $\boldsymbol{f}_{54} \in \mathbb{R}^{N_2 \times k \times D}$. Then, employing cross-attention mechanism, where $\tilde{\boldsymbol{f}}_4$ serves as query and $\boldsymbol{f}_{54}$ as both key and value, facilitates information flow across different point densities and effectively extract relevant details into the query points.

$$\hat{\tilde{\boldsymbol{f}}}_4 = \tilde{\boldsymbol{f}}_4 + \text{softmax}\left(\frac{\tilde{\boldsymbol{f}}_4 \boldsymbol{f}_{54}^T}{\sqrt{D}}\right) \boldsymbol{f}_{54}. \tag{5}$$

Leveraging geometric-guided feature propagation enables us to produce high-quality per-point embeddings denoted as $\boldsymbol{f}_{\mathcal{P}}$, laying the foundation for generating precise segmentation masks expressed as follows:

$$\boldsymbol{f}_{\mathcal{P}} = \mathcal{P}(\hat{\boldsymbol{f}}_{point}, \boldsymbol{g}_f), \quad \hat{\boldsymbol{m}} = \hat{\boldsymbol{h}}_{seg} \otimes \boldsymbol{f}_{\mathcal{P}}. \tag{6}$$

3.5 Training Objectives

Our model is trained end-to-end leveraging the text classification loss and the segmentation mask loss:

$$\mathcal{L} = \lambda_{txt}\mathcal{L}_{txt} + \lambda_{bce}\mathcal{L}_{bce} + \lambda_{dice}\mathcal{L}_{dice}, \tag{7}$$

where $\mathcal{L}_{txt}$ denotes the auto-regressive cross-entropy loss targeting text generation accuracy, segmentation mask loss includes both binary cross-entropy (BCE)

loss $\mathcal{L}_{bce}$ and DICE loss $\mathcal{L}_{dice}$, aims at refining segmentation quality. The weights λ_{txt}, λ_{bce} and λ_{dice} are utilized to balance the different loss items. The model's training is guided by the ground-truth labels $\boldsymbol{y}_{txt}$ for text and $\boldsymbol{M}$ for masks.

3.6 *Instruct3D* Dataset Collection

Although 3D instruction segmentation and 3D referring segmentation [1,4,79] are both language-based segmentation, 3D referring segmentation guides segmentation with explicit target object names, e.g., "chair", lacking more complicated reasoning instructions, e.g., "Where to sit in the room?". Besides, they also fall short in offering multi-target question-answer pairs with target descriptions directly connected to multiple segmentation masks, which cannot meet a common requirement in real-world scenarios, like "How to play computer games".

To enhance the assessment and analysis of instruction segmentation capabilities, we have developed a benchmark, referred to *Instruct3D*. This benchmark incorporates 280 scenes specifically selected for instruction segmentation tuning and evaluation, sourced from the recently introduced ScanNet++ [74] dataset. Each scene comes with approximately 10 different segmentation instructions, resulting in 2,565 instruction-point cloud pairings. This dataset is then divided into two splits: `train`, and `val`, containing 2,052, and 513 question-answer pairs, respectively. Our dataset is uniquely designed to encompass both multi-target and zero-target scenarios, addressing the real-world requirement of identifying multiple objects in response to text queries and accounting for situations where objects mentioned in the text may not be present in the paired point cloud. Besides, we take into account the characteristics of 3D scenes and incorporate diverse locations and view descriptions e.g., "something that is used for sitting while working at a desk. It is the one facing the window.". The model needs to have not only reasoning capabilities but also the ability to perceive views and directions in 3D scenes. These designs underscore the dataset's practical value.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. Our training data is composed of two types of datasets: (1) semantic segmentation dataset including ScanNet200 [53], and S3DIS [3]; (2) referring segmentation dataset consisting of ScanRefer [4], ReferIt3D [1](including Sr3D and Nr3D), and Multi3DRefer [79]. We design task-specific prompts to facilitate the joint training of various tasks within a unified framework.

Semantic Segmentation Dataset. We use two strategies to generate templates. 1) segment the specific category: "`USER: <POINT> Can you segment the {category} category in this point cloud? ASSISTANT: {category} <SEG>.`", where `category` is the random chosen category, and `<POINT>` denotes the placeholder for tokens of point cloud patches. 2) segment all the categories: "`USER: <POINT> Can you segment all the semantic masks in this point cloud and output separate masks for each category in the alphabetical order of the`

categories? ASSISTANT: {category} <SEG>, {category} <SEG>, ..." To simplify the output and ensure it has only one possible answer, we add the constraints "in the alphabetical order of the categories". To avoid generating class names not in the dataset, we incorporate category names in a dataset into the prompts during training and inference.

Referring Segmentation Dataset. We use template prompts: "USER: <POINT> Can you segment the object {description} in this point cloud? ASSISTANT: {category} <SEG>.", where {description} is the given explicit description from referring segmentation dataset. It is worth noting that during training, we also use other templates to generate the QA data to ensure data diversity. We add {category} in front of <SEG> to unify the output format so that when outputting semantic masks, the output category name is the label it predicts.

Evaluation Metrics. We follow most previous works on 3D segmentation [22, 27, 54] to adopt mIoU as primary metric. mIoU is defined by the average of all per-point cloud scene Intersection-over-Unions (IoUs). Besides, we employ accuracy (Acc) as a metric to evaluate whether the model accurately identifies targets with which the predictions have an IoU greater than 0.5.

4.2 Implementation Details

In our experiments, unless specified otherwise, we employ the LLaMA2-7B model [59] as the large language model $\mathcal{F}$ and Uni3D [81] as the point cloud processing backbone $\mathcal{E}$. The training stage leverages the deepspeed [51] engine for efficiency, with the AdamW [39] optimizer guiding the learning process. The learning rate and weight decay are set to 0.0003 and 0, respectively, enhanced by a WarmupDecayLR learning rate scheduler that initiates with 100 warmup iterations. The projection layer γ utilizes an MLP with channel sizes of [256, 4096, 4096]. We set balancing weight λ_{txt_gen}, λ_{bce}, and λ_{dice} to 1.0, 2.0, 2.0, respectively. The experiments utilize a total batch size of 16, distributed across 4 NVIDIA 80G A100 GPUs, and span 5,000 iterations, culminating in a training period of approximately 3 days. During training, we use all mentioned datasets in Sec. 4.1 for joint training by leveraging task-specific prompts. For evaluation on a specific dataset, we finetune the trained model on the corresponding dataset.

4.3 Results on *Instruct3D*

The instruction segmentation results, as detailed in Table 1, underscore a significant advancement: where existing methodologies fall short, our model demonstrates exceptional prowess, achieving a more than 15% improvement in mIoU for tasks requiring intricate reasoning. Unlike conventional referring segmentation tasks, instruction segmentation demands not just identification but also understanding, necessitating the model's reasoning capabilities and access to world knowledge. Existing approaches, confined to explicit references, struggle with implicit queries due to their lack of understanding, which further underscores the task's inherent challenges. In contrast, our model leverages LLMs to bridge this gap, demonstrating superior performance by comprehending and interpreting the queries accurately. Moreover, SegPoint configuration substantially outperforms SegPoint†, highlighting the critical role of our designed Geometric

Table 1: 3D instruction segmentation benchmark results on *Instruct3D* val split evaluated by Acc and mIoU. † denotes our vanilla baseline removing geometric enhancer module and geometric-guided feature propagation. ∗ represents adding an auxiliary mask head through our implementation.

Stage	Method	Reference	Acc	mIoU
Two	ScanRefer [4]	[ECCV'20]	12.0	6.9
Two	ReferIt3D [1]	[ECCV'20]	11.7	6.4
Two	M3DRef-CLIP [79]	[ICCV'23]	18.1	12.8
Single	TGNN [22]	[AAAI'21]	12.9	7.1
Single	BUTD-DETR [23]*	[ECCV'22]	16.3	10.9
Single	EDA [67]*	[CVPR'23]	16.6	12.1
Single	SegPoint†	[ECCV'24]	**21.8**	**16.1**
Single	SegPoint	[ECCV'24]	**31.6**	**27.5**

Table 2: 3D Semantic segmentation benchmark results on S3DIS [3], ScanNet [7], and ScanNet200 [53]. We evaluate on the Area 5 of S3DIS and validation split of ScanNet and ScanNet200.

Method	Reference	ScanNet	ScanNet200	S3DIS
PointNet++ [46]	[NeurIPS'17]	53.5	-	-
MinkUNet [6]	[CVPR'19]	72.2	25.0	65.4
PTv1 [80]	[ICCV'21]	70.6	27.8	70.4
PTv2 [66]	[NeurIPS'22]	75.4	30.2	71.6
PointNeXt [48]	[NeurIPS'22]	71.5	-	70.5
OctFormer [60]	[SIGGRAPH'23]	**75.7**	32.6	-
Swin3D [72]	[ArXiv]	75.5	-	**72.5**
SegPoint	[ECCV'24]	74.1	**35.3**	72.4

Enhancer Module and Geometric-guided Feature Propagation components. Notably, even in its baseline form, SegPoint† surpasses all competing methods, validating the effectiveness and rationale behind our pipeline design.

Besides, different from traditional two-stage approaches that first generate mask proposals using a pre-trained segmentor like Mask3D [54] and then apply language-aware networks for selection, SegPoint directly tackles the task, bypassing the need for preliminary mask proposals, enhancing its efficiency.

4.4 Results on Semantic Segmentation

Table 2 present SegPoint's performance on semantic segmentation, delivering competitive results across diverse datasets. Our model uses a simple yet effective answer format, `category <SEG>`, to use `category` name as predicted labels, achieving particularly stronger performance in datasets with various categories like ScanNet200 [53], where it surpasses SOTA methods by 2.1% mIoU. To ensure fair comparisons, we fine-tune our model on each semantic segmentation dataset to accommodate the varying class category definitions.

4.5 Results on Referring Segmentation

Table 3 presents results on referring segmentation datasets. SegPoint showcases outstanding performance in both single-target (e.g., ScanRefer [4], Nr3D [1]) and multi-target and zero-target contexts within the Multi3DRefer [79] dataset. For multi-targets, we aggregate masks into a single ground truth, and for zero-target, we use an empty mask, indicated by "`ASSISTANT`: There is no mask." SegPoint significantly surpasses other approaches, achieving 2.5% mIoU increase. The evaluation process of two-stage method M3DRef-CLIP is similar to Sec.4.3.

Table 3: 3D referring segmentation benchmark results on ScanRefer [4], Nr3D [1], and Multi3Drefer [79] evaluated by mIoU. * represents adding an auxiliary mask head through our implementation.

Stage	Method	Reference	ScanRefer	Nr3D	Multi3DRefer
Two	M3DRef-CLIP [79]	[ICCV'23]	35.7	27.0	32.6
Two	3D-STMN [63]	[AAAI'24]	39.5	-	-
Single	TGNN [22]	[AAAI'21]	27.8	-	-
Single	BUTD-DETR [23]*	[ECCV'22]	35.4	27.5	26.2
Single	EDA [67]*	[CVPR'23]	36.2	29.3	28.9
Single	X-RefSeg3D [49]	[AAAI'24]	29.9	-	-
Single	RefMask3D [18]	[ACMMM'24]	**44.8**	-	-
Single	SegPoint	[ECCV'24]	41.7	**32.2**	**36.1**

Table 4: 3D open-vocabulary semantic segmentation benchmark results on val split of ScanNet++ [74].

Type	Method	Reference	ScanNet++
Supervised	PointNet [45]	[CVPR'17]	7.0
	PointNet++ [46]	[NeurIPS'17]	15.0
	MinkUNet [6]	[CVPR'19]	28.0
	KPConv [58]	[ICCV'19]	30.0
Open-Vocabulary	OpenScene [42]	[CVPR'23]	12.8
	PLA [12]	[CVPR'23]	14.2
	RegionPLC [71]	[CVPR'24]	14.9
Unified	SegPoint	[ECCV'24]	**19.3**

4.6 Results on Open-vocabulary Semantic Segmentation

Table 4 shows our method's open-vocabulary segmentation performance which is directly evaluated on ScanNet++ [74] following the setting in prevalent methodologies in the 2D domain [13,69]. It demonstrates our superiority over both existing open-vocabulary techniques and even several supervised approaches, showing our model's robust generalization capabilities. It effectively aligns and interprets categories with visual scenes, underscoring the reasoning prowess of large language models. A notable issue is the potential misalignment between output categories of SegPoint and val split category names. To address this, we employ GPT-4 to match its most similar category names in the val split.

4.7 Ablation Study

We conduct extensive experiments to verify the effectiveness of our proposed components in Table 5 (a) on both *Instruct3D* and ScanRefer [4] dataset. We established a vanilla baseline as described in Sec. 3.2 following the paradigm of LISA [28], which cannot achieve satisfactory performance and only obtain 16.1%/30.3 mIoU on *Instruct3D*/ScanRefer, respectively. Further analysis, both qualitative and quantitative, of our proposed components reveals that their integration substantially outperforms the baseline.

Geometric Enhancer Module Integrating the Geometric Enhancer Module (GEM) into our point encoder results in a notable 5.3%/5.5% mIoU improvement on *Instruct3D*/ScanRefer, effectively addressing compatibility issues with dense prediction tasks. An ablation study, shown in Table 5 (b), shows that our improvement is not due to an increase in parameters. Our approach outperforms traditional full fine-tuning, LoRA [21] strategies, and the addition of

Table 5: Ablation studies on *Instruct3D* and ScanRefer.

Index	Components GEM	GFP	*Instruct3D* mIoU	ScanRefer mIoU
0	✗	✗	16.1	30.3
1	✓	✗	21.4	35.8
2	✗	✓	23.2	38.1
3	✓	✓	**27.5**	**41.7**

(a) Effect of different main components.

Method	*Instruct3D*	ScanRefer
Full tuning	26.2	39.1
LoRA [21]	24.8	38.5
MLP	23.1	36.8
GEM	**27.5**	**41.7**

(b) Effect of different tuning methods.

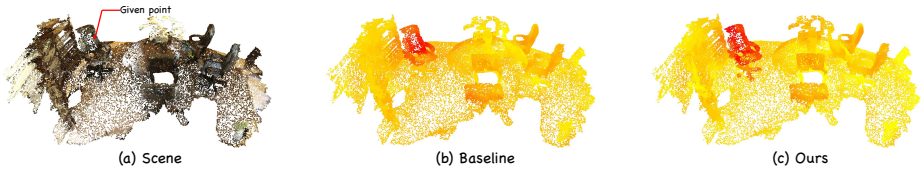

Fig. 4: (Best viewed in color) We visualize the feature responses between a given point (in red) and other points in the scene from per-point embeddings $\boldsymbol{f}_{\mathcal{P}}$ for the baseline and our SegPoint, respectively. The color changes from yellow to red, indicating increasing feature similarity.

MLP layers for feature adapter, underscoring its effectiveness in embedding 3D domain-specific knowledge into point encoder. Unlike adapter techniques common in language and 2D image processing, our GEM is designed to address the unique challenges in dense prediction tasks.

Geometric-guided Feature Propagation Introducing Geometric-guided Feature Propagation (GFP) results in a substantial improvement over the baseline, as shown in Table 5 (a) (index 2). This underscores our method's capability to minimize information loss and reduce noise during the upsampling phase, leading to higher-quality per-point embeddings.

4.8 Qualitative Visualization

Fig. 4 qualitatively illustrates the feature responses between a given point and others in a scene. From left to right, it presents the original scene, baseline method, and our SegPoint, respectively, with warmer colors indicating closer feature relationships. The baseline method predominantly highlights spatially proximate points, often missing important but distant features, such as chair cushions. In contrast, SegPoint leverages global information, allowing for the identification of both near and distant relevant features, such as cushions and casters. This demonstrates our model's superior ability to capture global context and recognize intricate structures within the scene.

As shown in Fig. 5, we provide some typical qualitative results from SegPoint on *Instruct3D*. Given an implicit instruction, *e.g.*, "*the object that is used for dispensing water during bathing*", SegPoint successfully infers the role of a shower. In other scenes, the instructions include queries requiring extensive knowledge of the world and complex reasoning, like "*What appliance is used for heating or cooking food quickly using electromagnetic radiation.*" SegPoint can still segment the microwave oven very well, which shows that SegPoint can make good use of the reasoning ability of LLM and provide high-quality masks.

Fig. 5: Qualitative results from val split of *Instruct3D*. SegPoint understand the human instruction and accurately segment the target object. We omitted the "please output segmentation mask" in the sentence for simplicity.

5 Conclusion

In this work, we introduce SegPoint, an effective model supported by LLM for point-level reasoning and segmentation. Benefiting from the proposed Geometric Enhancer Module and Geometric-guided Feature Propagation, SegPoint is adept at solving a variety of segmentation tasks in a unified framework. Additionally, we construct a comprehensive *Instruct3D* benchmark to bolster research area in segmentation via implicit and complex instructions, introducing more challenges to promote it closer to real-world applications. Through thorough experiments, SegPoint achieves promising results across multiple benchmarks.

Limitations Although SegPoint demonstrates notable success in tasks driven by text prompts, its current framework cannot process non-textual prompts, such as boxes and points. Future developments will explore the adoption of a prompt encoder, inspired by SAM model [26], to extend support for these formats.

Acknowledgements We thank the anonymous reviewers for their constructive suggestions. Following their advice, we have incorporated diverse location and view descriptions into our *Instruct3D*. This work was partially supported by the National Research Foundation Singapore Competitive Research Program (CRP29-2022-0003).

References

1. Achlioptas, P., Abdelreheem, A., Xia, F., Elhoseiny, M., Guibas, L.: Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In: ECCV (2020)
2. 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. In: NeurIPS (2022)
3. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: CVPR (2016)
4. Chen, D.Z., Chang, A.X., Nießner, M.: Scanrefer: 3d object localization in rgb-d scans using natural language. In: ECCV (2020)
5. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. In: ICLR (2023)
6. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: CVPR (2019)
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. Ding, H., Liu, C., He, S., Jiang, X., Loy, C.C.: MeViS: A large-scale benchmark for video segmentation with motion expressions. In: ICCV (2023)
9. Ding, H., Liu, C., He, S., Jiang, X., Torr, P.H., Bai, S.: MOSE: A new dataset for video object segmentation in complex scenes. In: ICCV (2023)
10. Ding, H., Liu, C., Wang, S., Jiang, X.: Vision-language transformer and query generation for referring segmentation. In: ICCV (2021)
11. Ding, H., Liu, C., Wang, S., Jiang, X.: VLT: Vision-language transformer and query generation for referring segmentation. IEEE TPAMI (2023)
12. Ding, R., Yang, J., Xue, C., Zhang, W., Bai, S., Qi, X.: Pla: Language-driven open-vocabulary 3d scene understanding. In: CVPR (2023)
13. Ding, Z., Wang, J., Tu, Z.: Open-vocabulary panoptic segmentation with maskclip. In: ICLR (2023)
14. 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)
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. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
17. He, S., Ding, H.: Decoupling static and hierarchical motion perception for referring video segmentation. In: CVPR (2024)
18. He, S., Ding, H.: RefMask3D: Language-guided transformer for 3d referring segmentation. In: ACM MM (2024)
19. He, S., Jiang, X., Jiang, W., Ding, H.: Prototype adaption and projection for few- and zero-shot 3d point cloud semantic segmentation. IEEE TIP (2023)
20. Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., Gan, C.: 3d-llm: Injecting the 3d world into large language models. In: NeurIPS (2023)
21. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: ICLR (2022)
22. Huang, P.H., Lee, H.H., Chen, H.T., Liu, T.L.: Text-guided graph neural networks for referring 3d instance segmentation. In: AAAI (2021)

23. Jain, A., Gkanatsios, N., Mediratta, I., Fragkiadaki, K.: Bottom up top down detection transformers for language grounding in images and point clouds. In: ECCV (2022)
24. Jia, B., Chen, Y., Yu, H., Wang, Y., Niu, X., Liu, T., Li, Q., Huang, S.: Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In: ECCV (2024)
25. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: Pointgroup: Dual-set point grouping for 3d instance segmentation. In: CVPR (2020)
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. In: ICCV (2023)
27. Kolodiazhnyi, M., Vorontsova, A., Konushin, A., Rukhovich, D.: Oneformer3d: One transformer for unified point cloud segmentation. In: CVPR (2024)
28. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. In: CVPR (2024)
29. Lai, X., Yuan, Y., Chu, R., Chen, Y., Hu, H., Jia, J.: Mask-attention-free transformer for 3d instance segmentation. In: ICCV (2023)
30. Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., Liu, Z.: Otter: A multi-modal model with in-context instruction tuning. arXiv:2305.03726 (2023)
31. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: NeurIPS (2023)
32. Lin, Z., Liu, C., Zhang, R., Gao, P., Qiu, L., Xiao, H., Qiu, H., Lin, C., Shao, W., Chen, K., et al.: Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. arXiv preprint arXiv:2311.07575 (2023)
33. Liu, C., Ding, H., Jiang, X.: GRES: Generalized referring expression segmentation. In: CVPR (2023)
34. Liu, C., Ding, H., Zhang, Y., Jiang, X.: Multi-modal mutual attention and iterative interaction for referring image segmentation. IEEE TIP (2023)
35. Liu, C., Jiang, X., Ding, H.: Instance-specific feature propagation for referring segmentation. IEEE TMM (2022)
36. Liu, C., Jiang, X., Ding, H.: Primitivenet: decomposing the global constraints for referring segmentation. Visual Intelligence **2**(1), 16 (2024)
37. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023)
38. Liu, Y., Kong, L., Cen, J., Chen, R., Zhang, W., Pan, L., Chen, K., Liu, Z.: Segment any point cloud sequences by distilling vision foundation models. In: NeurIPS (2023)
39. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
40. Nguyen, P.D., Ngo, T.D., Gan, C., Kalogerakis, E., Tran, A., Pham, C., Nguyen, K.: Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In: CVPR (2024)
41. Park, N., Kim, S.: How do vision transformers work? arXiv preprint arXiv:2202.06709 (2022)
42. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. In: CVPR (2023)
43. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Wei, F.: Kosmos-2: Grounding multimodal large language models to the world. In: ICLR (2024)
44. Pi, R., Gao, J., Diao, S., Pan, R., Dong, H., Zhang, J., Yao, L., Han, J., Xu, H., Zhang, L.K.T.: Detgpt: Detect what you need via reasoning. In: EMNLP (2023)
45. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017)
46. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)

47. Qi, Z., Fang, Y., Sun, Z., Wu, X., Wu, T., Wang, J., Lin, D., Zhao, H.: Gpt4point: A unified framework for point-language understanding and generation. In: CVPR (2024)
48. Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In: NeurIPS (2022)
49. Qian, Z., Ma, Y., Ji, J., Sun, X.: X-refseg3d: Enhancing referring 3d instance segmentation via structured cross-modal graph neural networks. In: AAAI (2024)
50. Rasheed, H., Maaz, M., Shaji, S., Shaker, A., Khan, S., Cholakkal, H., Anwer, R.M., Xing, E., Yang, M.H., Khan, F.S.: Glamm: Pixel grounding large multimodal model. In: CVPR (2024)
51. Rasley, J., Rajbhandari, S., Ruwase, O., He, Y.: Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In: KDD (2020)
52. Ren, Z., Huang, Z., Wei, Y., Zhao, Y., Fu, D., Feng, J., Jin, X.: Pixellm: Pixel reasoning with large multimodal model. In: CVPR (2024)
53. Rozenberszki, D., Litany, O., Dai, A.: Language-grounded indoor 3d semantic segmentation in the wild. In: ECCV (2022)
54. Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., Leibe, B.: Mask3d: Mask transformer for 3d semantic instance segmentation. In: ICRA (2023)
55. Shuai, X., Ding, H., Ma, X., Tu, R., Jiang, Y.G., Tao, D.: A survey of multimodal-guided image editing with text-to-image diffusion models. arXiv preprint arXiv:2406.14555 (2024)
56. Sun, J., Qing, C., Tan, J., Xu, X.: Superpoint transformer for 3d scene instance segmentation. In: AAAI (2023)
57. Takmaz, A., Fedele, E., Sumner, R.W., Pollefeys, M., Tombari, F., Engelmann, F.: Openmask3d: Open-vocabulary 3d instance segmentation. In: NeurIPS (2023)
58. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: ICCV (2019)
59. 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 preprint arXiv:2302.13971 (2023)
60. Wang, P.S.: Octformer: Octree-based transformers for 3d point clouds. In: SIGGRAPH (2023)
61. 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. In: NeurIPS (2023)
62. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pvt v2: Improved baselines with pyramid vision transformer. Computational Visual Media (2022)
63. Wu, C., Ma, Y., Chen, Q., Wang, H., Luo, G., Ji, J., Sun, X.: 3d-stmn: Dependency-driven superpoint-text matching network for end-to-end 3d referring expression segmentation. In: AAAI (2024)
64. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. In: ICCV (2021)
65. Wu, J., Li, X., Xu, S., Yuan, H., Ding, H., Yang, Y., Li, X., Zhang, J., Tong, Y., Jiang, X., Ghanem, B., Tao, D.: Towards open vocabulary learning: A survey. IEEE TPAMI (2024)
66. Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H.: Point transformer v2: Grouped vector attention and partition-based pooling. In: NeurIPS (2022)

67. Wu, Y., Cheng, X., Zhang, R., Cheng, Z., Zhang, J.: Eda: Explicit text-decoupling and dense alignment for 3d visual grounding. In: CVPR (2023)
68. Xiao, Z., Zhang, W., Wang, T., Loy, C.C., Lin, D., Pang, J.: Position-guided point cloud panoptic segmentation transformer. arXiv preprint arXiv:2303.13509 (2023)
69. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: CVPR (2023)
70. Xu, R., Wang, X., Wang, T., Chen, Y., Pang, J., Lin, D.: Pointllm: Empowering large language models to understand point clouds. In: ECCV (2024)
71. Yang, J., Ding, R., Deng, W., Wang, Z., Qi, X.: Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. In: CVPR (2024)
72. Yang, Y.Q., Guo, Y.X., Xiong, J.Y., Liu, Y., Pan, H., Wang, P.S., Tong, X., Guo, B.: Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. arXiv preprint arXiv:2304.06906 (2023)
73. Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al.: mplug-owl: Modularization empowers large language models with multimodality. arXiv:2304.14178 (2023)
74. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: ICCV (2023)
75. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.F., Yang, Y.: Ferret: Refer and ground anything anywhere at any granularity. In: ICLR (2024)
76. Zhang, H., Li, H., Li, F., Ren, T., Zou, X., Liu, S., Huang, S., Gao, J., Zhang, L., Li, C., et al.: Llava-grounding: Grounded visual chat with large multimodal models. arXiv preprint arXiv:2312.02949 (2023)
77. Zhang, H., Ding, H.: Prototypical matching and open set rejection for zero-shot semantic segmentation. In: ICCV (2021)
78. 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)
79. Zhang, Y., Gong, Z., Chang, A.X.: Multi3drefer: Grounding text description to multiple 3d objects. In: ICCV (2023)
80. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV (2021)
81. Zhou, J., Wang, J., Ma, B., Liu, Y.S., Huang, T., Wang, X.: Uni3d: Exploring unified 3d representation at scale. In: ICLR (2024)
82. Zhou, Z., Zhang, Y., Foroosh, H.: Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In: CVPR (2021)
83. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. In: ICLR (2024)
84. Zhu, Z., Ma, X., Chen, Y., Deng, Z., Huang, S., Li, Q.: 3d-vista: Pre-trained transformer for 3d vision and text alignment. In: ICCV (2023)