Uni3DL: A unified model for 3D vision-language understanding

Conference or Workshop Item

Accepted Version

Li, X. ORCID: https://orcid.org/0000-0002-9946-7000, Ding, J., Chen, Z. and Elhoseiny, M. (2024) Uni3DL: A unified model for 3D vision-language understanding. In: ECCV 2024, 29 Sep — 4 Oct 2024, Milan, Italy, pp. 74-92. doi: 10.1007/978-3-031-73337-6_5 Available at https://centaur.reading.ac.uk/119818/

It is advisable to refer to the publisher's version if you intend to cite from the work. See Guidance on citing.

To link to this article DOI: http://dx.doi.org/10.1007/978-3-031-73337-6_5

Publisher: Springer Nature Switzerland

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the End User Agreement.

www.reading.ac.uk/centaur

CentAUR

Central Archive at the University of Reading

University of Reading

Reading's research outputs online

Uni3DL: A Unified Model for 3D Vision-Language Understanding

Xiang Li[1,⋆], Jian Ding[1,⋆], Zhaoyang Chen[2,†], Mohamed Elhoseiny[1]

[1] King Abdullah University of Science and Technology
{xiang.li.1,jian.ding,mohamed.elhoseiny}@kaust.edu.sa
[2] École Polytechnique
zhaoyang.chen@polytechnique.edu

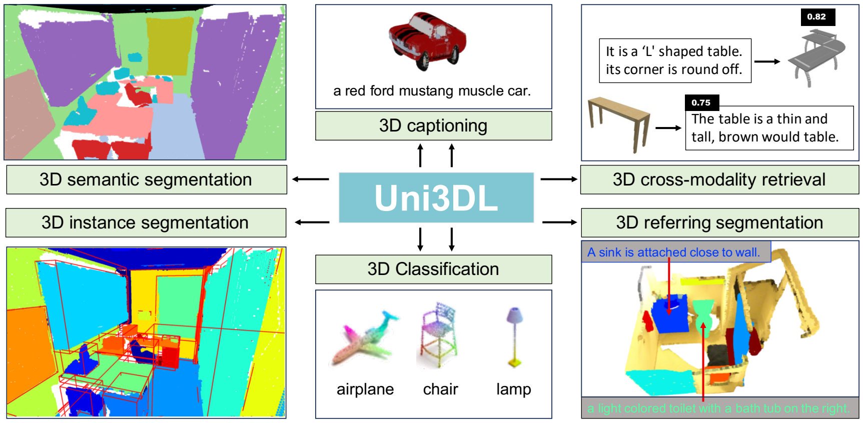

a red ford mustang muscle car.
3D captioning
0.82
It is a 'L' shaped table. its corner is round off.
0.75
The table is a thin and tall, brown would table.
3D semantic segmentation
Uni3DL
3D cross-modality retrieval
3D instance segmentation
3D referring segmentation
3D Classification
A sink is attached close to wall.
airplane
chair
lamp
a light colored toilet with a bath tub on the right.

Figure 1: With a unified architecture, Uni3DL supports diverse 3D vision-language understanding tasks, including semantic segmentation, object detection, instance segmentation, grounded segmentation, captioning, text-3D cross-modal retrieval, (zero-shot) 3D object classification.

Abstract. We present Uni3DL, a unified model for 3D Vision-Language understanding. Distinct from existing unified 3D vision-language models that mostly rely on projected multi-view images and support limited tasks, Uni3DL operates directly on point clouds and significantly broadens the spectrum of tasks in the 3D domain, encompassing both vision and vision-language tasks. At the core of Uni3DL, a query transformer is designed to learn task-agnostic semantic and mask outputs by attending to 3D visual features, and a task router is employed to selectively produce task-specific outputs required for diverse tasks. With a unified architecture, our Uni3DL model enjoys seamless task decomposition and substantial parameter sharing across tasks. Uni3DL has been rigorously evaluated across diverse 3D vision-language understanding tasks, including semantic segmentation, object detection, instance

⋆ Equal contribution

† This work was done when Zhaoyang Chen was an intern at KAUST.

segmentation, visual grounding, 3D captioning, and text-3D cross-modal retrieval. It demonstrates performance on par with or surpassing state-of-the-art (SOTA) task-specific models. We hope our benchmark and Uni3DL model will serve as a solid step to ease future research in unified models in the realm of 3D vision-language understanding. Project page: `https://uni3dl.github.io/`.

Keywords: 3D Vision-Language Understanding · Unified Model · Point Cloud Processing

1 Introduction

3D perception technology stands as a fundamental element in the automatic understanding and operation within the physical world. It enhances various applications, including autonomous driving, robotic navigation, object manipulation, and virtual reality. 3D perception encompasses a broad spectrum of vision and vision-language tasks, such as 3D instance segmentation [10,21,24,29,35,37,53, 66,70], semantic segmentation [30,45,47–49,60,67], visual grounding [5,25,73], object detection [31,68], retrieval [9,54] and captioning [41,63], and has witnessed remarkable advancements.

Despite these successes, task-specific models in 3D perception often lack generalizability, constraining their effectiveness across diverse tasks. In contrast, the broader scientific community, as exemplified by the grand unified theory (GUT) in physics [3,32], has consistently emphasized the importance of unification. Similarly, there is a growing trend towards unified models that integrate vision and language tasks, a concept that has demonstrated significant success in 2D domains [1, 34, 50, 58, 71, 78]. For example, CLIP [50] employs vision-language contrastive learning for zero-shot transfer across different classification tasks. Mask2former [13, 14] leverages a transformer-based architecture for unifying generic segmentation tasks. Moreover, XDecoder [78] and Uni-Perceiver v2 [34] adopt functionality unification modeling [33], covering both vision-only and vision-language tasks. These unified models exhibit greater versatility, efficient data utilization, and adaptability compared to task-specific models, resulting in heightened efficiency and conservation of resources during development.

Extending these successes of unified vision-language modeling in the 2D domain [34,50,71,78] to 3D perception tasks remains a formidable challenge. This difficulty primarily stems from the substantial architectural differences between 2D and 3D models, along with the limited availability of extensive 3D datasets for pre-training purposes. Recent studies [65,72,76], explore adapting CLIP for 3D vision-language modeling. They achieve this by matching projected multi-view images with text inputs. Nevertheless, these methods are mainly designed for 3D object classification. Point-LLM [63] and 3D-LLM [23] directly operate on raw point clouds and explore Large Language Models (LLMs) for 3D visual understanding tasks, including 3D object classification, captioning, and visual question-answering. 3D-VisTA [77] constructs a large-scale 3D scene-text pairs dataset, and perform vision-language pre-training for 3D data without the need of 2D pre-trained models.

Current unified vision-language models in 3D are summarized in Table 1, the scope of tasks supported by current 3D vision-language models is comparatively limited, with dense prediction tasks such as semantic and instance segmentation receiving less attention. Furthermore, many existing models require multi-view images rather than direct training on 3D point clouds. These approaches, while performing well, often result in the loss of critical information (e.g., 3D geometry) and lead to increased model complexity and overhead (multiple projected views required).

In response to these challenging issues, we introduce a unified model for 3D perception that operates on raw point clouds and language. Uni3DL starts with a 3D encoder to extract point features and a text encoder to extract text features, followed by a carefully designed query transformer that enables interaction between latent queries, point features, and text features. A task router with multiple highly shared *functional* heads is designed to selectively produce task-specific outputs for diverse 3D vision-only and vision-language tasks. Our contributions are summarized as:

- We present Uni3DL, a unified model tailored for 3D vision and language comprehension. Its versatile architecture allows for the processing of both point clouds and text inputs, generating diverse outputs including masks, classes, and texts. The model can be directly applied to 3D dense prediction tasks (e.g., instance segmentation).
- With a carefully designed query transformer decoder and task router, our model supports a wide range of vision-only and vision-language tasks within a single, unified architecture, and enjoys seamless task decomposition and substantial parameter sharing across tasks.
- Our results demonstrate enhanced or comparable performance against other multi-task and specialized models across a range of 3D vision-only and vision-language tasks.

2 Related Work

2.1 Unified Vision-Language Models in 2D

The pursuit of unified architectures across diverse tasks is a long-standing goal in computer vision and machine learning. Models like CLIP [50] and ALIGN [28] have made significant progress in merging vision and language through contrastive pre-training on extensive web-sourced image-text pairs, enabling natural language-based zero-shot transfer for various tasks. Yet, these methods have predominantly been applied to classification tasks, indicating room for broader application.

To broaden the scope, existing unified models can be classified into two predominant categories: *I/O unification* and *functional unification* [33]. Inspired by sequence-to-sequence (seq2seq) modeling in NLP [51], I/O unification employs a

Methods	MV	Pretrained FM	SemSeg	InstSeg	GndSeg	GndLoc	Class	Retr	Det	Capt
PointCLIP v2 [76]	✓	CLIP [50], GPT-3 [4]	✓				✓	✓	○	
UniT3D [12]	✓	BERT [19]				✓				✓
3DJCG [5]		Glove [46]				✓				✓
ULIP [64]	✓	CLIP [50]					✓	✓		
ULIP-2 [65]	✓	CLIP [50]					✓	✓		
3D-VisTA [77]		GPT-3 [4]				○				○
Point-LLM [63]		ULIP-2 [65], Vicuna [15]					✓			✓
Point-Bind [20]	✓	OpenCLIP [27]					✓	✓		
Uni3DL (Ours)			✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of various vision-language models in 3D, highlighting their capabilities across diverse tasks. It specifically indicates the utilization of Multi-View (MV) images and delineates the types of Pretrained Foundation Models (FMs) employed. ○ denotes the method is capable of doing the task but requires additional task-specific modules. The abbreviations employed in this comparison are as follows: SemSeg for Semantic Segmentation, InstSeg for Instance Segmentation, GndSeg for Grounded Segmentation, GndLoc for Grounded Localization, Class for Classification, Retr for Retrieval, Det for Detection, Capt for Captioning.

unified decoder to generate homogenous token sequences, which are subsequently processed by task-specific decoders. Notable methods such as Flamingo [1], OFA [58], and GIT [57] primarily focus on image-level tasks, such as image captioning and visual question answering (VQA). Following research like Pix2Seq v2 [11], Unitab [69], and Unified-IO [40] extend this approach by incorporating discrete coordinate tokens in seq2seq modeling for localization tasks. VisionLLM [59] and MiniGPT-2 [8] further enhance vision-language reasoning capabilities using pre-trained large language models. In contrast, *functional unification* models, exemplified by X-Decoder [78] and Uni-Perceiver v2 [34], generate heterogeneous outputs and utilize various routers or headers to produce final outputs for diverse tasks. These models typically comprise a vision encoder, a text encoder, and a unified decoder. Our work aligns with the *functional unification* approach, but with a special focus on 3D vision-language tasks, diverging from the conventional 2D paradigm.

2.2 Unified Vision-Language Models in 3D

Initial efforts in 3D vision-language modeling, such as PointCLIP [72], PointCLIP v2. [76], CLIP2Point [26], and ULIP [64], focus on adapting the CLIP [50] model for 3D applications. Rather than directly processing point clouds, these methods typically rely on projected multi-view images from point clouds during training or testing. Furthermore, these works are mainly designed for 3D object classification and require additional complex components, like 3DETR [42], for scene-level tasks, e.g., object detection.

Building on these developments, Point-LLM [63] marries 3D visual encoders with large language models (e.g., Vicuna [15]), and engages in a dual-stage training process of feature alignment and instruction tuning. This approach equips Point-LLM with proficiency in 3D object classification, captioning, and dialogue. UniT3D [12] introduces a unified transformer-based architecture for 3D vision-language alignment using both bi-directional and seq-to-seq training objectives, which is further fine-tuned for 3D dense captioning and visual grounding. Additionally, 3D-VisTA [77], a pre-trained transformer specialized in 3D vision and text alignment, demonstrates proficiency in multiple tasks including 3D visual grounding and question answering. A notable innovation of 3D-VisTA is the introduction of the Scanscribe dataset, a pioneering dataset for 3D vision-language pre-training. Nevertheless, 3D-VisTA [77] requires complex task-specific heads for different tasks.

In conclusion, current models mostly only support limited tasks and require complex task-specific module design, as summarized in Table 1. Furthermore, they generally depend on multi-view rendering images. Our Uni3DL, however, extensively extends the range of tasks it can handle, particularly emphasizing dense prediction tasks such as semantic segmentation, instance segmentation, and grounded segmentation, all within a unified architecture using highly shared parameters. A distinctive aspect of our approach is its direct operation on point clouds, thereby bypassing the need for multi-view images.

3 Uni3DL

3.1 Method overview

Uni3DL is a versatile architecture tailored for diverse 3D vision-language tasks, including 3D object classification, captioning, text-3D cross-modal retrieval, semantic and instance segmentation, and visual grounding. This architecture encompasses four integral modules: a **Text Encoder** for text feature extraction; a **Point Encoder** dedicated to point feature learning; a **Query Transformer Module** with a sequence of cross-attention and self-attention layers to learn relations among object and text queries and point features; and a **Task Router**, adaptable and comprising multiple functional heads, including a text generation head for generating text outputs, a class head for object classification, and a mask head for producing segmentation masks, a grounding head for text-to-object grounding, and a 3D-text matching head for 3D-text cross-modal matching. With the combination of these functional heads, the task router selectively combines functional heads for different tasks. For example, the instance segmentation task combines object classification and mask prediction heads.

Given an input point cloud $\mathbf{P}$, our Uni3DL leverages a 3D U-Net $\mathcal{E}_I$ to extract hierarchy point features $\mathbf{V}$, along with a text encoder $\mathcal{E}_T$ to obtain text features $\mathbf{F}_T \in \mathbb{R}^{L_T \times C}$. Point features, text features, along with learnable latent queries $\mathbf{F}_Q \in \mathbb{R}^{Q \times C}$ are fed into a unified decoder network to predict mask and semantic outputs, formulated as:

$$\mathbf{O}^m, \mathbf{O}^s = \mathcal{D}([\mathbf{F}_Q; \mathbf{F}_T], \mathbf{V}), \tag{1}$$

where $\mathbf{O}^m$ and $\mathbf{O}^s$ denote mask outputs and semantic outputs, $[;]$ denotes feature concatenation.

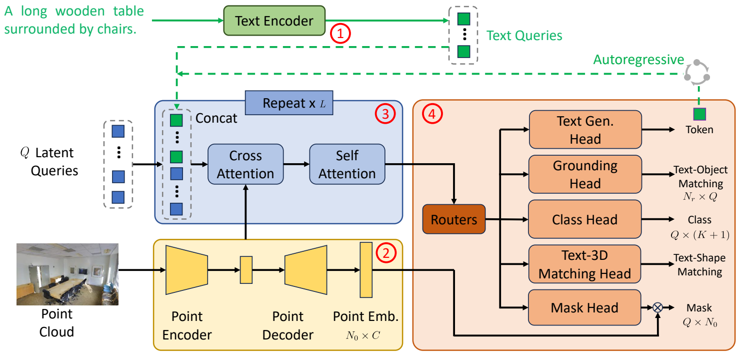


Fig. 2: Overview of the Uni3DL Model. The Uni3DL is engineered for multifaceted 3D data tasks, including classification, retrieval, captioning, semantic and instance segmentation, as well as visual grounding. The architecture is composed of four principal modules: ① a **Text Encoder** for text feature extraction; ② a **Point Encoder** for point feature learning; ③ a **Query Transformer** Module, which is the cornerstone of the system with a sequence of cross-attention and self-attention operations between latent queries, text queries and voxel features derived from the Point Encoder; and ④ a **Task Router** module, which comprises, as needed for the given task, text generation head for generating descriptive text, a grounding head for text-to-object grounding, a class head for object classification task, a mask head dedicated to segmentation, and a text-3D matching head for 3D-text cross-modal matching. The text generation head functions in an autoregressive manner and predicts one token at each forward step.

3.2 Point Cloud and Text Encoder

The architecture of our point feature extraction network employs a sparse 3D convolutional U-net structure based on the MinkowskiEngine framework [16], featuring both an encoder and a decoder network. A colored input point cloud, denoted as $\mathbf{P} \in \mathbb{R}^{N_0 \times 6}$, undergoes quantization into N_0 voxels represented as $\mathbf{V}_0 \in \mathbb{R}^{N_0 \times 3}$, with each voxel capturing the average RGB color from the points it contains as the initial voxel features. Several convolutional and downsampling layers are sequentially applied to extract high-level voxel features, followed by deconvolutional and upsampling layers to recover voxels to their original resolutions. Supposing the U-Net has S stages of feature blocks, at each stage $s \in [1, .., S]$, we can get voxel features $\mathbf{V}_s \in \mathbb{R}^{N_s \times C_s}$, where N_s denotes the number of valid voxels at stage s, and C_s denotes the corresponding feature dimension. We then project all voxel features to the same dimension D, resulting in a set of feature maps $\{\mathbf{V}_s \in \mathbb{R}^{N_s \times C}\}_{s=1}^{S}$. The last feature map ($\mathbf{V}_S$) is used as point embeddings to calculate per-point mask, while the remaining feature maps $\{\mathbf{V}_s\}_{s=1}^{S-1}$ are fed into the transformer module to enhance latent and text queries. For text inputs, we use the CLIP tokenizer [50] along with a transformer-based network for text feature learning.

3.3 Query Transformer Module

We follow query-based transformer architecture [6, 38, 52, 78] to design our decoder network. Given voxel features $\{\mathbf{V}_s\}_{s=1}^{S-1}$, our transformer module refines latent queries $\mathbf{F}_Q$ and text queries $\mathbf{F}_T$ by a sequence of L decoder layers. At each layer $l = [1..., L]$, we refine queries by cross-attending to voxel features, formulated as:

$$[\hat{\mathbf{F}}_Q^l; \hat{\mathbf{F}}_T^l] = \text{Cross-Att}([\mathbf{F}_Q^{l-1}; \mathbf{F}_T^{l-1}], \mathbf{V}_s). \tag{2}$$

We repeat this process for each feature level $s = [1, 2, ..., S-1]$.

Masked Attention. To enhance object localization capability, we follow the attention block design in Mask2Fomer [13] and use masked attention instead of vanilla cross-attention where each query only attends to masked voxels predicted by the previous layer.

Voxel Sampling. Point clouds in a batch usually have different numbers of points, leading to differing voxel quantities. Current transformer implementations generally require a fixed length of inputs in each batch entry. To enable efficient batch-wise training, for each feature level s, before feeding voxel features into the decoder network. The sampled voxel features are then utilized across all cross-attention layers following [52].

We further enhance object and text queries through self-attention layers and feed-forward layers, formulated as:

$$[\hat{\mathbf{F}}_Q^l; \hat{\mathbf{F}}_T^l] = \text{Self-Att}([\hat{\mathbf{F}}_Q^l; \hat{\mathbf{F}}_T^l]); \quad [\mathbf{F}_Q^l; \mathbf{F}_T^l] = \text{FFN}([\hat{\mathbf{F}}_Q^l; \hat{\mathbf{F}}_T^l]). \tag{3}$$

3.4 Task Router

To support diverse 3D vision-language tasks, we design multiple functional heads thus different tasks can be achieved by compositions of heads. As a result, there is a high degree of parameter sharing across different tasks. For instance, the mask head is utilized for semantic, instance, and grounded segmentation tasks. Specifically, the 3D instance segmentation task includes two functional heads, object classification, and mask prediction; while the 3D grounded segmentation task requires a mask head and a grounding head. Consequently, the Uni3DL model harnesses a consistent set of parameters, while applying unique routing strategies for each specific task, ensuring efficient task decomposition and substantial parameter reuse across different tasks. We show the head composition of different tasks in Table 2.

Object Classification Head. We select the first Q output semantic outputs for object classification. Given refined semantic queries $\mathbf{O}^s \in \mathbb{R}^{Q \times C}$, and K semantic classes with additional background class. We first feed all $K+1$ class names to the text encoder to get class embeddings $\mathbf{C}_{emb} \in \mathbb{R}^{(K+1) \times C}$, and calculate classification probabilities as $\mathbf{O}_c = \mathbf{O}^s \cdot \mathbf{C}_{emb}^T$, where $\cdot$ denotes the dot product between matrices.

During training, we calculate cross-entropy loss between predicted classification probabilities O_c and ground truth (GT) class labels C_{gt} to formulate classification loss as:

$$\mathcal{L}_{cls} = \lambda_{cls} \text{CE}(\mathbf{O}_c, C_{gt}), \tag{4}$$

Task	Obj-Cls	Mask	Grounding	Text-Gen	Matching
Semantic Segmentation	✓	✓			
Instance Segmentation	✓	✓			
Grounded Segmentation		✓	✓		
Captioning				✓	
Retrieval					✓
Shape Classification					✓

Table 2: Head compositions of different tasks. Obj-Cls denotes object classification head, Text-Gen denotes text generation head, and Matching denotes text-3D matching.

where CE denotes cross-entropy loss.

Mask Head. Given mask output $\mathbf{O}^m \in \mathbb{R}^{Q\times C}$, and full-resolution voxel features $\mathbf{V}_S \in \mathbb{R}^{N_0\times C}$, we calculate voxel mask as $\mathbf{O}_m = \mathbf{O}^m \cdot \mathbf{V}_S^T$. The output voxel mask $\mathbf{O}_m \in \mathbb{R}^{Q\times N_0}$, where each row denotes an object mask for the corresponding latent query.

During training, given ground truth object mask $\mathbf{M}_{gt}$, we calculate mask loss as:

$$\mathcal{L}_{mask} = \lambda_{bce}\text{BCE}(\mathbf{O}_m, \mathbf{M}_{gt}) + \lambda_{dice}\text{DICE}(\mathbf{O}_m, \mathbf{M}_{gt}), \tag{5}$$

where BCE and DICE denote binary cross-entropy loss and dice loss respectively.

Grounding Head. Visual grounding requires matching text descriptions to visual objects. We first generate text embeddings $\mathbf{T}_{emb} \in \mathbb{R}^{N_r\times C}$ by feeding all grounding sentences to the text encoder. We select the first Q output semantic queries $\mathbf{O}^s \in \mathbb{R}^{Q\times C}$ as object embeddings. Then, we calculate object-text similarity by

$$\mathbf{S}_t = \text{Softmax}(e^{\eta}\, \mathbf{T}_{emb} \cdot (\mathbf{O}^s)^T), \tag{6}$$

where $\mathbf{S}_t \in \mathbb{R}^{N_r\times Q}$ and η denotes a learnable scaling parameter. Softmax operation is applied on the last dimension.

Following DETR [6], we use Hungarian matching to get ground truth matching labels $T_{gt} \in \mathbb{R}^{N_r}$. We modified the original mask matching module in DETR to adapt it for voxel masks. We then calculate cross-entropy loss as:

$$\mathcal{L}_{gc} = \text{CE}(\mathbf{S}_t, T_{gt}). \tag{7}$$

Following the common practice of 3D visual grounding practice [5, 7], we design a lightweight classification head that takes text queries as inputs and predicts the existence of all K candidate object categories. Given input text queries $\mathbf{T}_{emb} \in \mathbb{R}^{N_r\times C}$, a single-layer MLP network is utilized to predict the probabilities matrix $\mathbf{T}_{cls} \in \mathbb{R}^{N_r\times K}$ over K candidate object categories. We then calculate multi-label classification loss as:

$$\mathcal{L}_{gtxt} = \text{BCE}(\mathbf{T}_{cls}, \mathbf{T}_{cls}^{gt}), \tag{8}$$

where $\mathbf{T}_{cls}^{gt} \in \mathbb{R}^{N_r\times K}$ denotes the ground truth labels of category existence.

Additional grounding mask $\mathcal{L}_{gmask}$ is calculated similarly to the mask head. The overall grounding loss is calculated as:

$$\mathcal{L}_{grd} = \lambda_{gc}\mathcal{L}_{gc} + \mathcal{L}_{gmask} + \mathcal{L}_{gtxt}. \tag{9}$$

Text Generation Head. In the context of 3D captioning, our method begins by generating text embeddings for each token within the vocabulary, which comprises V tokens, utilizing the text encoder. Subsequently, we use the last L_T semantic outputs generated by the decoder network and calculate the dot product against the token embeddings, resulting in an affinity matrix $\mathbf{S}_{cap} \in \mathbb{R}^{L_T \times V}$. The cross-entropy loss is calculated as:

$$\mathcal{L}_{cap} = \lambda_{cap} \mathrm{CE}(\mathbf{S}_{cap}, y_{cap}), \tag{10}$$

where y_{cap} is the ground truth token indices.

During training, a causal masking strategy is adopted in all self-attention layers of the decoder network. During inference, our model predicts one token at each time and gets 3D captions in an autoregressive way.

Text-3D Matching Head. This head can be used for text-3D cross-modal retrieval and (zero-shot) 3D shape classification tasks. To predict text-3D matching, the last output semantic token is used as the shape embedding with a dimension of $\mathbb{R}^{1 \times C}$. Given a batch of B text-shape pairs, the matching head computes the similarities between 3D shape embeddings and corresponding text embeddings as $\mathbf{S}_{ret} \in \mathbb{R}^{B \times B}$, and calculates retrieval loss as:

$$\mathcal{L}_{ret} = \lambda_{ret} \mathrm{CL}(\mathbf{S}_{ret}, y_{ret}), \tag{11}$$

where $y_{cap} \in \mathbb{R}^{1 \times B}$ denotes the ground truth matching indices. CL denotes vision-language contrastive loss defined in CLIP [50].

Multi-Task Training. During pretraining, we simultaneously train the whole network with all functional heads. The overall loss is formulated as:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{mask} + \mathcal{L}_{grd} + \mathcal{L}_{cap} + \mathcal{L}_{ret}. \tag{12}$$

4 Experiments

4.1 Dataset

ScanNet (v2) [17] captures more than 1,500 3D scans. Following the official benchmark, we use 1,201 scenes for training, 312 for validation. There are in total 20 semantic labels, 18 of which are instance classes.

ScanRefer [7] dataset contains 51,583 referring descriptions of 11,046 objects from 800 ScanNet scenes. We use 562 scenes for training and 141 scenes for evaluation.

Cap3D Objaverse [41] dataset, is derived from Objaverse with around 800K objects. It features 660K 3D-text pairs, created using an automated captioning process. We randomly select 80% for training and the remaining 20% for evaluation[3].

[3] To ensure a fair comparison with PointLLM, we filter out 200 objects used for benchmark evaluation from our training set and report the performance on the same 200 objects.

For model evaluation, we additionally use S3DIS [2] to evaluate semantic and instance segmentation, Text2Shape [9] to evaluate text-3D cross-modal retrieval.
S3DIS dataset contains 6 large-scale areas with 271 scenes, and 13 semantic categories are annotated. Following previous works, we use 68 scenes in Area 5 for validation and the others for model training.
Text2Shape [9] contains 8,447 table instances and 6,591 chair instances from the ShapeNet dataset, along with 75,344 natural language descriptions. We use the same training/test split as [9].

4.2 Implementation Details

In this work, we employ 150 latent queries and one additional latent query for scene-level tasks. The point encoder-decoder network is based on Minkowski Res16UNet34C [16] and pretraiend from Mask3D [52], and we use 12 transformer layers for the language encoder. Our Query Transformer module consists of 15 ($L = 15$) transformer layers. The segmentation weights $\lambda_{\text{cls}}, \lambda_{\text{bce}}, \lambda_{\text{dice}}$ are set 2.0, 5.0, 5.0, grounding classification weight λ_{gc} to 0.4, captioning and retrieval weight $\lambda_{\text{cap}}, \lambda_{\text{ret}}$ are set to 2.0.

During pretraining, we employ datasets including ScanNet (v2) instance segmentation, ScanRefer, and Cap3D Objaverse. Notably, ScanRefer is based on ScanNet, and Cap3D Objaverse shares numerous object categories with ScanNet. The alignment in object types and functional heads across these tasks justifies their combined processing in the same batch for joint training. The training process spans 50 epochs using the AdamW optimizer [39], taking approximately 20 hours on four NVIDIA A100 GPUs. During inference, the top 200 (for S3DIS) and top 500 (for ScanNet (v2)) instances with the highest classification scores are retained for the instance segmentation task. Details about the pretraining and task-specific fine-tuning setups can be found in the supplementary material.

4.3 3D Semantic/Instance Segmentation

We compare 3D semantic segmentation, object detection, and instance segmentation performance with previous STOA methods in Table 3. From the table, our Uni3DL method achieves better or comparable performance on general segmentation and detection tasks on S3DIS and ScanNet (v2)datasets. Figure 3 shows qualitative results of our Uni3DL on S3DIS and ScanNet (v2) datasets.

4.4 3D Visual Grounding

We compare the 3D grounded segmentation performance of our Uni3DL with TGNN (GRU) [25] in Table 3. Our method achieves significantly better performance than TGNN method as indicated by instance-average IoU, and accuracy at the IoU thresholds of 0.25 and 0.5. It should be noted we report grounded segmentation performance rather than grounded localization to ensure a fair comparison with TGNN. Grounded segmentation is more challenging than grounded

	Semantic Segmentation			Object Detection		Instance Segmentation				Grounded Segmentation			3D Captioning			3D Retrieval	
Method	S3DIS (Area 5)		SN Val	SN Val		SN Val		S3DIS (Area 5)		ScanRefer			Cap3D			Text2Shape	
	mIoU	mAcc	mIoU	bAP_{50}	bAP_{25}	mAP	mAP_{50}	mAP_{50}	mAP_{25}	mIoU	Acc@0.25	Acc@0.5	B-1	R	M	R@1	R@5
MinkowskiNet42 [16]	67.1	74.4	72.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FastPointTransformer [45]	68.5	76.5	72.1	-	-	-	-	-	-	-	-	-	-	-	-	-	
PointNeXt-XL [49]	71.1	77.2	71.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
StratifiedTransformer [30]	72.0	<u>78.1</u>	73.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PointTransformerV2 [60]	71.6	77.9	74.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EQ-Net [68]	71.3	*	<u>75.3</u>	-	-	-	-	-	-	-	-	-		-	-	-	-
Swin3D [67]	<u>72.5</u>	*	75.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Swin3D† [67]	**73.0**	*	75.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
VoteNet [62]		-	-	33.5	58.6	-	-	-	-	-	-	-	-	-	-	-	-
3DETR [43]	-	-	-	47.0	65.0	-	-	-	-	-	-	-	-	-	-	-	-
CAGroup3D [56]	-	-	-	61.3	<u>75.1</u>	-	-	-	-	-	-	-	-	-	-	-	-
PointGroup [29]	*	*	*	*	*	34.8	56.7	57.8	*	-	-	-	-	-	-	-	-
MaskGroup [75]	*	*	*	*	*	42.0	63.3	65.0	*	-	-	-	-	-	-	-	-
SSTNet [35]	*	*	*	*	*	49.4	64.3	59.3	*	-	-	-	-	-	-	-	-
SoftGroup [55]	*	*	*	59.4	71.6	50.4	<u>76.1</u>	66.1	*	-	-	-	-	-	-	-	-
Mask3D [52]	*	*	*	56.2	70.2	55.2	73.7	<u>68.4</u>	<u>75.2</u>	-	-	-	-	-	-	-	-
Mask-Att-Free† [31]	*	*	*	<u>63.9</u>	73.5	<u>58.4</u>	75.9	**69.1**	**75.7**	-	-	-	-	-	-	-	-
TGNN (GRU) [25]	-	-	-	-	-	-	-	-	-	26.1	35.0	29.0	-	-	-	-	-
TGNN (BERT) [25]	-	-	-	-	-	-	-	-	-	<u>27.8</u>	<u>37.5</u>	<u>31.4</u>	-	-	-	-	-
InstructBLIP-7B [18]	-	-	-	-	-	-	-	-	-	-	-	-	11.2	13.9	14.9	*	*
InstructBLIP-13B [18]	-	-	-	-	-	-	-	-	-	-	-	-	<u>12.6</u>	<u>15.0</u>	**16.0**	*	*
PointLLM-7B [63]	-	-	-	-	-	-	-	-	-	-	-	-	8.0	11.1	15.2	*	*
PointLLM-13B [63]	-	-	-	-	-	-	-	-	-	-	-	-	9.7	12.8	15.3	*	*
FTST [9]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.2	1.6
FMM [9]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.2	2.4
Y2S [22]	-	-	-	-	-	-	-	-	-	-	-	-	*	*	*	2.9	9.2
Parts2Words (no parts) [54]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	<u>5.1</u>	<u>17.2</u>
Ours	**72.7**	**79.3**	**76.2**	**67.7**	**77.1**	**60.9**	**80.9**	65.3	74.3	**32.3**	**39.4**	**36.4**	**31.6**	**33.1**	<u>14.4</u>	**5.7**	**19.7**

Table 3: Performance of our Uni3DL on different segmentation and VL tasks. Uni3DL achieves the best performance on 14 out of 17 metrics. 'SN' denotes the ScanNet (v2) dataset. '*' indicates the model is capable of the task without a reported metric, and '-' signifies the model lacks this specific capability. The results highlighted in **bold** and <u>underline</u> denote the best and second-best outcomes, respectively, for each column. Note that Swin3D† uses extra training data (Structure3D [74]).

localization because minor boundary inaccuracies in segmentation masks minimally impact segmentation IOU, but can significantly alter bounding box locations. Figure 4 shows qualitative results of our method. More qualitative results and grounded localization performance are presented in the supplementary file.

4.5 3D Captioning

From Table 3, our Uni3DL model outperforms existing methods in 3D captioning on the Cap3D Objaverse dataset. On the BLEU-1 [44] and ROUGE-L [36] scores, our method beats precious STOA methods by a large margin (more than 20%). Qualitative analyses, illustrated in Figure 5, demonstrate our caption predictions closely align with the ground truth. Additional qualitative results are presented in the supplementary file.

4.6 Text-to-3D Retrieval

We evaluate text-to-3D retrieval performance on the Text2Shape ShapeNet subset. From Table 3, our Uni3DL model achieves better text-to-3D retrieval performance than previous STOA task-specific methods, including FTST [9], FMM [9], Y2S [22], and Parts2Words [54], as indicated by recall scores R@1 and R@5. Note that for the Parts2Words method, we primarily compare its performance without using part information for a fair comparison. Qualitative results are provided in the supplementary file.

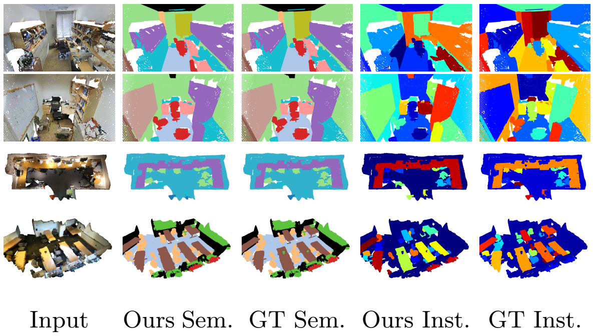
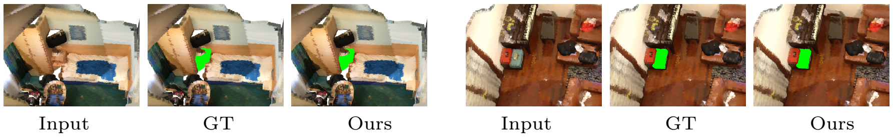


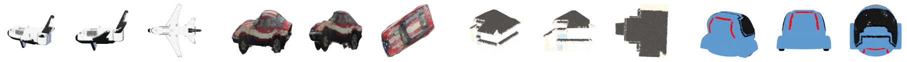
Fig. 3: 3D Segmentation results on S3DIS (top) and ScanNet (bottom) datasets.



Refer: a brown wooden nightstand. it's between the end of the bed and close to the wall.

Refer: this is a green toolbox. the green toolbox is in front of a red toolbox on the floor next to a piano.

Fig. 4: Results of grounded segmentation on the ScanRefer dataset. Grounded masks are shown in green.



GT: a small white NASA space shuttle airplane flying in the sky.
Ours: a small white airplane flying in the air

GT: an old red and white car with an American flag painted on it.
Ours: an old red and white race car with its rear paintings featuring stickers

GT: a white house with a roof.
Ours: a white house with a roof and stairs

GT: a small blue toy car with red accents and a helmet on top.
Ours: a small blue toy vehicle, resembling a car with wheels

Fig. 5: 3D captioning results on Cap3D Objaverse dataset.

4.7 Zero-Shot 3D Object Classification

We evaluate the zero-shot 3D classification performance on the ModelNet10/40 dataset [61]. Experiments demonstrate that our Uni3DL model achieves competitive performance compared to previous SOTA methods. Additional details can be found in the supplementary file.

4.8 Ablation Study

Effect of Pretraining. We evaluate the impact of pretraining on downstream tasks. Ablation experiments are conducted by training separate models from scratch for various tasks, including ScanNet (v2) semantic segmentation, S3DIS instance segmentation, ScanRefer grounded segmentation, and Text2Shape retrieval. As evidenced in Table 4, the pretraining stage significantly enhances performance across all downstream tasks. We show the qualitative comparison of the baseline model trained from scratch and our finetuned model on the S3DIS instance segmentation dataset in Figure 6. From this figure, the baseline model fails to capture the geometry structures of objects and may produce noisy masks; our finetuned model can better extract objects with consistent boundaries.

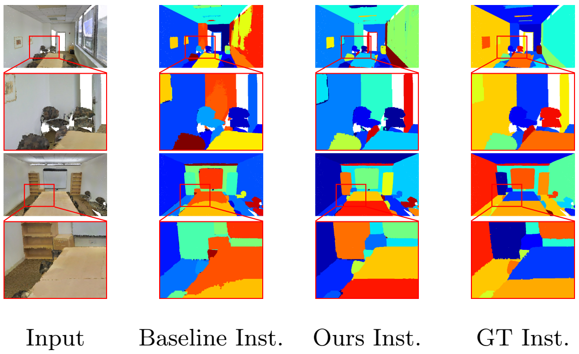


Fig. 6: Instance (Inst.) segmentation results on S3DIS dataset. We show results of the baseline method trained from scratch and our finetuned model.

Task	Semantic Segmentation	Instance Segmentation	Grounded Segmentation	Retrieval
	SN Val	S3DIS (Area 5)	ScanRefer	Text2Shape
	mIoU/mAcc	mAP_{50} / mAP_{25}	Acc@0.25/Acc@0.5	R@1/R@5
From scratch	72.3/81.8	61.7/71.7	33.8/31.4	2.4/7.7
Ours	**76.2/84.8**	**65.3/74.3**	**39.4/36.4**	**5.7/19.7**

Table 4: Ablation of pertaining.

Effect of different pertaining tasks. We further investigate the effect of each pertaining task, including instance/grounded segmentation, 3D captioning, and text-to-3D retrieval. In Table 5, we keep grounded segmentation while evaluating the significance of remaining pretraining tasks. From Table 5, we have the following findings: 1) Instance segmentation benefits both grounded segmentation and text-3D cross-modal retrieval. Without instance segmentation task, the grounded segmentation Acc@0.25 drops from 37.8% to 35.8%. This is because the grounding task itself is based on instance identification. Instance segmentation also helps to better learn object-text alignment and benefits text-to-3D cross-modal retrieval. 2) Caption and retrieval tasks benefit each other. Without pertaining on the captioning task, the text-3D cross-modal retrieval accuracy drops from 5.5% (resp., 15.5%) to 5.0% (resp., 12.8%) in terms of text-to-shape R@1 (resp., R@5) on the Cap3D dataset. Without pertaining on the retrieval task, the captioning performance drops from 16.8% (resp., 13.7) to 13.7% (resp., 11.2) in terms of B-1 (resp., ROUGE-L) scores on the Cap3D dataset.

Task	Grounded Segmentation	Captioning	Retrieval
	ScanRefer	Cap3D	Cap3D
	Acc@0.25/Acc@0.5	B-1/R	T2S R@1/R@5
Ours (β=1)	37.8/34.2	16.8/13.7	5.5/15.5
- Retrieval	38.8/35.8	13.5/11.2	N/A
- Captioning	38.3/35.5	N/A	5.0/12.8
- Instance Segmentation	35.8/31.0	18.2/14.9	4.0/11.0
Ours (β=0.5)	38.1/36.5	15.7/10.3	5.5/10.5
Ours (β=2)	36.4/34.0	18.3/13.4	6.0/16.0
Ours (β=5)	35.2/31.3	17.7/12.0	4.0/15.5
Ours + alt. (β=1)	36.8/33.6	14.8/14.4	5.0/13.0

Table 5: Ablation of pertaining tasks and scene-object task balance. Ours + alt. means our model with alternative training.

Scene-object task balance. During the pretraining phase, we include both object understanding (including object captioning, and text-to-3D cross-modal retrieval) and scene understanding (specifically, instance and grounded segmentation) tasks. Achieving a proper balance between these two types of tasks—each characterized by unique data distributions—is crucial in our multi-task training framework. To manage this balance, we modulate the weights assigned to the object understanding tasks ($\lambda_{cap}, \lambda_{ret}$) using different scaling factors ($\beta = 0.5, 1, 2, 5$), where $\beta = 1$ represents the baseline setting. As demonstrated in the middle section of Table 5, increasing parameter β from 0.5 to 2 slightly improves performance in two object understanding tasks; however, further increment to 5 hurts the performance. Meanwhile, increasing β from 0.5 to 5 marginally diminishes the results in scene understanding tasks. It should be highlighted that adjusting the parameter β from 0.5 to 5 has a small impact on performance across tasks, mostly less than a 2% change, demonstrating Uni3DL's robustness to variations in balancing weights.

Alternative training. We explore alternate training between object and scene-level tasks. Results in the lower section of Table 5 indicate that alternate training results in marginally inferior performance compared to joint training.

5 Conclusion

We introduce Uni3DL, a unified model for generalized 3D vision and language understanding tasks. We design a query transformer to attentively align 3D features with latent and text queries. A task router module with multiple functional heads is designed to support diverse vision-language tasks, including 3D object classification, 3D semantic/instance segmentation, 3D object detection, 3D grounded segmentation, 3D captioning, and text-3D cross-modal retrieval. Experiments on multiple benchmark datasets show comparable or even superior performance of our Uni3DL model compared to the previous SOTA methods.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems **35**, 23716–23736 (2022)
2. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1534–1543 (2016)
3. Baez, J., Huerta, J.: The algebra of grand unified theories. Bulletin of the American Mathematical Society **47**(3), 483–552 (2010)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
5. Cai, D., Zhao, L., Zhang, J., Sheng, L., Xu, D.: 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16464–16473 (2022)
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
7. Chen, D.Z., Chang, A.X., Nießner, M.: Scanrefer: 3d object localization in rgb-d scans using natural language. In: European conference on computer vision. pp. 202–221. Springer (2020)
8. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. arXiv preprint arXiv:2310.09478 (2023)
9. Chen, K., Choy, C.B., Savva, M., Chang, A.X., Funkhouser, T., Savarese, S.: Text2shape: Generating shapes from natural language by learning joint embeddings. In: Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14. pp. 100–116. Springer (2019)
10. Chen, S., Fang, J., Zhang, Q., Liu, W., Wang, X.: Hierarchical aggregation for 3d instance segmentation. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15447–15456 (2021). `https://doi.org/10.1109/ICCV48922.2021.01518`
11. Chen, T., Saxena, S., Li, L., Lin, T.Y., Fleet, D.J., Hinton, G.E.: A unified sequence interface for vision tasks. Advances in Neural Information Processing Systems **35**, 31333–31346 (2022)
12. Chen, Z., Hu, R., Chen, X., Nießner, M., Chang, A.X.: Unit3d: A unified transformer for 3d dense captioning and visual grounding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18109–18119 (2023)
13. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1290–1299 (2022)
14. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Advances in Neural Information Processing Systems **34**, 17864–17875 (2021)

15. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023) (2023)
16. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3075–3084 (2019)
17. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
18. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning (2023)
19. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
20. 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)
21. Han, L., Zheng, T., Xu, L., Fang, L.: Occuseg: Occupancy-aware 3d instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2940–2949 (2020)
22. Han, Z., Shang, M., Wang, X., Liu, Y.S., Zwicker, M.: Y2seq2seq: Cross-modal representation learning for 3d shape and text by joint reconstruction and prediction of view and word sequences. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 126–133 (2019)
23. Hong, Y., Zhen, H., Chen, P., Zheng, S., Du, Y., Chen, Z., Gan, C.: 3d-llm: Injecting the 3d world into large language models. arXiv preprint arXiv:2307.12981 (2023)
24. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4421–4430 (2019)
25. Huang, P.H., Lee, H.H., Chen, H.T., Liu, T.L.: Text-guided graph neural networks for referring 3d instance segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1610–1618 (2021)
26. Huang, T., Dong, B., Yang, Y., Huang, X., Lau, R.W., Ouyang, W., Zuo, W.: Clip2point: Transfer clip to point cloud classification with image-depth pretraining. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22157–22167 (2023)
27. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: Openclip (Jul 2021). https://doi.org/10.5281/zenodo.5143773, https://doi.org/10.5281/zenodo.5143773, if you use this software, please cite it as below.
28. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021)
29. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: Pointgroup: Dual-set point grouping for 3d instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition. pp. 4867–4876 (2020)

30. Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J.: Stratified transformer for 3d point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8500–8509 (2022)
31. Lai, X., Yuan, Y., Chu, R., Chen, Y., Hu, H., Jia, J.: Mask-attention-free transformer for 3d instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3693–3703 (October 2023)
32. Langacker, P.: Grand unified theories and proton decay. Physics Reports **72**(4), 185–385 (1981)
33. Li, C., Gan, Z., Yang, Z., Yang, J., Li, L., Wang, L., Gao, J.: Multimodal foundation models: From specialists to general-purpose assistants. arXiv preprint arXiv:2309.10020 **1** (2023)
34. Li, H., Zhu, J., Jiang, X., Zhu, X., Li, H., Yuan, C., Wang, X., Qiao, Y., Wang, X., Wang, W., et al.: Uni-perceiver v2: A generalist model for large-scale vision and vision-language tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2691–2700 (2023)
35. Liang, Z., Li, Z., Xu, S., Tan, M., Jia, K.: Instance segmentation in 3d scenes using semantic superpoint tree networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2783–2792 (October 2021)
36. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)
37. Liu, S.H., Yu, S.Y., Wu, S.C., Chen, H.T., Liu, T.L.: Learning gaussian instance segmentation in point clouds. arXiv preprint arXiv:2007.09860 (2020)
38. Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. Advances in Neural Information Processing Systems **33**, 11525–11538 (2020)
39. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
40. Lu, J., Clark, C., Zellers, R., Mottaghi, R., Kembhavi, A.: Unified-io: A unified model for vision, language, and multi-modal tasks. arXiv preprint arXiv:2206.08916 (2022)
41. Luo, T., Rockwell, C., Lee, H., Johnson, J.: Scalable 3d captioning with pretrained models. arXiv preprint arXiv:2306.07279 (2023)
42. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2906–2917 (2021)
43. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2906–2917 (2021)
44. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
45. Park, C., Jeong, Y., Cho, M., Park, J.: Fast point transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16949–16958 (2022)
46. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
47. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)

48. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
49. Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: Pointnext: Revisiting pointnet++ with improved training and scaling strategies. Advances in Neural Information Processing Systems **35**, 23192–23204 (2022)
50. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021)
51. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
52. Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., Leibe, B.: Mask3d: Mask transformer for 3d semantic instance segmentation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 8216–8223. IEEE (2023)
53. Sun, J., Qing, C., Tan, J., Xu, X.: Superpoint transformer for 3d scene instance segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2393–2401 (2023)
54. Tang, C., Yang, X., Wu, B., Han, Z., Chang, Y.: Parts2words: Learning joint embedding of point clouds and texts by bidirectional matching between parts and words. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6884–6893 (2023)
55. Vu, T., Kim, K., Luu, T.M., Nguyen, T., Yoo, C.D.: Softgroup for 3d instance segmentation on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2708–2717 (June 2022)
56. Wang, H., Dong, S., Shi, S., Li, A., Li, J., Li, Z., Wang, L., et al.: Cagroup3d: Class-aware grouping for 3d object detection on point clouds. Advances in Neural Information Processing Systems **35**, 29975–29988 (2022)
57. Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C., Wang, L.: Git: A generative image-to-text transformer for vision and language. arXiv preprint arXiv:2205.14100 (2022)
58. Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., Yang, H.: Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In: International Conference on Machine Learning. pp. 23318–23340. PMLR (2022)
59. Wang, W., Chen, Z., Chen, X., Wu, J., Zhu, X., Zeng, G., Luo, P., Lu, T., Zhou, J., Qiao, Y., et al.: Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. arXiv preprint arXiv:2305.11175 (2023)
60. Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H.: Point transformer v2: Grouped vector attention and partition-based pooling. Advances in Neural Information Processing Systems **35**, 33330–33342 (2022)
61. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
62. Xie, Q., Lai, Y.K., Wu, J., Wang, Z., Lu, D., Wei, M., Wang, J.: Venet: Voting enhancement network for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3712–3721 (2021)
63. Xu, R., Wang, X., Wang, T., Chen, Y., Pang, J., Lin, D.: Pointllm: Empowering large language models to understand point clouds. arXiv preprint arXiv:2308.16911 (2023)

64. Xue, L., Gao, M., Xing, C., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S.: Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1179–1189 (2023)
65. Xue, L., Yu, N., Zhang, S., Li, J., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S.: Ulip-2: Towards scalable multimodal pre-training for 3d understanding. arXiv preprint arXiv:2305.08275 (2023)
66. Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N.: Learning object bounding boxes for 3d instance segmentation on point clouds. Advances in neural information processing systems **32** (2019)
67. 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)
68. Yang, Z., Jiang, L., Sun, Y., Schiele, B., Jia, J.: A unified query-based paradigm for point cloud understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8541–8551 (2022)
69. Yang, Z., Gan, Z., Wang, J., Hu, X., Ahmed, F., Liu, Z., Lu, Y., Wang, L.: Unitab: Unifying text and box outputs for grounded vision-language modeling. In: European Conference on Computer Vision. pp. 521–539. Springer (2022)
70. Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3947–3956 (2019)
71. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021)
72. Zhang, R., Guo, Z., Zhang, W., Li, K., Miao, X., Cui, B., Qiao, Y., Gao, P., Li, H.: Pointclip: Point cloud understanding by clip. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8552–8562 (2022)
73. Zhao, L., Cai, D., Sheng, L., Xu, D.: 3dvg-transformer: Relation modeling for visual grounding on point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2928–2937 (2021)
74. Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., Zhou, Z.: Structured3d: A large photo-realistic dataset for structured 3d modeling. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16. pp. 519–535. Springer (2020)
75. Zhong, M., Chen, X., Chen, X., Zeng, G., Wang, Y.: Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2022)
76. Zhu, X., Zhang, R., He, B., Guo, Z., Zeng, Z., Qin, Z., Zhang, S., Gao, P.: Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2639–2650 (2023)
77. Zhu, Z., Ma, X., Chen, Y., Deng, Z., Huang, S., Li, Q.: 3d-vista: Pre-trained transformer for 3d vision and text alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2911–2921 (2023)
78. Zou, X., Dou, Z.Y., Yang, J., Gan, Z., Li, L., Li, C., Dai, X., Behl, H., Wang, J., Yuan, L., Peng, N., Wang, L., Lee, Y.J., Gao, J.: Generalized decoding for pixel, image, and language. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15116–15127 (June 2023)