Uni3DL: Unified Model for 3D and Language Understanding

Xiang Li[1,*], Jian Ding[1,*], Zhaoyang Chen[2†], Mohamed Elhoseiny[1]
[1] King Abdullah University of Science and Technology
[2] Ecole Polytechnique
{xiang.li.1,jian.ding,zhaoyang.chen,mohamed.elhoseiny}@kaust.edu.sa

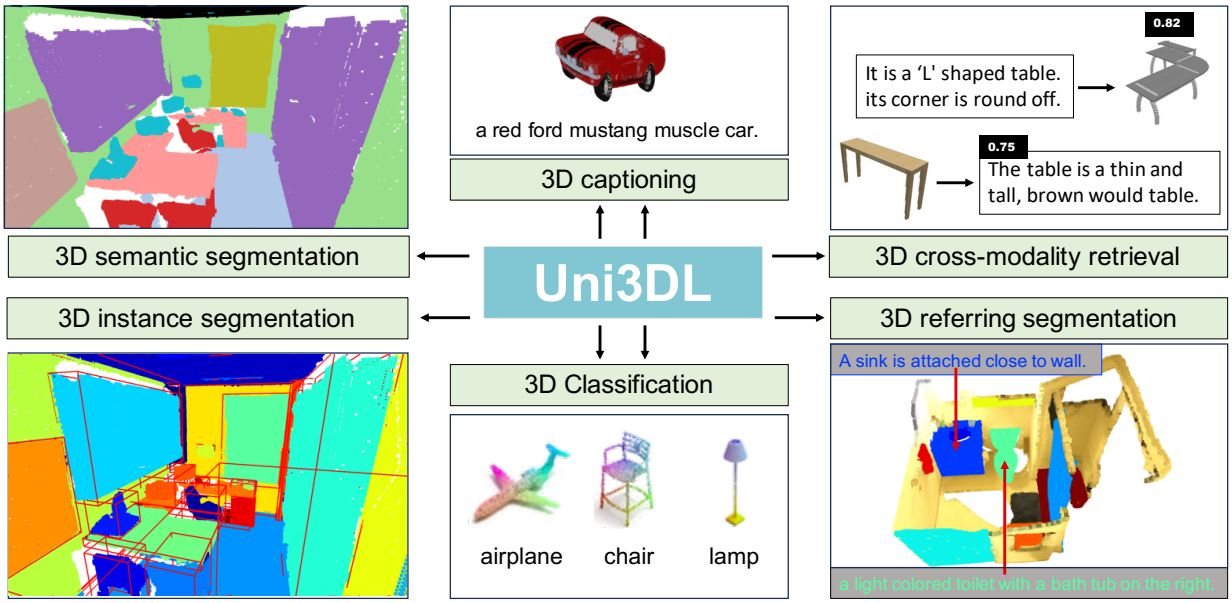


Figure 1: With a unified architecture, Uni3DL supports diverse 3D vision-language understanding tasks, including semantic segmentation, object detection, instance segmentation, grounded segmentation, captioning, text-3D cross-modality retrieval, (zero-shot) 3D object classification.

Abstract

In this work, we present Uni3DL, a unified model for 3D and Language understanding. Distinct from existing unified vision-language models in 3D which are limited in task variety and predominantly dependent on projected multi-view images, Uni3DL operates directly on point clouds. This approach significantly expands the range of supported tasks in 3D, encompassing both vision and vision-language tasks in 3D. 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 generate 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 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 and language understanding. Project page: https://uni3dl.github.io/.

1. Introduction

3D perception technology stands as a fundamental element in the automatic understanding and operation within the

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

physical world. It enhances various applications, including robotic navigation, object manipulation, autonomous driving, and virtual reality, thus improving machine-world interactions. 3D perception encompasses a broad spectrum of vision and vision-language tasks, such as 3D instance segmentation [13, 25, 27, 32, 40, 42, 57, 70, 74], semantic segmentation [33, 38, 49, 51–53, 64, 71], visual grounding [4, 7, 28, 79], object detection [34, 73], retrieval [12, 58] and captioning [45, 67], this technology has witnessed remarkable advancements.

Despite these successes, task-specific models in 3D perception often lack generalizability, constraining their effectiveness across varied tasks. In contrast, the broader scientific community, as exemplified by the grand unified theory (GUT) in physics [3, 35], 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, 37, 39, 55, 62, 75, 84]. For example, CLIP [55] employs vision-language contrastive learning for zero-shot transfer across different classification tasks. Mask2former [16, 17] leverages a transformer-based architecture for unifying generic segmentation tasks. Moreover, XDecoder [84] and Uni-Perceiver v2 [37] adopt *functionality unification modeling* [36], 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 for 2D tasks [37, 55, 75, 84] to 3D 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. Several studies, including [69, 77, 82], have explored adapting CLIP, originally pre-trained with 2D images, for 3D vision-language modeling. They achieve this by using projected multi-view images of point clouds in either training or testing phases. However, these methods have been mainly developed for classification tasks. 3D-VisTA [83] constructed large-scale 3D scene-text pairs dataset, and performed a vision-language pre-training for 3D without the need for 2D pre-trained models. However, it still requires fine-tuning for specific tasks, and the parameters of each task head are totally different.

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, most existing models necessitate the use of multi-view images rather than direct training on 3D point clouds. This approach, while performing well, often results in the loss of critical information (e.g., 3D geometry) and leads to increased model complexity and overhead (multiple projected views required).

In response to these improvement opportunities, we introduce a unified model for 3D perception that leverages both point clouds and language. Uni3DL designs a query transformer that enables latent and textural queries to softly attend to 3D visual features. The integration of these elements allows for the processing of point features, textual queries, and latent queries, and generates semantic features and mask features. Then we designed a task router with multiple *functional* heads, which takes semantic features or mask features to predict diverse outputs. By combining the outputs from these functional heads, we are able to obtain the final results for various 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 textual inputs, generating diverse outputs including masks, classes, and texts. The model can be directly applied to dense prediction tasks (e.g., instance segmentation), even without task-specific finetuning.
- 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 with shared parameters across tasks is a key goal in computer vision and machine learning. Models like CLIP [55] and ALIGN [31] 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, their use has predominantly been confined to classification, indicating room for broader application.

To broaden the scope, modeling in this domain has branched into two main categories: *I/O unification* and *functional unification* [36]. Inspired by sequence-to-sequence (seq2seq) modeling in NLP [54], I/O unification employs decoders to generate homogenous token sequences that are further processed by task-specific decoders. Promi-

Methods	MV	Pretrained FM	Sem Seg	Inst Seg	Gnd Seg	Gnd Loc	Class	Retr	Det	Capt
PointCLIP v2 [82]	✓	CLIP [55], GPT-3 [6]	✓				✓	✓	○	
UniT3D [15]	✓	BERT [22]				✓				✓
3DJCG [7]		Glove [50]				✓				✓
ULIP [68]	✓	CLIP [55]					✓	✓		
ULIP-2 [69]	✓	CLIP [55]					✓	✓		
3D-VisTA [83]		GPT-3 [6]				○				○
Point-LLM [67]		ULIP-2 [69], Vicuna [18]					✓			✓
Point-Bind [24]	✓	OpenCLIP [30]					✓	✓		
Uni3DL (Ours)			✓	✓	✓	✓	✓	✓	✓	✓

Table 1. Comparison of various vision-language models in 3D, highlighting their capabilities across a multitude of 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: Sem Seg for Semantic Segmentation, Inst Seg for Instance Segmentation, Gnd Seg for Grounded Segmentation, Gnd Loc for Grounded Localization, Class for Classification, Retr for Retrieval, Det for Detection, Capt for Captioning.

nent models such as Flamingo [1], OFA [62], and GIT [61] have primarily concentrated on image-level tasks like image captioning and visual question answering (VQA). Models like Pix2Seq v2 [14], Unitab [72], and Unified-IO have extended this approach by incorporating discrete coordinate tokens in seq2seq modeling for localization tasks, with Vision-LLM [63] and MiniGPT-2 [11] enhancing this capability using pre-trained large language models. In contrast, *functional unification* models, exemplified by X-Decoder [84] and Uni-Perceiver v2 [37], predict heterogeneous outputs and utilize various routers or headers to deliver final outputs for diverse tasks. These models typically comprise a vision encoder, a text encoder, and a general decoder. Our work aligns with the *functional unification* approach, but with a novel 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 those by PointCLIP [77], PointCLIP v2. [82], CLIP2Point [29], and ULIP [68], focus on adapting the 2D-based CLIP [55] model for 3D applications. These works contribute to enhancing classification tasks for point clouds but often *require additional components, like 3DETR [46], for tasks such as object detection*. Furthermore, rather than directly processing point clouds, they typically rely on projected multi-view images from point clouds during training or testing.

Building on these developments, Point-LLM [67], evolving from ULIP-2 [69] and incorporating the Vicuna [18] language model, engages in a dual-stage training process of feature alignment and instruction tuning. This approach equips Point-LLM with proficiency in classification, captioning, and dialogue. UniT3D [15] introduces a unified framework employing transformer technology for 3D dense captioning and visual grounding, signifying a leap in simplifying 3D vision-language tasks. Further, 3D-VisTA [83], a pre-trained transformer adept in 3D vision and text Alignment, excels in various tasks including 3D visual grounding and question answering. A key innovation of 3D-VisTA is the Scanscribe dataset, a novel contribution to 3D-VL pre-training, which uniquely operates without the need for multi-view images. However, 3D-VisTA [83] still requires fine-tuning of task-specific heads for different applications.

In conclusion, current models face notable limitations. They only support limited tasks, *and often require additional task-specific module design and fine-tuning*, as summarized in Table 1. Furthermore, they generally depend on multi-view images. Our method, 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 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

The Uni3DL is a versatile architecture tailored for diverse 3D vision-language tasks, including 3D object classification, text-to-3D retrieval, 3D captioning, 3D semantic and instance segmentation, and 3D visual grounding. This architecture encompasses four integral modules: a **Text Encoder** for textual 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 voxel features from the Point Encoder; 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.

Given an input point cloud $\mathbf{P}$, our Uni3DL leverages a 3D U-Net $\mathcal{E}_I$ to extract hierarchy voxel features $\mathbf{V}$, along with a text encoder $\mathcal{E}_T$ to obtain textural features $\mathbf{F}_T \in \mathbb{R}^{L_T \times C}$. Voxel features, textural 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.

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 [19], 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 [55] along with a transformer-based network for textural feature learning.

3.3. Query Transformer Module

We follow query-based transformer architecture [8, 43, 56, 84] 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 $\{\mathbf{V}_s\}_{s=1}^{S-1}$, 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 [17] 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 [56].

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

$$< \hat{\mathbf{F}}_Q^l, \hat{\mathbf{F}}_T^l >= \text{Self-Att}(< \hat{\mathbf{F}}_Q^l, \hat{\mathbf{F}}_T^l >), \tag{3}$$

$$< \mathbf{F}_Q^l, \mathbf{F}_T^l >= \text{FFN}(< \hat{\mathbf{F}}_Q^l, \hat{\mathbf{F}}_T^l >). \tag{4}$$

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. For example, the 3D instance segmentation task includes two heads, object classification, and mask prediction. 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.

Task	Obj Cls.	Mask	Grounding	Text Gen.	Text-3D 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.

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 matrixes.

During training, we calculate cross-entropy loss between predicted classification probabilities O_c and ground truth

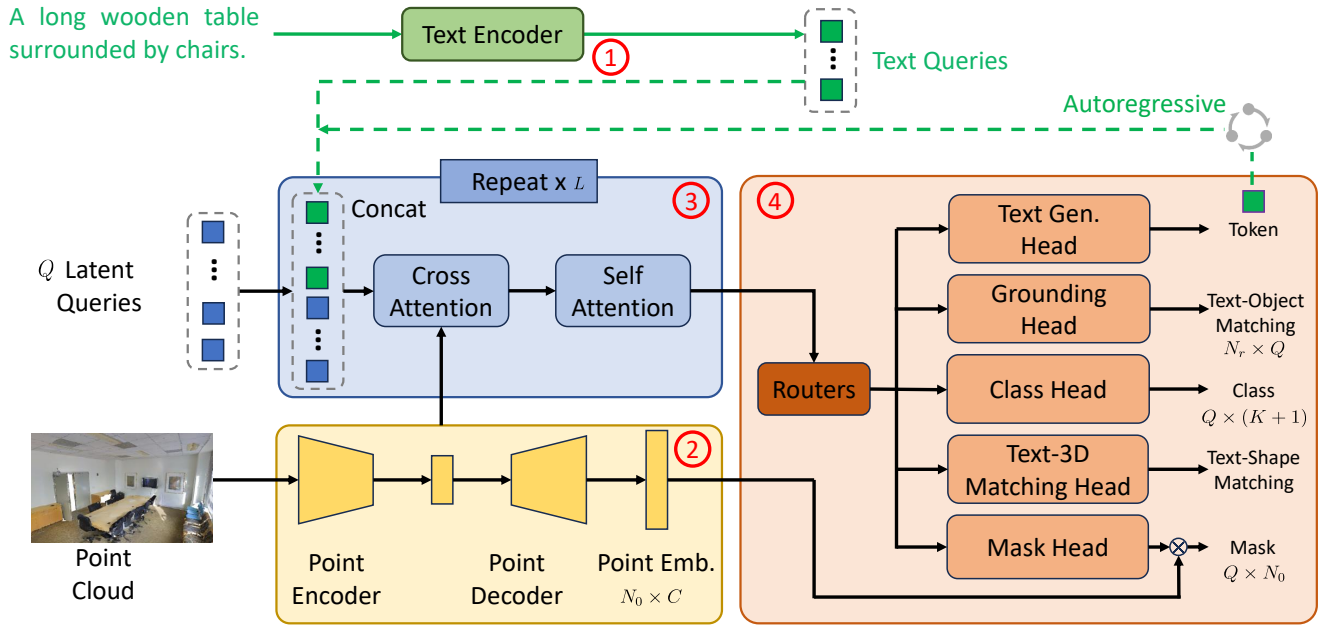


Figure 2. Overview of the Uni3DL Model. The Uni3DL is engineered for multifaceted 3D data tasks, including classification, retrieval, captioning, semantic and instance segmentation, as well as visual grounding. The architecture is composed of four principal modules: ① a **Text Encoder** for textual feature 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.

(GT) class labels C_{gt} to formulate classification loss as:

$$\mathcal{L}_{cls} = \lambda_{cls}\text{CE}(\mathbf{O}_c, C_{gt}), \tag{5}$$

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 a mask for the corresponding latent query.

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

$$\mathcal{L}_{mask} = \lambda_{bce}\text{BCE}(\mathbf{O}_m, \mathbf{M}_{gt}) + \lambda_{dice}\text{DICE}(\mathbf{O}_m, \mathbf{M}_{gt}), \tag{6}$$

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 textural embeddings $\mathbf{T}_{emb} \in \mathbb{R}^{N_r\times C}$ by feeding all referring 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

$$\mathbf{S}_t = \text{Softmax}(e^{\eta}\ \mathbf{T}_{emb} \cdot (\mathbf{O}^s)^T), \tag{7}$$

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 [8], 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:

$$\mathcal{L}_{gc} = \text{CE}(\mathbf{S}_t, T_{gt}). \tag{8}$$

Following the common practice of 3D visual grounding practice [7, 9], we design a lightweight classification that takes textural embeddings as inputs and predicts the existence of all K candidate object categories. Given input textural embeddings $\mathbf{T}_{emb} \in \mathbb{R}^{N_r\times C}$, we use a single-layer MLP network to calculate the probabilities matrix $\mathbf{T}_{cls} \in \mathbb{R}^{N_r\times K}$ over K candidate object categories and calculate multi-label classification loss as:

$$\mathcal{L}_{gtxt} = \text{BCE}(\mathbf{T}_{cls}, \mathbf{T}_{cls}^{gt}), \tag{9}$$

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 calcu-

lated as:

$$\mathcal{L}_{grd} = \lambda_{gc}\mathcal{L}_{gc} + \mathcal{L}_{gtxt} + \mathcal{L}_{gmask}. \quad (10)$$

Text Generation Head. In the context of 3D captioning, our method begins by generating textural 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:

$$\mathcal{L}_{cap} = \lambda_{cap}\text{CE}(\mathbf{S}_{cap}, y_{cap}), \quad (11)$$

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. Our Uni3DL uses decoupled point and text encoder networks. 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 retrieval 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:

$$\mathcal{L}_{ret} = \lambda_{ret}\text{CL}(\mathbf{S}_{ret}, y_{ret}), \quad (12)$$

where $y_{cap} \in \mathbb{R}^{1\times B}$ denotes the ground truth matching indices. CL denotes contrastive loss defined in CLIP [55].

Multi-Task Training. During pretraining, we simultaneously train the whole network with both object classification head, mask head, grounding head, text generation head, and text-3D matching head. The overall loss is formulated as:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{mask} + \mathcal{L}_{grd} + \mathcal{L}_{cap} + \mathcal{L}_{ret}. \quad (13)$$

4. Experiments

4.1. Dataset

We pretrain our Uni3DL on three datasets, including ScanNet (v2) [20] for instance segmentation, ScanRefer [9] for visual grounding, and Cap3D Objaverse [45] dataset for 3D captioning and text-3D cross-modal retrieval.

ScanNet (v2) [20] captures RGB-D videos with 2.5 million views from more than 1,500 3D scans. Following the official benchmark, we use 1,201 scenes for training, 312 for validation, and 100 indoor scenes for online evaluation. There are in total 20 semantic labels, 18 of which are instance classes.

ScanRefer [9] 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 [45] dataset, is derived from Objaverse, one of the largest 3D datasets 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[1].

For model evaluation, other than ScanNet (v2), ScanRefer, Cap3D, we use additional S3DIS [2] to evaluate both semantic and instance segmentation, Text2Shape [12] to evaluate text-to-3D 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 [12] 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 [12].

4.2. Implementation Details

In this work, we employ 150 latent queries and an additional latent query for scene-level tasks. The point encoder-decoder network is based on Minkowski Res16UNet34C [19] and pretraiend from Mask3D [56], 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 to λ_{gc} to 0.4, captioning and retrieval weight $\lambda_{\text{cap}}, \lambda_{\text{ret}}$ are set to 2.0.

During pretraining, the voxel size is set to 0.02m for 3D scans (e.g., ScanNet (v2)) and 0.01 for normalized 3D shapes (e.g., Cap3D Objaverse), with a batch size of 8 for 3D scans and 12 for 3D-text pairs. Input scenes are augmented by random flips along the X and Y axes, and rotations along the X, Y, and Z axes. Color augmentations, including jittering, brightness, and contrast adjustments, are also applied. The training process spans 50 epochs using the AdamW optimizer [44], taking approximately 20 hours on four NVIDIA A100 GPUs. Details about pertaining and task-specific finetuning can be found in Appendix A.

During inference, the top 200 (for S3DIS) and 500 (for ScanNet (v2)) instances with the highest classification scores are retained for the instance segmentation task.

4.3. 3D Semantic/Instance Sementation

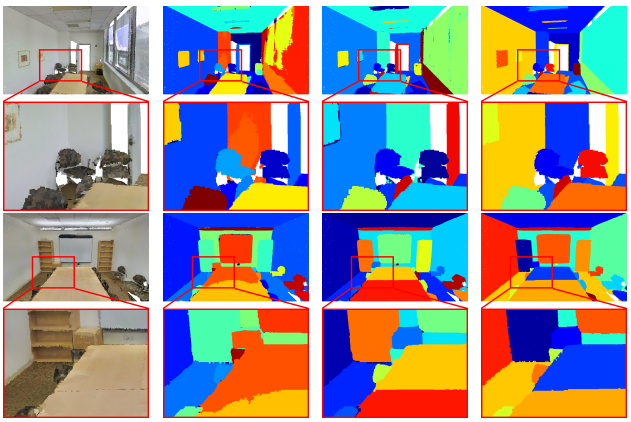
We compare 3D semantic segmentation, object detection, and instance segmentation performance with previous STOA methods in Table 3. From the table, our Uni3DL method achieves comparable or superior performance than previous STOA methods on semantic/instance segmentation on S3DIS and ScanNet (v2) datasets. Specifically, our

[1]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.

Method	Semantic Segmentation			Object Detection		Instance Segmentation				Grounded Segmentation			3D Captioning			3D Retrieval	
	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[19]	67.1	74.4	72.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FastPointTransformer[49]	68.5	76.5	72.1	-	-	-	-	-	-	-	-	-	-	-	-	-	
PointNeXt-XL[53]	71.1	77.2	71.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
StratifiedTransformer [33]	72.0	78.1	73.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PointTransformerV2 [64]	71.6	77.9	74.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EQ-Net[73]	71.3	*	75.3	-	-	-	-	-	-	-	-	-		-	-	-	-
Swin3D[71]	72.5	*	75.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Swin3D†[71]	**73.0**	*	75.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
VoteNet [66]		-	-	33.5	58.6	-	-	-	-	-	-	-	-	-	-	-	-
3DETR [47]	-	-	-	47.0	65.0	-	-	-	-	-	-	-	-	-	-	-	-
CAGroup3D [60]	-	-	-	61.3	75.1	-	-	-	-	-	-	-	-	-	-	-	-
PointGroup[32]	*	*	*	*	*	34.8	56.7	57.8	*	-	-	-	-	-	-	-	-
MaskGroup[81]	*	*	*	*	*	42.0	63.3	65.0	*	-	-	-	-	-	-	-	-
SSTNet[40]	*	*	*	*	*	49.4	64.3	59.3	*	-	-	-	-	-	-	-	-
SoftGroup[59]	*	*	*	59.4	71.6	50.4	76.1	66.1	*	-	-	-	-	-	-	-	-
Mask3D[56]	*	*	*	56.2	70.2	55.2	73.7	68.4	75.2	-	-	-	-	-	-	-	-
Mask-Att-Free†[34]	*	*	*	63.9	73.5	58.4	75.9	**69.1**	**75.7**	-	-	-	-	-	-	-	-
TGNN (GRU) [28]	-	-	-	-	-	-	-	-	-	26.1	35.0	29.0	-	-	-	-	-
TGNN (BERT) [28]	-	-	-	-	-	-	-	-	-	27.8	37.5	31.4	-	-	-	-	-
InstructBLIP-7B [21]	-	-	-	-	-	-	-	-	-	-	-	-	11.2	13.9	14.9	*	*
InstructBLIP-13B [21]	-	-	-	-	-	-	-	-	-	-	-	-	12.6	15.0	**16.0**	*	*
PointLLM-7B [67]	-	-	-	-	-	-	-	-	-	-	-	-	8.0	11.1	15.2	*	*
PointLLM-13B [67]	-	-	-	-	-	-	-	-	-	-	-	-	9.7	12.8	15.3	*	*
FTST [12]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.2	1.6
FMM [12]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.2	2.4
Y2S [26]	-	-	-	-	-	-	-	-	-	-	-	-	*	*	*	2.9	9.2
Parts2Words (no parts) [58]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5.1	17.2
Parts2Words† [58]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**12.7**	33.0
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**	14.4	**5.8**	**19.7**

Table 3. Performance of our Uni3DL on different segmentation and VL tasks. '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 underline denote the best and second-best outcomes, respectively, for each column. Note that Swin3D† uses extra training data (Structure3D [80]), and Parts2Words† [58] uses additional part labels. Mask-Att-Free† uses several task-specific designs, including a center regression module and position-aware components, to improve the performance.

method archives the best performance on ScanNet (v2) semantic segmentation, with a mIoU of 76.2. Figure 7 shows qualitative results of our method. Additional qualitative results are presented in the Appendix C.2.



Input Baseline Inst. Ours Inst. GT Inst.

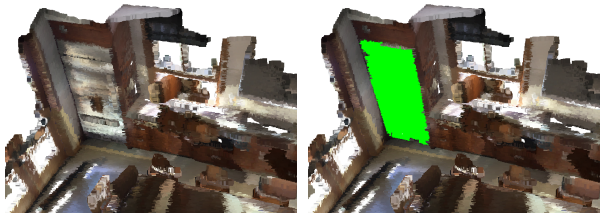
Figure 3. Instance (Inst.) segmentation results on S3DIS dataset. We show results of the baseline method trained from scratch and our finetuned model.

4.4. 3D Visual Grounding

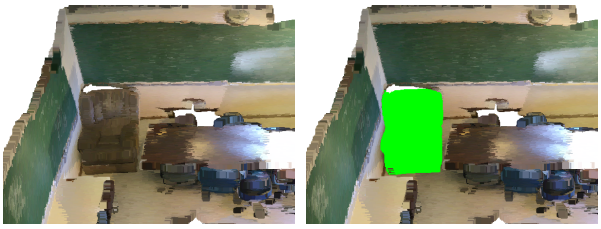
We compare the 3D grounded segmentation performance of our Uni3DL with previous STOA methods TGNN (GRU) [28] and TGNN (BERT) [28] in Table 3. Our method achieves significantly better performance than previous SOTA methods as indicated by instance-average IoU, and accuracy at the IoU thresholds of 0.25 and 0.5. Figure 4 shows qualitative results of our method. More qualitative results are presented in Appendix C.3. Grounded localization performance can be found in Appendix B.2.

4.5. 3D Captioning

From Table 3, our Uni3DL model outperforms existing methods in 3D captioning on the Cap3D Objaverse dataset, as evidenced by its superior BLEU-1 (B-1) [48], ROUGE-L(R) [41], and METEOR (M) [5] scores. Specifically, on the BLEU-1 and ROUGE-L scores, our method beats precious STOA methods by a large margin (more than 20%). Qualitative analyses, illustrated in Figure 6, demonstrate our caption predictions closely align with the ground truth. Additional qualitative results are presented in the Appendix C.1. We use this finetuned model to evaluate zero-shot 3D classification performance on ModelNet40/10 dataset and results are provided in B.1.



Refer: It is a stainless steel refrigerator. the refrigerator sits to the left of the windows above the counter.

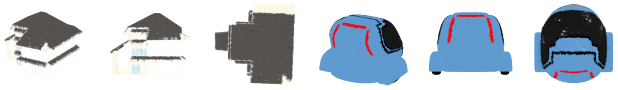


Refer: This is the brown easy chair at the back of the room where the two chalk boards meet. it is a brown easy chair.

Figure 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

Figure 5. 3D captioning results on Cap3D Objaverse dataset.

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 comparable text-to-3D retrieval performance with STOA task-specific methods, including FTST [12], FMM [12], Y2S [26], and Parts2Words [58], as indicated by recall scores R@1 and R@5. For the Parts2Words method, we primarily compare its performance without using part information for a fair comparison. Qualitative results are provided in Appendix C.4.

4.7. Ablation Study

Effect of Pretraining. We assess 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 S3DIS instance segmentation in Figure 7.

Task	Sem Seg SN Val mIoU/mAcc	Inst Seg S3DIS (Area 5) mAP_{50} / mAP_{25}	Gnd Seg ScanRefer Acc@0.25/Acc@0.5	Ret Text2Shape 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**	**4.6/18.0**

Table 4. Ablation of pertaining.

Task	Gnd Seg ScanRefer Acc@0.25/Acc@0.5	Captioning Cap3D B-1/R	Retrieval Cap3D T2S R@1/S2T R@1
Ours	37.8/34.2	15.4/**18.6**	**5.5/8.0**
- Inst Seg	33.8/31.3	**20.9**/17.8	3.0/4.0
- Retrieval	37.7/34.5	19.6/15.8	n/a
- Captioning	**37.9/35.9**	n/a	5.0/3.5

Table 5. Ablation of pertaining tasks. T2S for Text-to-Shape retrieval, and S2T for Shape-to-Text retrieval.

Effect of 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 33.8%. 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-3D cross-modal retrieval. 2) Caption and retrieval benefit each other. Without pertaining on the captioning task, the text-3D cross-modal retrieval accuracy drops from 8.0% 5o 3.5% in terms of shape-to-text R@1 on the Cap3D retrieval task. Without pertaining on the retrieval task, the captioning performance drops from 18.6% to 15.8% in terms of ROUGE scores on the Cap3D captioning task.

5. Conclusion

In this work, we introduce a unified model named Uni3DL for generalized 3D vision and language understanding tasks. We design a query transformer module to attentively align 3D features with latent and textural queries. A task router module with multiple functional heads is faithfully designed to support diverse vision-language tasks, including 3D object classification, 3D semantic/instance segmen-

tation, 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 STOA method.

6. Acknowledgement

We extend our sincere gratitude to Xueyan Zou from the University of Wisconsin-Madison for the helpful and insightful discussions that contributed to our work.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2, 3

[2] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 6

[3] John Baez and John Huerta. The algebra of grand unified theories. *Bulletin of the American Mathematical Society*, 47 (3):483–552, 2010. 2

[4] Eslam Bakr, Yasmeen Alsaedy, and Mohamed Elhoseiny. Look around and refer: 2d synthetic semantics knowledge distillation for 3d visual grounding. *Advances in Neural Information Processing Systems*, 35:37146–37158, 2022. 2

[5] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 7

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[7] Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 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*, pages 16464–16473, 2022. 2, 3, 5, 14

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 4, 5

[9] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020. 5, 6, 14

[10] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X Chang. D 3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding. In *European Conference on Computer Vision*, pages 487–505. Springer, 2022. 14

[11] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023. 3

[12] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. 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*, pages 100–116. Springer, 2019. 2, 6, 7, 8

[13] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15447–15456, 2021. 2

[14] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey E Hinton. A unified sequence interface for vision tasks. *Advances in Neural Information Processing Systems*, 35:31333–31346, 2022. 3

[15] Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18109–18119, 2023. 3, 14

[16] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 2

[17] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 2, 4

[18] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023. 3

[19] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019. 4, 6, 7

[20] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 6

[21] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-

purpose vision-language models with instruction tuning, 2023. 7

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[23] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59:167–181, 2004. 13

[24] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, 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. 3

[25] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2940–2949, 2020. 2

[26] Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. 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*, pages 126–133, 2019. 7, 8

[27] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4421–4430, 2019. 2

[28] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1610–1618, 2021. 2, 7, 13, 14

[29] Tianyu Huang, Bowen Dong, Yunhan Yang, Xiaoshui Huang, Rynson WH Lau, Wanli Ouyang, and Wangmeng Zuo. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22157–22167, 2023. 3, 14

[30] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 3

[31] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 2

[32] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020. 2, 7

[33] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. 2, 7

[34] Xin Lai, Yuhui Yuan, Ruihang Chu, Yukang Chen, Han Hu, and Jiaya Jia. Mask-attention-free transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3693–3703, 2023. 2, 7

[35] Paul Langacker. Grand unified theories and proton decay. *Physics Reports*, 72(4):185–385, 1981. 2

[36] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. Multimodal foundation models: From specialists to general-purpose assistants. *arXiv preprint arXiv:2309.10020*, 1, 2023. 2

[37] Hao Li, Jinguo Zhu, Xiaohu Jiang, Xizhou Zhu, Hongsheng Li, Chun Yuan, Xiaohua Wang, Yu Qiao, Xiaogang Wang, Wenhai Wang, 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*, pages 2691–2700, 2023. 2, 3

[38] Xiang Li, Lingjing Wang, Mingyang Wang, Congcong Wen, and Yi Fang. Dance-net: Density-aware convolution networks with context encoding for airborne lidar point cloud classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:128–139, 2020. 2

[39] Xiang Li, Congcong Wen, Yuan Hu, and Nan Zhou. Rs-clip: Zero shot remote sensing scene classification via contrastive vision-language supervision. *International Journal of Applied Earth Observation and Geoinformation*, 124:103497, 2023. 2

[40] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2783–2792, 2021. 2, 7

[41] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 7

[42] Shih-Hung Liu, Shang-Yi Yu, Shao-Chi Wu, Hwann-Tzong Chen, and Tyng-Luh Liu. Learning gaussian instance segmentation in point clouds. *arXiv preprint arXiv:2007.09860*, 2020. 2

[43] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 4

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 6

[45] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023. 2, 6

[46] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceed-*

ings of the IEEE/CVF International Conference on Computer Vision, pages 2906–2917, 2021. 3

[47] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 7

[48] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 7

[49] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16949–16958, 2022. 2, 7

[50] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 3

[51] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[52] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[53] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022. 2, 7

[54] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 2

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2, 3, 4, 6

[56] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023. 4, 6, 7

[57] Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3d scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2393–2401, 2023. 2

[58] Chuan Tang, Xi Yang, Bojian Wu, Zhizhong Han, and Yi Chang. 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*, pages 6884–6893, 2023. 2, 7, 8

[59] Thang Vu, Kookhoi Kim, Tung M. Luu, Thanh Nguyen, and Chang D. Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2708–2717, 2022. 7

[60] Haiyang Wang, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, Liwei Wang, et al. Cagroup3d: Class-aware grouping for 3d object detection on point clouds. *Advances in Neural Information Processing Systems*, 35: 29975–29988, 2022. 7

[61] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. 3

[62] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 2, 3

[63] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023. 3

[64] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 2, 7

[65] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 13

[66] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Dening Lu, Mingqiang Wei, and Jun Wang. Venet: Voting enhancement network for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3712–3721, 2021. 7

[67] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023. 2, 3, 7

[68] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. 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*, pages 1179–1189, 2023. 3, 14

[69] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. *arXiv preprint arXiv:2305.08275*, 2023. 2, 3

[70] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point

clouds. *Advances in neural information processing systems*, 32, 2019. 2

[71] Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint arXiv:2304.06906*, 2023. 2, 7

[72] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *European Conference on Computer Vision*, pages 521–539. Springer, 2022. 3

[73] Zetong Yang, Li Jiang, Yanan Sun, Bernt Schiele, and Jiaya Jia. A unified query-based paradigm for point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8541–8551, 2022. 2, 7

[74] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. 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*, pages 3947–3956, 2019. 2

[75] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2

[76] Zhihao Yuan, Xu Yan, Yinghong Liao, Ruimao Zhang, Sheng Wang, Zhen Li, and Shuguang Cui. Instancerefer: Cooperative holistic understanding for visual grounding on point clouds through instance multi-level contextual referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1791–1800, 2021. 14

[77] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022. 2, 3, 14

[78] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15225–15236, 2023. 14

[79] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2928–2937, 2021. 2, 14

[80] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. 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*, pages 519–535. Springer, 2020. 7

[81] Min Zhong, Xinghao Chen, Xiaokang Chen, Gang Zeng, and Yunhe Wang. Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 7

[82] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2639–2650, 2023. 2, 3, 14

[83] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2921, 2023. 2, 3

[84] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee, and Jianfeng Gao. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15116–15127, 2023. 2, 3, 4

Uni3DL: Unified Model for 3D and Language Understanding

Supplementary Material

A. Experimental Settings

A.1. Pretraining

As described in the main paper, we use the ScanNet (v2), ScanRefer, and Cap3D Objaverse datasets for joint pretraining. For the Cap3D Objaverse caption dataset, we only include objects whose captions contain any object name from the ScanNet, S3DIS, or ModelNet categories. The Uni3DL model is pretrained for 50 epochs. We set the initial learning rate to 1e-4 and reduce it by 0.1 after 50% and 70% of the total training steps. A linear warmup is applied for the first 10 iterations.

A.2. Finetuning

Finetuning for 3D semantic/instance Segmentation. For the S3DIS dataset, we randomly crop $5m \times 5m \times 5m$ blocks from each scene, ensuring a minimum of 25,000 points per scene. The Uni3DL model is finetuned for 25 epochs with an initial learning rate of 2e-5, which is multiplied by 0.1 after 50% and 70% of the total training steps. For ScanNet Segmentation, we finetune our Uni3DL model for 30 epochs on ScanNet semantic/instance segmentation with the same learning rate strategy as in S3DIS segmentation.

Current state-of-the-art instance segmentation methods, including Mask3D and Mask-Att-Free, use additional segment labels obtained from an unsupervised graph-based segmentation method [23] during training and evaluation. To ensure a fair comparison, we report the performance of our Uni3DL model using segment information.

Finetuning for Grounded Segmentation. For 20 epochs, we finetune the Uni3DL model on Grounded Segmentation with an initial learning rate of 1e-5, decaying it by 0.1 after reaching 50% and 70% of the total training steps.

Finetuning for 3D Captioning. The Uni3DL model is finetuned for 30 epochs on the Cap3D Objaverse dataset. The learning rate starts at 1e-4 and is reduced by 0.2 after 50% and 70% of the training steps.

Finetuning for Text-3D Cross-Modal Retrieval. We finetune the Uni3DL model for 30 epochs on the Text2Shape retrieval task, following a similar learning rate strategy as in 3D Captioning. For data augmentation, we applied random scaling to the training shapes, using a scale factor uniformly sampled from the range [0.8, 1.2]. Additionally, we randomly rotated the shapes along the z-axis, selecting rotation angles within the range $[-\pi/2, \pi/2]$.

B. More quantative results

B.1. Zero-Shot 3D Classification

We use our Uni3DL model fine-tuned on the Cap3D Objaverse dataset to evaluate zero-shot 3D classification performance on ModelNet40 and ModelNet10 datasets. ModelNet40 includes 40 different categories with 12, 311 CAD models, while ModelNet10, a smaller subset, consists of 10 categories with 4, 899 models. We use the same validation set as [65] for performance evaluation.

Table 6 summarizes the performance on ModelNet10 and ModelNet40 test datasets. From this Table, we can see that our method achieves a competitive performance on both datasets, with a classification accuracy of 70.4% on ModelNet10 and 57.0% on ModelNet40. Specifically, our Uni3DL trained achieves the best top-5 classification accuracy. *It should be noted that all compared methods rely on projecting 3D data to multiview 2D images and use a pretrained CLIP for image-text alignment; while our method does not require view projection.*

B.2. Grounded Localization

In the main paper, we report the performance of our Uni3DL model for grounded *segmentation*. Previous methods have also explored the grounded *localization* task. To produce grounded object location, we directly use grounded object masks to calculate their bounding boxes. Table 7 summarizes the performance of Uni3DL and previous state-of-the-art methods for grounded localization. It should be noted that all compared methods except TGNN [28] employ a dual-stage process, where a 3D object detector identifies potential bounding box candidates, followed by a disambiguation module employed to fuse visual and textural features and determine the precise target bounding box. *In contrast, our Uni3DL model is a single-stage model, without using second-stage object-text fusion modules.* Specifically, our Uni3DL model achieves better performance than another single-stage model TGNN [28] which also generates bounding boxes from object segmentation masks.

C. More qualitative results

C.1. 3D Captioning

We show more qualitative results of 3D captioning on the Cap3D objaverse dataset in Figure 6. As shown in the figure, our Uni3DL can generate text descriptions well aligned with ground truth captions.

Method	Input	Pretraing dataset	Pretrained FM	ModelNet10	ModelNet40	
				top-1	top-1	top-5
PointCLIP[77]	MV Images	ShapeNet	Yes (CLIP)	30.2	23.8	-
CLIP2Point[29]	MV Images	ShapeNet	Yes (CLIP)	66.6	49.4	-
PointCLIP V2[82]	MV Images	ShapeNet	Yes (CLIP+GPT3)	**73.1**	<u>64.2</u>	-
ULIP [68]	MV Images	ShapeNet	Yes (CLIP)	-	60.4	<u>84.0</u>
ULIP [68]	MV Images	Cap3D Objaverse	Yes (CLIP)	-	**67.2**	83.1
Ours	Point Cloud	Cap3D Objaverse	No	<u>70.4</u>	57.0	**88.8**

Table 6. Zero-shot 3D shape classification performance on ModelNet10 and ModelNet40 datasets. We show input types, pretrained datasets, and foundation model (FM) requirements for detailed comparison. Our method does not require projected multiview images as inputs and does not require pretrained foundation models. The results highlighted in **bold** and <u>underline</u> denote the best and second-best performance, respectively.

Model	Single Stage	Detector	Overall	
			Acc@0.25	Acc@0.5
ScanRefer [9]	✗	VoteNet	39.0	26.1
InstanceRefer [76]	✗	PointGroup	38.2	31.4
3DVG-Transformer [79]	✗	VoteNet	45.9	34.5
3DJCG [7]	✗	VoteNet	**47.6**	36.1
D3Net [10]	✗	PointGroup	-	35.6
UniT3D [15]	✗	PointGroup	-	36.5
M3DRef [78]	✗	PointGroup	-	**40.4**
TGNN [28]	✓	N/A	37.4	29.7
Uni3DL (Ours)	✓	N/A	**37.8**	**33.7**

Table 7. Comparative analysis of grounded localization performance on the ScanRefer [9] dataset. We report the ratios of correctly predicted bounding boxes with IoU thresholds of 0.25 and 0.5. We report the performance of all comparing methods with only 3D point clouds as inputs.

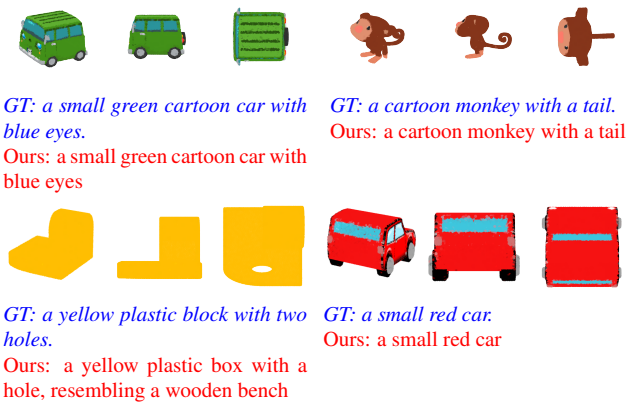


Figure 6. 3D captioning results on Cap3D Objaverse dataset.

C.2. 3D Segmentation

We show more instance segmentation results on both S3DIS and ScanNet validation set in Figure 7. From the figure, we can see that our Uni3DL model produces satisfying results for both semantic and instance segmentation tasks.

C.3. Grounded Segmentation

Figure 8 presents additional grounded segmentation results obtained using the ScanRefer dataset. As illustrated, our Uni3DL model accurately predicts the grounded masks corresponding to each referring sentence.

C.4. Text-3D cross-modal retrieval

We show text-to-3D and 3D-to-text retrieval results in Figure 9 and Figure 10 respectively. From the two figures, our Uni3DL model learns satisfying text-3D feature alignments and produces satisfying cross-modal retrieval results.

D. Limitation and Future Work

In this study, we introduced Uni3DL, a novel unified model for understanding both 3D structures and language, operating directly on raw point clouds. This approach marks a departure from conventional 3D vision-language models that predominantly rely on projected multi-view images. While these projection-based methods are limited by their handling of geometric information, their integration with powerful 2D pretrained foundation models, such as CLIP, has yielded promising results.

To leverage the benefits of both point-based and

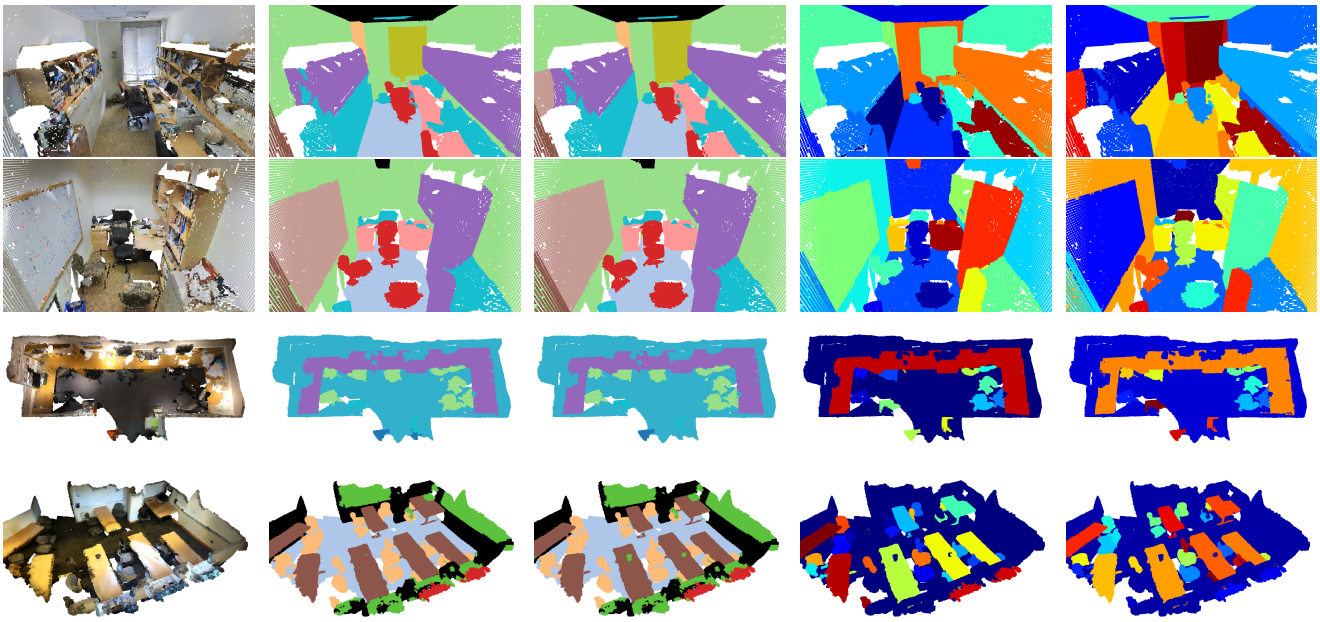


Figure 7. 3D Segmentation results on S3DIS (top) and ScanNet (bottom) datasets.

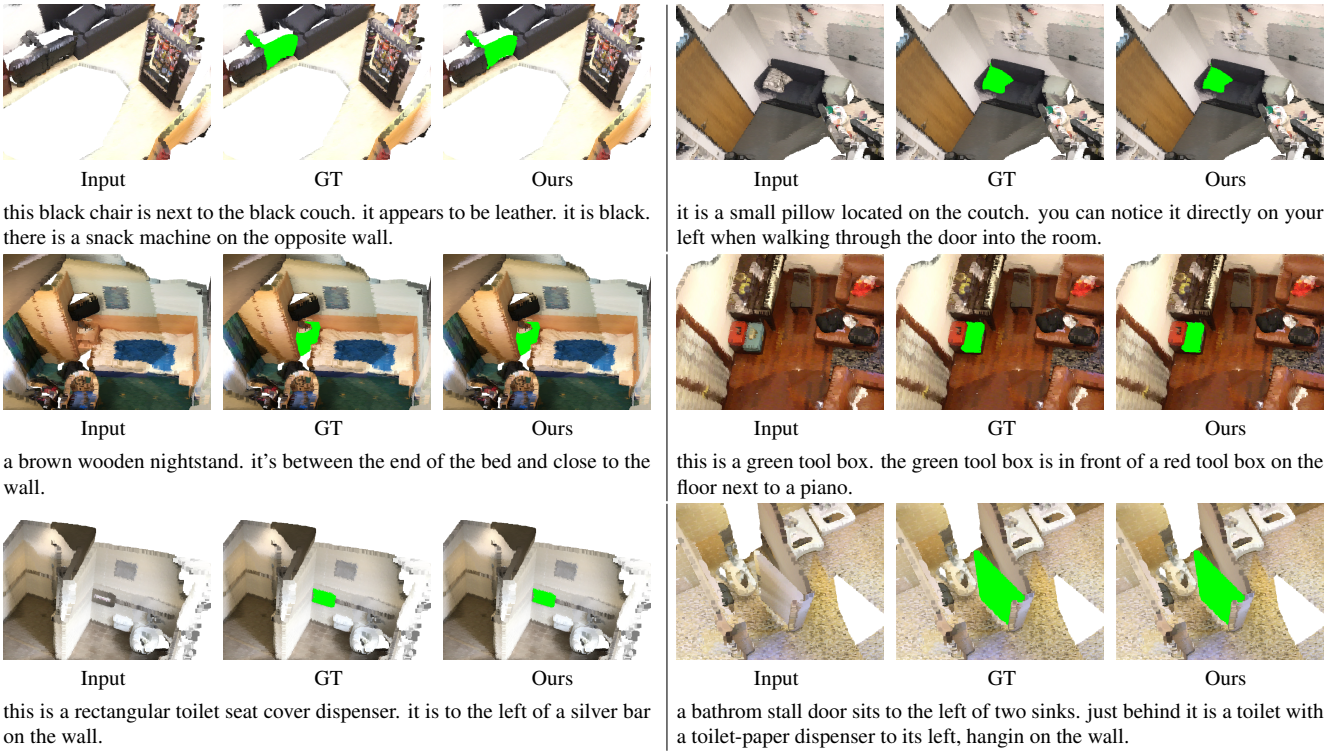


Figure 8. Results of grounded segmentation on ScanRefer dataset.

projection-based techniques, our future work will focus on a hybrid approach. This strategy aims to concurrently learn joint 2D and 3D features, integrating insights from 2D foundation models. This advancement is expected to signifi-

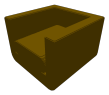
	top1	top2	top3	top4	top5
a round table with differnt type of look and is good	 **0.91(GT)**	 0.90	 0.90	 0.90	 0.88
it is an oblong table with distressed wooden top and six spindle shaped legs.	 **0.91(GT)**	 0.86	 0.86	 0.83	 0.83
a red sofa that is sitting on a black carpet. the sofa is round and ovalular.	 0.90	 **0.90(GT)**	 0.87	 0.86	 0.82
a unique design brown wooden table with white color at top is great for outdoor	 0.94	 0.87	 **0.83(GT)**	 0.82	 0.81

Figure 9. Text-to-Shape Retrieval results on Text2Shape dataset, For each query sentence, we show the top-5 ranked shape, the scores of ground truth shape are marked in red.

cantly enhance the sophistication and accuracy of 3D language understanding in upcoming versions of our model.

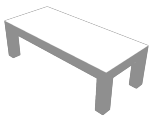
Query Shape	Retrieval Results
	**1. it is an oblong table with distressed wooden top and six spindle shaped legs. (Prob: 0.91, GT)** 2. elliptical table with brown wooden top and grey straight legs (Prob: 0.88) 3. a brown oblong wooden topped table with four grey supporting legs (Prob: 0.86) 4. oval table with shape oval , 4 legs and high qualit wood from alaska that will make you happy (Prob: 0.86) 5. this is a dining table that is oval with the insert, but could collapse down to a circle table. it has 4 legs. (Prob: 0.85)
	1. a grey rectangular shaped wooden table with four short legs. (Prob: 0.91) **2. grey colored, wooden table. four short solid legs with rectangular top. (Prob: 0.89, GT)** 3. a grey rectangular short table with four short grey legs. (Prob: 0.89) 4. a white colored rectangular table which has rectangular top painted in white and has four short legs colored in black. (Prob: 0.89) **5. a low and long grey table with four legs. (Prob: 0.89, GT)**
	**1. a white conference table with legs (Prob: 0.92, GT)** 2. a table with a white colored oval type top and four grey colored plate type legs (Prob: 0.88) 3. simple white table. lunch room table. 4 legs. metal legs. formica top. wide. (Prob: 0.87) 4. an ash colored oval shaped steel coffee table which has skinny rectangular shaped long four legs. (Prob: 0.87) 5. outdoor table, wooden, gray, oval shape, with four legs. (Prob: 0.86)
	**1. red colour plastic chair with u shape iron legs and chair was looking variety (Prob: 0.90, GT)** 2. a red chair with curved back legs. probably made of plastic. (Prob: 0.89) 3. a basket backed, red seated high bar stool with thin metal legs (Prob: 0.88) **4. red high back chair made of plastic. four legs are made of metal. (Prob: 0.88, GT)** **5. this is a red molded chair with back and no arms. the chair has 4 metal/plastic legs. (Prob: 0.87, GT)**

Figure 10. Shape-to-Text Retrieval results on Text2Shape dataset, For each query shape, we show the top-5 ranked sentences, the ground truth sentences are marked in red.