PointCLIP V2: Prompting CLIP and GPT for Powerful 3D Open-world Learning

Xiangyang Zhu*[1], Renrui Zhang*†‡[2,3], Bowei He[1], Ziyu Guo[2,3], Ziyao Zeng[5], Zipeng Qin[2]
Shanghang Zhang[4], Peng Gao[3]

* Equal contribution † Project leader ‡ Corresponding author

[1]City University of Hong Kong [2]The Chinese University of Hong Kong
[3]Shanghai Artificial Intelligence Laboratory [4]Peking University [5]Yale University

{xiangyzhu6-c, boweihe2-c}@my.cityu.edu.hk,
{zhangrenrui, gaopeng}@pjlab.org.cn, shanghang@pku.edu.cn

Abstract

*Large-scale pre-trained models have shown promising open-world performance for both vision and language tasks. However, their transferred capacity on 3D point clouds is still limited and only constrained to the classification task. In this paper, we first collaborate **CLIP** and **GPT** to be a unified 3D open-world learner, named as **PointCLIP V2**, which fully unleashes their potential for zero-shot 3D classification, segmentation, and detection. To better align 3D data with the pre-trained language knowledge, PointCLIP V2 contains two key designs. For the visual end, we prompt CLIP via a shape projection module to generate more realistic depth maps, narrowing the domain gap between projected point clouds with natural images. For the textual end, we prompt the GPT model to generate 3D-specific text as the input of CLIP's textual encoder. Without any training in 3D domains, our approach significantly surpasses PointCLIP by **+42.90%**, **+40.44%**, and **+28.75%** accuracy on three datasets for zero-shot 3D classification. On top of that, V2 can be extended to few-shot 3D classification, zero-shot 3D part segmentation, and 3D object detection in a simple manner, demonstrating our generalization ability for unified 3D open-world learning. Code is available at https://github.com/yangyangyang127/PointCLIP_V2.*

1. Introduction

The advancement of spatial sensors has stimulated widespread attention in recent years for both academia and industry. To effectively understand point clouds, the major data form in 3D, many related tasks are put forward and gained great progress, including 3D classification [35, 51, 66], segmentation [36, 54, 48, 52], detection [55, 29], and self-supervised learning [65, 17, 61, 11]. Importantly, for the complexity and diversity of open-world circumstances, the collected 3D data normally contains a large number of 'unseen' objects, namely, not ever defined and trained by the already deployed 3D systems. Given the human-laboring data annotations, how to recognize such 3D shapes of new categories has become a hot-spot issue, which still remains to be fully explored.

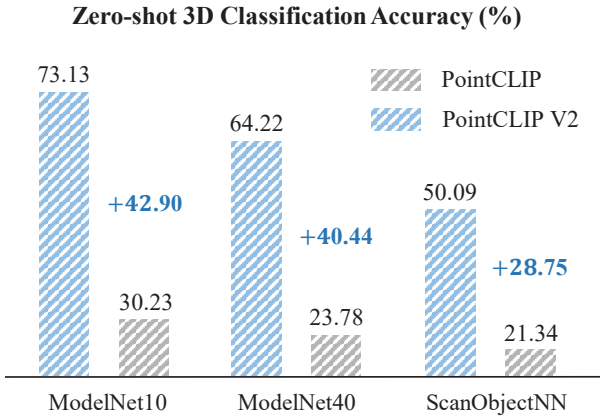

Figure 1. **Zero-shot Performance of PointCLIP V2.** On different 3D datasets, our approach achieves significant accuracy enhancement for zero-shot 3D classification over PointCLIP [62].

Recently, large-scale pre-trained vision and language models, *e.g.*, CLIP [37] and GPT-3 [3], have obtained a strong capacity to process data in both modalities. However, limited efforts have focused on their application in the point cloud, and existing work only explores the possibility of CLIP on the 3D classification task, without con-

Figure 2. **Comparison of Visual Projection.** PointCLIP V2 (Bottom) generates more realistic depth maps with denser point distribution and smoother depth values.

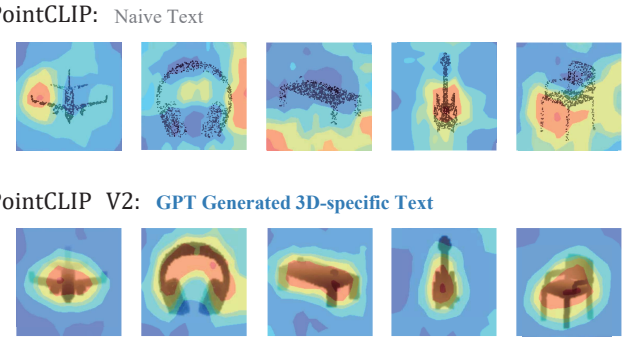

Figure 3. **Comparison of Textual Input.** We visualize the similarity score maps of the encoded textual and visual features, where PointCLIP V2 (Bottom) shows better alignment.

sidering other 3D open-world tasks. PointCLIP [62], for the first time, indicates that CLIP can be adapted for zero-shot point cloud classification without any 3D training. It projects the 'unseen' 3D point cloud sparsely into 2D depth maps, and leverages CLIP's image-text alignment for depth map recognition. However, as a preliminary work, the performance of PointCLIP is far from satisfactory as shown in Figure 1, which cannot be put into actual use. More importantly, PointCLIP only draws support from pre-trained CLIP, without considering the powerful large-scale language model (LLM). Therefore, we ask the question: *Can we properly unify CLIP and LLM to fully unleash their potentials for unified 3D open-world understanding?*

We observe that PointCLIP mainly suffers from two factors concerning the 2D-3D domain gap. **(1) Sparse Projection.** PointCLIP simply projects 3D point clouds onto depth maps as sparsely distributed points with depth values (Figure 2). Though simple, the scatter-style figures are dramatically different from the real-world pre-training images for both appearances and semantics, which severely confuses CLIP's visual encoder. **(2) Naive Text.** PointCLIP mostly inherits CLIP's 2D text input, "`a photo of a [CLASS].`" and only appends simple 3D-related words, "`a depth map`". As visualized in Figure 3, the textual features extracted by CLIP can hardly focus on the target object with high similarity scores. Such naive text cannot fully describe 3D shapes and harms the pre-trained language-image alignment.

In this paper, we integrate the advantage of CLIP and the GPT-3 [3] model and propose **PointCLIP V2**, a powerful framework for unified 3D open-world understanding, including zero-shot/few-shot 3D classification, zero-shot part segmentation, and zero-shot 3D object detection. Without 'seeing' any 3D training data, V2 can project point clouds into realistic 2D figures and align them with 3D-aware text, which fully unleashes CLIP's pre-trained knowledge in the 3D domain.

Firstly, we propose to **Prompt CLIP with Realistic Projection**, which generates CLIP-preferred images from 3D point clouds. Specifically, we transform the irregular point cloud into grid-based voxels and then apply non-parametric 3D local filtering on top. By this, the projected 3D shapes are composed of denser points with smoother depth values. As shown in Figure 2, our generated figures are more visually similar to real-world images and can highly unleash the representation capacity of CLIP's pre-trained visual encoder. Secondly, we **Prompt GPT with 3D Command** to generate text with rich 3D semantics as the input of CLIP's textual encoder. By feeding heuristic 3D-oriented command into GPT-3, *e.g.*, "`Give a caption of a table depth map:`", we leverage its language-generative knowledge to obtain a series of 3D-specific text, *e.g.*, "`A height map of a table with a top and several legs.`". A group of language commands is customized to prompt GPT-3 to produce diverse text with 3D shape information. As shown in Figure 3, the textual features of PointCLIP V2 exert stronger matching properties to the projected maps, largely boosting CLIP's image-text alignment for 3D point clouds.

With our prompting schemes, PointCLIP V2 exhibits superior performance for zero-shot 3D classification, surpassing PointCLIP by $+42.90\%$, $+40.44\%$, and $+28.75\%$ accuracy, respectively on ModelNet10 [53], ModelNet40 [53], and ScanObjectNN [44] datasets. Further, our approach can be adapted for more no-trivial 3D open-world tasks by marginal modifications, such as a learnable 3D smoothing for 3D few-shot classification, a back-projection head for zero-shot segmentation, and a 3D region proposal network for zero-shot detection. This fully indicates the power of V2 for general 3D open-world understanding.

Our contributions are summarized as follows:

- We propose PointCLIP V2, a powerful cross-modal learner unifying CLIP and GPT-3 to transfer the pre-trained vision-language knowledge into 3D domains.
- We introduce a realistic projection to prompt CLIP and 3D-oriented command to prompt GPT-3 to effectively mitigate the domain gap among 2D, 3D, and language.

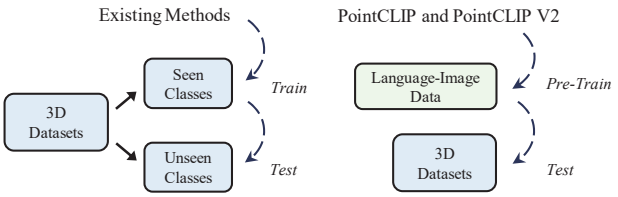

Figure 4. **Comparison of Open-world Settings.** Existing methods still depend on prerequisite 3D training to recognize the 'unseen' point clouds. In contrast, we require no training in the 3D domain and directly conduct 3D open-world understanding.

Method	Latency	ModelNet40	ScanObjectNN
Phong Shading [42]	107.2	57.30	29.33
Height Map [43]	87.7	54.73	26.25
Silhouette Map [43]	87.9	48.40	20.91
PointCLIP [62]	**11.3**	42.53	26.37
PointCLIP V2	16.7	**64.22**	**35.36**

Table 1. **Comparison of Different Projection Methods.** We report zero-shot classification results (%) on two datasets [53, 44], and compare the inference latency (ms) by projecting 10-view images from an input point cloud.

- As the first work for unified 3D open-world learning, our PointCLIP V2 can be further extended for zero-shot part segmentation and 3D object detection.

2. Related Works

3D Open-world Learning. Traditional methods for 3D open-world learning still require 3D training data as a pre-training stage. The series of work of Cheraghian *et al*. train zero-shot classifiers on 'seen' 3D categories by maximizing inter-class divergence in latent space, and then test on 'unseen' ones [6, 8, 7]. Some recent works [30, 24, 32, 27] also investigate open-world semantic segmentation and 3D object detection for more complex 3D scenes. Inspired by CLIP-based adaption methods [60, 13, 23], PointCLIP [62] achieves zero-shot point cloud recognition without any training on 3D datasets. By transferring the pre-trained CLIP model [37], the 2D knowledge can be effectively utilized for recognizing 3D data. CLIP2Point [19] further improves the adaption performance of CLIP on point clouds by an additional 3D pre-training. In this paper, we propose PointCLIP V2 and follow the open-world setting of PointCLIP, which is more challenging than previous methods as compared in Figure 4. We require no 'seen' 3D training and, for the first time, simultaneously conduct zero-shot 3D part segmentation and object detection, achieving unified 3D open-world understanding.

Projection for Point Clouds. Concurrent to point-based methods [35, 36, 28], projection-based point cloud analysis aims to utilize plentiful 2D networks for 3D domains by projecting point clouds into 2D images [42, 41, 58, 47, 50, 1]. Therein, PointCLIP [62] follows SimpleView [14] to conduct perspective transformation as 3D-to-2D projection, which achieves high efficiency but limited classification accuracy. Under 3D open-world settings, we are motivated to develop more efficient and realistic projection methods for prompting CLIP on point cloud data. In Table 1, we compare our approach with existing advanced projection methods for latency and accuracy. For a fair comparison, we implement all prior works under the pipeline of our V2, namely, with our GPT prompting approach to fully reveal their effectiveness. As shown, our realistic projection exhibits faster inference speed than other approaches and attains higher zero-shot performance than PointCLIP.

Prompt Learning in Vision. Prompt engineering first derives from natural language processing, where a textual template, termed as prompt, is generated to narrow the domain gap between the pre-training pre-text task and downstream scenarios [25, 22, 46, 22]. Inspired by this, CoOp [69] firstly introduces learnable prompting into 2D vision-language classification, and the follow-up CoCoOp [68] extends it for 2D domain generalization. CuPL [34] and CaFo [63] leverage GPT-3 [3] to enhance the downstream performance of CLIP on various 2D datasets. From another perspective, visual prompting methods propose to append input images with learnable visual pixels [21, 2, 5, 12] or embeddings [21, 16, 64], and improve pre-trained vision backbones without downstream fine-tuning. In this paper, we seek to prompt both CLIP's visual encoder by realistic projection and textual encoder by GPT-3 to improve its zero-shot prediction.

GPT-3 Model. The Generative Pre-trained Transformer (GPT) models [38, 39, 3] have achieved a progressive improvement in processing natural languages. Among them, GPT-3 demonstrates a remarkable proficiency in both language comprehension and generation, compared to its predecessors [38, 39, 26, 56, 40]. GPT-3 is a large-scale autoregressive language model of 175 billion trainable parameters. Although not open-sourced, some efforts have explored its application to downstream tasks, such as PICa [57] for visual question-answering, CuPL [34] for 2D zero-shot recognition, and CaFo [63] for 2D few-shot learning. In this work, we, for the first time, prompt GPT-3 [3] to boost open-world 3D tasks via 3D-related command.

3. Methods

The overall framework of PointCLIP V2 is shown in Figure 7. Inheriting from CLIP [37], our framework consists of two pre-trained visual and textual encoders. To bridge the modal gap, we introduce a realistic projection (Sec. 3.1)

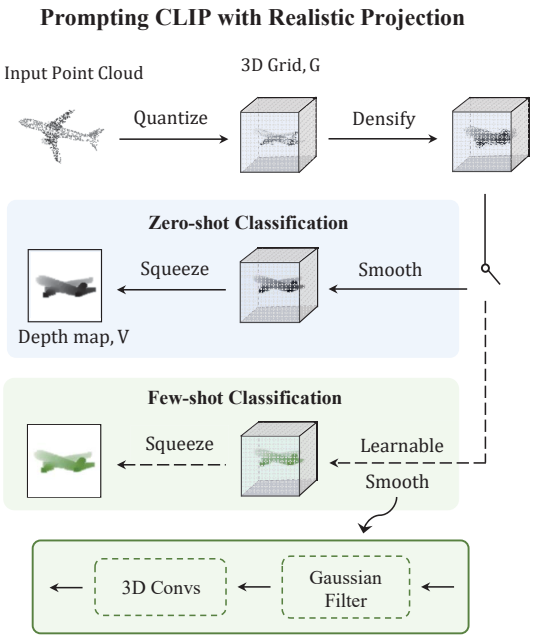

Figure 5. **Prompting CLIP with Realistic Projection.** We present the projection pipeline for one of the views. The switch selects zero- or few-shot classification with learnable smoothing.

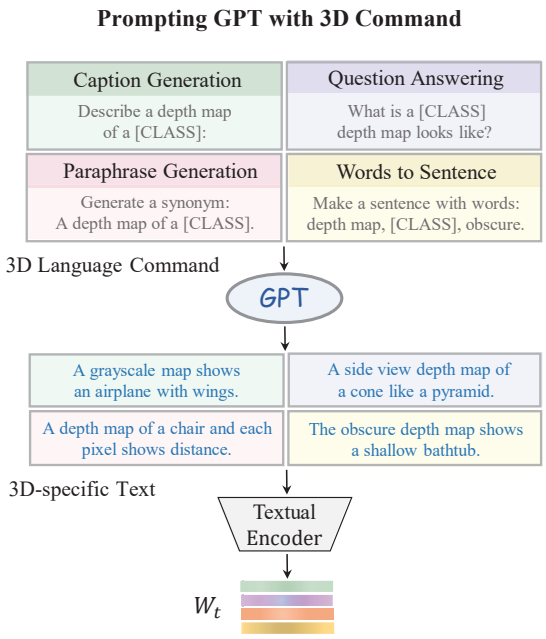

Figure 6. **Prompting GPT with 3D Command.** We feed four types of language command into the pre-trained GPT-3, which generates a series of 3D-specific text for CLIP's textual encoder.

from 3D to depth maps, and GPT-generated 3D-specific text (Sec. 3.2) to align depth maps with languages. PointCLIP V2 can also be extended to various 3D tasks for unified 3D open-world learning (Sec. 3.3).

3.1. Prompting CLIP with Realistic Projection

To generate more realistic 2D input from 3D data for CLIP and also achieve time efficiency, we project 3D point clouds into depth maps by four steps: Quantize, Densify, Smooth, and Squeeze, as shown in Figure 5.

Quantize. For different M views to be projected, we respectively create a zero-initialized 3D grid $G \in \mathbb{R}^{H \times W \times D}$, where H, W, D denote its spatial resolutions and D specially represents the depth dimension vertical to the view plane. Taking one view as an example, we normalize the 3D coordinates of the input point cloud into $[0, 1]$ and project a point $p = (x, y, z)$ into a voxel in the grid by

$$G(\lceil sHx \rceil, \lceil sWy \rceil, \lceil Dz \rceil) = z, \tag{1}$$

where the voxels are assigned with different depth values, and $s \in (0, 1]$ denotes a scale factor to adjust the projected shape size. For multiple points projected into the same voxel, we simply assign the minimum depth value. This is because, from the perspective of the target image plane, the points with a smaller depth value z would occlude the larger ones. Then, we obtain a 3D grid G containing sparse depth values, most voxels of which are empty due to the sparsity of point clouds.

Densify. To tackle such unreal scattering, we densify the grid via a local mini-value pooling operation to guarantee visual continuity. We reassign every voxel in G by the minimum voxel value within a local spatial window. Likewise, compared to the average and max pooling, preserving the minimum depth values accords with the occluded visual appearances on the projected maps. In this way, the originally vacant voxels between the sparse points can be effectively filled with reasonable depth values, while the background voxels still remain empty, which derives dense and solid spatial shape representations.

Smooth. As the local pooling operation might introduce artifacts on some 3D surfaces, we adopt a non-parametric Gaussian kernel for shape smoothing and noise filtering. With a proper kernel size and variance, the filtering can not only remove the spatial noises caused by densification but also preserve the sharpness of edges and corners in the original 3D shapes. By this, we acquire a more compact and smooth shape represented by the 3D grid.

Squeeze. As the final step, we simply squeeze the depth dimension of G to acquire the projected depth map $V \in \mathbb{R}^{H \times W}$. We extract the minima of every depth channel as the value for each pixel location and repeat it for three times as the RGB intensity. Our grid-based projection can be simply achieved by a minimum pooling along the depth channel of G, more friendly for hardware implementation.

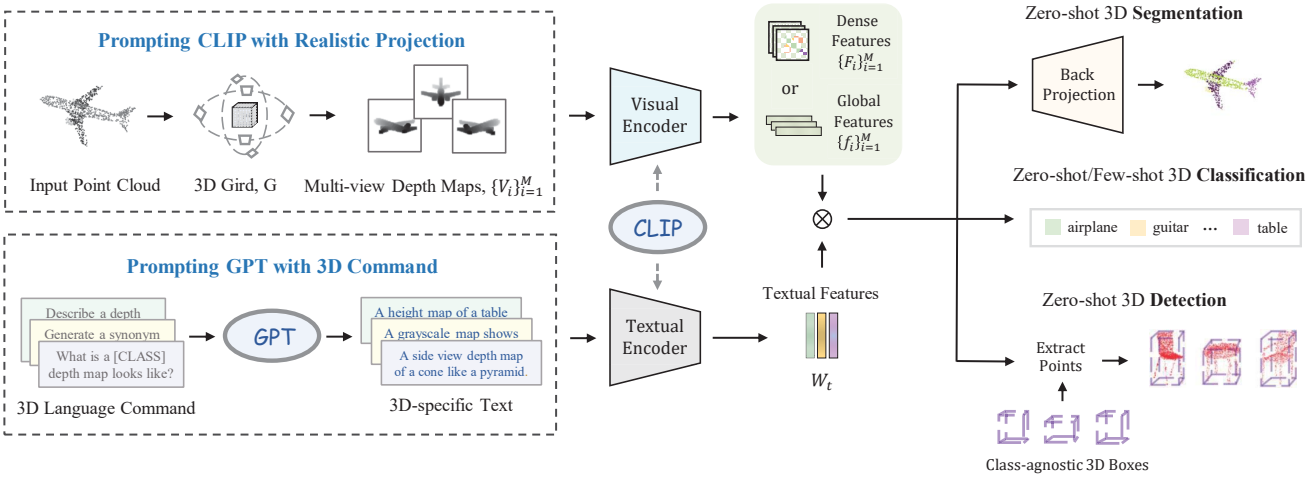

Figure 7. **The Unified Framework of PointCLIP V2 for 3D Open-world Learning.** We first generate high-quality depth maps via a realistic projection to prompt CLIP's [37] visual encoder. Then, we design 3D language command to prompt GPT-3 [3] for 3D-specific text into CLIP's textual encoder. V2 can also be extended to 3D segmentation and detection by simple modifications.

3.2. Prompting GPT with 3D Command

To better activate CLIP's textual encoder to align with our depth maps, we aim to utilize 3D-specific description with category-wise shape characteristics as the textual input of CLIP, instead of using the general "`a photo of a [CLASS]:`". Considering the powerful descriptive capacity of LLMs, we leverage GPT-3 [3] to generate 3D-specific text with sufficient 3D semantics for CLIP's textual encoder as shown in Figure 6. Normally, GPT-3 receives a language command and outputs a response via pre-trained knowledge. To fully adapt GPT-3 to 3D domains, we propose the following four series of heuristic command:

Caption Generation. Given a descriptive command, GPT-3 synthesizes general captions for the target projected 3D shape, *e.g.*, Input: "`Describe a depth map of a [window]:`"; GPT-3: "`It depicts the [window] as a dark pane.`".

Question Answering. GPT-3 produces descriptive answers to the 3D-related question, *e.g.*, Input: "`How to describe a depth map of a [table]?`"; GPT-3: "`The [table] may have a rectangular or circular flat top and legs.`".

Paraphrase Generation. For a depth map description, GPT-3 is expected to generate a synonymous sentence. *e.g.*, Input: "`Generate a synonym for the sentence: A grayscale depth map of an inclined [bed].`"; GPT-3: "`An monochrome depth map of an oblique [bed].`".

Words to Sentence. Based on a group of keywords, GPT-3 is requested to organize them into a complete sentence and enrich additional shape-related contents, *e.g.*, Input: "`Make a sentence using these words: a [table], depth map, smooth.`"; GPT-3: "`This smooth depth map shows a [table] at the corner.`". The adjective "`smooth`" here depicts the natural appearance caused by the smoothing operation.

For a K-category 3D dataset, we place K category names at the "`[CLASS]`" position of each command and feed them into GPT-3, which generates 3D-specific descriptions with rich category-wise semantics. Finally, we integrate the descriptions of each category and regard them as the input for CLIP's textual encoder.

3.3. Unified Open-world Learning

By introducing the realistic projection and 3D-specific text, PointCLIP V2 exhibits strong generalization capacity and can be adapted for different 3D open-world tasks.

3D Zero-shot Classification. For all M views in the visual branch, we feed the projected depth maps $\{V_i\}_{i=1}^M$ into CLIP's visual encoder and obtain the multi-view features $\{f_i\}_{i=1}^M$, where $f_i \in \mathbb{R}^{1\times C}$. For the textual branch, we leverage CLIP's textual encoder to extract the category feature $W_t \in \mathbb{R}^{K\times C}$, which serves as the zero-shot classification weights. Then, the final zero-shot classification logits are calculated by aggregating the multi-view alignment between $\{f_i\}_{i=1}^M$ and W_t, formulated as

$$\text{logits} = \sum\nolimits_{i=1}^{M} \alpha_i \cdot f_i W_t^T \in \mathbb{R}^{1\times K}, \tag{2}$$

where α_i denotes a hyper-parameter weighing the importance of view i.

Method	2D Pre-train	3D Pre-train	ModelNet10	ModelNet40	S-OBJ_ONLY	S-OBJ_BG	S-PB_T50_RS
CLIP2Point [19]	✓	✓	66.63	49.38	35.46	30.46	23.32
Cheraghian [7]	-	✓	68.50	-	-	-	-
PointCLIP [62]	✓	-	30.23	23.78	21.34	19.28	15.38
PointCLIP V2	✓	-	**73.13**	**64.22**	**50.09**	**41.22**	**35.36**
Improvement			+42.90	+40.44	+28.75	+21.94	+19.98

Table 2. **Zero-shot 3D Classification (%) ModelNet10 [53], ModelNet40 [53] and ScanObjectNN [44].** We report the performance of other methods with their ***best-performing settings,* e.g.*, visual encoder, projected view number, and textual input***. "2D Pre-train" denotes the pre-training of CLIP on image-language pairs, and "3D Pre-train" denotes the training on 3D datasets.

3D Few-shot Classification. Given a small set of 3D training data, we can modify our smoothing operation of the realistic shape projection to be learnable, as shown in Figure 5. Specifically, as the irregular point clouds have been converted into grid-based voxels, we adopt two 3D convolutional layers after the Gaussian filter. Such learnable modules can summarize the 3D geometric knowledge from the few-shot dataset, and further adapt the 3D shape to be more CLIP-friendly. During training, we freeze the two encoders of CLIP to preserve the pre-trained knowledge and avoid over-fitting on small-scale few-shot data.

3D Zero-shot Part Segmentation. Besides shape classification, we propose a zero-shot segmentation pipeline for our framework, which can also work for the existing PointCLIP. Instead of the global features $\{f_i\}_{i=1}^{M}$, we adopt CLIP's visual encoder to extract dense features $\{F_i\}_{i=1}^{M}$ from $\{V_i\}_{i=1}^{M}$, where $F_i \in \mathbb{R}^{H\times W\times C}$. Specifically, we output the feature maps from the visual encoder before its final pooling operation and upsample the features into the original depth map size. For our 3D-specific text, we utilize GPT-3 to generate the descriptions for different part categories. As an example, for a part category "`[PART]`" within object "`[CLASS]`", we construct the command as "`Describe the [PART] part of a [CLASS] in a depth map:`". Then, for view i, we conduct dense alignment between each pixel and the textual feature W_t, *i.e.*, segmenting different parts of the shape on multi-view depth maps, formulated as

$$\text{logits}_i = F_i W_t^T \in \mathbb{R}^{H\times W\times K}. \tag{3}$$

Each element in logits_i denotes the pixel-wise classification logits. After this, we back-project the logits of different views into the 3D space according to the 2D-3D correspondence. As one view can only depict a partial point cloud due to occlusion, we average the prediction across different views for each point, where we acquire the final part segmentation logits in 3D space. Via the geometric back projection, the segmentation task in 3D can be tackled in a zero-shot manner.

Zero-shot 3D Object Detection. For 3D object detection, we follow the settings of 2D open-world detection [15, 67] to equip our V2 as a zero-shot classification head on top of pre-trained region proposal networks (RPN). We first utilize 3DETR [31] as the 3D RPN to generate class-agnostic 3D box candidates. Then, we extract the raw points within each 3D box and feed them into V2 for zero-shot classification. By this, the V2-based 3DETR can detect 'unseen' objects in a zero-shot manner.

4. Experiments

In this section, we first illustrate the detailed network configurations of PointCLIP V2, and then present our open-world performance on different 3D tasks.

4.1. Implementation Details

CLIP Prompting. We follow PointCLIP [62] to project a point cloud into depth maps of 10 views. We set the size of grid G as $H \times W \times D = 112 \times 112 \times 8$, and the projected depth map is upsampled to 224×224. The point cloud is placed at the center of the grid, and the scale factor s is set to 0.7 for better visual appearances. The window size of the minimum pooling for densifying is $(6, 6, 2)$. The kernel size of the Gaussian filter is set as $(3, 3, 1)$. We randomly sample 1024 points as input and adopt Vision Transformer [10] with patch size 16×16 as default, denoted as ViT-B/16.

GPT Prompting. We design 50 different 3D language commands, containing 13 for caption generation, 13 for question answering, 12 for paraphrase generation, and 12 for words-to-sentence. Each command triggers GPT-3 to produce 20 3D-specific descriptions, and we finally obtain around 250 descriptions for each command type and 1000 descriptions in total for one category. We use "text-davinci-002" GPT-3 engine and set the temperature constant to 0.7. The largest length of a 3D-specific description is set to 40. For the textual encoder, a 12-layer transformer [45] is adopted to encode our generated text.

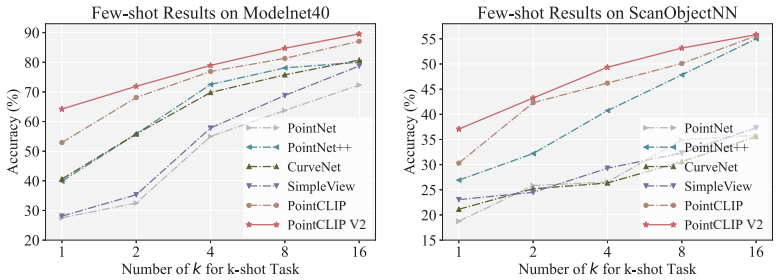

Figure 8. **Few-shot 3D Classification on ModelNet40 [53] and ScanObjectNN [44].** We adopt the PB_T50_RS split of ScanObjectNN for comparison.

Learnable Smooth	View Weighing	GPT Prompting	16-shot
-	-	-	85.52
✓	-	-	86.22
✓	✓	-	87.11
✓	-	✓	89.55
✓	✓	✓	89.55

Table 3. **Ablation Study of Few-shot Learning on ModelNet40 [53].** We report the 16-shot classification accuracy (%).

Quantize	Densify	Smooth	Squeeze	Zero-shot
-	Min	✓	-	57.35
✓	-	-	✓	44.50
✓	Min	-	✓	59.64
✓	-	✓	✓	50.20
✓	Max	✓	✓	57.35
✓	Avg	✓	✓	60.71
✓	Min	✓	✓	64.22

Table 4. **Ablation Study of Realistic Shape Projection** on ModelNet40 [53] zero-shot classification (%). We compare the four steps in our projection module.

Caption	Question	Paraphrase	Words	Zero-shot
-	-	-	-	39.11
✓	-	-	-	61.67
✓	✓	-	-	60.86
✓	-	✓	-	61.12
✓	✓	✓	-	63.29
✓	✓	-	✓	61.26
✓	✓	✓	✓	64.22

Table 5. **Ablation Study of GPT Prompting** on ModelNet40 [53] zero-shot classification (%). We compare four types of language command to generate the 3D-specific text.

4.2. Zero-shot Classification

Settings. The zero-shot classification performance is evaluated on three widely-used benchmarks: ModelNet10 [53], ModelNet40 [53] and ScanObjectNN [44]. Three splits of the ScanObjectNN dataset are investigated: OBJ_ONLY, OBJ_BG, and PB_T50_RS. Following the zero-shot principle, we directly test the classification performance on the full test set without learning from the training set. We compare existing methods under their best settings. Specifically, ViT-B/16 is adopted for both our model and CLIP2Point [19]. For PointCLIP, we utilize ResNet-101 [18], ResNet-50×4 [37], and ViT-B/16, respectively for ModelNet10, ModelNet40, and ScanObjectNN datasets, which is to fully achieve its best performance.

Main Results. In Table 2, we compare the zero-shot classification performance with existing approaches. Some models require extra pre-training on 3D point cloud datasets. CLIP2Point trains a depth map encoder on ShapeNet dataset [4], and then uses it for a 3D zero-shot classification task. Cheraghian *et al*. [7] directly extracts point cloud features with a 3D encoder. They sample 'seen' categories in the dataset to pre-train the model, and validate on the 'unseen' categories. In contrast, PointCLIP and our V2 discard any 3D training and can directly test on 3D datasets. For all three benchmarks, our approach outperforms existing works by significant margins. V2 achieves 73.13% and 64.22% accuracy on ModelNet10 and ModelNet40, respectively, surpassing PointCLIP by +42.90% and +40.44%. V2 also achieves 35.36% on PB_T50_RS split of the ScanObjectNN dataset, demonstrating our effectiveness under noisy real-world scenes.

Ablation Study. In Table 4, we conduct an ablation study of PointCLIP V2 concerning four steps of the realistic projection module. When we directly project the point cloud into 2D images via orthogonal projection, the zero-shot accuracy performs 57.35%, reduced by −6.87%. If the quantizing step is adopted, the densifying and smoothing operation can improve zero-shot performance by +15.14% and +5.7%, respectively, indicating the importance of these two steps. We also compare alternative pooling operations for the densifying step, including maximum, minimum, and average pooling. We observe that the minimum pooling achieves the best performance, which is consistent with the occlusion effect in the real world. In Table 5, we show the effect of four command types in the GPT prompting module. Under different command combinations, the zero-shot performance is improved with various degrees. If using all four types, the 3D-specific text improves the zero-shot performance by +25.11%, indicating the great significance of better language-image alignment.

4.3. Few-shot Classification

Settings. We test k-shot classification performance on ModelNet40 [53] and ScanObjectNN [44] datasets, where

	mIoU$_I$	Airplane	Bag	Cap	Chair	Earphone	Guitar	Knife	Laptop	Mug	Rocket	Skate	Table
# Shapes	2874	341	14	11	704	14	159	80	83	38	12	31	848
PointCLIP*	31.0	22.0	44.8	13.4	18.7	28.3	22.7	24.8	22.9	48.6	22.7	42.7	45.5
PointCLIP V2	**49.5**	**33.5**	**60.4**	**52.8**	**51.5**	**56.5**	**71.5**	**66.7**	**61.6**	**48.0**	**49.6**	**43.9**	**61.1**

Table 6. **Zero-shot Part Segmentation (%) on ShapeNetPart [59].** We implement PointCLIP by our proposed segmentation pipeline.

	Method	Mean	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Counter	Desk	Sink	Bathtub
AP$_{25}$	PointCLIP*	6.00	3.99	4.82	45.16	4.82	7.36	4.62	2.19	1.02	4.00	13.40	6.46
	PointCLIP V2	**18.97**	**19.32**	**20.98**	**61.89**	**15.55**	**23.78**	**13.22**	**17.42**	**12.43**	**21.43**	**14.54**	**16.77**
AP$_{50}$	PointCLIP*	4.76	1.67	4.33	39.53	3.65	5.97	2.61	0.52	0.42	2.45	5.27	1.31
	PointCLIP V2	**11.53**	**10.43**	**13.54**	**41.23**	**6.60**	**15.21**	**6.23**	**11.35**	**6.23**	**10.84**	**11.43**	**10.14**

Table 7. **Zero-shot 3D Object Detection (%) on ScanNet V2 [9].** We implement PointCLIP by our proposed detection pipeline.

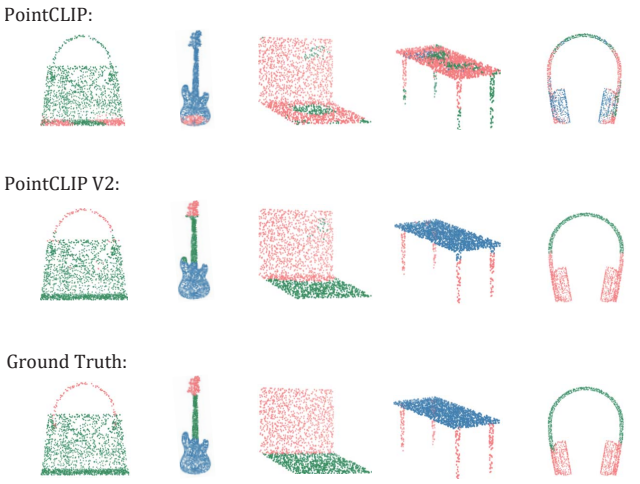

Figure 9. **Visualization of Zero-shot Part Segmentation on ShapeNetPart [59].** Our V2 exhibits better fine-grained segmentation than PointCLIP.

$k \in \{1, 2, 4, 8, 16\}$. We adopt the same 3D-specific text used in the zero-shot task as textual input and jointly train the learnable smoothing (Figure 5) and inter-view adapter [62]. The 3D convolution layers adopt a $5 \times 5 \times 3$ kernel size and are followed by a batch normalization [20] with a ReLU non-linear activation [33].

Main Results. In Figure 8, we show the few-shot classification results of V2, comparing with PointCLIP and other four representative 3D networks: PointNet [35], PointNet++ [36], SimpleView [14], and CurveNet [54]. As shown, V2 outperforms all other methods by few-shot training and shows a more significant improvement on 1-shot classification. V2 surpasses PointCLIP's 1-shot accuracy by $+12\%$ on ModelNet40 and $+7\%$ on ScanObjectNN. In addition, our approach achieves a 16-shot accuracy of 89.55% on ModelNet40 dataset, even approaching the fully supervised PointNet[35].

Ablation Study. In Table 3, we report the impact of different modules on few-shot V2 with 16-shot results, including the learnable smoothing, the view weighing following PointCLIP, and 3D-specific text from GPT prompting. We find that the learnable 3D projection module improves 16-shot accuracy by $+0.7\%$ than the fixed one, and adopting 3D-specific text improves accuracy by $+3.33\%$.

4.4. Zero-shot Part Segmentation

Settings. We evaluate the zero-shot segmentation performance on the ShapeNetPart dataset [59], which includes 16 categories and 50 annotated parts. Following prior methods [36, 48, 28], we sample 2048 points from each point cloud, and test on the official test split. *For comparison, we implement PointCLIP via our proposed zero-shot segmentation pipeline and report the best-performing results.*

Main Results. We show the mean intersection of union score across instances (mIoU$_I$) in Table 6. Our method surpasses PointCLIP by $+17.4\%$ for overall mIoU$_I$ and performs consistently better on different object categories. We also visualize the segmentation results in Figure 9, which further demonstrates our effectiveness to capture fine-grained 3D patterns in a zero-shot manner.

4.5. Zero-shot 3D Object Detection.

Settings. ScanNet V2 dataset [9] is utilized to evaluate the detection performance, which contains 18 object categories. We adopt the pre-trained 3DETR-m [31] model as the region proposal network and extract 1024 points within each 3D box. We report the zero-shot detection performance on the validation set using mean Average Precision (mAP) at two different IoU thresholds of 0.25 and 0.5, denoted as AP$_{25}$ and AP$_{50}$. *Also, PointCLIP is implemented by our efforts for zero-shot 3D detection and we report the best-performing results.*

Method	PointCLIP	CLIP2Point	PointCLIP V2
GFLOPs	16.46	16.46	16.51
Memory (GB)	2901	3006	2967
Accuracy (%)	16.94	49.38	55.92

Table 8. **Comparison of Accuracy and Computation Overhead** with other approaches on ModelNet40 [53].

Main Results. Table 7 shows our zero-shot 3D detection results compared with PointCLIP. We observe that PointCLIP V2 achieves mAP_{25} and mAP_{50} of 18.97% and 11.53%, outperforming PointCLIP by $+12.97\%$ and $+6.77\%$, respectively. This verifies that V2 is superior to recognize 3D open-world objects in real-world scenes and obtains great potential for general 3D open-world learning.

4.6. Other Experiments

Computation Burden. We have compared the latency of inference in Table 1. Additionally, we compare the computation complexity to PointCLIP [62] and CLIP2Point [19] in Table 8. We test the computation overhead of each inference on 1 RTX A6000 with ViT-B/32 backbone. From the table, V2 causes a similar overhead to PointCLIP and achieves superior zero-shot accuracy on ModelNet40. Thus we achieve a better accuracy-efficiency trade-off.

More Ablations for Zero-shot Classification. In Table 9 and 10, we additionally investigate 2 factors that influence the zero-shot classification performance: the visual encoder backbone and the number of sampled points. **1) Different Backbones.** In Table 9, we examine the results with different backbones on ModelNet40 [53] and ScanObjectNN [44] datasets. We observe that the default ViT-B/16 backbone achieves the best overall performance. **2) Sample Rate of Points.** Table 10 presents the effect of different numbers of sampled points. Note that the officially released ModelNet40 dataset contains only 2048 points per point cloud, so we adopt a resampled version of ModelNet40 from [49], which contains 8192 points per point cloud. We observe improvements when increasing the sampling rate of points.

5. Conclusion

We propose PointCLIP V2, a powerful and unified 3D open-world learner, which surpasses the existing PointCLIP with significant margins. We propose to prompt CLIP with a realistic projection module for producing high-quality depth maps from 3D, and prompt GPT-3 model to generate 3D-specific descriptions. The visual and language representations achieve better alignment via prompting. Besides classification, V2 can generalize to various challenging tasks with promising performance, including 3D few-shot classification, 3D zero-shot part segmentation, and object detection. For future work, we will further explore how to adapt CLIP to wider open-world applications, *e.g.*, outdoor 3D detection and visual grounding.

Datasets	RN50	RN101	ViT-B/32	ViT-B/16	RN.×4
ModelNet40	46.45	49.34	60.00	**64.22**	56.28
ScanObjectNN	33.21	31.47	**35.36**	34.91	34.98

Table 9. **Ablation Study on Visual Encoders** for Zero-shot Classification (%) on ModelNet40 [53] and ScanObjectNN [44].

Point Number	1024	2048	3072	4096	8192
ModelNet40	64.22	65.28	66.17	66.45	**68.56**
ScanObjectNN	34.91	36.05	36.26	37.27	**38.90**

Table 10. **Ablation Study on Point Number** for Zero-shot Classification (%).

Acknowledgement. This work is partially supported by the National Natural Science Foundation of China (Grant No.62206272), and by the National Key R&D Program of China (NO.2022ZD0160100).

References

[1] Syeda Mariam Ahmed, Pan Liang, and Chee Meng Chew. EPN: Edge-aware PointNet for object recognition from multi-view 2.5D point clouds. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3445–3450, 2019. 3

[2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 2022. 3

[3] 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. 1, 2, 3, 5

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 7

[5] Aochuan Chen, Peter Lorenz, Yuguang Yao, Pin-Yu Chen, and Sijia Liu. Visual prompting for adversarial robustness. *arXiv preprint arXiv:2210.06284*, 2022. 3

[6] Ali Cheraghian, Shafin Rahman, Dylan Campbell, and Lars Petersson. Mitigating the hubness problem for zero-shot learning of 3D objects. *arXiv preprint arXiv:1907.06371*, 2019. 3

[7] Ali Cheraghian, Shafin Rahman, Townim F Chowdhury, Dylan Campbell, and Lars Petersson. Zero-shot learning on 3d point cloud objects and beyond. *International Journal of Computer Vision*, pages 1–21, 2022. 3, 6, 7

[8] Ali Cheraghian, Shafin Rahman, and Lars Petersson. Zero-shot learning of 3D point cloud objects. In *IEEE International Conference on Machine Vision Applications*, pages 1–6, 2019. 3

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 8

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[11] Kexue Fu, Peng Gao, ShaoLei Liu, Renrui Zhang, Yu Qiao, and Manning Wang. Pos-bert: Point cloud one-stage bert pre-training. *arXiv preprint arXiv:2204.00989*, 2022. 1

[12] Yulu Gan, Yan Bai, Yihang Lou, Xianzheng Ma, Renrui Zhang, Nian Shi, and Lin Luo. Decorate the newcomers: Visual domain prompt for continual test time adaptation. *arXiv preprint arXiv:2212.04145*, 2023. 3

[13] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. CLIP-Adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 3

[14] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. *International Conference on Machine Learning*, 2021. 3, 8

[15] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 6

[16] Ziyu Guo, Yiwen Tang, Renrui Zhang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. Viewrefer: Grasp the multi-view knowledge for 3d visual grounding with gpt and prototype guidance. *arXiv preprint arXiv:2303.16894*, 2023. 3

[17] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzhi Li, and Pheng Ann Heng. Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training. *IJCAI 2023*, 2023. 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[19] 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. *arXiv preprint arXiv:2210.01055*, 2022. 3, 6, 7, 9

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. 8

[21] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727, 2022. 3

[22] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020. 3

[23] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 388–404. Springer, 2022. 3

[24] Bo Liu, Shuang Deng, Qiulei Dong, and Zhanyi Hu. Segmenting 3D hybrid scenes via zero-shot learning. *arXiv preprint arXiv:2107.00430*, 2021. 3

[25] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 3

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 3

[27] Yuheng Lu, Chenfeng Xu, Xiaobao Wei, Xiaodong Xie, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. Open-vocabulary 3D detection via image-level class and debiased cross-modal contrastive learning. 2022. 3

[28] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *arXiv preprint arXiv:2202.07123*, 2022. 3, 8

[29] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Yunde Jia, and Luc Van Gool. Towards a weakly supervised framework for 3D point cloud object detection and annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[30] Björn Michele, Alexandre Boulch, Gilles Puy, Maxime Bucher, and Renaud Marlet. Generative zero-shot learning for semantic segmentation of 3D point clouds. In *IEEE International Conference on 3D Vision*, pages 992–1002, 2021. 3

[31] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *International Conference on Computer Vision*, pages 2906–2917, 2021. 6, 8

[32] Muhammad Ferjad Naeem, Evin Pınar Örnek, Yongqin Xian, Luc Van Gool, and Federico Tombari. 3D compositional zero-shot learning with decompositional consensus. In *European Conference on Computer Vision*, pages 713–730, 2022. 3

[33] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve Restricted Boltzmann machines. In *International Conference on Machine Learning*, 2010. 8

[34] Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? Generating customized prompts for zero-shot image classification. *arXiv preprint arXiv:2209.03320*, 2022. 3

[35] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 1, 3, 8

[36] 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. 1, 3, 8

[37] 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. 1, 3, 5, 7

[38] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 3

[39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3

[40] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 3

[41] Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai. DeepPano: Deep panoramic representation for 3-D shape recognition. *IEEE Signal Processing Letters*, 22(12):2339–2343, 2015. 3

[42] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *International Conference on Computer Vision*, pages 945–953, 2015. 3

[43] Jong-Chyi Su, Matheus Gadelha, Rui Wang, and Subhransu Maji. A deeper look at 3D shape classifiers. In *European Conference on Computer Vision Workshops*, 2018. 3

[44] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *International Conference on Computer Vision*, pages 1588–1597, 2019. 2, 3, 6, 7, 9

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 6

[46] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Nov. 2019. 3

[47] Chu Wang, Marcello Pelillo, and Kaleem Siddiqi. Dominant set clustering and pooling for multi-view 3D object recognition. *arXiv preprint arXiv:1906.01592*, 2019. 3

[48] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions On Graphics*, 38(5):1–12, 2019. 1, 8

[49] Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. P2P: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. *arXiv preprint arXiv:2208.02812*, 2022. 9

[50] Xin Wei, Ruixuan Yu, and Jian Sun. View-GCN: View-based graph convolutional network for 3D shape analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1850–1859, 2020. 3

[51] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 1

[52] Yanmin Wu, Xinhua Cheng, Renrui Zhang, Zesen Cheng, and Jian Zhang. Eda: Explicit text-decoupling and dense alignment for 3d visual and language learning. *arXiv preprint arXiv:2209.14941*, 2022. 1

[53] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 2, 3, 6, 7, 9

[54] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *International Conference on Computer Vision*, pages 915–924, 2021. 1, 8

[55] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *European Conference on Computer Vision*, pages 1–19, 2020. 1

[56] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[57] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of GPT-3 for few-shot knowledge-based VQA. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089, 2022. 3

[58] Ze Yang and Liwei Wang. Learning relationships for multi-view 3D object recognition. In *International Conference on Computer Vision*, pages 7505–7514, 2019. 3

[59] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics*, 35(6):1–12, 2016. 8

[60] Renrui Zhang, Rongyao Fang, Peng Gao, Wei Zhang, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-Adapter: Training-free CLIP-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 3

[61] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *arXiv preprint arXiv:2205.14401*, 2022. 1

[62] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022. 1, 2, 3, 6, 8, 9

[63] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Hongsheng Li, Yu Qiao, and Peng Gao. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. *arXiv preprint arXiv:2303.02151*, 2023. 3

[64] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. *arXiv preprint arXiv:2305.03048*, 2023. 3

[65] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. *arXiv preprint arXiv:2212.06785*, 2022. 1

[66] Renrui Zhang, Liuhui Wang, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Parameter is not all you need: Starting from non-parametric networks for 3d point cloud analysis. *arXiv preprint arXiv:2303.08134*, 2023. 1

[67] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022. 6

[68] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 3

[69] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 2022. 3