PointCLIP V2: Adapting CLIP for Powerful 3D Open-world Learning

Xiangyang Zhu∗1, Renrui Zhang∗†2,3, Bowei He1, Ziyao Zeng2, Shanghang Zhang4, Peng Gao2

1City University of Hong Kong  2Shanghai AI Laboratory  3The Chinese University of Hong Kong  4Peking University

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

Abstract

Contrastive Language-Image Pre-training (CLIP) has shown promising open-world performance on 2D image tasks, while its transferred capacity on 3D point clouds, i.e., PointCLIP, is still far from satisfactory. In this work, we propose PointCLIP V2, a powerful 3D open-world learner, to fully unleash the potential of CLIP on 3D point cloud data. First, we introduce a realistic shape projection module to generate more realistic depth maps for CLIP’s visual encoder, which is quite efficient and narrows the domain gap between projected point clouds with natural images. Second, we leverage large-scale language models to automatically design a more descriptive 3D-semantic prompt for CLIP’s textual encoder, instead of the previous hand-crafted one. Without introducing 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. Furthermore, PointCLIP V2 can be extended to few-shot classification, zero-shot part segmentation, and zero-shot 3D object detection in a simple manner, demonstrating our superior generalization ability for 3D open-world learning. Code will be available at https://github.com/yangyangyang127/PointCLIP_V2.

1. Introduction

The advancement of spatial sensors and systems has stimulated widespread attention in recent years for both academia and industry, e.g., autonomous driving [7,18], indoor navigation [57], and stereo reconstruction [31,42]. To effectively understand point clouds, the major data form in 3D, many related tasks are put forward and gained great progress, including shape classification [34,50], scene segmentation [36,48,52], and 3D object detection [26,53]. 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.

Recently, the versatile CLIP [37] has been proposed for zero-shot 2D image classification, which is pre-trained by large-scale image-text pairs and obtains strong open-world recognition capacity. Inspired by this, PointCLIP [56], for the first time, indicates that CLIP can also be adapted for zero-shot point cloud classification without any 3D training. To bridge the modal gap between the 2D pre-trained CLIP and the 3D input, PointCLIP introduces two modules for visual and textual branches, respectively. The visual one projects the ‘unseen’ 3D point cloud sparsely into 2D depth maps, and the textual one modifies the general 2D prompt into handcrafted 3D descriptions. However, as a
preliminary work, the zero-shot classification performance of PointCLIP is far from satisfactory. As shown in Figure 1, on the widely adopted ModelNet40 [51] and ScanObjectNN [45] datasets, PointCLIP only achieves 23.78% and 21.34% classification accuracy, which cannot be put into actual use. Therefore, we ask the question: what actually restricts the performance of CLIP on point clouds and how to fully unleash it for 3D open-world understanding?

We observe that PointCLIP mainly suffers from two factors concerning the 2D-3D domain gap. (1) Sparse Visual Projection. PointCLIP simply projects 3D point clouds onto depth maps as sparsely distributed points with single 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 Textual Prompting. PointCLIP mostly inherits CLIP’s 2D prompt, “a photo of a [CLASS].” and appends simple domain-specific words, e.g., “a depth map”. As visualized in Figure 3, the textual features extracted by CLIP’s textual encoder can hardly focus on the target object with high similarity scores. Such naive 3D prompting cannot fully describe 3D shapes and harms the pre-trained language-image alignment in the embedding space.

To tackle this issue, we propose PointCLIP V2, a powerful zero-shot learner for 3D open-world understanding. Without ‘seeing’ any 3D training data, we effectively transfer the 2D pre-trained CLIP for zero-shot 3D classification, part segmentation, and object detection. Considering the aforementioned drawbacks in PointCLIP, our V2 further mitigates the 2D-3D domain gap from two aspects.

Firstly, to generate more CLIP-preferred images from 3D point clouds, an intuitive solution is to utilize more advanced projection methods. Besides PointCLIP’s perspective depth scatters [10], existing literature has exploited the Phong shading [43], height map [44], and silhouette map [44] fashions. However, these methods include complicated processing steps, which are quite computation-expensive and time-consuming, impeding their real-time application as compared in Table 1. To this end, we propose Realistic Shape Projection, which is quite efficient, i.e., 67× faster than Phong shading, and produces more realistic images than PointCLIP. Specifically, we transform the irregular point cloud into grid-based voxels with depth values and then apply a 3D local pooling along with a Gaussian filtering kernel 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, as the original prompt used in PointCLIP lacks detailed 3D-specific descriptions, we turn to the success of prompt engineering in natural language processing [16, 21, 22]. Motivated by automatic prompt designs [20, 33, 39], we adopt the large-scale language models (LLM), e.g., GPT-3 [2], to generate a prompt with rich 3D semantics, named as LLM-assisted 3D Prompting. By feeding customized commands into LLM, e.g., “Give a caption of a table depth map:”, we leverage its pre-trained language-generative knowledge to obtain a series of 3D-specific prompt, e.g., “A height map of a table with four legs.”. Also, we specialize the generated prompt for different shape categories, which better captures the unique characters of 3D shapes with distinctive spatial structures. As shown in Figure 3, the textual features of PointCLIP V2 exert stronger matching properties to the projected depth maps, largely preserving the pre-trained image-text alignment in 3D domains.

With our delicately improved projection and prompting schemes, PointCLIP V2 significantly surpasses PointCLIP for zero-shot 3D classification, i.e., +42.90%, +40.44%, and +28.75% accuracy, respectively on ModelNet10 [51], ModelNet40 [51], and ScanObjectNN [45] datasets. Further, our approach can also be adapted for more no-trivial 3D tasks in a zero-shot manner: part segmentation and 3D object detection. By appending a dense back-projection
head and a pre-trained 3D region proposal network, PointCLIP V2 exhibits strong zero-shot segmentation and detection performance, e.g., 48.4\% mIoU on ShapeNetPart [55] and 38.97\% AP$_{25}$ on ScanNet V2 [8]. This indicates the great potential of PointCLIP V2 for general 3D open-world understanding.

Our contributions are summarized as follows:

- We propose PointCLIP V2, an efficient cross-modal adaption method for CLIP to transfer the pre-trained 2D knowledge into 3D domains.
- We introduce realistic shape projection and LLM-assisted 3D prompting to effectively mitigate the 2D-3D domain gap.
- As the first work, our PointCLIP V2 can be extended for zero-shot 3D part segmentation and object detection without any 3D-domain training.

2. Related Works

3D Open-world Learning. Existing methods for 3D open-world learning mainly focus on classification tasks. The series efforts of Cheraghian et al. train zero-shot classifiers on ‘seen’ categories by maximizing inter-class divergence in latent space, and test on ‘unseen’ categories [4–6]. As the first attempt, PointCLIP [56] 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 [14] further improves the adaption performance of CLIP on point clouds by an additional 3D pre-training. In this paper, we propose PointCLIP V2 which follows the paradigm of PointCLIP, but significantly enhances its capacity for zero-shot 3D classification. Furthermore, some recent works [19, 23, 27, 29] investigate open-world semantic segmentation and 3D object detection for 3D scenes. Similar to Cheraghian et al., these methods all depend on 3D ‘seen’ data with segmentation and detection labels as shown in Figure 4. In contrast, our PointCLIP V2 still requires no ‘seen’ 3D training and, for the first time, directly conducts zero-shot 3D segmentation and detection, achieving complete 3D open-world understanding.

| Method                  | Latency | ModelNet40 | ScanObjectNN |
|-------------------------|---------|------------|--------------|
| Phong Shading [41]      | 107.2   | 57.30      | 29.33        |
| Height Map [44]         | 87.7    | 54.73      | 26.25        |
| Silhouette Map [44]     | 87.9    | 48.40      | 20.91        |
| PointCLIP [56]          | 1.2     | 42.53      | 26.37        |
| PointCLIP V2            | 1.6     | 64.22      | 35.36        |

Table 1. Comparison of Different Projection Methods. We report zero-shot classification results (%) on two datasets [45, 51], and compare the inference latency (ms) by projecting 10-view images from an input point cloud.

Point Cloud Projection. Concurrent to point-based 3D models [17, 25, 34, 36, 48, 50, 52], projection-based point cloud analysis aims to utilize plentiful 2D networks for 3D domains by projecting point clouds into 2D images for shape classification [1, 9, 11, 12, 24, 38, 40, 41, 43, 47, 49, 54]. Therein, Qi et al. [35] conduct spherical voxelization for point clouds and utilize Phong shading [32] to render different image views. Su et al. [44] find that the simple height map and binary silhouette generalize well without large-scale dataset training. Notably, PointCLIP [56] and SimpleView [10] conduct perspective transformation for 3D-to-2D projection, which achieves high efficiency with promising classification accuracy. Given the wide applications of view projection, we are inspired to develop more efficient and realistic projection methods for PointCLIP V2 under 3D open-world settings. 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 PointCLIP V2, namely, with our LLM-assisted 3D prompting to fully reveal their effectiveness. As shown, our realistic shape projection exhibits much faster inference speed than [35, 43, 44] and attains higher zero-shot performance than PointCLIP, indicating our superiority.

3. Methods

In this section, we first briefly revisit PointCLIP [56] for zero-shot 3D classification (Sec. 3.1). Then, we specifically introduce our proposed realistic shape projection (Sec. 3.2) and the LLM-assisted 3D prompting (Sec. 3.3). Finally, we implement our approach for various 3D open-world tasks (Sec. 3.4). The whole framework of PointCLIP V2 is shown in Figure 5.

3.1. A Revisit of PointCLIP

Inherited from CLIP [37], PointCLIP consists of two pre-trained encoders for visual and textual encoding, respectively. To bridge the modal gap, PointCLIP projects...
3D point clouds sparsely into depth maps as the visual input and customizes the general 2D prompt with 3D-related words as the textual input.

**Sparse Visual Projection.** Given an input point cloud, PointCLIP follows SimpleView [10] to conduct multi-view perspective projection. Specifically for each view, a point \( p \) with 3D coordinate \((x, y, z)\) is projected into a pixel \([x/z, y/z]\) on the 2D image plane, where \([\cdot]\) denotes the ceiling-integer operation. Then, the depth value \( z \) is regarded as the pixel intensity and repeated three times for RGB channels. Without further processing, the generated \( M \) depth maps, \( \{V_i\}_{i=1}^M \), are composed of scattering points with discontinuous pixel values. Then, PointCLIP feeds such sparse depth maps into CLIP’s visual encoder and obtains the global visual features \( \{f_i\}_{i=1}^M \) from \( M \) projection views, where \( f_i \in \mathbb{R}^{K \times C} \).

**Naive Textual Prompting.** Concurrently, the CLIP’s textual encoder takes as input a handcrafted prompt, e.g., “a depth map of a [CLASS].” and extracts the textual feature \( W_t \in \mathbb{R}^{K \times C} \), that is, the zero-shot classification weights. \( K \) denotes the number of classes and each column vector in \( W_t \) corresponds to a certain shape category. By using the word “a depth map”, PointCLIP expects the pre-trained textual encoder to capture more 3D-related clues in the sentences, which can better align with the projected depth maps. On top of this, 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

\[
\logits = \sum_{i=1}^M \alpha_i \cdot f_i W_t^T, \tag{1}
\]

where \( \alpha_i \) serves as a hyper-parameter weighing the importance of view \( i \).

**3.2. Realistic Shape Projection**

To generate more realistic depth maps from 3D point clouds and also achieve time efficiency, the projection in our PointCLIP V2 includes four steps: Voxelize, Densify, Smooth, and Squeeze, as shown in Figure 6.

**Voxelize.** For different \( M \) views, 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 represent 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 the point \( p = (x, y, z) \) into a voxel by

\[
G([sHx], [sWy], [Dz]) = z, \tag{2}
\]

where \( s \in (0, 1] \) 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 for the corresponding voxel. 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 denser and smoother spatial 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 variances, the filtering can not only removes 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 also repeat it three times as the RGB intensity. Compared to the 3D-to-2D perspective transformation in PointCLIP [56], our grid-based orthogonal projection is more friendly for hardware implementation.

### 3.3. LLM-assisted 3D Prompt

PointCLIP adopts prior handcrafted templates as the prompt fed into the textual encoder, which lacks 3D-specific descriptions of the projected depth maps and category-wise shape characteristics. Therefore, considering the powerful descriptive capacity of large-scale language models (LLMs), we leverage GPT-3 [2] to generate the textual prompt for CLIP with sufficient 3D semantics as shown in Figure 7. Normally, GPT-3 receives a customized language command and outputs a response via the pre-trained knowledge. To fully adapt GPT-3 to 3D domains, we propose the following four series of 3D-related language commands:

**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] is a rectangular shape with a flat top and four 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 [CLASS]."; GPT-3: "An monochrome depth map of an oblique [CLASS]."

**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, obscure."; GPT-3: "The obscure depth map shows a [table] at the corner."

For $K$ classes, we place their category names at the "[CLASS]" position to capture category-wise shape prop-
erties. Then, we feed all language commands into GPT-3 and acquire \( L \) generated prompt for each command type. By this, we obtain \( 4 \cdot L \) 3D-specific descriptions with rich semantics for each category and integrate them as the category-wise prompt for CLIP’s textual encoder.

### 3.4. Open-world Understanding

**Zero-shot Classification.** After cross-modal adaption by projection and prompting, we feed the multi-view depth maps and the LLM-assisted 3D prompt into CLIP’s encoders, which produces the visual features \( \{ f_i \}_{i=1}^M \) and textual feature \( W_i \). Then, we follow PointCLIP to conduct language-image alignment to obtain the final classification logits via Equation 1.

**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 6. 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 learn to summarize the 3D-domain knowledge from the few-shot dataset, and further adapt the 3D shape to be more CLIP-friendly. For a fair comparison, we follow PointCLIP to adopt a multi-view adapter and freeze the pre-trained encoders during training.

**Zero-shot Part Segmentation.** For a more general 3D understanding, we propose a zero-shot segmentation pipeline to enable both PointCLIP and our V2. For the projected views, different from shape classification that obtains the global features \( \{ f_i \}_{i=1}^M \), we extract the dense feature maps from CLIP’s visual encoder before its final pooling operation, and upsample the features into the original depth map size, denoted as \( \{ F_i \}_{i=1}^M \), where \( F_i \in \mathbb{R}^{H \times W \times C} \). Then, for view \( i \), we conduct dense alignment between each feature pixel and the textual feature \( W_i \), formulated as

\[
\logits_i = F_i W_i^T \in \mathbb{R}^{H \times W \times K}.
\] (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 and average them to acquire the part segmentation logits for each point. 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, our PointCLIP V2 can serve as a zero-shot classification head to recognize ‘unseen’ objects in the scene. Specifically, we leverage a pre-trained 3D region proposal network, e.g., 3DETR [28], to generate candidate 3D boxes. Then, the raw points within each box are fed into PointCLIP V2 for zero-shot classification.

### 4. Experiments

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

#### 4.1. Implementation Details

**Realistic Shape Projection.** We set the default size of grid \( G \) as \( H \times W \times D = 224 \times 224 \times 112 \), so the projected depth map is of size \( 224 \times 224 \). The point cloud is placed at the center of \( G \), and the scale factor \( s \) is set to 0.7 to guarantee better visual appearances. The window size of the minimum pooling for densifying is \( (10, 10, 5) \). The kernel size of the Gaussian filter is set as \((7, 7, 5)\). We follow PointCLIP [56] to take 1024 points as input and project the point cloud into 10 views. For the visual encoder, we adopt Vision Transformer [46] with patch size \( 16 \times 16 \) as default, denoted as ViT-B\(\setminus 16\).

**LLM-assisted Prompting.** The powerful language model, GPT-3 [2], is requested to produce \( L = 250 \) shape prompt for each command type. We use “text-davinci-002” GPT-3 engine and set the temperature constant to 0.99. The largest length of a 3D-specific prompt is set to 40. For the textual encoder, a 12-layer transformer is adopted to encode the prompt [37].

#### 4.2. Zero-shot Classification

**Settings.** The zero-shot classification performance is evaluated on three widely-used benchmarks: ModelNet10 [51], ModelNet40 [51] and ScanObjectNN [45]. Three splits of the ScanObjectNN dataset are investigated: OBJ_ONLY, OBJ_BG, and PB_T50_RS. Following the zero-shot manner, we directly test the classification performance on the full test set without involving the training set. We compare existing methods under their best settings. Specifically, ViT-B\(\setminus 16\) is adopted for both PointCLIP V2 and CLIP2Point [14]. For PointCLIP, we utilize ResNet-101 [15], ResNet-50×4 [37], and ViT-B\(\setminus 16\), 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 [3], and then uses it for a 3D zero-shot classification task. Cheraghian et al. [5] 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. PointCLIP 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 3, we conduct an ablation study of PointCLIP V2 concerning four steps of the realistic shape projection module. When we directly project the point cloud into 2D images via orthogonal projection, the zero-shot accuracy is 57.35%, reduced by −6.87%. If the voxelization is used, the densifying and smoothing operation can improve zero-shot performance by +15.14% and +5.7% from 44.50%, 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 4, we show the effect of four types of language commands in the 3D prompting module. Under different command combinations, the zero-shot performance is improved with various degrees. If all four types of commands are utilized, the 3D-specific prompt improves the zero-shot classification performance on ModelNet40 by +21.07%, indicating the great significance of better language-image alignment.

4.3. Few-shot Classification

Settings. We test k-shot classification performance on ModelNet40 [51] and ScanObjectNN [45] datasets, where $k \in \{1, 2, 4, 8, 16\}$. We adopt the same 3D-specific prompt used in the zero-shot task as textual input, and jointly train the inter-view adapter and the learnable smoothing (Figure 6). The 3D convolution layers adopt the $5 \times 5 \times 3$ kernel size and are followed by a batch normalization [15] and a nonlinear activation layer [30].

Main Results. In Figure 8, we show the few-shot classification results of PointCLIP V2 and compare it with PointCLIP and four representative 3D networks: PointNet [34], PointNet++ [36], SimpleView [10], and CurveNet [52]. As shown, PointCLIP V2 outperforms other methods in few-shot classification and shows a more obvious improvement on 1-shot classification. Compared with PointCLIP, V2 improves 1-shot accuracy by +12% on ModelNet40 and +7% on ScanObjectNN. In addition, PointCLIP V2 achieves a 16-shot accuracy of 89.55% on ModelNet40, approaching some fully supervised approaches [34].

Ablation Study. Table 7 shows the impact of different modules on PointCLIP V2 for 16-shot classification results, including the learnable smoothing, multi-view weighing following PointCLIP, and LLM-assisted prompting. We find that the learnable 3D projection module can improve...
Table 5. Zero-shot Part Segmentation (%) on ShapeNetPart [55]. We implement PointCLIP by our proposed segmentation pipeline.

| Method      | AP25 | AP50 |
|-------------|------|------|
| PointCLIP   | 31.0 | 48.4 |
| PointCLIP V2| 48.4 | 50.2 |

Table 6. Zero-shot 3D Object Detection (%) on ScanNet V2 [8]. We implement PointCLIP by our proposed detection pipeline.

Table 7. Ablation Study of Few-shot Learning on ModelNet40 [51]. We report the 16-shot classification accuracy (%).

4.4. Zero-shot Part Segmentation

**Settings.** We evaluate the zero-shot segmentation performance on ShapeNetPart dataset [55], which includes 16 categories and 50 annotated parts. Following prior fully trained methods [25,36,48], we randomly sample 2048 points from each point cloud, and adopt the official train/validation/test splits. We adopt the ViT-B/16 visual encoder and extract the feature maps from the last transformer block as feature map $F_i, i = 1, \ldots, M$. For comparison, we also 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 5. Our method achieves an overall mIoU$_I$ of 48.4%, surpassing PointCLIP by +17.4%, demonstrating the effectiveness of our PointCLIP V2 for capturing fine-grained 3D patterns in a zero-shot manner.

4.5. Zero-shot 3D Object Detection

**Settings.** The ScanNet V2 dataset [8] is utilized to evaluate our performance, which contains 18 object categories. We adopt the pre-trained 3DETR-m [28] model as the region proposal network to predict candidate boxes. We extract the points in each box and conduct zero-shot classification via PointCLIP V2. A “None” category is added to the prior 18 classes to measure the objectness score of each candidate 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.

**Main Results.** Table 6 shows our zero-shot 3D detection results. 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 PointCLIP V2 has great potential to recognize 3D open-world objects in real-world scenes.
5. Conclusion

We propose PointCLIP V2, a powerful 3D open-world learner, which improves the performance of PointCLIP with significant margins. We propose an efficient realistic shape projection module for high-quality depth maps, and adopt the LLM-assisted 3D prompt to achieve better alignment between visual and language representations. Besides classification, PointCLIP V2 also conducts zero-shot part segmentation and 3D object detection with promising performance. For future work, we will develop more advanced methods which adapt CLIP to wider open-world applications, e.g. outdoor 3D segmentation and detection.

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.

[2] 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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):722–739, 2021.

[8] 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.

[9] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. GVCNN: Group-view convolutional neural networks for 3D shape recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2018.

[10] 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.

[11] Haiyun Guo, Jingqiao Wang, Yue Gao, Jianqiang Li, and Hanqing Lu. Multi-view 3D object retrieval with deep embedding network. *IEEE Transactions on Image Processing*, 25(12):5526–5537, 2016.

[12] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D Wegner, Konrad Schindler, and Marc Pollefeys. Semantic3D.net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] 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.

[15] 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.

[16] Takeshi Kojima, Shixiang Shane Gu, Machiel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022.

[17] Ruifui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. PointAugment: An auto-augmentation framework for point cloud classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6378–6387, 2020.

[18] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3D object detection. In *European Conference on Computer Vision*, pages 641–656, 2018.

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

[20] Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*, 2021.

[21] Jiacheng Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? *arXiv preprint arXiv:2101.06804*, 2021.

[22] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroyuki 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.

[23] 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. *arXiv preprint arXiv:2207.01987*, 2022.

[24] Chao Ma, Yulan Guo, Jungang Yang, and Wei An. Learning multi-view representation with LSTM for 3D shape recognition and retrieval. *IEEE Transactions on Multimedia*, 21(5):1169–1182, 2018.

[25] 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.

[26] 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.

[27] Björn Michele, Alexandre Boulch, Gilles Puy, Maxime Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 38(5):1–12, 2019.

[28] Riccardo Roveri, Lukas Rahmann, Cengiz Oztireli, and Markus Gross. A network architecture for point cloud classification via automatic depth images generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4176–4184, 2018.

[29] Shibani Santurkar, Yann Dubois, Rohan Taori, Percy Liang, and Tatsunori Hashimoto. Is a caption worth a thousand images? A controlled study for representation learning. *arXiv preprint arXiv:2207.07635*, 2022.

[30] 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.

[31] Ishaan 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.

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

[33] 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.

[34] 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.

[35] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view CNNs for 3D object classification on 3D data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5648–5656, 2016.

[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.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021.

[38] Kripasindhu Sarkar, Basavaraj Hampiholi, Kiran Varanasi, and Dilie Stricker. Learning 3D shapes as multi-layered height-maps using 2D convolutional networks. In *European Conference on Computer Vision*, pages 74–89, 2018.

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

[40] 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.

[41] 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.

[42] 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.

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

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

[45] 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.

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

[47] 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 4176–4184, 2018.
[51] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Lingguang Zhang, Xiaou 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.

[52] 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.

[53] 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.

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

[55] 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.

[56] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunzhang 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.

[57] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In IEEE International Conference on Robotics and Automation, pages 3357–3364, 2017.