PointCLIP: Point Cloud Understanding by CLIP

Renrui Zhang*[1,3], Ziyu Guo*[2], Wei Zhang[1], Kunchang Li[1], Xupeng Miao[2]
Bin Cui[2], Yu Qiao[1], Peng Gao†[1], Hongsheng Li[3,4]
[1]Shanghai AI Laboratory
[2]School of CS and Key Lab of HCST, Peking University
[3]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
[4]Centre for Perceptual and Interactive Intelligence (CPII)
{zhangrenrui, gaopeng}@pjlab.org.cn hsli@ee.cuhk.edu.hk

Abstract

*Recently, zero-shot and few-shot learning via Contrastive Vision-Language Pre-training (CLIP) have shown inspirational performance on 2D visual recognition, which learns to match images with their corresponding texts in open-vocabulary settings. However, it remains under explored that whether CLIP, pre-trained by large-scale image-text pairs in 2D, can be generalized to 3D recognition. In this paper, we identify such a setting is feasible by proposing **PointCLIP**, which conducts alignment between CLIP-encoded point clouds and 3D category texts. Specifically, we encode a point cloud by projecting it onto multi-view depth maps and aggregate the view-wise zero-shot prediction in an end-to-end manner, which achieves efficient knowledge transfer from 2D to 3D. We further design an inter-view adapter to better extract the global feature and adaptively fuse the 3D few-shot knowledge into CLIP pre-trained in 2D. By just fine-tuning the adapter under few-shot settings, the performance of PointCLIP could be largely improved. In addition, we observe the knowledge complementary property between PointCLIP and classical 3D-supervised networks. Via simple ensemble during inference, PointCLIP contributes to favorable performance enhancement over state-of-the-art 3D networks. Therefore, PointCLIP is a promising alternative for effective 3D point cloud understanding under low data regime with marginal resource cost. We conduct thorough experiments on ModelNet10, ModelNet40 and ScanObjectNN to demonstrate the effectiveness of PointCLIP. Code is available at https://github.com/ZrrSkywalker/PointCLIP.*

1. Introduction

Deep learning has dominated computer vision tasks of both 2D and 3D domains in recent years, such as image

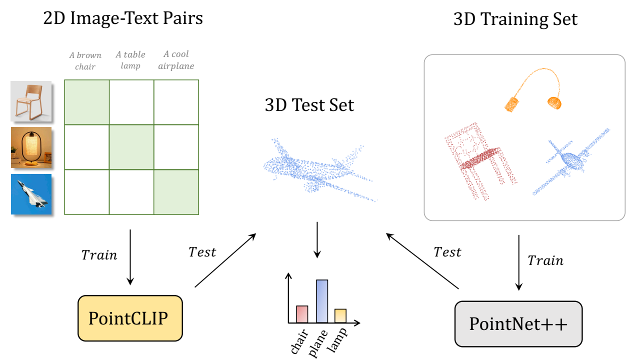

Figure 1. **Comparison of Training-testing Schemes between PointCLIP and PointNet++.** Different from classical 3D networks, our proposed PointCLIP is pre-trained by 2D image-text pairs and directly conducts zero-shot classification on 3D datasets without 3D training, which achieves efficient cross-modality knowledge transfer.

classification [12,17,22,28,37,41], object detection [1,4,13,29,47,67], semantic segmentation [3,25,35,36,64,68], point cloud recognition and part segmentation [19,42,44,45,56]. With 3D sensing technology developing rapidly, the growing demand for processing 3D point cloud data has boosted many advanced deep models with better local feature aggregator [30, 32, 50], geometry modeling [20, 40, 60] and projection-based processing [21, 34, 49]. Different from grid-based 2D image data, 3D point clouds suffer from space sparsity and irregular distribution, which hinder the direct transfer of methods from 2D domain. More importantly, a large number of newly captured point clouds contain objects of "unseen" categories to the deployed models. In this scenario, even the best-performing classifier might fail to recognize them and it is unaffordable to re-train the models every time when "unseen" objects arise.

Similar issues have been dramatically mitigated in 2D vision by Contrastive Vision-Language Pre-training

* Indicates equal contributions, † Indicates corresponding author

(CLIP) [46], which proposes to learn transferable visual features with natural language supervisions. For zero-shot classification of "unseen" categories, CLIP utilizes the pre-trained correlation between vision and language to conduct open-vocabulary recognition and achieves promising performance. To enhance the accuracy in few-shot settings, CoOp [69] adopts learnable tokens to encode the textual inputs and avoids the tuning for hand-crafted prompt. From another perspective, CLIP-Adapter [16] appends a lightweight residual-style adapter with two linear layers for better adapting image features and Tip-Adapter [66] further boosts its performance while greatly reduces the training time. Consequently, the problem of recognizing new unlabeled objects has been well explored on 2D images, and the proposed methods achieve significant improvements over zero-shot CLIP. However, for the more challenging point clouds, a question is naturally raised: Could such CLIP-based models be transferred to 3D domain and realize zero-shot classification for "unseen" 3D objects?

To address this issue, we propose **PointCLIP**, which transfers CLIP's 2D pre-trained knowledge to 3D point cloud understanding. The first concern is to bridge the modal gap between unordered point clouds and the grid-based images that CLIP can process. Considering the real-time need for some applications, such as autonomous driving [4, 13, 29, 43] and indoor navigation [71], we propose to adopt online perspective projection [19] without any post rendering [49], i.e., simply projecting raw points onto pre-defined image planes to generate scatter depth maps. The cost of this projection process is marginal in both time and computation, but reserves the original property of the point cloud from multiple views. On top of that, we apply CLIP's pre-trained visual encoder to extract multi-view features of the point cloud and then obtain each view's zero-shot prediction by the text-generated classifier. Therein, we place 3D category names into a hand-crafted template and produce the zero-shot classifier by CLIP's pre-trained textual encoder. As different views contribute differently to the understanding, we obtain the final prediction for the point cloud by weighted aggregation between views.

Although PointCLIP achieves cross-modality zero-shot classification without any 3D training, its performance still falls behind classical point cloud networks well-trained on full datasets. To eliminate this gap, we introduce a learnable inter-view adapter with bottleneck linear layers to better extract features from multiple views in few-shot settings. Specifically, we concatenate all views' features and summarize the compact global feature of the point cloud by cross-view interaction and fusion. Based on the global representation, the adapted feature of each view is generated and added to their original CLIP-encoded features via a residual connection. In this way, each view is aware of global information and also combines new knowledge from the 3D few-shot dataset with the 2D knowledge of pre-trained CLIP. During training, we only fine-tune this adapter and freeze both CLIP's visual and textual encoders to avoid over-fitting, since only a few samples per class are insufficient for training CLIP. By few-shot fine-tuning, PointCLIP with an inter-view adapter largely improves the zero-shot performance and exerts a good trade-off between performance and cost.

Additionally, we observe that CLIP's 2D knowledge, supervised by contrastive image-text pairs, is complementary to 3D close-set supervisions. PointCLIP with the inter-view adapter can be utilized to improve the performance of classical fully-trained 3D networks. For PointNet++ [45] with an accuracy of 89.71%, we adopt PointCLIP of 87.20% fine-tuned by 16-shot ModelNet40 [58] and directly ensemble their predicted classification logits during inference. The performance is enhanced by +2.32%, from 89.71% to 92.03%. Also for CurveNet [60], the state-of-the-art 3D recognition network, the knowledge ensemble contributes to performance boost from 93.84% to 94.08%. In contrast, simply ensemble between two models fully trained on ModelNet40 without PointCLIP cannot lead to performance improvement. Therefore, PointCLIP could be regarded as a drop-in multi-knowledge ensemble module, which promotes 3D networks via 2D contrastive knowledge with marginal few-shot training.

The contributions of our paper are as follows:

- We propose PointCLIP to extend CLIP for handling 3D point cloud data, which achieves cross-modality zero-shot recognition by transferring 2D pre-trained knowledge into 3D.
- An inter-view adapter is introduced upon PointCLIP via feature interaction among multiple views and largely improves the performance by few-shot fine-tuning.
- PointCLIP can be utilized as a multi-knowledge ensemble module to enhance the performance of existing fully-trained 3D networks.
- Comprehensive experiments are conducted on widely adapted ModelNet10, ModelNet40 and the challenging ScanObjectNN, which indicate PointCLIP's potential for effective 3D understanding.

2. Related Work

Zero-shot Learning in 3D. The objective of zero-shot learning is to enable the recognition of "unseen" objects, which are not adopted as training samples. Although zero-shot learning has drawn much attention on 2D classification [27, 46, 59], only a few works explore how to conduct it in 3D domain. As the first attempt on point clouds, [7] divides the 3D dataset into two parts consisting of "seen" and "unseen" samples, respectively. By leaning a projection function from point cloud feature space to the category semantic space, [7] trains PointNet [44] by the former and tests it on the latter. Based on this prior work, [5] further mitigates the hubness problem [65] resulted from low-quality 3D features and [6] introduces a triplet loss for better performance in transductive settings, which allows to utilize unlabeled "unseen" data for training. Different from all above settings, which train the network by part of 3D samples and predict on the others, PointCLIP only pre-trains from 2D data and achieves direct zero-shot recognition on "unseen" 3D samples without any 3D training. Thus, our setting is more challenging considering the domain gap from 2D to 3D and is more urgent for practical problems.

Transfer Learning. Transfer learning [9, 63] aims to utilize the knowledge from data-abundant domains to help with the learning on data-scarce domains. For general vision, ImageNet [9] pre-training can greatly benefit various downstream tasks, such as object detection [1, 18, 47] and semantic segmentation [35]. Also in natural language processing, representations pre-trained on web-crawled corpus via Mask Language Model [10] achieve leading performance on machine translation [39] and natural language inference [8]. Without any fine-tuning, the recently introduced CLIP [46] shows superior image understanding ability for "unseen" datasets. CoOp [69], CLIP-Adapter [16], Tip-Adapter [66] and so on [54, 57, 70] further indicate that the performance of CLIP can be largely improved by infusing domain-specific supervisions. Although the successes stories are encouraging, besides Image2Point [61], most of the existing methods conduct knowledge transfer within the same modality, namely, image to image [9], video to video [2] or language to language [10]. Different from them, our PointCLIP is able to efficiently transfer representations learned from 2D images to the disparate 3D point clouds, which motivates future research on transfer learning across different modalities.

Deep Neural Networks for Point Clouds. Existing deep neural networks for point clouds can be categorized into point-based and projection-based methods. Point-based models process on raw points without any pre-transformation. PointNet [44] and PointNet++ [45] firstly encode each point with a Multi-layer Perceptron (MLP) and utilize max pooling operation to ensure the permutation invariance. Recent point-based methods have proposed more advanced architecture designs along with geometry extractors [30, 50, 60] for better point cloud parsing. Other than raw points, projection-based methods understand point clouds by transferring them into volumetric [38] or multi-view [49] data forms. Therein, multi-view methods project point clouds onto images of multiple views and process them with 2D Convolution Neural Networks (CNN) [22] pre-trained on ImageNet [28], such as MVCNN [49] and others [14, 15, 21, 26, 62]. Normally, such view-projection methods operate on offline-generated images projected from 3D meshes [55] or require post-rendering [48] for shades and textures, which are costly and impractical to be adopted for real-time applications. On the contrary, we follow SimpleView [19] to naively project raw points onto image planes without processing and set their pixel values by the vertical distances. Such depth-map projection results in marginal time and computation costs, which meets the demand for efficient end-to-end zero-shot recognition.

3. Method

In Section 3.1, we first revisit Contrastive Vision-Language Pre-training (CLIP) for 2D zero-shot classification. Then in Section 3.2, we introduce our PointCLIP, which transfers 2D pre-trained knowledge into 3D point clouds. In Section 3.3, we provide an inter-view adapter for better few-shot performance. In Section 3.4, we propose to ensemble PointCLIP with fully-trained classical 3D networks for multi-knowledge complementation.

3.1. A Revisit of CLIP

CLIP is pre-trained to match images with their corresponding natural language descriptions. There are two independent encoders in CLIP for visual and textual features encoding, respectively. During training, given a batch of images and texts, CLIP extracts their features and learns to align them in the embedding space with a contrastive loss. To ensure comprehensive learning, 400 million training image-text pairs are collected from the internet, which enables CLIP to align images with any semantic concepts in an open vocabulary for zero-shot classification.

Specifically, for an "unseen" dataset of K classes, CLIP constructs the textual inputs by placing all category names into a pre-defined template, known as the prompt. Then, the zero-shot classifier is obtained by the C-dimensional textual features of category texts, the weights of which we denote as $W_t \in \mathbb{R}^{K\times C}$. Each of the K row vectors in W_t encodes the pre-trained category knowledge. Meanwhile, the feature of the test image is encoded by CLIP's visual encoder as $f_v \in \mathbb{R}^{1\times C}$ and the classification logits $\in \mathbb{R}^{1\times K}$ are computed as,

$$\mathrm{logits} = f_v W_t^T;\;\; p = \mathrm{SoftMax}(\mathrm{logits}), \tag{1}$$

where $\mathrm{SoftMax}(\cdot)$ and p denote the softmax function and the predicted probabilities for K categories. The whole process does not require any new training images and achieves promising zero-shot classification performance by the pre-trained encoders.

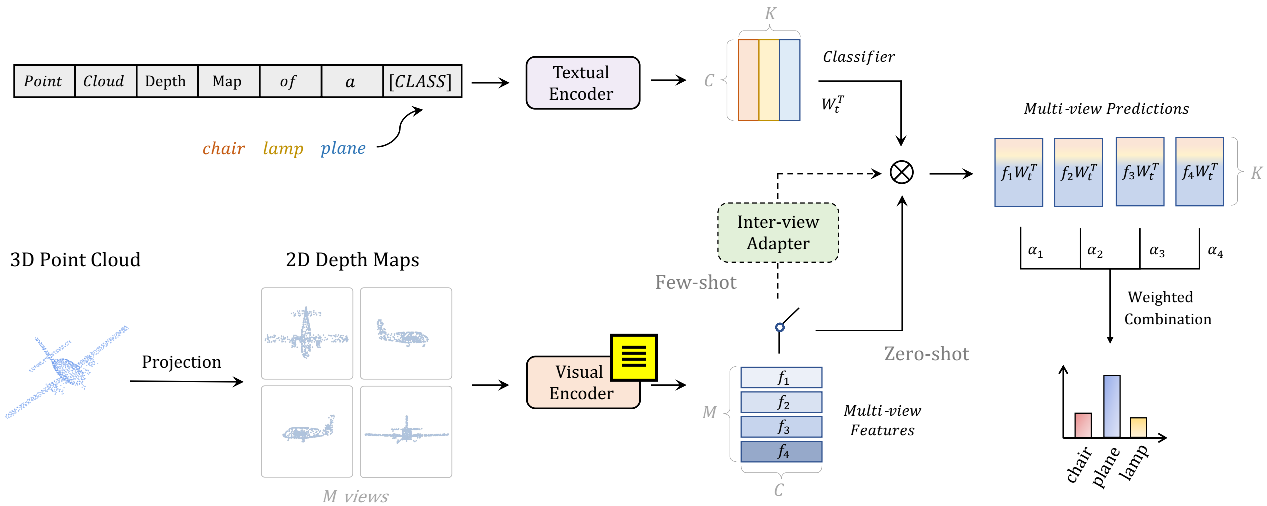

Figure 2. **The Pipeline of PointCLIP.** To bridge the modal gap, PointCLIP projects the point cloud onto multi-view depth maps and conducts 3D recognition via CLIP pre-trained in 2D. The switch provides alternatives for direct zero-shot classification and few-shot classification with the inter-view adapter, respectively in solid and dotted lines.

3.2. Point Cloud Understanding by CLIP

A variety of large-scale datesets [28, 31] in 2D provide abundant samples to pre-train models [11, 22] for extracting high-quality and robust 2D features. In contrast, the widely-adopted 3D datasets are comparatively much smaller and include limited object categories, e.g., ModelNet40 [58] with 9,843 samples and 40 classes v.s. ImageNet [28] with 1 million samples and 1,000 classes. Thus, it is very difficult to obtain well-performed pre-trained 3D networks for transfer learning. To alleviate this problem and explore the cross-modality power of CLIP, we propose PointCLIP to conduct zero-shot learning on point clouds based on the pre-trained CLIP.

Bridging the Modal Gap. Point cloud data is a set of unordered points scattering around the 3D space, whose sparsity and distribution greatly differ from grid-based 2D images. To convert point clouds into CLIP-accessible representations, we generate point-projected images from multiple views to eliminate the modal gap between 3D and 2D. In detail, if the coordinate of a point is denoted as (x, y, z), taking the bottom view as an example, its projected location on the image plane is $(\lceil x/z \rceil, \lceil y/z \rceil)$ following [19]. In this way, the projected point cloud is a foreshortened figure, namely, small in the distance but big on the contrary, which is more similar to that in real photos. Other than [19] applying one convolution layer to pre-process the one-channel depth map into a three-channel feature map, we do not adopt any pre-transformation and repeat the pixel values z for all three channels. Also, we apply no off-line processing [49, 55] and acquire projected depth maps directly from raw points without color information, which leads to marginal time and computation cost. With this lightweight cross-modality cohesion, CLIP's pre-trained knowledge can be then utilized for point cloud understanding.

Zero-shot Classification. Based on projected images from M views, we use CLIP to extract their visual features $\{f_i\}$, for $i = 1, \dots, M$ by the visual encoder. For the textual branch, we place K category names into the class token position of a pre-defined template: "point cloud depth map of a [CLASS]." and encode their textual features as the zero-shot classifier $W_t \in \mathbb{R}^{K\times C}$. On top of that, the classification logits_i of each view are separately calculated and the final logits of point cloud are acquired by their weighted summation,

$$\begin{aligned} \mathrm{logits}_i &= f_i W_t^T, \text{ for } \ i = 1, \dots, M, \\ \mathrm{logits} &= \sum\nolimits_{i=1}^{M} \alpha_i \mathrm{logits}_i, \end{aligned} \tag{2}$$

where α_i is a hyper-parameter weighing the importance of view i. Each view's f_i encodes a different perspective of the point cloud and is capable of independent zero-shot classification. Their aggregation further complements the information from different perspectives to achieve an overall understanding. The whole process of PointCLIP is non-parametric for the "unseen" 3D dataset, which pairs each point cloud with its category via CLIP's pre-trained 2D knowledge without any 3D training.

3.3. Inter-view Adapter for PointCLIP

Although PointCLIP achieves efficient zero-shot classification on point clouds, its performance is incomparable to those fully-trained 3D neural networks [44, 45]. We then

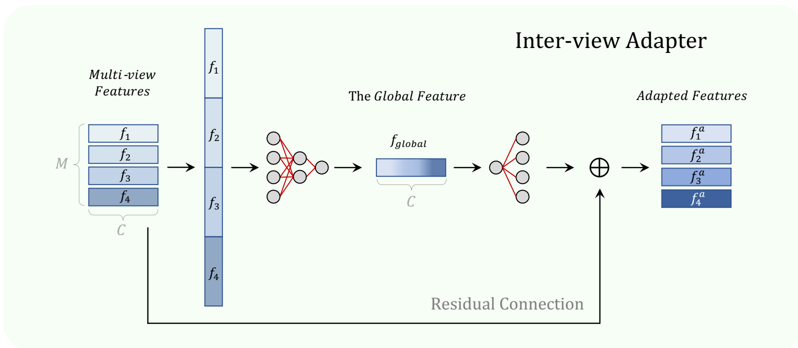

Figure 3. **Detailed Structure of Inter-view Adapter.** Given multi-view features of a point cloud, the adapter extracts its global representation and generates view-wise adapted features. Via a residual connection, the newly-learned 3D knowledge is fused into the pre-trained CLIP.

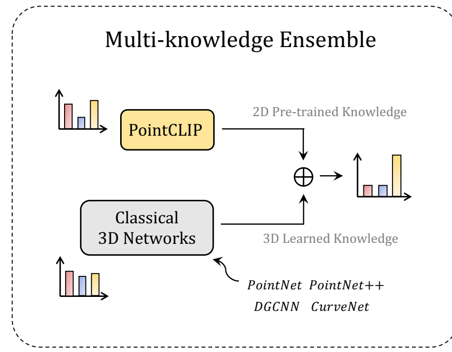

Figure 4. PointCLIP could provide **Complementary 2D Knowledge** to classical 3D networks and serve as a plug-and-play enhancement module.

consider a more common scenario where a few objects of each "unseen" category are contained in the newly collected data, and networks are required to recognize them under such few-shot settings. It is impractical to fine-tune the entire CLIP, since the enormous parameters and insufficient training samples would easily lead to over-fitting. Therefore, referring to [24] in Natural Language Processing (NLP) and CLIP-Adapter [16] for fine-tuning CLIP on downstream tasks, we append a three-layer Multi-layer Perceptron (MLP) on top of PointCLIP, named inter-view adapter, to further enhance its performance under few-shot settings. During training, we freeze both CLIP's visual and textual encoders and only fine-tune the learnable adapter via cross-entropy loss.

To be specific, given CLIP-encoded M-view features of a point cloud, we concatenate them along the channel dimension as $\text{Concate}(f_{1\sim M}) \in \mathbb{R}^{1\times MC}$, and then obtain the compact global representation via two linear layers of the inter-view adapter as

$$f_{\text{global}} = \text{ReLU}(\text{Concate}(f_{1\sim M})W_1^T)W_2^T, \quad (3)$$

where $f_{\text{global}} \in \mathbb{R}^{1\times C}$ and W_1, W_2 stand for two-layer weights in the adapter. By this inter-view aggregation, features from multiple perspectives are fused into a summative vector. Based on that, the view-wise adapted feature is generated from the global feature and added to its original CLIP-encoded feature via a residual connection as

$$f_i^a = f_i + \text{ReLU}(f_{\text{global}}W_{3i}^T), \quad (4)$$

where $W_{3i} \in \mathbb{R}^{C\times C}$ denotes the i-th part of W_3 for view i, and $W_3^T = [W_{31}^T; W_{32}^T; \cdots W_{3M}^T] \in \mathbb{R}^{C\times MC}$. The inter-view adapter exhibits two benefits: for one, f_i^a blends global-guided adapted feature with f_i for an overall understanding of the point cloud; for the other, the newly-learned 3D few-shot knowledge is infused into 2D pre-trained CLIP, which further promotes the cross-modality performance with 3D-specific supervisions.

After the inter-view adapter, each view conducts classification with the adapted feature f_i^a and the textual classifier W_t. Same as zero-shot classification, all M logits from M views are summarized to construct the final prediction. Surprisingly, just fine-tuning this additive adapter with few-shot samples contributes to significant performance improvement, e.g., from 20.18% to 87.20% on ModelNet40 [58] with 16 samples per category, less than 1/10 of the full data. This inspirational boost demonstrates the effectiveness and importance of feature adaption on 3D few-shot data, which greatly facilitates the knowledge transfer from 2D to 3D. Consequently, PointCLIP with inter-view adapter provides a promising alternative solution for point cloud understanding. Especially for some applications, where there is no condition to train the entire model by large-scale fully annotated data, just fine-tuning the three-layer adapter of PointCLIP with few-shot data can achieve competitive performance.

3.4. Multi-knowledge Ensemble

Classical point cloud networks, from the early PointNet [44] to the recent CurveNet [60], are trained from scratch on 3D datasets by close-set supervisions, but PointCLIP mostly inherits the pre-trained priors from 2D vision-language learning and contains a different aspect of knowledge. We then investigate if the two forms of knowledge can be ensembled together for better joint inference. In practice, we select two models: PointNet++ [45] and our PointCLIP under 16-shot fine-tuning, and directly ensemble their predicted logits by simple addition as the final output. Beyond our expectation, aided by PointCLIP's 87.20%, PointNet++ of 89.71% is enhanced to 92.03% with a significant improvement of +2.32%. In other words, the ensemble of two low-score models can produce a much stronger one, which fully demonstrates the complementary interaction of two kinds of knowledge. In contrast, ensemble between a pair of classical full-trained models would not bring perfor-

Zero-shot Performance of PointCLIP			
Datesets	Accuracy	Proj. Settings	View Weights
ModelNet10 [58]	30.23%	1.7, 100	2,5,7,10,5,6
ModelNet40 [58]	20.18%	1.6, 121	3,9,5,4,5,4
ScanObjectNN [52]	15.38%	1.8, 196	3,10,7,4,1,0

Table 1. Zero-shot Performance of PointCLIP on ModelNet10, ModelNet40 and ScanObjectNN with the best-performing settings. Proj. Settings include projection distances and the side length of depth maps.

View Numbers of Projection						
Numbers	1	4	**6**	8	**10**	12
Zero-shot	14.95	18.68	**20.18**	16.98	14.91	13.65
16-shot	75.53	82.17	84.24	85.48	**87.20**	86.35
Importance of Each View						
View	Front	**Right**	Back	**Left**	Top	Down
Zero-shot	18.64	**19.57**	18.92	19.12	17.46	17.63
16-shot	84.91	85.69	85.03	**85.76**	84.44	84.35

Table 2. Ablation studies (%) of projection view numbers and importance for zero-shot and 16-shot PointCLIP on ModelNet40.

mance boost, indicating the importance of complementarity. We further ensemble PointCLIP with other state-of-the-art 3D networks and observe similar performance boosts. Therefore, PointCLIP can be utilized as a plug-and-play enhancement module to achieve more robust point cloud recognition.

4. Experiments

4.1. Zero-shot Classification

Settings. We evaluate the zero-shot classification performance of PointCLIP on three well-known datasets: ModelNet10 [58], ModelNet40 [58] and ScanObjectNN [52]. For each dataset, we require no training data and adopt the full test set for evaluation. For the pre-trained CLIP model, we adopt ResNet-50 [22] as the visual encoder and the transformer [53] as the textual encoder by default. We then project the point cloud from 6 orthogonal views: front, right, back, left, top and bottom, and each view has a relative weight value ranging from 1 to 10, shown in the fourth column of Table 1. As the point coordinates are normalized from -1 to 1, we set the 6 image planes at a fixed distance away from the coordinate center $(0, 0)$. This distance is shown as the first value of Proj. Settings in Table 1, where the larger distance leads to the denser points distribution on the image. The side length of projected square depth maps varies to different datasets, which is presented as the second value in Proj. Settings, and the larger side length results in a smaller projected object size. We then upsample all images to $(224, 224)$ for alignment with CLIP's settings. For the zero-shot classifier from the textual encoder, we set the textual template as "point cloud depth map of a [CLASS]." to cater to the visual features of point clouds.

Prompts	Zero-shot	16-shot
"a photo of a [CLASS]."	17.02%	85.98%
"a point cloud photo of a [CLASS]."	16.41%	86.02%
"point cloud of a [CLASS]."	18.68%	86.06%
"point cloud of a big [CLASS]."	19.21%	**87.20**%
"point cloud depth map of a [CLASS]."	**20.18**%	85.82%
"[Learnable Tokens] + [CLASS]"	-	73.63%

Table 3. Performance of PointCLIP with different prompt designs on ModelNet40. [CLASS] denotes the class token, and [Learnable Tokens] denotes learnable prompts with fixed length.

Different Visual Encoders						
Models	RN50	**RN101**	ViT/32	ViT/16	RN.×4	**RN.×16**
Zero-shot	20.18	17.02	16.94	21.31	17.02	**23.78**
16-shot	85.09	**87.20**	83.83	85.37	85.58	85.90

Table 4. Performance (%) of PointCLIP with different visual encoders on ModelNet40. RN50 and ViT-B/32 denote ResNet-50 and vision transformer with 32 × 32 patch embeddings. RN.×16 denotes ResNet-50 with 16 times more computations from [46].

Performance. In Table 1, we present the performance of zero-shot PointCLIP on three datasets with their best-performing settings. Without any 3D training, PointCLIP is able to achieve a promising 30.23% on ModelNet10, which demonstrates our effective knowledge transfer from 2D to 3D. For ModelNet40 of 4 times the number of categories and ScanObjectNN with noisy real-world scenes, PointCLIP achieves slightly worse performance: 20.18% and 15.38%, respectively, due to the lack of 3D-specific downstream adaptions. As for the projection distances and image resolutions of Proj. Settings, their variances accord with the properties of different datasets. Compared to indoor ModelNet10, PointCLIP on ModelNet40 requires more details to recognize complex outdoor objects, such as airplanes and plants, and thus performs better with more scattered points and larger object size, namely, larger perspective projection distance and resolutions. In contrast, for ScanObjectNN, denser points and larger resolutions are required for filtering out the noise and reserving complex real-scene information. With respect to view weights, ModelNet10 and ModelNet40 of synthetic objects require all 6 views' contributions to the final classification with different importance, but for ScanObjectNN which contains noisy points of floors and ceilings, the top and bottom views could hardly provide any information.

Ablations. In Table 2, We conduct ablation studies of zero-shot PointCLIP concerning projection view numbers

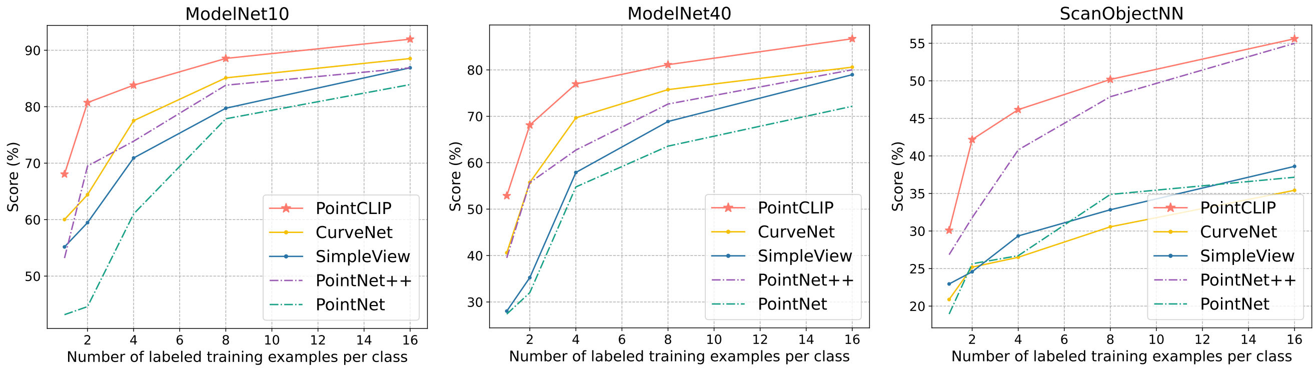

Figure 5. Few-shot performance comparison between PointCLIP and other classical 3D networks on ModelNet10, ModelNet40 and ScanObjectNN. Our PointCLIP shows consistent superiority to other models under 1, 2, 4, 8 and 16-shot settings.

and the importance of each view on ModelNet40. For the number of views, we try 1, 4, 6, 8, 10 and 12 views, for increasingly capturing the multi-view information of point clouds, but more than 6 views would bring redundancy and lead to performance decay. To explore how different views impact the performance, we unify all relative weights to 3 and respectively increase each view's weight to 9. As is shown in the table, projection from the right achieves the highest performance, which indicates its leading role, and both top and down views contribute relatively less to the classification. In Table 4, we implement different visual backbones including ResNet [22] and vision transformer [11], where RN50×16 [46] achieves the best performance of 23.78%.

Prompt Design. We present five prompt designs for zero-shot PointCLIP in Table 3. We observe that the naive "a photo of a [CLASS]." achieves 17.02% on ModelNet40 and simply inserting the word "point cloud" into it would hurt the performance. We then remove "a photo" and directly utilize "point cloud" as the subject, which benefits the accuracy by +1.66%. As the projected point cloud normally covers most of the image area, appending an adjective "big" could bring further performance improvement. Also, we add the "depth map" to describe the projected images more relevantly, which contributes to the best-performing 20.18%, demonstrating the importance of prompt choices.

4.2. Few-shot Classification

Settings. We experiment PointCLIP with the inter-view adapter under 1, 2, 4, 8, 16 shots also on ModelNet10 [58], ModelNet40 [58] and ScanObjectNN [52]. For N-shot settings, we randomly sample N point clouds from each category of the training set. Considering both efficiency and performance, we adopt ResNet-101 [22] as CLIP's pre-trained visual encoder for stronger feature extraction and increase the projected view numbers to 10, adding the views of upper/bottom-front/back-left corners, since the left view is proven to be the most informative for few-shot recognition in Table 2. In addition, we modify the prompt to "point cloud of a big [CLASS].", which performs better in the few-shot experiments. For the inter-view adapter, we construct a residual-style Multi-layer Perceptron (MLP) consisting of three linear layers, as described in Section 3.3.

Performance. In Figure 5, we present the few-shot performance of PointCLIP and compare it with 4 representative 3D networks: PointNet [44], PointNet++ [45], SimpleView [19] and the state-of-the-art CurveNet [60]. As we can see, PointCLIP with inter-view adapter surpasses all other methods for the few-shot classification. When there are only a small number of samples per category, PointCLIP has distinct advantages, exceeding PointNet by 25.49% and CurveNet by 12.29% on ModelNet40 with 1 shot. When given more training samples, PointCLIP still leads the performance, but the gap becomes smaller due to the frozen encoders and limited fitting capacity of the only three-layer adapter.

Ablations. In Table 2, we show the 16-shot PointCLIP under different projection views and explore how each view contributes to ModelNet40. Differing from the zero-shot version, 10 views of 16-shot PointCLIP performs better than 6 views, probably because the newly-added adapter is able to better utilize the information from more views and adaptively aggregate them. For the importance of views, we follow the configurations of our zero-shot experiments but observe the reversed conclusion: the left view is the most informative one. For different visual encoders in Table 4, ResNet-101 achieves the highest accuracy with less parameters than vision transformer or ResNet-50×16. Table 3 lists the performance influences caused by prompt designs. The learnable prompt following CoOp [69] performs worse than hand-crafted designs and the "point cloud of a big [CLASS]." performs the best.

Models	Before En.	After En.	Gain	Ratio
PointNet [44]	88.78	90.76	+1.98	0.60
PointNet++ [45]	89.71	92.10	+2.39	0.70
RSCNN [33]	92.22	92.59	+0.37	0.70
DGCNN [56]	92.63	92.83	+0.20	0.70
SimpleView [19]	93.23	93.87	+0.64	0.60
CurveNet [60]	93.84	**94.08**	+0.24	0.15

Table 5. The enhancement (%) of multi-knowledge ensemble by 16-shot PointCLIP, which achieves 87.20% on ModelNet40. Before and After En. denote models with and without PointCLIP's ensemble.

4.3. Multi-knowledge Ensemble

Settings. To verify the complementarity of blending pre-trained 2D priors with 3D knowledge, we aggregate the fine-tuned 16-shot PointCLIP of 87.20% on ModelNet40 with the fully-trained PointNet [44], PointNet++ [45], DGCNN [56], SimpleView [19] and CurveNet [60], respectively. All checkpoints of other models are obtained from [23, 51] without any voting. We manually modulate the fusion ratio between PointCLIP and each model, and report the performance with the best Ratio in Table 5, which represents PointCLIP's relative weight to the whole.

Performance. As shown in Table 5, the ensemble with PointCLIP improves the performance of all classical fully-trained 3D networks. The results fully demonstrate the complementarity of PointCLIP to existing 3D models. It is worth noting that the performance gain is not simply achieved by the ensemble between two models, because the accuracy of 16-shot PointCLIP is lower than other fully-trained models, but could still benefit their already-high performance to be higher. Therein, the largest accuracy improvement is on PointNet++ from 89.71% to 92.10%, and combining PointCLIP with the state-of-the-art CurveNet achieves the best 94.08%. Also, we observe that, for models with lower baseline performance, PointCLIP's logits need to account for a larger proportion, but for the well-performing ones, such as CurveNet, their knowledge is supposed to play a dominant role in the ensemble.

Ablations. We conduct ablation studies of the ensemble of two models fully trained on ModelNet40 without PointCLIP, and fuse their logits with the same ratio for simplicity. As is presented in Table 6, aggregating PointNet++ lowers the performance of RSCNN and CurveNet, and the ensemble between the highest two models, SimpleView and CurveNet, could not achieve better performance. Also, the paired ensemble of PointCLIP would hurt the original performance. Hence, simple ensemble of two models with the same training schemes normally leads to performance degradation, which demonstrates the significance of multi-

En. Model 1		En. Model 2	After En.
PointNet++ [45], 89.71	+	RSCNN [33], 92.22	92.14
PointNet++, 89.71	+	CurveNet [60], 93.84	91.61
SimpleView [19], 93.23	+	CurveNet, 93.84	93.68
PointCLIP, 87.20	+	PointCLIP, 87.14	87.06

Table 6. Ablation studies (%) of ensemble between models with the same training schemes.

Ensemble with CurveNet [60]						
Shots	0	8	16	32	64	128
PointCLIP	20.18	81.96	87.20	87.83	88.95	**90.02**
After En.	93.88	93.89	**94.08**	94.00	93.92	93.88

Table 7. Enhancement performance (%) of PointCLIP under different few-shot settings for CurveNet on ModelNet40.

knowledge interaction. In Table 7, we fuse PointCLIP fine-tuned by zero-shot, 8, 16, 32, 64 and 128 shots, respectively with CurveNet to explore their enhancement abilities. As reported, zero-shot PointCLIP with only 20.18% could promote CurveNet by +0.04%. However, too much training on 3D datasets would adversely influence the ensemble accuracy. This is possibly caused by the over-much knowledge similarity between two models, which cannot provide complementary information as expected.

5. Conclusion

We propose PointCLIP, which conducts cross-modality zero-shot recognition on point clouds without any 3D training. Via multi-view projection, PointCLIP efficiently transfers CLIP's pre-trained 2D knowledge into the 3D domain. Furthermore, we design an inter-view adapter to aggregate multi-view features and fuse the 3D learned knowledge into pre-trained CLIP under few-shot settings. By fine-tuning the adapter and freezing all other modules, the performance of PointCLIP is largely improved. In addition, PointCLIP could serve as a plug-and-play module to provide complementary knowledge for the classical 3D networks, which leads to favorable performance boost. Besides recognition, our future work will focus on generalizing CLIP for wider 3D applications.

Acknowledgement

This work is supported in part by Centre for Perceptual and Interactive Intelligence Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants (Nos. 14204021, 14207319), in part by CUHK Strategic Fund, in part by the Shanghai Committee of Science and Technology, China (Grant No.21DZ1100100).

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1, 3

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 3

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 1, 2

[5] 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

[6] Ali Cheraghian, Shafinn Rahman, Townim F Chowdhury, Dylan Campbell, and Lars Petersson. Zero-shot learning on 3d point cloud objects and beyond. *arXiv preprint arXiv:2104.04980*, 2021. 3

[7] Ali Cheraghian, Shafin Rahman, and Lars Petersson. Zero-shot learning of 3d point cloud objects. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6. IEEE, 2019. 2, 3

[8] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017. 3

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

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

[11] 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. 4, 7

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[13] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017. 1, 2

[14] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019. 3

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

[16] 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. 2, 3, 5

[17] Peng Gao, Jiasen Lu, Hongsheng Li, Roozbeh Mottaghi, and Aniruddha Kembhavi. Container: Context aggregation network. *arXiv preprint arXiv:2106.01401*, 2021. 1

[18] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. *arXiv preprint arXiv:2101.07448*, 2021. 3

[19] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. *arXiv preprint arXiv:2106.05304*, 2021. 1, 2, 3, 4, 7, 8

[20] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 1

[21] Abdullah Hamdi, Silvio Giancola, Bing Li, Ali K. Thabet, and Bernard Ghanem. MVTN: multi-view transformation network for 3d shape recognition. *CoRR*, abs/2011.13244, 2020. 1, 3

[22] 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. 1, 3, 4, 6, 7

[23] Ankit Goyal Hei Law. Simpleview. `https://github.com/princeton-vl/SimpleView`, 2021. 8

[24] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019. 5

[25] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. 1

[26] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019, 2018. 3

[27] Nour Karessli, Zeynep Akata, Bernt Schiele, and Andreas Bulling. Gaze embeddings for zero-shot image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4525–4534, 2017. 2

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1, 3, 4

[29] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 1, 2

[30] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31:820–830, 2018. 1, 3

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[32] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. Meteornet: Deep learning on dynamic 3d point cloud sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9246–9255, 2019. 1

[33] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 8

[34] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *arXiv preprint arXiv:1907.03739*, 2019. 1

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1, 3

[36] Xianzheng Ma, Zhixiang Wang, Yacheng Zhan, Yinqiang Zheng, Zheng Wang, Dengxin Dai, and Chia-Wen Lin. Both style and fog matter: Cumulative domain adaptation for semantic foggy scene understanding. *arXiv preprint arXiv:2112.00484*, 2021. 1

[37] Mingyuan Mao, Renrui Zhang, Honghui Zheng, Peng Gao, Teli Ma, Yan Peng, Errui Ding, Baochang Zhang, and Shumin Han. Dual-stream network for visual recognition. *arXiv preprint arXiv:2105.14734*, 2021. 1

[38] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 3

[39] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. *arXiv preprint arXiv:1708.00107*, 2017. 3

[40] Guanghua Pan, Jun Wang, Rendong Ying, and Peilin Liu. 3dti-net: Learn inner transform invariant 3d geometry features using dynamic gcn. *arXiv preprint arXiv:1812.06254*, 2018. 1

[41] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 1

[42] Haotian Peng, Bin Zhou, Liyuan Yin, Kan Guo, and Qinping Zhao. Semantic part segmentation of single-view point cloud. *Science China Information Sciences*, 63(12):224101, 2020. 1

[43] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2

[44] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 3, 4, 5, 7, 8

[45] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 1, 2, 3, 4, 5, 7, 8

[46] 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. *arXiv preprint arXiv:2103.00020*, 2021. 2, 3, 6, 7

[47] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1, 3

[48] Kripasindhu Sarkar, Basavaraj Hampiholi, Kiran Varanasi, and Didier Stricker. Learning 3d shapes as multi-layered height-maps using 2d convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–86, 2018. 3

[49] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 1, 2, 3, 4

[50] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. 1, 3

[51] Yuchen Li Tiange Xiang. curvenet. `https://github.com/tiangexiang/CurveNet`, 2021. 8

[52] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1588–1597, 2019. 6, 7

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 6

[54] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 3

[55] Pengyu Wang, Yuan Gan, Panpan Shui, Fenggen Yu, Yan Zhang, Songle Chen, and Zhengxing Sun. 3d shape segmentation via shape fully convolutional networks. *Computers & Graphics*, 76:182–192, 2018. 3, 4

[56] 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 (tog)*, 38(5):1–12, 2019. 1, 8

[57] Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021. 3

[58] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2, 4, 5, 6, 7

[59] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 69–77, 2016. 2

[60] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. *arXiv preprint arXiv:2105.01288*, 2021. 1, 2, 3, 5, 7, 8

[61] Chenfeng Xu, Shijia Yang, Bohan Zhai, Bichen Wu, Xiangyu Yue, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Image2point: 3d point-cloud understanding with pretrained 2d convnets. *arXiv preprint arXiv:2106.04180*, 2021. 3

[62] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. 3

[63] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018. 3

[64] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018. 1

[65] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2021–2030, 2017. 3

[66] 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. 2, 3

[67] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020. 1

[68] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021. 1

[69] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021. 2, 3, 7

[70] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. *arXiv preprint arXiv:2203.05557*, 2022. 3

[71] 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 *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017. 2