PointCLIP: Point Cloud Understanding by CLIP

Renrui Zhang*[1], Ziyu Guo*[2], Wei Zhang[1], Kunchang Li[1], Xupeng Miao[2]
Bin Cui[2], Yu Qiao[1], Peng Gao†[1], Hongsheng Li†[3]
[1]Shanghai AI Laboratory [2]Peking University
[3]The Chinese University of Hong Kong
{zhangrenrui, gaopeng, qiaoyu}@pjlab.org.cn
2101210573@pku.edu.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 cloud and 3D category texts. Specifically, we encode a point cloud by projecting it into multi-view depth maps without rendering, and aggregate the view-wise zero-shot prediction to achieve knowledge transfer from 2D to 3D. On top of that, we design an inter-view adapter to better extract the global feature and adaptively fuse the few-shot knowledge learned from 3D into CLIP pre-trained in 2D. By just fine-tuning the lightweight adapter in the few-shot settings, the performance of PointCLIP could be largely improved. In addition, we observe the complementary property between PointCLIP and classical 3D-supervised networks. By simple ensembling, PointCLIP boosts baseline's performance and even surpasses state-of-the-art models. Therefore, PointCLIP is a promising alternative for effective 3D point cloud understanding via CLIP under low resource cost and data regime. We conduct thorough experiments on widely-adopted ModelNet10, ModelNet40 and the challenging ScanObjectNN to demonstrate the effectiveness of PointCLIP. The code is released 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 classification [12, 17, 21, 28, 36, 41], object detection [1, 4, 13, 29, 46, 64], semantic segmentation [3, 24, 35, 61, 65], point cloud recognition and part segmentation [19, 43, 44, 55]. 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, 49], geometry modeling [20, 39, 40] and projection-based processing [20, 34, 48]. Different from grid-based 2D image data, 3D point clouds suffer from space sparsity and irregular distribution, which hinder direct methods transfer from 2D domain. Additionally, large-scale newly captured point cloud data contain a large number of objects of "unseen" categories to the trained classifier. In this scenario, even the best-performing models might fail to recognize them and it is unaffordable to re-train every time when "unseen" objects arise.

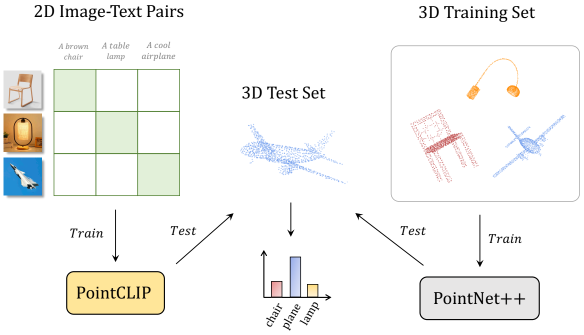

Figure 1. **A Comparison of Training-testing schemes between PointCLIP and PointNet++.** Different from classical 3D networks, our proposed PointCLIP is pre-trained by 2D image-text pairs, but conducts zero-shot classification on 3D datasets, which achieves cross-modality knowledge transfer.

Similar issues have been dramatically mitigated in 2D vision by Contrastive Vision-Language Pre-training (CLIP) [45], which proposed 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

* Equal contribution. † Corresponding author.

open-vocabulary recognition and achieves promising performance. To further enhance the accuracy in few-shot settings, CoOp [66] adopted learnable tokens to encode the text prompts, so that the classifier weights can be adaptively formed. From another perspective, CLIP-Adapter [16] appends a lightweight residual-style adapter with two linear layers for better adapting image features. Tip-Adapter [63] further boosts its performance while greatly reduces the training time. Both methods achieve significant improvements over zero-shot CLIP. Consequently, the problem of recognizing new unlabeled objects has been explored by CLIP in 2D. However, a question is naturally arised: 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 could process. Considering the need for real-time prediction in various scenarios, such as autonomous driving [4, 13, 29, 42] and indoor navigation [67], we propose to adopt online perspective projection [19] without any post rendering [48], i.e., simply projecting each point onto a series of 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 to encode multi-view features of point cloud by the CLIP pre-trained visual encoder and obtain each view's text-matched prediction independently via zero-shot classifier. Following CLIP, we place 3D category names into a hand-crafted template as prompts and generate the zero-shot classifier by CLIP's textual encoder. As different views contribute differently to the recognition of entire scene, we obtain the final prediction for 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 extract the compact global feature of the point cloud via interacting and summarizing cross-view information. Based on the global representation, adapted feature of each view is generated and added to their original CLIP-encoded feature via a residual connection. In this way, each view is equipped with the fused global feature and also combines newly adapted feature from the 3D few-shot dataset with 2D pre-trained CLIP's encoding. During training, we only fine-tune this lightweight adapter and freeze CLIP's both visual and textual encoders to avoid over-fitting, since only a few samples per class are given. Surprisingly, PointCLIP with an inter-view adapter with few-shot fine-tuning achieves comparable performance with some previous models well-trained with full datasets, which is a good trade-off between performance and cost.

Additionally, we observe that CLIP's 2D knowledge, supervised by contrastive loss, is complementary to the close-set 3D supervisions. The PointCLIP with an inter-view adapter can be fine-tuned under few-shot settings to improve the performance of classical fully-trained 3D networks. Taking PointCLIP in 16-shot ModelNet40 [57] and fully-trained PointNet++ [44] as an example, we directly ensemble their predicted logits for testing. Surprisingly, the performance of PointNet++'s 89.71%, is enhanced to 92.03% by PointCLIP with an accuracy of 87.20%. Furthermore, we select CurveNet [39], the state-of-the-art 3D recognition model, as the ensembling baseline, and achieve performance boost from 93.84% to 94.08%. In contrast, simply ensembling two models fully trained on ModelNet40 without PointCLIP only leads to performance loss. Therefore, PointCLIP could be regraded as a multi-knowledge ensembling module, which promotes 3D networks via 2D contrastive knowledge with limited additional 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 improves the performance of few-shot fine-tuning.
- PointCLIP can be utilized as a multi-knowledge ensembling module for enhancing performance of existing fully-trained 3D networks, which surpasses state-of-the-art performances.
- Comprehensive experiments are conducted on widely adapted ModelNet10, ModelNet40 and the challenging ScanObjectNN, which indicate PointCLIP's potential for 3D understanding.

2. Related Work

Zero-shot Learning in 3D. The objective of zero-shot learning is to enable recognition of "unseen" objects which are not adopted during training. Although zero-shot learning has drown much attention on 2D classification [26, 45, 58], only a few works explore how to conduct it in 3D domain. As the first attempt on point cloud, [7] divides the 3D dataset into two parts: "seen" and "unseen" samples, and trains PointNet [43] on the former but tests on the latter by measuring cosine similarities with category semantics. Based on this prior work, [5] further mitigates the hubness problem [62] resulted from low-quality extracted 3D features and [6] introduces a triplet loss for better performance in transductive settings, which allows to utilize unlabeled "unseen" data at training time. Different from all above settings, which train the network on part of the 3D samples and predict on the others, PointCLIP achieves direct zero-shot recognition without any 3D training and conducts prediction on the whole point cloud datasets. Thus, our setting is more challenging for the domain gap between 2D pre-training and 3D application, but more urgent for practical problems.

Transfer Learning. Transfer learning [9, 60] aims to utilize the knowledge from data-abundant domains to help the learning on data-scarce domains. For general vision, ImageNet [9] pre-training can greatly assist downstream tasks, such as object detection [1, 18, 46] and semantic segmentation [35]. Also in natural language processing, representations pre-trained on web-crawled corpus via Mask Language Model [10] achieves leading performance on machine translation [38] and natural language inference [8]. Without any fine-tuning, the recently introduced CLIP [45] shows superior image understanding ability for "unseen" datasets. CLIP-Adapter [16], Tip-Adapter [63], ActionCLIP [53] and WiSE-FT [56] further indicate that the performance of CLIP can be largely improved by infusing domain-specific supervisions. Although the successes stories are encouraging, 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 researches on transfer learning across different modalities.

Deep Neural Networks for Point Cloud. Existing deep neural networks for point cloud can be divided into point-based and projection-based methods. Point-based models process on raw points without any pre-transformation. PointNet [43] and PointNet++ [44] firstly encode each point with a Multi-layer Perceptron (MLP) and utilize max pooling operation to realize permutation invariance. Recent point-based methods propose more advanced local aggregators and architecture designs [30,49]. Other than raw points, projection-based methods understand point cloud by transferring it to volumetric [37] or multi-view [48] data forms. Therein, multi-view methods project point cloud into images of multiple views and process them with 2D Convolution Neural Networks (CNN) [21] pre-trained on ImageNet [28], such as MVCNN [48] and others [14,15,25,59]. Normally, such view-projected methods operate on offline-generated images which are projected from point-converted 3D meshes [54] or required post-rendering [47] for shades and textures, so they 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 and set their pixel values according to the vertical distance. Such depth-map generation 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. In Section 3.3, we provide PointCLIP with inter-view adapter for better performance under few-shot settings. In Section 3.4, we propose to ensemble PointCLIP with fully-trained classic 3D networks for multi-knowledge ensembling, which can achieve state-of-the-art performance.

3.1. A Revisit of CLIP

CLIP is trained to match images with their corresponding natural language descriptions. There are two independent encoders in CLIP, respectively for visual and textual features encoding. 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 prompt. Then, the zero-shot classifier, denoted as $W_t \in \mathbb{R}^{K\times C}$, is obtained by the C-dimensional textual feature of category prompts. Each of the K row vectors in W_t encodes the pre-trained category weights. Meanwhile, the feature of every test image is encoded by CLIP's visual encoder to $f_v \in \mathbb{R}^{1\times C}$ and the classification logits $\in \mathbb{R}^{1\times K}$ are computed as,

$$\mathrm{logits} = f_v W_t^T; \;\; p_i = \mathrm{softmax}_i(\mathrm{logits}), \qquad (1)$$

where $\mathrm{softmax}_i(\cdot)$ and p_i denote the softmax function and predicted probability for category i. The whole process does not require new training images, but achieves promising zero-shot classification performance only by frozen pre-trained encoders.

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, 21] for high-quality and robust 2D features extraction. In contrast,

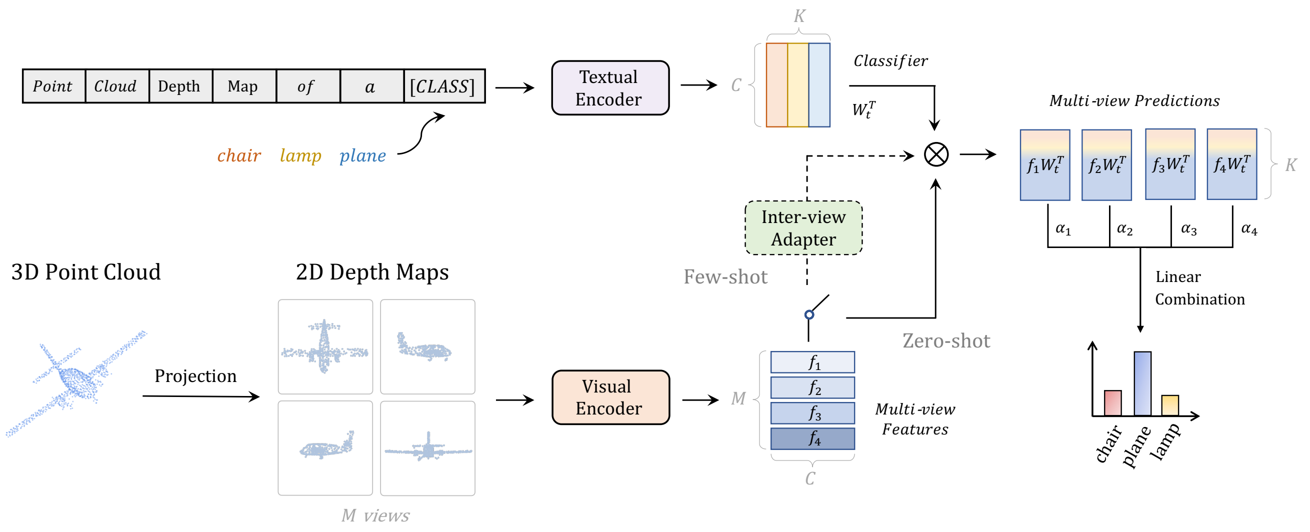

Figure 2. **The Pipeline of PointCLIP.** To bridge the modal gap, PointCLIP projects the point cloud onto multi-view depth maps, and conducts 3D recognition via CLIP pre-trained in 2D. The switch provides alternatives for direct zero-shot classification and few-shot classification with inter-view adapter,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

the widely-adopted 3D datasets are comparatively much smaller and have limited categories, e.g. ModelNet40 [57] with 9,843 samples and 40 classes vs. ImageNet [28] with 1 million samples and 1,000 classes. Thus, it is very difficult to obtain good 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 is a set of unordered points scattering in the 3D space, its 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) in the 3D space, taking the bottom projection view as an example, its 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 convolution layers to pre-processing the one-channel depth map into three, we do not adopt any pre-convolution and directly set the pixel value equaling to z in all three channels. Also, different from other off-line projection methods, whose projected images are generated from meshes [54] or CAD models [48], our projected depth maps are from raw points and contain no color information but scattered depth values, 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 $i = 1, \dots, M$. For the textual branch, we place K category names to 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, classification logits_i of each view are separately calculated and the final logits_p of point cloud are acquired by their weighted summation,

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

where α_i is a hyper-parameter weighing the importance of view i. Each view f_i encodes a different perspective of the point cloud feature, which is capable for independent zero-shot classification. Their summation further complements the information of different perspectives to obtain 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 and without any 3D training.

3.3. Inter-view Adapter for PointCLIP

Although PointCLIP achieves efficient zero-shot classification on point clouds, its performance is still incomparable to fully-trained 3D neural networks [43, 44]. We then 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

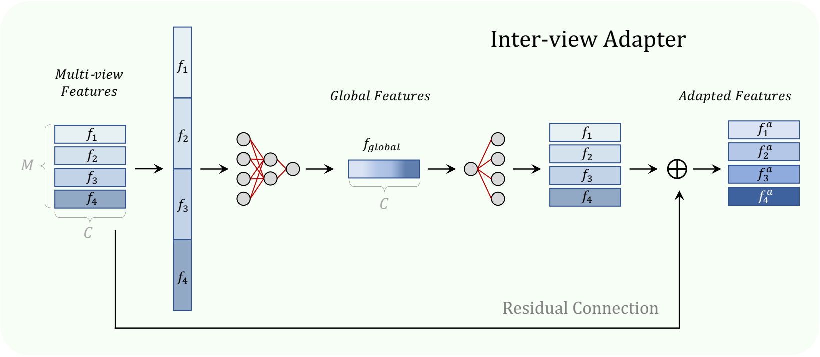

Figure 3. Detailed structure of the proposed **Inter-view Adapter.** Given multi-view features of a point cloud, the adapter extracts its global representation and generates view-wise adapted features. Via a residual connection, the newly-learned 3D knowledge is fused into the pre-trained CLIP.

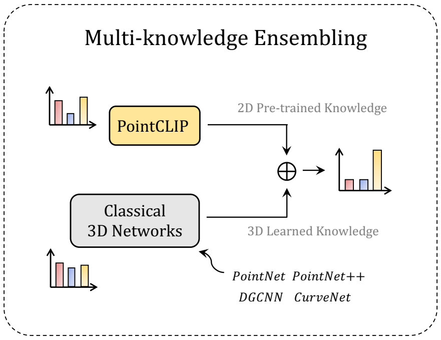

Figure 4. PointCLIP could provide complimentary 2D knowledge to classical 3D networks and serve as a plug-and-play enhancement module.

such few-shot settings. It is impractical to fine-tune the whole model, since the enormous parameters and insufficient samples would easily result in over-fitting. Therefore, referring to [23] in Natural Language Processing (NLP) and CLIP-Adapter [16] for fine-tuning pre-trained models 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. For training, we freeze CLIP's both visual and textual encoders and fine-tune the learnable adapter via a 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 acquire the compact global feature of point cloud via the first two layers of the inter-view adapter as

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

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 fuse into a summative representation. After 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

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

where $W_{3i} \in \mathbb{R}^{C\times C}$ denotes the i-th part of W_3 for view i, and $W_3^T = [W_{31}^T; W_{32}^T; \cdots W_{3M}^T] \in \mathbb{R}^{C\times MC}$. On the one hand, f_i^a blends global-guided adapted feature into f_i for the overall understanding of the point cloud and, thus, better view-wise prediction. On the other hand, the residual-style adapter infuses newly-learned 3D few-shot knowledge with that of 2D pre-trained CLIP, which further promotes the cross-modality knowledge transfer.

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 all views are summarized to construct the final prediction, and the view weights α_i can be learnable parameters here for more adaptive aggregation. Surprisingly, just fine-tuning this lightweight adapter with few-shot samples contributes to significant performance improvement, e.g. from 20.18% to 87.20% on ModelNet40 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 knowledge transfer from 2D to 3D. Consequently, PointCLIP with inter-view adapter provides a promising alternative solution for point cloud understanding. In some applications, there is no condition to train the entire model with large-scale fully annotated data, and fine-tuning only the three-layer adapter with few-shot data can achieve comparable performance.

3.4. Multi-knowledge Ensembling

Classical point cloud networks, such as the early PointNet [43] and the recent CurveNet [39], are trained from scratch on 3D datasets by close-set supervision. In contrast, PointCLIP mostly inherits pre-trained priors from 2D vision-language learning, containing different aspects of knowledge. We then investigate if the two forms of knowledge can be ensembled together for joint inference. In practice, we first obtain the classical model, e.g. PointNet++ [44] pre-trained from [22], and PointCLIP of either zero-shot or the adapter version. We conduct inferences of the two models and ensemble their predicted logits by simple addition as the final output. Beyond our expectation, aided by 16-shot fine-tuned PointCLIP of 87.20%, PointNet++ of 89.71% is enhanced to 92.03% with a significant improvement of +2.32%. In other words, ensembling of two low-score models can produce a much stronger one, which fully demonstrates the complimentary interaction of knowledge from the two models. Also, even

Zero-shot Performance of PointCLIP			
Datesets	Accuracy	Proj. Settings	View Weights
ModelNet10 [57]	30.23%	1.7, 100	2,5,7,10,5,6
ModelNet40 [57]	20.18%	1.6, 121	3,9,5,4,5,4
ScanObjectNN [51]	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 consist of projection distances and side length of the projected depth maps. View Weights are the relative values from 1 to 10.

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 (%) concerning projected view numbers and each view's importance for zero-shot and 16-shot PointCLIP on ModelNet40.

with the zero-shot PointCLIP of 20.18%, PointNet++ can still be improved to 92.10%. In contrast, ensembling a pair of classical full-trained models would not enhance the performance, which indicates the importance of complimentary knowledge. We also implement this ensembling with other advanced networks and observe similar performance boosts, some of which achieve state-of-the-art performances. Therefore, PointCLIP can be utilized as a plug-and-play enhancement module to achieve robust point cloud understanding.

4. Experiments

4.1. Zero-shot Classification

Settings. We evaluate the zero-shot classification performance of PointCLIP on three well-known datasets: ModelNet10 [57], ModelNet40 [57] and ScanObjectNN [51]. 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 [21] as the visual encoder and transformer [52] 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 ranged 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, and the larger distance leads to the denser points distributions 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 larger side length results in smaller projected object size. We then upsample all images to $(224, 224)$ for alignment with CLIP's settings. Also, 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. Performances 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. Performances (%) of PointCLIP for different visual encoders on ModelNet40. RN50 denotes ResNet-50, and ViT-B/32 represents vision transformer with 32 × 32 patch embeddings, and RN.×16 denotes ResNet-50 with 16 times more computations from [45].

Performance. In Table 1, we present performances of zero-shot PointCLIP for three datasets with their best-performing settings. Without any 3D training, PointCLIP is able to achieve a promising 30.23% on ModelNet10, which demonstrates the effective knowledge transfer from 2D to 3D. For ModelNet40 with 4 times the number of categories and ScanObjectNN of noisy real-world scenes, PointCLIP achieves slightly worse performances, 20.18% and 15.38%, respectively, due to the lack of 3D 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 fi-

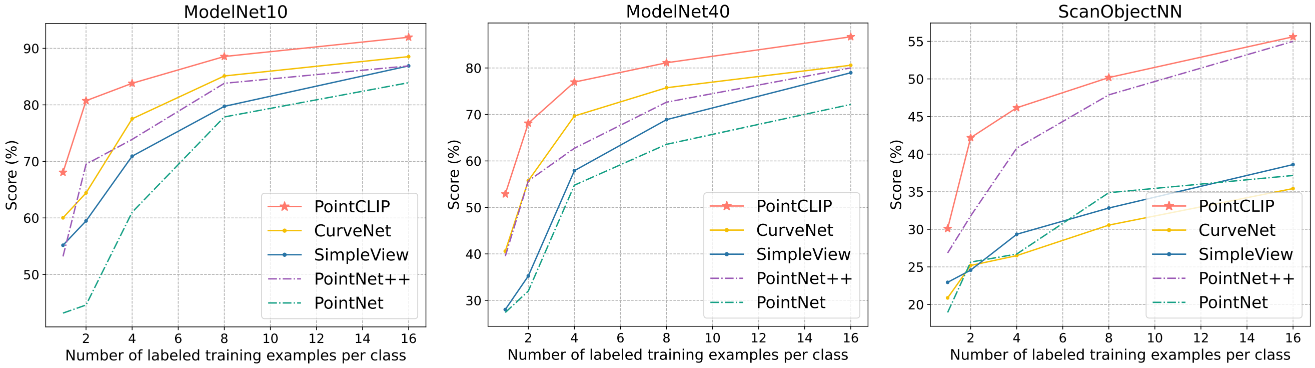

Figure 5. Few-shot performance comparison between PointCLIP and other classical 3D networks, including the state-of-the-art CurveNet, on ModelNet10, ModelNet40 and ScanObjectNN. Our PointCLIP shows consistent superiority to other models under 1, 2, 4, 8 and 16-shot settings.

nal 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 and the importance of each view on ModelNet40. For the number of projected views, we try 1, 4, 6, 8, 10 and 12[1] 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 the top and down views contribute relatively less to the zero-shot classification. In Table 4, we implement different visual backbones from ResNet [21] to vision transformer [11], and RN50×16 [45] achieves the best performance of 23.78%, which has 16 times more computations than ResNet-50. However, upgrading ResNet-50 to ResNet-101 with more parameters and deeper layers would not provide higher classification accuracy.

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, but 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%. Also, as the projected point cloud normally covers most of the image area, appending an adjective "big" could bring further performance improvement. Furthermore, 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 in the three datasets: ModelNet10 [57], ModelNet40 [57] and ScanObjectNN [51]. For K-shot settings, we randomly sample K point clouds from each category of the training set. We inherit the best projection settings from zero-shot experiments in Section 4.1. In contrast, considering both efficiency and performance, we adopt ResNet-101 [21] 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 performances of PointCLIP and compare it with 4 representative 3D networks: PointNet [43], PointNet++ [44], SimpleView [19] and the state-of-the-art CurveNet [39]. 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 limited fitting capacity of the lightweight three-layer adapter. For the detailed training settings, please refer to the Appendix.

Ablations. In Table 2, we show the 16-shot PointCLIP under different projection views and explore how each view

[1] The settings of views are in the Appendix.

Models	Before En.	After En.	Gain	Ratio
PointNet [43]	88.78	90.76	+1.98	0.60
PointNet++ [44]	89.71	92.10	+2.39	0.70
RSCNN [33]	92.22	92.59	+0.37	0.70
DGCNN [55]	92.63	92.83	+0.20	0.70
SimpleView [19]	93.23	93.87	+0.64	0.60
CurveNet [39]	93.84	**94.08**	+0.24	0.15

Table 5. The enhancement ability (%) of 16-shot PointCLIP, which achieves 87.20%, on multiple classical 3D networks in ModelNet40. Before and After En. denote models with and without PointCLIP ensembling, respectively.

contributes on 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 version and observe the reversed conclusion that, the left view is the most informative here. Surprisingly, 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 influence caused by prompt designs, and the "point cloud of a big [CLASS]." performs the best, which is slightly different from the analysis in Paragraph 4.1.

4.3. Multi-knowledge Ensembling

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, respectively with fully-trained PointNet [43], PointNet++ [44], DGCNN [55], SimpleView [19] and CurveNet [39], whose trained models are obtained from [22, 50] 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, ensembling with PointCLIP improves the performances of all classical fully-trained 3D networks. The results fully demonstrate the complementarity of PointCLIP to existing fully-trained 3D models, and the performance gain is not simply achieved by ensembling models. These are surprising results to us, because the accuracy of 16-shot PointCLIP is lower than all other models trained with full datasets, but could still benefit their already high performances 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 further achieves 94.08%. Also, we observe that, for models with low baseline performances, PointCLIP's logits need to account for a large proportion, but for the well-performing ones, such as CurveNet, their knowledge is supposed to play a dominant role in the ensembling.

En. Model 1		En. Model 2	After En.
PointNet++ [44], 89.71	+	RSCNN [33], 92.22	92.14
PointNet++, 89.71	+	CurveNet [39], 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 ensembling models both trained on ModelNet40 or pre-trained in 2D.

Ensembling with CurveNet [39]						
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.

Ablations. We conduct ablation studies of ensembling two models fully trained on ModelNet40 without PointCLIP, and fuse their logits with the same ratio for simplicity. As is presented in Table 6, ensembling PointNet++ lowers the performance of RSCNN and CurveNet, and aggregating the highest two models, SimpleView and CurveNet, could not achieve better performance. Also, a pair of PointCLIP would hurt the performance. Hence, simply ensembling two models with the same training scheme normally leads to performance degradation, which demonstrates the significance of multi-knowledge interaction. In Table 7, we fuse zero-shot PointCLIP and the model fine-tuned by 8, 16, 32, 64 and 128 shots, respectively with CurveNet to explore their ensembling performances. As reported, zero-shot PointCLIP with only 20.18% could enhance CurveNet by +0.04%. However, too much training on 3D dataset would adversely influence the ensembling accuracy. This is possibly caused by the too high similarity between two models, which cannot provide complementary knowledge as expected.

5. Conclusion and Limitation

We propose PointCLIP, which conducts cross-modality zero-shot recognition on point cloud without any 3D training. Via multi-view projection, PointCLIP efficiently transfers CLIP's pre-trained 2D knowledge into the 3D domain. Under few-shot settings, we design a lightweight inter-view adapter to aggregate multi-view representations and generate adapted features. By fine-tuning such 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 complimentary information for the classical 3D networks, which surpasses state-of-the-art performance. Although PointCLIP realizes the transfer learning from 2D to 3D, how to utilize CLIP's knowledge for other 3D tasks is still under explored. Our future work will focus on generalizing CLIP for wider 3D applications.

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

[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. 3, 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, 12

[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] 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, 6, 7, 12

[22] Ankit Goyal Hei Law. Simpleview. `https://github.com/princeton-vl/SimpleView`, 2021. 5, 8

[23] 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

[24] 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

[25] 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

[26] 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

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12

[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. 3

[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] 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

[37] 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

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

[39] AAM Muzahid, Wanggen Wan, Ferdous Sohel, Lianyao Wu, and Li Hou. Curvenet: Curvature-based multitask learning deep networks for 3d object recognition. *IEEE/CAA Journal of Automatica Sinica*, 8(6):1177–1187, 2020. 1, 2, 5, 7, 8

[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] 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

[43] 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, 2, 3, 4, 5, 7, 8, 12

[44] 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, 13

[45] 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. 1, 2, 3, 6, 7

[46] 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

[47] 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

[48] 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

[49] 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

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

[51] 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, 12

[52] 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

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

[54] 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

[55] 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, 12

[56] 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

[57] 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, 6, 7, 12

[58] 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

[59] 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

[60] 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

[61] 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

[62] 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

[63] 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

[64] 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

[65] 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

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

[67] 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

Appendix

A. Datasets

We evaluate our PointCLIP on three well-known datasets: ModelNet10 [57], ModelNet40 [57] and ScanObjectNN [51]. Therein, ModelNet10 consists of 4,899 synthetic meshed CAD models with 10 indoor categories, 3,991 for training and 908 for testing. ModelNet40 is larger and contains 12,311 samples of 40 common categories, 9,843 for training and 2,468 for testing. In both datasets, we uniformly sample 1,024 points from each object as the network input. ScanObjectNN contains 2,321 training and 581 testing point clouds of 15 categories collected directly from real-world scans. Different from synthetic data with complete profiles, objects in ScanObjectNN are occluded at different levels and disturbed with background noise, so it is more challenging for accurate recognition.

B. Implementation Details

For ablation studies of projected view numbers, we adopt different settings for zero-shot and few-shot PointCLIP. As the right view is the most important for zero-shot PointCLIP, we set the 12 views to: front, right, back, left, top, bottom, upper/lower right diagonal front/back (4 views) and upper left diagonal front/back (2 views). In contrast, few-shot PointCLIP achieves higher performance with left views, so we replace all the "left" settings above into "right". For both versions, the view number of M represents picking the first M views for experiments.

For PointCLIP with inter-view adapter, we fine-tune it under 1, 2, 4, 8 and 16 shots with batch size 32 and learning rate 0.01 for 250 epochs. Stochastic Gradient Decent (SGD) [27] with momentum 0.9 is adopted as the optimizer. We utilize a cosine scheduler for learning rate decay and Smooth Loss [55] following [19]. In ModelNet10 and ModelNet40, We apply random scaling and translation for training augmentation, but in the challenging ScanObjectNN, we append jitter and random rotation following [43]. During training, we freeze CLIP's both visual and textual encoders, and only fine-tune the inter-view adapter. For other compared models, we unfreeze all the parameters, and adopt the same data augmentation and loss functions reported in the papers.

C. Supplementary Ablations

Inter-view Adapter. We adopt the inter-view adapter with three linear layers: one for global extraction and two for view-wise adapted features generation. Here, we explore other architectures of the adapter on 16-shot PointCLIP for ModelNet40 in Table 8. Specifically, w/o global denotes the adapter processing each view separately without interaction, and the w/o view-wise version repeats the global feature as each view's adapted feature. The 2-layer adapter removes the linear layer after the global representation and the pre-layer version moves it before the global extraction. The results show that dropping or changing the original modules in the adapter would all hurt the performance, especially the inter-view extraction of global feature.

	Architectures of Inter-view Adapter			
original	w/o global	w/o view-wise	2-layer	pre-layer
87.20	83.87	85.93	86.48	86.78

Table 8. Architectures of the inter-view adapter.

Adapted Features Fusion. The view-wise adapted feature is generated by the adapter and then added to the original CLIP-encoded feature via a residual connection. On ModelNet40, we evaluate the performance of 16-shot PointCLIP with different fusion ratios β, which denotes the proportion of adapted features. To show the effect of β, we set all view weights the same. From the results in Table 9, different ratios actually lead to little performance variance and the β of 0.6 perfoms better than others. Thus, we adopt 0.6 as the fusion ratio by default, which indicates the comparable contributions between 2D pre-trained knowledge and 3D learned knowledge.

					Adapter Fusion Ratios β					
0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
9.56	85.74	85.78	85.66	85.76	85.98	86.13	85.91	85.85	85.74	85.53

Table 9. Different fusion ratios of adapted features.

Full Training Set. We also fine-tune PointCLIP on full training set of ModelNet40 [57] and present the results in Table 10. Likewise, we freeze both pre-trained visual and textual encoders in CLIP and only train the inter-view adapter. As expected, visual encoders with more parameters lead to higher accuracy, and only fine-tuning the appended lightweight adapter could achieve the performance of 92.01%.

			Fine-tuning on Full ModelNet40 [57]			
Models	RN50	RN101	ViT/32	ViT/16	RN.×4	**RN.×16**
Accuracy	86.42	91.69	91.76	90.70	91.93	**92.01**

Table 10. Performances (%) for fine-tuning PointCLIP on full training set of ModelNet40 with different visual encoders.

Fine-tuning Settings. Under full training set of ModelNet40 [57], we further fine-tune different modules of PointCLIP in Table 11. Therein, we adopt ResNet-101 [21] as the

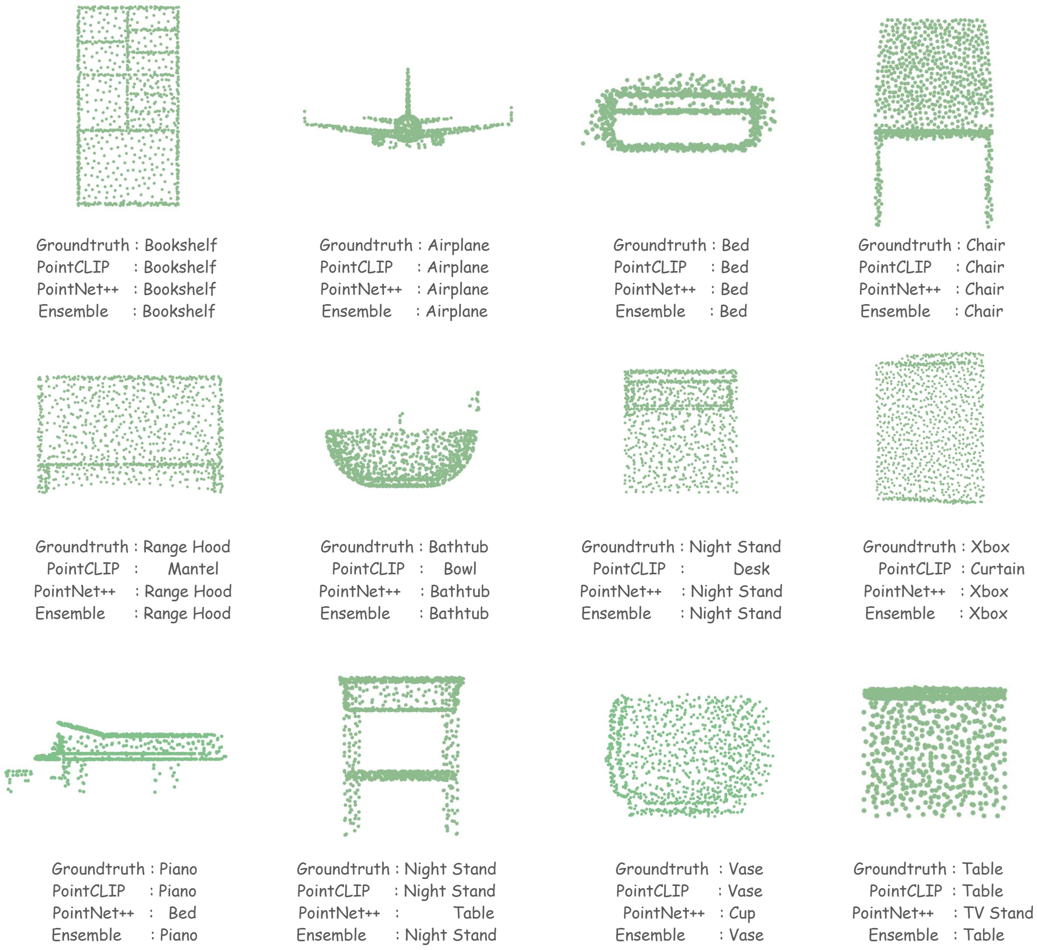

Figure 6. Visualizations of predictions by PointCLIP, PointNet++ [44] and the ensembled model.

visual encoder, and fine-tuning without the adapter represent unfreezing the visual or textual encoder upon the zero-shot PointCLIP. As presented, unfreezing just the textual encoder normally hurts the performance, but training both encoders and all modules of PointCLIP achieves better performance of 91.40% and 91.89%, respectively.

Visual Encoder	Textual Encoder	Inter-view Adapter	Accuracy(%)
✓	-	-	91.01
-	✓	-	73.89
✓	✓	-	91.49
-	-	✓	91.69
✓	-	✓	90.99
-	✓	✓	88.82
✓	✓	✓	**91.89**

Table 11. Ablations of PointCLIP fine-tuning different modules. ✓ denotes fine-tuning the module and symbol - denotes freezing.

D. Visualizations

We visualize some cases of ensembling PointCLIP with PointNet++ [44] to reveal the effectiveness of enhancement. As shown in Figure 6, two models both predict correctly for the four samples, and the ensembled model preserves the prediction. As for samples in the second and the third rows, PointCLIP and PointNet++ show the complementary properties that the ensembled model would rectify one of their wrong predictions, which demonstrates the importance of knowledge interaction.