CLIP2POINT: TRANSFER CLIP TO POINT CLOUD CLASSIFICATION WITH IMAGE-DEPTH PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-training across 3D vision and language remains under development because of limited training data. Recent works attempt to transfer vision-language pre-training models to 3D vision. PointCLIP converts point cloud data to multi-view depth maps, adopting CLIP for shape classification. However, its performance is restricted by the domain gap between rendered depth maps and images, as well as the diversity of depth distributions. To address this issue, we propose CLIP2Point, an image-depth pre-training method by contrastive learning to transfer CLIP to the 3D domain, and adapt it to point cloud classification. We introduce a new depth rendering setting that forms a better visual effect, and then render 52,460 pairs of images and depth maps from ShapeNet for pre-training. The pre-training scheme of CLIP2Point combines cross-modality learning to enforce the depth features for capturing expressive visual and textual features and intra-modality learning to enhance the invariance of depth aggregation. Additionally, we propose a novel Dual-Path Adapter (DPA) module, i.e., a dual-path structure with simplified adapters for few-shot learning. The dual-path structure allows the joint use of CLIP and CLIP2Point, and the simplified adapter can well fit few-shot tasks without post-search. Experimental results show that CLIP2Point is effective in transferring CLIP knowledge to 3D vision. Our CLIP2Point outperforms PointCLIP and other self-supervised 3D networks, achieving state-of-the-art results on zero-shot and few-shot classification.

1 INTRODUCTION

Vision-language (V-L) pre-training has achieved great success in computer vision. Benefiting from large-scale data, V-L pre-trained models (Radford et al., 2021; Yao et al., 2021) transfer language knowledge to visual understanding, which can be fine-tuned to multiple downstream tasks. However, pre-training across 3D vision and language remains an open question, due to the lack of sufficient training data. For example, Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) takes more than 400M image-text pairs as training data. In contrast, few studies have been given to pre-training across 3D vision and language. Moreover, even conventional 3D pre-training method PointContrast (Xie et al., 2020) is trained on ScanNet (Dai et al., 2017) with only 100k pairs of point clouds in 1,513 scenes. Due to the limitation of 3D pre-training, most existing 3D deep networks (Qi et al., 2017; Wang et al., 2019) are trained from scratch on specific downstream datasets.

One remedy is to leverage the existing successful V-L pre-trained model for 3D vision tasks. To this end, one may first convert the 3D point clouds to multi-view 2D depth maps (Su et al., 2015; Goyal et al., 2021; Wang et al., 2022). By simply treating 2D depth maps as images, PointCLIP (Zhang et al., 2022) applies CLIP to 3D tasks, providing zero-shot and few-shot settings in the point cloud classification with textual prompting. However, its results are still limited since the rendered depth maps are much different from the image domain of the CLIP training dataset. And the sparsity and disorder of point cloud data result in various depth distributions from multiple views, further confusing the aggregation of CLIP. Existing pre-training works focus on the domain gap (Afham et al., 2022) or multi-view consistency (Xie et al., 2020) of point clouds, while we intend to tackle similar issues based on depth maps. In addition, a solution of adapting pre-training knowledge to downstream tasks should be included in the V-L transfer.

In order to transfer CLIP to the 3D domain, we propose CLIP2Point, a pre-training scheme with two learning mechanisms: 1) *cross-modality learning* for the contrastive alignment of RGB image and depth map, 2) *intra-modality learning* in the depth modality to enhance the invariance of depth aggregation In particular, the image encoder E_i is directly from CLIP weights and is frozen during pre-training. While the depth encoder E_d is trained to 1) align depth features with CLIP image features in cross-modality learning and 2) encourage the depth aggregation to be invariant to view changes in intra-modality learning. With pre-training, the depth features can then be well aligned with the visual CLIP features. As for the training data, we do not adopt the depth maps in the existing RGB-D datasets as they are densely sampled and are contradicted to the sparsity of rendered depth maps. Instead, we reconstruct multi-view images and depth maps from 3D models directly. Specifically, we render 10 views of RGB images from ShapeNet (Chang et al., 2015), which covers 52,460 3D models for 55 object categories. Meanwhile, we generate corresponding depth maps, with a new rendering setting that forms a better visual effect for CLIP encoding. Experiments show that our CLIP2Point can significantly improve the performance of zero-shot point cloud classification.

To further adapt our CLIP2Point to few-shot learning, we propose a novel Dual-Path Adapter (DPA) module. Since our pre-training is to align the instance-level depth map, it can be complementary with CLIP pre-training knowledge that focuses on category-level discrimination. We propose a dual-path structure, where both our pre-trained depth encoder E_d and the CLIP visual encoder E_i are utilized. A learnable simplified adapter is attached to each encoder to extract a global feature from multiple views. And the final logits can be calculated by the combination of two encoders.

To sum up, our main contributions can be summarized as:

- We propose a CLIP2Point method by contrastive learning to transfer CLIP knowledge to the 3D domain. For the training data, we pre-process ShapeNet, reconstructing 52,460 pairs of rendered images and depth maps with a better depth rendering setting. Experiments show that CLIP2Point significantly improves the performance of zero-shot point cloud classification.
- We propose a novel Dual-Path Adapter (DPA) module, a dual-path structure with a simplified adapter for extending CLIP2Point to few-shot classification.
- Extensive experiments are conducted on ModelNet10, ModelNet40, and ScanobjectNN. In comparison to PointCLIP and self-supervised 3D networks, CLIP2Point achieves state-of-the-art results on both zero-shot and few-shot point cloud classification tasks.

2 Related Work

2.1 Vision-language Pre-training

Vision-language (V-L) pre-training has been a growing interest in multi-modal tasks. Pre-trained by large-scale image-text (Chen et al., 2020b) or video-text (Sun et al., 2019) pairs, those models can be applied to multiple downstream tasks, *e.g.*, visual question answering, image/video captioning, and text-to-image generation. CLIP (Radford et al., 2021) further leverages V-L pre-training to transfer cross-modal knowledge, allowing natural language to understand visual concepts. Nonetheless, pre-training across 3D vision and language is restricted by insufficient 3D-text data pairs. And 3D downstream tasks like shape retrieval (Han et al., 2019) and text-guided shape generation (Liu et al., 2022) suffer from limited performance. Considering the vacancy between 3D vision and language, we attempt to transfer CLIP pre-trained knowledge to the 3D domain, making language applicable to point cloud classification.

2.2 Self-supervised Pre-training

Self-supervised pre-training has become an important issue in computer vision. Since task-related annotations are not required, it can leverage large-scale data and pretext tasks to learn general representation. In particular, contrastive learning (He et al., 2020; Chen et al., 2020a) and masked auto-encoding (He et al., 2022; Zhou et al., 2021; Devlin et al., 2018) are two popular self-supervised schemes. Instead of directly applying masked auto-encoding to 3D point completion (Yu et al., 2022; Pang et al., 2022), Li & Heizmann (2022) show that contrastive learning in 3D vision can vary from granularity (point/instance/scene) or modality (point/depth/image). In this work, we aim

to adopt image-depth contrastive learning to bridge the domain gap between depth features and visual CLIP features, thereby allowing to transfer CLIP knowledge to the 3D domain.

2.3 DOWNSTREAM FINE-TUNING

Fine-tuning has been widely used in downstream tasks to fit pre-trained weights to specific training datasets (Zhai et al., 2019; Lin et al., 2014; Zhou et al., 2017). One common practice is to update the entire parameters during training, while it may be overfitted if the scale of training data is limited. Instead, partial tuning (Cai et al., 2020; Zhang et al., 2021) is a data-efficient way to fit downstream data. Recently, prompt tuning has been applied to language (Brown et al., 2020; Li & Liang, 2021) and vision (Dosovitskiy et al., 2020; Jia et al., 2022) models. Prompt tuning provides several learnable token sequences and specific task heads for the adaptation, without the full tuning of pre-trained parameters. Note that pre-trained models in 3D vision are still in early exploration, and existing deep networks in point cloud (Qi et al., 2017; Wang et al., 2019; Mohammadi et al., 2021) all follow a full tuning paradigm. In contrast, we propose a novel Dual-Path Adapter module for a lightweight fine-tuning. With CLIP textual prompts, a few-shot setting is available by tuning simplified adapters only.

3 CLIP-BASED TRANSFER LEARNING IN 3D

3.1 REVIEW OF CLIP AND POINTCLIP

CLIP (Radford et al., 2021) is a vision-language pre-training method that matches images and texts by contrastive learning. It contains two individual encoders: a visual encoder and a language encoder, to respectively extract image features $\mathbf{F}^I \in \mathbb{R}^{1\times C}$ and textual features $\mathbf{F}^T \in \mathbb{R}^{1\times C}$. Here, C is the embedding dimension of encoders. For zero-shot transfer, the cosine similarity of $\mathbf{F}^I$ and $\mathbf{F}^T$ implies the matching results. Taking a K-category classification task as an example, textual prompts are generated with the category names and then encoded by CLIP, extracting a list of textual features $\{\mathbf{F}_k^T\}_{k=1}^K \in \mathbb{R}^{K\times C}$. For each image feature $\mathbf{F}^I$, we can calculate the classification logits as follows,

$$\mathrm{logits} = \mathrm{softmax}(\mathbf{F}^I \{\mathbf{F}_k^T\}^T). \tag{1}$$

where $\mathrm{logits}^{(k)}$ denotes the predicted probability of the k-th category.

PointCLIP (Zhang et al., 2022) applies CLIP to 3D point cloud data. It renders multi-view depth maps from point clouds, and then extracts the depth map features $\{\mathbf{F}_v^D\}_{v=1}^N$ with the CLIP visual encoder, where N is the number of views. Logits of the zero-shot classification can be calculated similarly to Eq. (1), while multi-view features are gathered with searched weights. PointCLIP also proposes an inter-view adapter for the few-shot classification. It adopts a residual form, which concatenates multi-view features $\{\mathbf{F}_v^D\}_{v=1}^N$ for a global representation $\mathbf{G}^D \in \mathbb{R}^{1\times C}$ and then add $\mathbf{G}^D$ back to extract adapted features $\hat{\mathbf{F}}_v^D \in \mathbb{R}^{1\times C}$. The adapter can be formulated as,

$$\mathbf{G}^D = f_2(\mathrm{ReLU}(f_1(\mathrm{concat}(\{\mathbf{F}_v^D\}_{v=1}^N)))), \tag{2}$$

$$\hat{\mathbf{F}}_v^D = \mathrm{ReLU}(\mathbf{G}^D \mathbf{W}_v^T), \tag{3}$$

$$\mathrm{logits} = \mathrm{softmax}(\sum_{v=1}^{N} \alpha_v ((\mathbf{F}_v^D + \hat{\mathbf{F}}_v^D)\{\mathbf{F}_k^T\}^T)), \tag{4}$$

where $\mathrm{concat}(\cdot)$ denotes the concatenation on channel dimensions, f_1 and f_2 are two-layer MLPs, and $\mathbf{W}_v \in \mathbb{R}^{C\times C}$ and α_v denote the view transformation and the summation weights of the v-th view, respectively. f_1, f_2 and $\mathbf{W}_v$ are learnable during the few-shot learning, and α_v is post-searched.

However, depth maps are representations of geometry information, which lack natural texture information. Therefore, it is inappropriate to directly apply CLIP visual encoder for the extraction of depth features, leaving some leeway for boosting zero-shot point cloud classification.

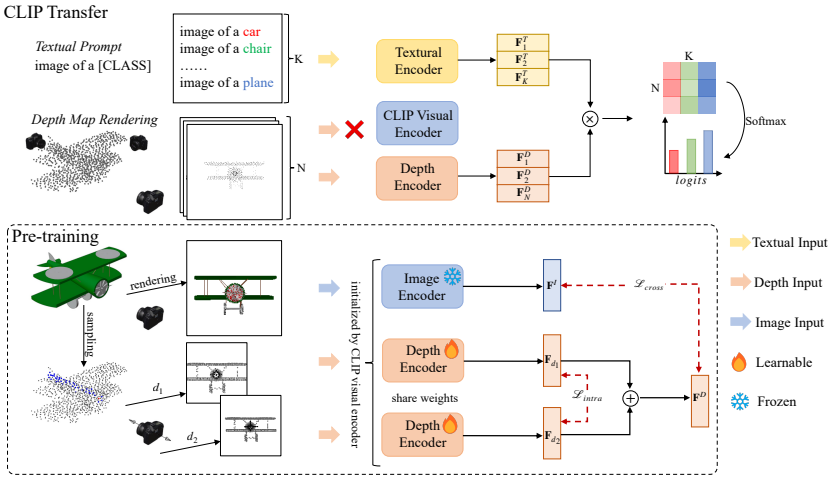

Figure 1: Overall architecture of our CLIP transfer learning. We improve the depth rendering setting and avoid inefficient post-search. Most importantly, we replace CLIP visual encoder with our pre-trained depth encoder. We propose a self-supervised pre-training scheme with intra-modality and cross-modality contrastive learning to align depth features with CLIP visual features. We randomly choose a camera view for each 3D model and modify the distances of the view to construct a pair of rendered depth maps. We adopt one NT-Xent loss between pairs of depth features extracted from the depth encoder and the other between image features and average depth features. We freeze the image encoder during training, enforcing the depth features by depth encoder to be aligned with the image features by CLIP visual encoder.

3.2 Aligning Multi-view Depth Features with CLIP Visual Features

Instead of directly applying CLIP visual encoder to depth maps, we suggest to learn a depth encoder for aligning depth features with CLIP visual features. In other words, we expect the extracted features of a rendered depth map to be consistent with CLIP visual features of the corresponding image. Then, CLIP textual prompts can be directly adopted to match the depth features. Moreover, since depth maps are presented in multiple views, the consistency of depth distribution needs maintaining as well.

Contrastive learning is a self-supervised pre-training method that aligns features of each sample with its positive samples, and satisfies our expectations of minimizing the distance between image and depth features, as well as enhancing the consistency of multi-view depth features. We reconstruct a pre-training dataset from ShapeNet, which consists of pairs of rendered RGB images and corresponding depth maps. To generate depth maps in a better visual effect for CLIP encoding, a new depth rendering setting is adopted. We propose a self-supervised pre-training scheme with intra-modality and cross-modality contrastive learning. The pre-trained depth encoder can well adapt to CLIP prompts. In the following, we explain these modules in more detail.

3.2.1 Depth Rendering

To convert point cloud data into rendered depth images,we need to project 3D coordinates $(X, Y, Z) \in \mathbb{R}^3$ to 2D coordinates $(\hat{X}, \hat{Y}) \in \mathbb{Z}^2$ in a specific view. Here we choose rendering from the front view as an example to illustrate the projection. Specifically, a point at (x, y, z) can match the corresponding pixel at $(\lceil x/z \rceil, \lceil y/z \rceil)$ by perspective projection. However, there are still two issues: 1) multiple points can be projected to the same pixel in a specific plane; 2) a large area of the rendered depth maps remains blank since no points are in the background. For the first issue, existing works (Goyal et al., 2021; Zhang et al., 2022) prefer weighted summation of multiple points,

$$d(\hat{x}, \hat{y}) = \frac{\sum_{(x,y,z)} z/(z+\epsilon)}{\sum_{(x,y,z)} 1/z}, \tag{5}$$

where (x, y, z) is the set of points matching $(\hat{x}, \hat{y})$, and ϵ denotes a minimal value, e.g., $1e{-}12$. We argue that the minimum depth value of those points is more intuitive in 2D vision, as we cannot watch an object perspectively with naked eyes. For the second issue, few pixels can be covered due to the sparsity of point clouds. We extend each point to its neighborhood pixels, in order that the visual continuity of the depth value can be refined. We set the dilation rate R to 2, thus obtaining the final rendered value as follows:

$$d(\hat{x}, \hat{y}) = \min(z|(x, y, z) \in \mathbf{P}, \hat{x} - \frac{R}{2} \leq \lceil x/z \rceil < \hat{x} + \frac{R}{2}, \hat{y} - \frac{R}{2} \leq \lceil y/z \rceil < \hat{y} + \frac{R}{2}), \quad (6)$$

where $\min(\cdot)$ denotes the minimum value of the input set, and $\mathbf{P}$ denotes the set of point clouds. We visualize the rendering process in the bottom of Fig. 1, where we take the value of the red point in the airplane as the depth in $(0, 0)$, but previous works additionally consider all the blue points.

3.2.2 Pre-training Scheme

As shown in Fig. 1, our pre-training network includes a depth encoder E_d and an image encoder E_i. Given the input dataset $S = \{\mathbf{I}_i\}_{i=1}^{|S|}$, where $\mathbf{I}_i \in \mathbb{R}^{3\times H\times W}$ is the i-th rendered image in a random camera view, we render the corresponding depth maps $\mathbf{D}_{i,d_1}$ and $\mathbf{D}_{i,d_2}$ in the same view angle but with different distances d_1 and d_2. We first adopt a intra-modality aggregation among $\{(\mathbf{D}_{i,d_1}, \mathbf{D}_{i,d_2})\}_{i=1}^{|S|}$ with E_d, and then extract image features from $\{\mathbf{I}_i\}_{i=1}^{|S|}$ with E_i, enforcing E_d to keep consistent with E_i in a cross-modality aspect. E_d and E_i are both initialized with the weights of the visual encoder in CLIP. We freeze the parameters of E_i on training, while E_d is learnable.

Intra-modality Learning. Considering the sparsity and disorder of point clouds in the 3D space, distributions of depth values for different views vary a lot, even though we render depth maps at the same distance. To keep the invariance of distance aggregation in E_d, intra-modality contrastive learning is adopted. For each input depth map $\mathbf{D}_i$, we randomly modify the distance of the camera view but keep the view angle, generating two augmented depth maps $\mathbf{D}_{i,d_1}$ and $\mathbf{D}_{i,d_2}$. $\mathbf{D}_{i,d_1}$ and $\mathbf{D}_{i,d_2}$ are then fed into E_d, extracting depth features $\mathbf{F}_{i,d_1}^D, \mathbf{F}_{i,d_2}^D \in \mathbb{R}^{1\times C}$, Following the NT-Xent loss in SimCLR (Chen et al., 2020a), the intra-modality contrastive loss $\mathcal{L}_{intra}$ can be formulated as,

$$l_{intra}(i; d_1, d_2) = -\log \frac{\exp(s(\mathbf{F}_{i,d_1}^D, \mathbf{F}_{i,d_2}^D)/\tau)}{\sum_{k=1}^{N} \mathbb{1}_{[k\neq i]} \exp(s(\mathbf{F}_{i,d_1}^D, \mathbf{F}_{k,d_1}^D)/\tau) + \sum_{k=1}^{N} \exp(s(\mathbf{F}_{i,d_1}^D, \mathbf{F}_{k,d_2}^D)/\tau)}, \quad (7)$$

$$\mathcal{L}_{intra} = \frac{1}{2N} \sum_{i=1}^{N} (l_{intra}(i; d_1, d_2) + l_{intra}(i; d_2, d_1)), \quad (8)$$

where N denotes the batch size, $s(\cdot, \cdot)$ denotes the cosine similarity, and τ denotes the temperature coefficient. We set $\tau = 0.7$. And the final depth feature map $\mathbf{F}_i^D$ is the mean of $\mathbf{F}_{i,d_1}^D$ and $\mathbf{F}_{i,d_2}^D$.

Cross-modality Learning. For a set of rendered RGB-D data, cross-modality contrastive learning aims to minimize the distance between rendered images and depth maps in the same pair, while maximizing the distance of others. For each input image $\mathbf{I}_i$, we extract the image features $\mathbf{F}_i^I \in \mathbb{R}^{1\times C}$, which is exactly the same as CLIP visual features. Together with depth features $\mathbf{F}_i^D$, we have the cross-modality contrastive loss $\mathcal{L}_{cross}$ as follows,

$$l_{cross}(i; D, I) = -\log \frac{\exp(s(\mathbf{F}_i^D, \mathbf{F}_i^I)/\tau)}{\sum_{k=1}^{N} \mathbb{1}_{[k\neq i]} \exp(s(\mathbf{F}_i^D, \mathbf{F}_k^D)/\tau) + \sum_{k=1}^{N} \exp(s(\mathbf{F}_i^D, \mathbf{F}_k^I)/\tau)}, \quad (9)$$

$$\mathcal{L}_{cross} = \frac{1}{2N} \sum_{i=1}^{N} (l_{cross}(i; D, I) + l_{cross}(i; I, D)). \quad (10)$$

$\mathcal{L}_{intra}$ and $\mathcal{L}_{cross}$ are independently propagated, and $\mathcal{L}_{intra}$ drops much faster than $\mathcal{L}_{cross}$ during our pre-training. Thus, we adopt a multi-task loss (Kendall et al., 2018) to balance the two terms. The overall loss function $\mathcal{L}$ is formulated as,

$$\mathcal{L} = \frac{1}{\sigma^2} \mathcal{L}_{intra} + \mathcal{L}_{cross} + \log(\sigma + 1), \quad (11)$$

where σ is a learnable balance parameter.

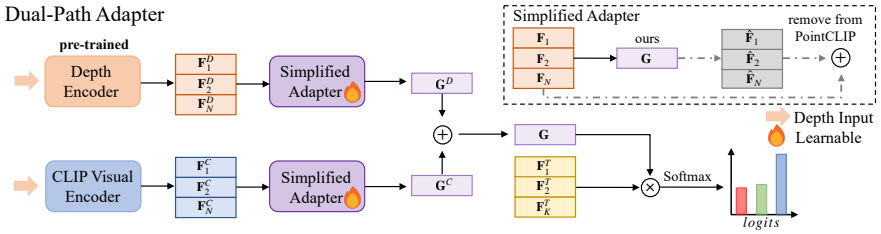

Figure 2: Dual-Path Adapter (DPA) module for few-shot learning. We design a dual-path structure, combining our pre-trained depth encoder with CLIP visual encoder. We propose a simplified adapter and attach it to each encoder, which is parameter-efficient for few-shot training. DPA allows a combination of knowledge in CLIP and our pre-training, and meanwhile avoid a post-search procedure.

3.3 Zero-Shot Classification

With newly rendered depth maps and a pre-trained depth encoder, we can obtain better performance of zero-shot classification with a similar pipeline in PointCLIP. And since we have narrowed the gap between depth maps and images after pre-training, depth features have a similar distribution to image features. We can simply use the prompt, *i.e.*, "**image of a [class name]**" as the textual prompts. After extracting depth features $\{\mathbf{F}_v^D\}_{v=1}^N$, we calculate the average logits of all the views as follows,

$$\text{logits} = \frac{1}{N}\text{softmax}(\sum_{v=1}^{N}(\mathbf{F}_v^D\{\mathbf{F}_k^T\}^T)). \tag{12}$$

Note that PointCLIP exploits post-search to find a set of view weights $\{\alpha_v\}_{v=1}^N$ that achieves the highest accuracy. We argue that post-search is a time-consuming procedure, which is typically unfair for zero/few-shot tasks that require efficiency. Hence, we avoid post-search during training and evaluation, replacing it with the mean of multi-view logits.

4 Dual-Path Adapter for Few-Shot Learning

Albeit zero-shot learning is an efficient transfer pipeline to downstream tasks, lightweight few-shot learning is also very useful for further refining the prediction accuracy. For example, PointCLIP improves the classification accuracy from 23.78% to 87.20% by a 16-shot learning. However, the few-shot pipeline in PointCLIP still depends on post-search. To avoid it, we simplify its adapter and propose a Dual-Path Adapter (DPA) module for few-shot learning. DPA allows a combination of pre-trained knowledge in CLIP and our pre-training, thereby enhancing the adaptation ability of CLIP2Point.

4.1 Simplified Adapter

The cross-view adapter in PointCLIP adopts a residual structure, which can maintain the dimension of input multi-view depth features $\mathbb{R}^{N\times C}$. However, the expansion of $\mathbf{G}^D$ in Eq. (3) requires extra weights, which may easily be overfitted in few-shot learning. Besides, summation weights remain a problem since features exist in multiple views. Simply calculating the average logits of the multi-view depth features (Eq. (12)) is not competitive with post-search that we have dismissed (Eq. (4)). Instead, $\mathbf{G}^D$ is the global feature of multiple views, which can be directly used to estimate a global logits vector. With such simplification, we reduce the learnable parameters and avoid post-search. The simplified adapter can be formulated as,

$$\text{logits} = \text{softmax}(\mathbf{G}^D\{\mathbf{F}_k^T\}^T). \tag{13}$$

4.2 Dual-Path Adapter

CLIP2Point has achieved a significant improvement on zero-shot point cloud classification, as our pre-training narrows the domain gap between depth maps and images. While in few-shot learning, lightweight adapters also help transfer domains in a more direct way somehow, focusing on

minimizing the category-level distance. That is the reason why PointCLIP can enjoy a promising accuracy in few-shot classification. However, the domain transfer in our pre-training is based on instance-level discrimination, extracting and comparing global features. Thus, our pre-trained depth encoder and the CLIP visual encoder can be complementary, where the depth encoder can adjust to an appropriate feature domain, and the visual encoder can pay more attention to category selection. We design a dual-path structure with these two encoders. For each path, an independent adapter is attached to the encoder. Finally, DPA can be formulated as,

$$\mathbf{G}^C = f_2^C(\mathrm{ReLU}(f_1^C(\mathrm{concat}(\{\mathbf{F}_v^C\}_{v=1}^N)))), \tag{14}$$

$$\mathbf{G}^D = f_2^D(\mathrm{ReLU}(f_1^D(\mathrm{concat}(\{\mathbf{F}_v^D\}_{v=1}^N)))), \tag{15}$$

$$\mathrm{logits} = \mathrm{softmax}(\frac{1}{2}(\mathbf{G}^C + \mathbf{G}^D)\{\mathbf{F}_k^T\}^T), \tag{16}$$

where the superscripts C and D of features/weights are related to CLIP and our pre-training, respectively. We use cross-entropy (De Boer et al., 2005) loss for supervision.

5 Experiments

5.1 Datasets

Pre-training Datasets. Numerous RGB-D datasets are available now, while depth images in those datasets cannot replace rendered depth maps, as they are densely annotated. To align images with sparsely marked depth maps, we have to directly convert 3D point clouds to depth maps. ShapeNet (Chang et al., 2015) is a large-scale dataset of 3D shape, with 52,460 3D models in 55 categories. Previous works (Xu et al., 2019; Choy et al., 2016) render a subset of ShapeNet in limited views. Instead, we render RGB images in 10 views with shapes and texture information from the complementary set of ShapeNet. The implementation follows MVTN (Hamdi et al., 2021) on Pytorch3D (Lassner & Zollhöfer, 2020). Meanwhile, we sample the farthest 1,024 points of corresponding 3D models, and then render those points to depth maps as Eq. (6). To access the CLIP representation, the size of rendered images and depth maps is 224×224. Following the separation of the classification benchmark on ShapeNet, we have 41,943 pairs for training and 10,517 pairs for validation. For each training sample in the batch, we randomly choose a view out of the ten views. To evaluate the rendering quality, we conduct zero-shot classification experiments. The accuracy of RGB images and depth maps in our validation set are 54.21% and 19.98%, respectively.

Downstream Datasets. Following PointCLIP, we evaluate zero-shot classification on ModelNet10 (Wu et al., 2015), ModelNet40 (Wu et al., 2015), and ScanObjectNN (Uy et al., 2019), 16-shot classification on ModelNet40. ModelNet is a synthetic indoor 3D dataset, where ModelNet10 and ModelNet40 are both its subsets for classification. In ModelNet10, there are 4,899 orientation-aligned CAD models from 10 categories, including 3,991 for training and 908 for testing. While ModelNet40 contains 12,311 CAD models from 40 categories, with 9,843 for training and 2,468 for testing. Since the original ModelNet40 is not aligned in orientation, we use the aligned version (Sedaghat et al., 2016). ScanObjectNN is a real-world dataset, which contains 2,902 samples of point cloud data from 15 categories. Different from clean CAD models in ModelNet, objects in ScanObjectNN are partially presented and attached with backgrounds. Thus, it is much harder than ModelNet. For all the three datasets, we sample 1,024 points of each model as the input point cloud.

5.2 Implementation Details

We implement our framework on PyTorch (Paszke et al., 2019) and use the basic version of Vision Transformer (Dosovitskiy et al., 2020) with a patch size of 32 (namely ViT-B/32) as our visual encoders. In pre-training, we use LAMB (You et al., 2019) optimizer with a weight decay of 1×10^{-4} and initialize the learning rate to 6×10^{-3}. Our pre-training takes 100 epochs with a batch size of 256. We choose the checkpoint with the highest accuracy in our evaluation set as the final weights for downstream tasks. In few-shot learning, we use AdamW (Loshchilov & Hutter, 2017) optimizer with a weight decay of 1×10^{-4} and initialize the learning rate to 1×10^{-3}. The training batch size is 32. Following PointCLIP, we use 6 orthogonal views: front, back, left, right, top, and bottom for zero-shot, and add four corner views for pre-training and few-shot learning. The view distance is initialized as 1, and the random range of distance in pre-training is $[0.9, 1.1)$.

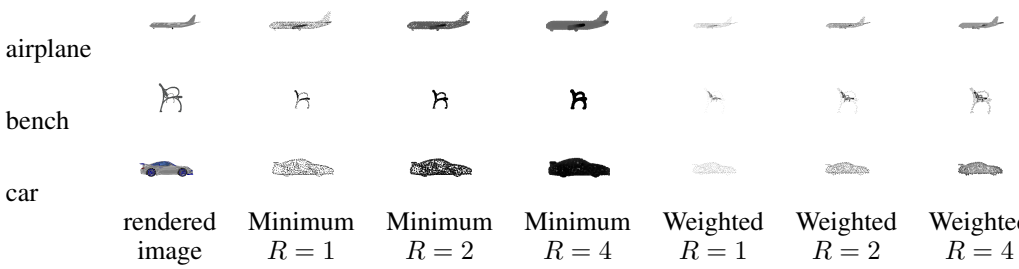

Figure 3: Visualization results of our rendered images with different rendering settings.

5.3 Zero-shot Classification

To the best of our knowledge, PointCLIP is the only attempt to conduct zero-shot classification on the whole 3D dataset. Previous works (Cheraghian et al., 2019; 2022) divide 3D datasets into two parts: "seen" and "unseen" categories. Models are trained on the former and evaluated on the latter, which is easier than the zero-shot setting in PointCLIP. To evaluate the effectiveness of our depth rendering setting and pre-training transfer, we compare PointCLIP on ModelNet10, ModelNet40, and ScanobjectNN.

As shown in Tab. 1, even without pre-training, our method can still outperform PointCLIP, simply by using the newly rendered depth maps. Especially in ModelNet40, we almost have a 10% gain on accuracy. After pre-training, the accuracy is significantly improved in ModelNet10 and ModelNet40, by 36.12% and 19.67%. Nonetheless, a 5.14% gain can also be attained on ScanObjectNN. While the improvement on ScanObjectNN is relatively small. We think that is because we generate our pre-training dataset from ShapeNet, which is a clean synthetic dataset like ModelNet.

Table 1: Quantitative results of zero-shot classification. Our pre-training significantly improves the accuracy, especially on Model10 and Model40.

Models	ModelNet10	ModelNet40	ScanObjectNN
PointCLIP	30.23	20.18	15.38
Ours w/o pre-training	30.51	29.71	18.18
Ours w/ pre-training	66.63	49.38	23.32

5.4 16-shot Classification

To further evaluate the transfer ability of our pre-training and verify our few-shot pipeline, we compare with PointCLIP, as well as two self-supervised pre-training methods: CrossPoint (Afham et al., 2022) and Point-MAE (Pang et al., 2022) in 16-shot classification. We choose a DGCNN (Wang et al., 2019) backbone for CrossPoint, and a 12-layer Transformer for Point-MAE. Although we only adopt ViT-B/32 as our encoder, PointCLIP in ResNet101 is included in the experiments.

We present the quantitative results of our few-shot experiments in Tab. 2. Initialized by CLIP weights, our few-shot pipeline w/o pre-training has already outperformed other methods, thanks to the simplified adapter and the dual-path structure. And our pre-trained version can reach an accuracy of 89.21%, which is very close to some traditional supervised networks such as PointNet++ (Qi et al., 2017).

Table 2: Quantitative results of few-shot classification. Our few-shot pipeline has already achieved state-of-the-art results, and the pre-trained version can further improve the performance.

Method & Encoder	CrossPoint & DGCNN	Point-MAE & Transformer	PointCLIP & ViT-B/32	PointCLIP & ResNet101	Ours & ViT-B/32
w/o pre-training	81.56	79.70	83.83	87.20	87.46
w/ pre-training	84.48	84.20	-	-	89.21

5.5 ABLATION STUDY

Depth Rendering. To analyze the depth rendering setting, we evaluate several settings for zero-shot classification in Tab. 3. "Weighted" and "Minimum" represent the depth values described in Eq. (5) and Eq. (6), respectively. "Dilation Rate" is R in Eq. (6), in which 1, 2, 4 are selected for ablation studies. The range definition of (x, y, z) in Eq. (5) is the same as Eq. (6) when adding a dilation rate.

As shown in Tab. 3, using the minimum depth value has much higher accuracy in CLIP zero-shot classification. We think that is because the visual effects can be close to CLIP pre-training images. While the larger is not the better in the setting of dilation rates. A too large dilation rate blurs depth maps, especially near the corners of objects. According to the results of zero-shot classification, we finally choose "Minimum" with a dilation rate of 2 as our depth rendering setting. The visualization in Fig. 3 further demonstrates that our setting has the best visual effect.

Intra-modality Learning. To evaluate the effectiveness of our intra-modality learning, we conduct a pre-training experiment with cross-modality only, in which the accuracy of zero-shot classification is only 38.29%. Regardless of random view distances, we simply extract the features of original depth maps as $\mathbf{F}_i^D$. The final loss can be formulated as Eq. 10. We keep the same pre-training setting, while the result of zero-shot classification in this version of pre-training is 11.09% lower than the version with intra-modality. Our intra-modality contrastive learning allows the depth encoder to keep a depth invariance among different camera views. Without randomized distances and corresponding contrastive restrictions, the encoder may easily fail when depth values vary a lot in different views.

Dual-Path Adapter. To evaluate the design of our Gual-Path Adapter module, we compare our simplified adapter, as well as the dual-path structure, with PointCLIP. For the experiments of the original adapter, we calculate the average logits, which is similar to Eq. (12).

As shown in Tab. 4, the simplified adapter surpasses the original one in PointCLIP. We have explained that extra weights in the original adapter make few-shot training much easier to overfit. The gathering of multi-view features is another problem for the original adapter since post-search is avoided in our evaluation. Additionally, the dual-path structure improves the performance as well, especially in our simplified adapter. The results demonstrate that our pre-trained encoder and the CLIP visual encoder are complementary. While the improvement in the original adapter is relatively small, we think multi-view gathering in a single encoder may disturb the combination of encoders.

Table 3: Quantitative results of zero-shot classification in different depth rendering settings.

Dilation Rate	1	2	4
Weighted	16.86	17.63	21.11
Minimum	24.87	29.71	28.36

Table 4: Quantitative results of few-shot classification with different components.

Dual-Path	Original	Simplified
✗	86.06	87.32
✓	86.18	89.21

6 CONCLUSION

In this paper, we propose CLIP2Point, which pre-trains a depth encoder for adapting CLIP knowledge to the 3D domain. We introduce a depth-image pre-training method, which consists of both intra-modality and cross-modality contrastive learning to bridge the domain gap between depth features by depth encoder and image features by CLIP visual encoder, and to maintain the invariance of multi-view depth distribution. For the pre-training data, we render 52,560 images from 3D models in ShapeNet, and meanwhile generate corresponding depth maps with a new depth rendering setting. After pre-training, the performance of zero-shot point cloud classification is significantly improved. To further adapt our pre-trained weights to 3D downstream tasks, we propose Dual-Path Adapter for few-shot classification. With a simplified adapter and a dual-path structure, we achieve state-of-the-art results in comparison with PointCLIP and other pre-trained 3D networks.

Although CLIP2Point successfully transfers CLIP knowledge to 3D vision, we observe that training data greatly influence the quality of our pre-training and the performance of downstream tasks, *e.g.*, synthetic pre-training data has a limited improvement on real-world downstream datasets. In future, we will improve the rendering data and explore more real-world 3D tasks with CLIP2Point.

REFERENCES

Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9902–9912, 2022.

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

Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. Tinytl: Reduce memory, not parameters for efficient on-device learning. *Advances in Neural Information Processing Systems*, 33:11285–11297, 2020.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pp. 104–120. Springer, 2020b.

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)*, pp. 1–6. IEEE, 2019.

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

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

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.

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.

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

Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Mvtn: Multi-view transformation network for 3d shape recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1–11, 2021.

Zhizhong Han, Mingyang Shang, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Y2seq2seq: Cross-modal representation learning for 3d shape and text by joint reconstruction and prediction of view and word sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 126–133, 2019.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Christoph Lassner and Michael Zollhöfer. Pulsar: Efficient sphere-based neural rendering. *arXiv:2004.07484*, 2020.

Lanxiao Li and Michael Heizmann. A closer look at invariances in self-supervised pre-training for 3d vision. *arXiv preprint arXiv:2207.04997*, 2022.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

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*, pp. 740–755. Springer, 2014.

Zhengzhe Liu, Yi Wang, Xiaojuan Qi, and Chi-Wing Fu. Towards implicit text-guided 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17896–17906, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Seyed Saber Mohammadi, Yiming Wang, and Alessio Del Bue. Pointview-gcn: 3d shape classification with multi-view point clouds. In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 3103–3107. IEEE, 2021.

Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. *arXiv preprint arXiv:2203.06604*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

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

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

Nima Sedaghat, Mohammadreza Zolfaghari, Ehsan Amiri, and Thomas Brox. Orientation-boosted voxel nets for 3d object recognition. *arXiv preprint arXiv:1604.03351*, 2016.

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*, pp. 945–953, 2015.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7464–7473, 2019.

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*, pp. 1588–1597, 2019.

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.

Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. *arXiv preprint arXiv:2208.02812*, 2022.

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*, pp. 1912–1920, 2015.

Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, pp. 574–591. Springer, 2020.

Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in Neural Information Processing Systems*, 32, 2019.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19313–19322, 2022.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark, 2019.

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.

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

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

A SUPERVISED CLASSIFICATION

CLIP2Point is a transfer learning paradigm in 3D vision, which achieves state-of-the-art results in zero-shot and few-shot downstream tasks. While it is available to supervised tasks with the same pipeline in few-shot learning. We conduct a supervised classification experiment on ModelNet40, comparing with 4 relative state-of-the-art networks: MVCNN (Su et al., 2015), SimpleView (Goyal et al., 2021), MVTN (Hamdi et al., 2021), and P2P Wang et al. (2022). Similar to CLIP2Point, these networks convert point cloud data to 2D depth maps, leveraging 2D pre-trained backbones to extract corresponding shape features.

As shown in Tab. 5, CLIP2Point achieves an equal accuracy to P2P (HorNet-L), but with much lower input requirements and evaluation computation costs. Since P2P infers a single view at one time, its evaluation cost needs to multiply the number of views 40. Additionally, our training only fine-tunes learnable adapters, which is more efficient than those full tuning methods.

Table 5: Quantitative results of supervised classification on ModelNet40. The numbers of input points and views are respectively presented in Data Type.

Methods	Data Type	Acc.(%)	Eval. MACs(G)	Tr. Param.(M)
MVCNN	image, 12	90.1	43.72	11.20
SimpleView	1,024, 6	93.4	53.38	12.76
MVTN	2,048, 12	93.8	45.97	27.06
P2P: ResNet-101	4,096, 40	93.1	11.96($\times$40)	0.25
P2P: ConvNeXt-L	4,096, 40	93.2	38.51($\times$40)	0.14
P2P: HorNet-L	4,096, 40	94.0	38.72($\times$40)	1.01
CLIP2Point (Ours)	1,024, 10	94.0	88.23	5.78

B RENDERING DETAILS

Following MVTN (Hamdi et al., 2021), we render 3D models to RGB images with Pytorch3D (Lassner & Zollhöfer, 2020). We first load mesh objects with texture information from ShapeNetCore v2. We choose 10 views in a spherical configuration, and then use **MeshRasterizer** and **HardPhongShader** in Pytorch3D.render, with the colors of backgrounds and lights both white. We visualize ten views of an airplane in Fig. 4.

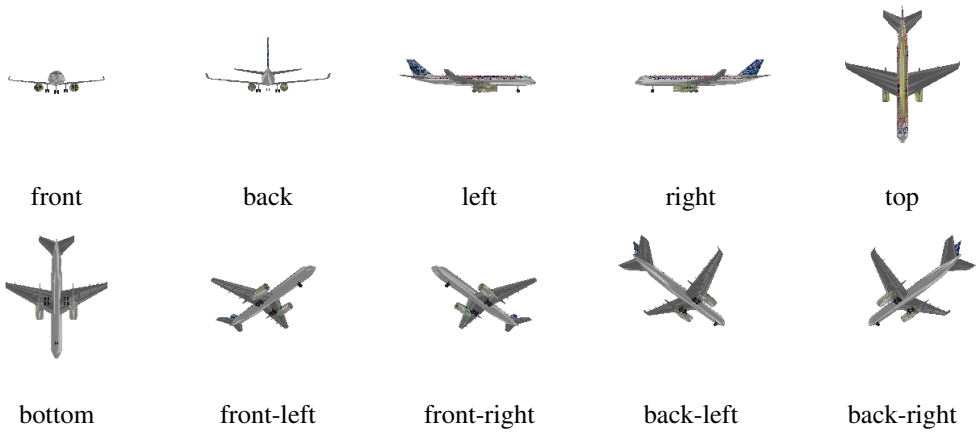

Figure 4: Visualization of multi-view RGB images for an airplane.

C VISUALIZATION

We provide more visualization results in Fig. 5, 6, 7. For each category in ShapeNet, we have a rendered RGB image and a corresponding depth map.

Figure 5: Rendered RGB images of Category 1 $\sim$ Category 20 on ShapeNet.

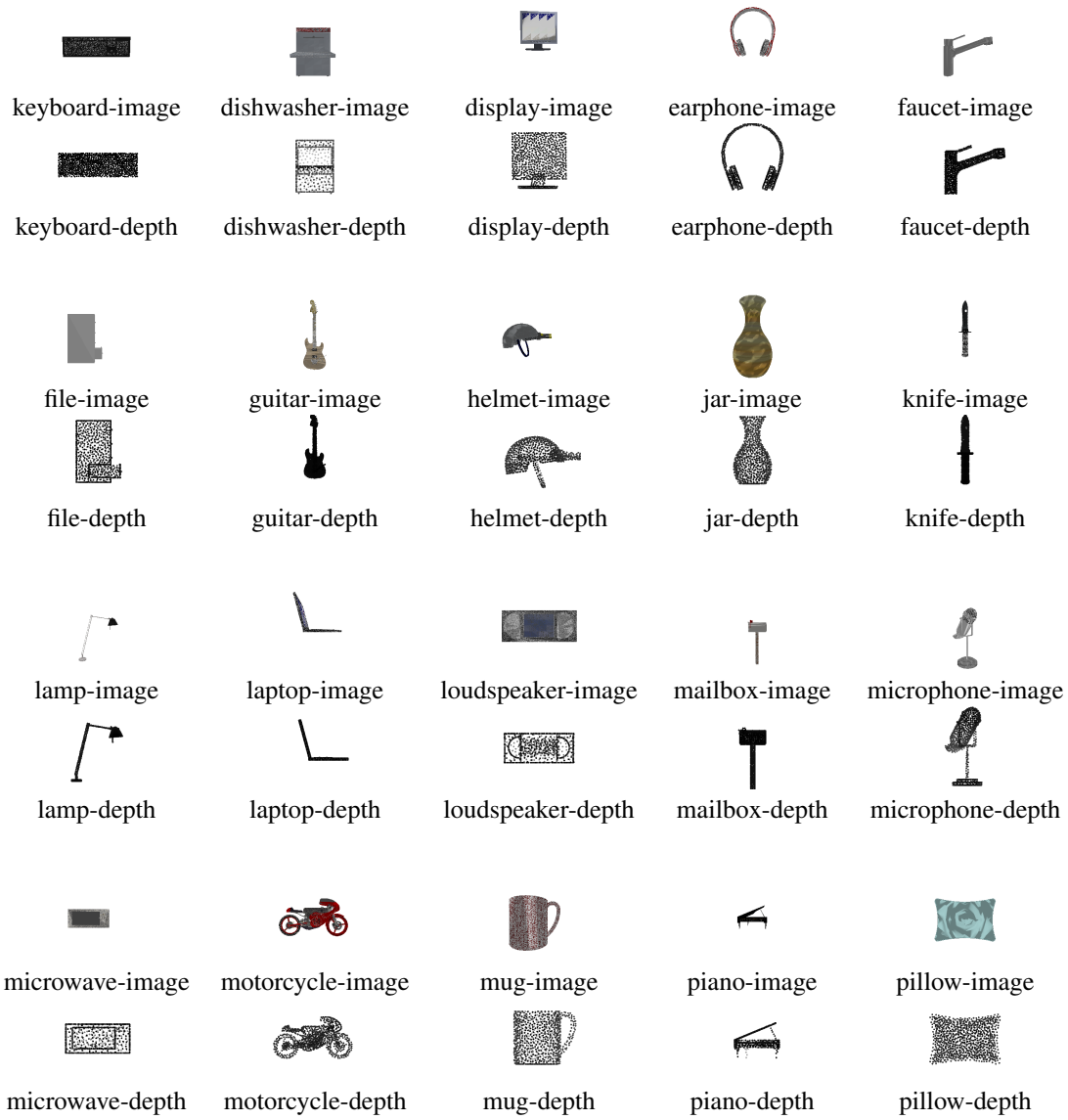

Figure 6: Rendered RGB images of Category 21 $\sim$ Category 40 on ShapeNet.

Figure 7: Rendered RGB images of Category 41 $\sim$ Category 55 on ShapeNet.