Supplementary - CLIP2Point: Transfer CLIP to Point Cloud Classification with Image-Depth Pre-training

Tianyu Huang[1,3] Bowen Dong[1] Yunhan Yang[1,4] Xiaoshui Huang[2]
Rynson W.H. Lau[3] Wanli Ouyang[2] Wangmeng Zuo[1,5†]
[1]Harbin Institute of Technology [2]Shanghai AI Laboratory [3]City University of Hong Kong
[4]The University of Hong Kong [5]Peng Cheng Laboratory
tyhuang0428@gmail.com, rynson.lau@cityu.edu.hk, wanli.ouyang@sydney.edu.au, wmzuo@hit.edu.cn

1. Details of Loss Function

$l^i_{intra}(\cdot)$ and $l^i_{cross}(\cdot)$ are InfoNCE-based loss. They are formulated as follows,

$$s^i_{intra}(d_1, d_2) = \sum_{k=1}^{N} e(\mathbf{F}^D_{i,d_1}, \mathbf{F}^D_{k,d_1}) + e(\mathbf{F}^D_{i,d_1}, \mathbf{F}^D_{k,d_2}), \tag{1}$$

$$l^i_{intra}(d_1, d_2) = -\log \frac{e(\mathbf{F}^D_{i,d_1}, \mathbf{F}^D_{i,d_2})}{s^i_{intra}(d_1, d_2) - e(\mathbf{F}^D_{i,d_1}, \mathbf{F}^D_{i,d_1})}. \tag{2}$$

$$s^i_{cross}(D, I) = \sum_{k=1}^{N} e(\mathbf{F}^D_i, \mathbf{F}^D_k) + e(\mathbf{F}^D_i, \mathbf{F}^I_k), \tag{3}$$

$$l^i_{cross}(D, I) = -\log \frac{e(\mathbf{F}^D_i, \mathbf{F}^I_i)}{s^i_{cross}(D, I) - e(\mathbf{F}^D_i, \mathbf{F}^D_i)}. \tag{4}$$

Here, $e(a, b) = \exp(a \cdot b^T / \tau)$. We set the temperature coefficient $\tau = 0.7$.

2. Complexity Analysis

Based on the experiments on fully-supervised classification, we further report the computation costs for evaluation and the parameter sizes for training in Tab. 1 to analyze the complexity of compared models.

Since PointCLIP achieves its best accuracy when using ResNet50x16, its computation cost is higher than our CLIP2Point with ViT-B/32. While the computation cost of P2P is even higher, its cost needs to be multiplied by 40 (the number of views) as P2P infers a single view at one time. Our CLIP2Point achieves a higher accuracy than P2P (HorNet-L) with a much lower computation cost.

†Corresponding Author: Wangmeng Zuo (wmzuo@hit.edu.cn)

On the other hand, embedded with GDPA module, CLIP2Point contains fewer training parameters than those full-tuning methods. Only by tuning lightweight adapters, CLIP2Point can outperform the state-of-the-art pre-training method Point-MAE.

Note that, the low computation cost of Transformer is because the grouping and gathering mechanisms for point cloud are not included in the calculation of MACs. Thus, comparing the MACs values with 3D networks is not fair.

Table 1. The computation costs for evaluation and the parameter sizes for training, based on fully-supervised classification.

Methods	Eval. MACs(G)	Tr. Param.(M)
MVCNN	43.72	11.20
SimpleView	53.38	12.76
MVTN	45.97	27.06
PointCLIP	227.42	5.51
P2P: ResNet-101	11.96(×40)	0.25
P2P: ConvNeXt-L	38.51(×40)	0.14
P2P: HorNet-L	38.72(×40)	1.01
Transformer	2.40	22.10
+ Point-BERT	2.40	22.10
+ Point-MAE	2.40	22.10
CLIP2Point (Ours)	88.23	5.78

3. Application on Scene-Level Tasks

Analogous to CLIP, we design our pre-training pipeline in an instance-level paradigm. We note that the knowledge in CLIP is more suitable to be used in image classification and retrieval, and it is also hard to directly leverage fine-grained knowledge from CLIP. Thus, existing works [2, 1] usually adopt extra modules to provide possible proposals, and CLIP still works as a classifier. Following [1], we conduct open-vocabulary 3D detection experiments, using our CLIP2Point to classify the bounding box generated by 3D detectors. In Tab. 2, CLIP2Point outperforms two 3D detec-

Table 2. Open-Vocabulary 3D Detection on ScanNet.

Method	Training Data	mAP_{25}
VoteNet	Seen(Train)-Unseen(Test)	0.04
3DETR		1.11
Image2Point	Zero-Shot	0.84
PointCLIP		3.09
CLIP2Point		3.71
OV-3DETIC	2D&3D Detection Data	12.65

tion networks and two cross-modal methods based on pre-training, indicating that **CLIP2Point can adapt to open-world scene-level tasks**. In contrast to OV-3DETIC specifically trained on 3D detection datasets with the distillation of a 2D detector, our mAP is relatively low, owing to noised point cloud data in proposed bounding boxes. Nonetheless, the result verifies the feasibility of such knowledge transfer in scene-level tasks. In the conclusion of the main text, we also mention that real-world training data can further enhance tasks related to real scenes in future work. Due to that CLIP2Point is effective in classifying objects, it can be naturally applied to scene-level tasks with proper proposals.

4. Why Not Applying CLIP to 3D Networks?

Since 3D backbones can be applied to downstream tasks more easily, a natural idea is whether CLIP pre-training knowledge can be directly applied to 3D networks. In fact, previous works [3, 6] have already demonstrated the effectiveness of such 2D-3D transfer. However, these methods achieve a sound result only after well fine-tuning on specific downstream datasets (*e.g.*, DGCNN pre-trained by CrossPoint even cannot surpass PointCLIP in the few-shot experiment in our main text), which means they can hardly adapt to 3D tasks with 2D knowledge alone. We replace the depth encoder in CLIP2Point with a 3D encoder Point Transformer [7], while getting a **20.83%** accuracy in ModelNet40 zero-shot classification after pre-training. We summarize two reasons for the bad result: 1) Features extracted by 2D and 3D encoders have different granularities: 2D encoders extract single-view features, while 3D encoders can aggregate a complete 3D object; 2) The gap between the parameter sizes of 2D and 3D encoders is large (*e.g.*, ViT-B/32 in CLIP contains 87.85M parameters, but DGCNN only contains 0.98M parameters): 2D pre-training knowledge cannot completely transfer to small 3D encoders.

Nonetheless, it is still promising future work to directly transfer CLIP knowledge to 3D networks.

5. Rendering Details

Following MVTN [4], we render 3D models to RGB images with Pytorch3D [5]. 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. For zero-shot evaluation, we use 6 orthogonal views: front, back, left, right, top, and bottom. We add four corner views for pre-training and downstream learning. The view distance is initialized as 1, and the random range of distance in pre-training is $[0.9, 1.1)$. We visualize ten views of an airplane in Fig. 1.

6. Dataset Visualization

We provide more visualization results in Fig. 2, 3, 4. For each category in ShapeNet, we have a rendered RGB image and a corresponding depth map.

References

[1] Open-vocabulary 3d detection via image-level class and debiased cross-modal contrastive learning. *Arxiv2207.01987*. 1

[2] A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *ECCV 2022*. 1

[3] 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*, pages 9902–9912, 2022. 2

[4] 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*, pages 1–11, 2021. 2

[5] Christoph Lassner and Michael Zollhöfer. Pulsar: Efficient sphere-based neural rendering. *arXiv:2004.07484*, 2020. 2

[6] Lanxiao Li and Michael Heizmann. A closer look at invariances in self-supervised pre-training for 3d vision. *arXiv preprint arXiv:2207.04997*, 2022. 2

[7] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 2

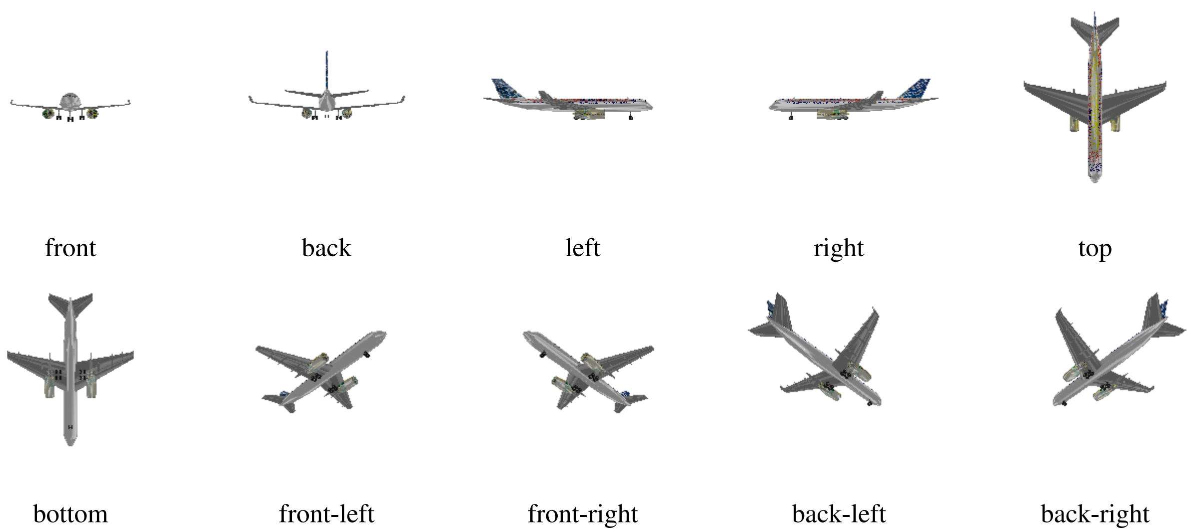

Figure 1. Visualization of multi-view RGB images for an airplane.

Figure 2. Rendered RGB images of Category 1 $\sim$ Category 20 on ShapeNet.

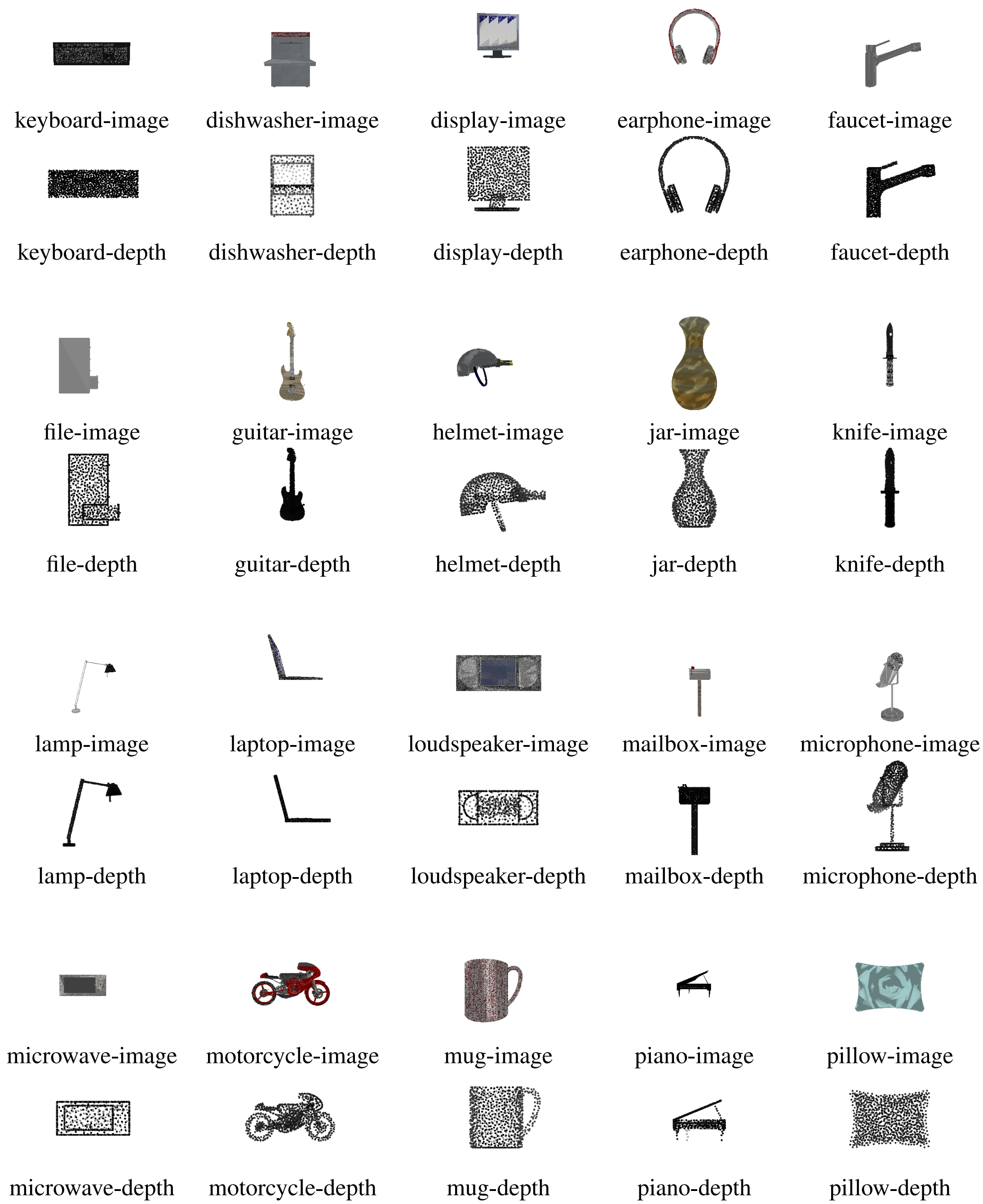

Figure 3. Rendered RGB images of Category 21 $\sim$ Category 40 on ShapeNet.

Figure 4. Rendered RGB images of Category 41 $\sim$ Category 55 on ShapeNet.