Supplementary Material

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

A. Datasets

We evaluate our PointCLIP on three well-known datasets: ModelNet10 [8], ModelNet40 [8] and ScanObjectNN [6]. ModelNet10 consists of 4,899 synthetic meshed CAD models with 10 indoor categories, 3,991 for training and 908 for testing. ModelNet40 is larger with 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's input. ScanObjectNN includes 2,321 training and 581 testing point clouds of 15 categories collected directly by real-world scans. Different from synthetic data with complete profiles, objects in ScanObjectNN are occluded at different levels and disturbed with background noise, which makes them more challenging for accurate recognition.

B. Implementation Details

For ablation studies of projection 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 as: 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) [3] with momentum 0.9 is adopted as the optimizer. We utilize Smooth Loss [7] following [1] and the cosine scheduler for learning rate decay. In ModelNet10 and ModelNet40, We apply random scaling and translation for training augmentation, but for the challenging ScanObjectNN, we append jitter and random rotation following [4]. 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 data augmentation and loss functions referring to their papers.

[*] Indicates equal contributions, † Indicates corresponding author

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 1. Specifically, w/o global denotes the adapter processing each view separately without interaction, and w/o view-wise simply repeats the global feature as the adapted feature. The 2-layer adapter removes the linear layer after the global representation and the pre-layer moves it before the global extraction. The results indicate the significance of inter-view extraction for the global feature and the view-wise adapted feature, without which hurt the performance by -3.33% and -1.27%, respectively.

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 1. Ablation studies (%) concerning different architectures of the inter-view adapter for 16-shot PointCLIP on ModelNet40.

Adapted Features Fusion. The view-wise adapted feature is generated by the adapter and then blended with the original CLIP-encoded feature via a residual connection. On ModelNet40, we evaluate the performance of 16-shot PointCLIP with different fusion ratios, which denotes the relative proportion of fusing adapted features. To show the effect of ratio, we set all view weights the same. As shown

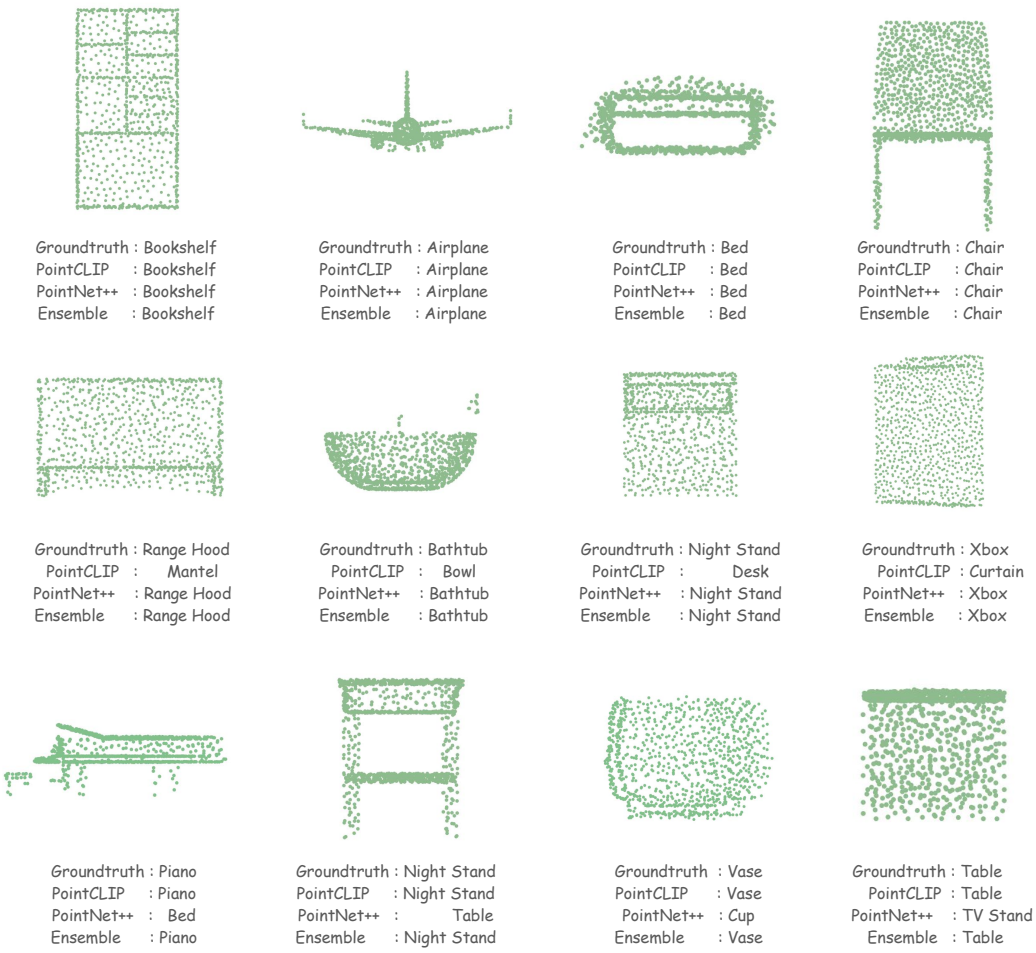

Figure 1. Visualization of recognition results from PointCLIP, PointNet++ [5] and the model with multi-knowledge ensemble.

in Table 2, different ratios lead to slight performance variance and the default ratio of 0.6 performs better than others. This indicates the comparable contributions between 2D pre-trained knowledge and 3D learned knowledge.

Adapter Fusion Ratios					
0.0	0.2	0.4	0.6	0.8	1.0
9.56%	85.78%	85.76%	86.13%	85.85%	85.53%

Table 2. Different fusion ratios of view-wise adapted feature for 16-shot PointCLIP on ModelNet40.

Full Training Set. We also fine-tune PointCLIP on the full training set of ModelNet40 [8] and present the results in Table 3. 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.

Fine-tuning on Full ModelNet40 [8]					
RN50	RN101	ViT/32	ViT/16	RN.×4	**RN.×16**
91.09%	91.69%	90.70%	91.76%	91.93%	**92.01%**

Table 3. Fine-tuning PointCLIP on full training set of ModelNet40 with different visual encoders.

Fine-tuning Settings. Under full training set of ModelNet40 [8], we further fine-tune different modules of PointCLIP in Table 4, where we adopt ResNet-101 [2] as the visual encoder. The fine-tuning without the inter-view adapter represents unfreezing the visual or textual encoder upon the zero-shot PointCLIP. As presented, unfreezing the textual encoder normally hurts the performance, and only fine-tuning the inter-view adapter obtains the best accuracy.

Visual Encoder	Textual Encoder	Inter-view Adapter	Acc.
✓	-	-	91.01
-	✓	-	73.89
-	-	✓	**91.69**
✓	-	✓	90.99
-	✓	✓	88.82

Table 4. Ablations of PointCLIP fine-tuning different modules. ✓denotes fine-tuning and - denotes freezing.

D. Visualization

We visualize some cases of multi-knowledge ensemble of PointCLIP and PointNet++ [5] to reveal the effectiveness of the enhancement. As shown in Figure 1, when two models both predict correctly for the first four samples, the ensemble model would preserve the prediction. As for samples in the other two rows, PointCLIP and PointNet++ show the complementary properties that the ensemble model would rectify one of their wrong predictions.

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.