Self-Supervised Learning with Multi-View Rendering for 3D Point Cloud Analysis – Supplementary Material –

Bach Tran[1], Binh-Son Hua[1], Anh Tuan Tran[1], and Minh Hoai[1,2]

[1] VinAI Research, Hanoi, Vietnam
[2] Stony Brook University, New York, 11794, USA
{v.bachtx12,v.sonhb,v.anhtt152,v.hoainm}@vinai.io

Abstract. In this document, we provide more details for our proposed method. Firstly, we present the complete experiment of using different rendering styles, including position encoding, silhouette, and shading for both PointNet and DGCNN (Section 1). We also include additional results for rest of object classification task with PointNet backbone(Section 2.1), 6-fold cross validation results for the semantic segmentation task on S3DIS dataset (Section 2.3). Secondly, we also report the performance of training with limited data on both ModelNet40 and ScanObjectNN (Section 3). Finally, we report detail settings, runtime statistics and more insights into the proposed method by analyzing the t-SNE embedding and the critical point and upper-bound point visualizations (Section 4).

1 Evaluation of multi-view rendering

We report the performance of PointNet [7] and DGCNN [12] on different rendering styles in Table 1. It can be seen that shaded images yield slightly higher performance than other renderings on both datasets. However, other rendering styles such as position encoding (RGB) and silhouette still produce competitive results. It implies that in cases where only point clouds are available for pre-training, RGB and silhouette rendering can be used while not causing a significant performance difference compared to mesh-based rendering.

2 Details of downstream tasks

2.1 Object classification

Similar to the comparison with the DGCNN backbone in the main paper, we provide comparisons with the PointNet backbone. The results are shown in Table 2. As can be seen, our method outperforms random inititalization as well as other pre-training methods, including Jigsaw [8], OcCo [11], and CM [5].

	PointNet			DGCNN		
	RGB	Silhouette	Shading	RGB	Silhouette	Shading
ModelNet40 [13]	88.3 ± 0.2	**88.9 ± 0.2**	**88.9 ± 0.1**	**92.5 ± 0.2**	**92.5 ± 0.2**	**92.5 ± 0.1**
ScanObjectNN [10]	**79.7 ± 0.5**	78.8 ± 0.6	79.3 ± 0.3	**82.8 ± 0.5**	82.0 ± 0.2	**82.8 ± 1.0**
ScanObjectNN BG [10]	75.1 ± 0.3	75.6 ± 0.4	**75.7 ± 0.5**	81.0 ± 0.2	81.8 ± 0.9	**82.6 ± 0.7**

Table 1: Effect of different rendering techniques to our pre-training

	PointNet				
	Random	Jigsaw	OcCo	CM	Ours
MN40 [13]	88.9±0.0	89.2±0.0	89.2±0.1	89.1±0.1	**89.5±0.2**
SO [10]	78.2±0.1	79.4±0.1	79.5±0.1	79.3±0.5	**80.5±0.4**
SO BG [10]	76.4±0.0	76.4±0.4	76.4±0.1	74.1±0.2	**78.5±0.5**

Table 2: Comparison among random, Jigsaw [8], OcCo [11], CM [5], and our initialization to the object classification downstream task. We reported the mean and standard deviation at the best epoch over three runs.

2.2 A note on the OcCo baseline

It can be seen that in our paper, some experiment results of OcCo are lower than the results reported by its original paper. We did our best to reproduce the results of OcCo but unfortunately, we were not able to match the results with the original paper. We confirmed this issue by using the docker image provided by the OcCo authors and rerun the experiments, but still could not reproduce the results exactly as in the OcCo paper. For fair comparison and reproducibility, we decided to report the results based on our own runs. Additionally, the pre-training time of OcCo is about 7x slower than our method.

2.3 Semantic segmentation

In additional to the Area-5 results reported in the main paper, we further report the results of 6-fold cross-validation over the 6 areas on the S3DIS dataset. For completeness, all results are shown in Table 3 (Area-5), and Table 4 (6 folds). In both cases, we can see that models initialized by our method outperform others in both PointNet [7] and DGCNN [12].

2.4 Details of PointContrast baseline

Semantic segmentation: We evaluate on two datasets S3DIS [1] and ScanNet [3]. We use SGD optimizer with the initial learning rate 0.1 and 0.8 for S3DIS and ScanNet respectively. We use PolynomialLR scheduler with a power factor 0.9.For ScanNet dataset, we train the model with 15000 iterations and batch size

	PointNet				DGCNN			
	Random	Jigsaw	OcCo	Ours	Random	Jigsaw	OcCo	Ours
mAcc	83.9	82.5	83.6	**85.0**	86.8	86.8	**87.0**	**87.0**
mIoU	43.6	43.6	44.5	**46.7**	49.3	48.2	49.5	**49.9**

Table 3: Fold 1 of overall point accuracy (mAcc) and mean Intersection-over-Union (mIoU) on the S3DIS (Stanford Area 5 Test) [1]

	PointNet				DGCNN			
	Random	Jigsaw	OcCo	Ours	Random	Jigsaw	OcCo	Ours
mAcc	82.8	82.8	82.7	**83.2**	86.9	86.6	87.1	**87.5**
mIoU	50.6	51.4	51.1	**52.1**	58.4	58.1	58.7	**59.0**

Table 4: Average of 6-fold cross validation of overall point accuracy (mAcc) and mean Intersection-over-Union (mIoU) on the S3DIS [1]

48 on 4 GPUs. For S3DIS dataset, we train the model with 20000 iterations and batch size 32 on 1 GPU.
Object detection: For object detection task, in the training we follow the configuration of PointContrast [14]. We use Adam optimizer with the initial learning rate 0.001 and train the model on 1 GPU with 180 epochs. Specifically, we train the model with batch size 32 and 64 for ScanNet and SUN RGB-D, respectively. Before voxelization, we sample 40000 and 20000 points from original point of ScanNet and SUN RGB-D and the voxel sizes are 2.5 cm and 5 cm respectively.

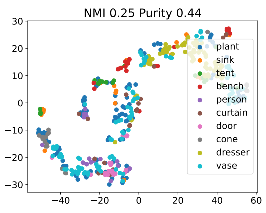
3 Training with limited data

To prove the effectiveness of our pre-training, we supervise the downstream task with fewer data when the network is pre-trained and compare to other initializations. We show both results on ModelNet40 (also reported in the paper) and ScanObjectNN. In this experiment, we decrease the number of training samples to 5%, 10%, 20%, 50%, and 80%, and evaluate the model on the original test set. The results are reported in Table 5, which shows that the performance of our method outperforms Random, Jigsaw [8], and OcCo [11] in most cases except DGCNN on 80% of ScanObjectNN.

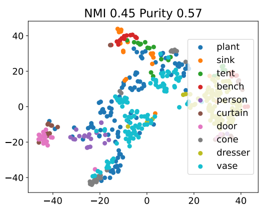
4 Visualization

4.1 t-SNE embedding

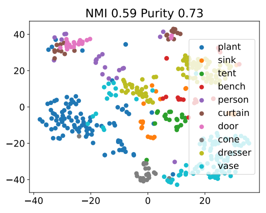
We further visualize learned object embeddings of the ModelNet40 test set by using t-SNE with perplexity 15 and 1000 iterations in Figure 1. We observe that



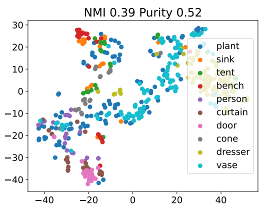
(a) Jigsaw [8] on PointNet



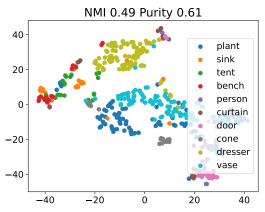
(b) OcCo [11] on PointNet



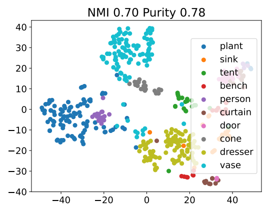
(c) Ours on PointNet



(d) Jigsaw [8] on DGCNN



(e) OcCo [11] on DGCNN



(f) Ours on DGCNN

Fig. 1: t-SNE visualization of the object embedding of the test data of ModelNet40. Our method has better cluster quality measured by NMI and purity.

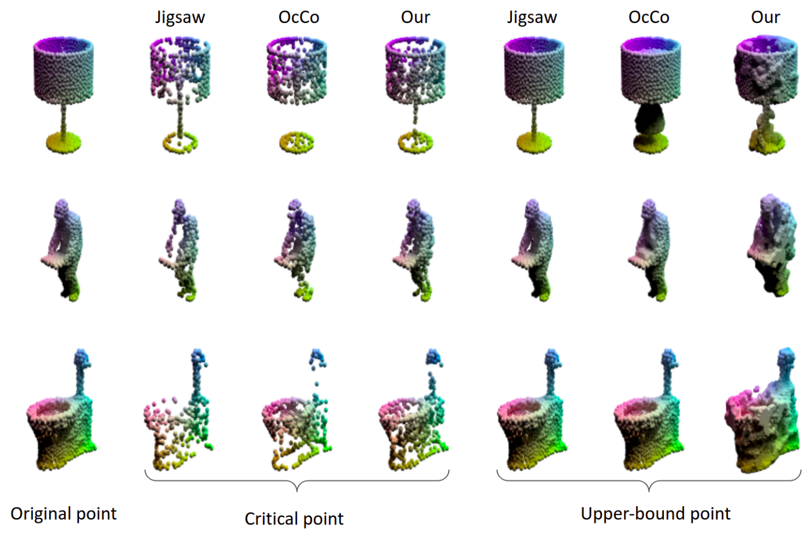


Fig. 2: Critical and upper-bound point visualizations for models pretrained with Jigsaw, OcCo, and our method.

ModelNet40 [13]	PointNet				DGCNN			
	Random	Jigsaw	OcCo	Ours	Random	Jigsaw	OcCo	Ours
5%	73.2	73.8	73.9	**77.9**	82.0	82.1	82.3	**84.9**
10%	75.2	77.3	75.6	**79.0**	84.7	84.1	84.9	**86.6**
20%	81.3	82.9	81.6	**84.6**	89.4	89.2	89.1	**90.2**
50%	86.6	86.5	87.1	**87.6**	91.6	91.8	91.7	**92.4**
70 %	88.3	88.4	88.4	**88.7**	92.3	92.4	92.5	**92.8**
ScanObjectNN [10]	PointNet				DGCNN			
	Random	Jigsaw	OcCo	Ours	Random	Jigsaw	OcCo	Ours
5%	52.1	51.8	53.7	**60.8**	48.3	46.7	51.4	**60.9**
10%	63.0	62.3	62.5	**69.0**	58.7	58.0	61.5	**69.5**
20%	69.0	68.5	67.1	**72.2**	69.8	68.7	71.6	**74.7**
50%	73.7	75.1	72.6	**77.0**	76.3	77.1	78.0	**81.6**
80 %	76.1	77.9	76.7	**78.4**	79.9	78.1	80.8	**82.1**

Table 5: Performance of the object classification task trained with fewer data on ModelNet40 [13] and ScanObjectNN [10]. Our method has significant gains compared to other initialization methods. We reported the mean at the best epoch over three runs

the embeddings learned from using our initialization for different classes are well clustered than those acquired with OcCo and Jigsaw initialization indicated by normalized mutual information (NMI) and purity [6].

4.2 Critical point sets

We then visualize the critical point sets and upper-bound shapes by following PointNet [7] for selected samples in Figure 2. To recap, a critical point set is a set of points that contribute directly to the last max pooled feature, i.e., the global feature. Perturbing the critical point set can lead to changes in the global features and thus classification results. The upper-bound shape is the largest possible point set that does not directly affect the global feature of the original shape. From Figure 2, we found that in our method, the critical point sets can represent the object skeleton more faithfully (e.g., the toilet example) than other methods. Jigsaw sometimes causes sparse critical points, and OcCo tends to discard points along thin geometric features. We also found that the upper-bound shape of our initialization appears thicker than that of Jigsaw and OcCo, which we hypothesize that our model can be more robust to point perturbations than Jigsaw and OcCo.

5 Running time

Following the request, we provide the pre-training time of three methods on **an NVIDIA Tesla V100 GPU** in Table 6. As can be seen, the pre-training time of our method is slightly longer than Jigsaw and significantly shorter than OcCo. Despite such, our method achieves better performance than the others.

	Jigsaw	OcCo	Ours
PointNet	2.6	24.8	3.8
DGCNN	4.1	35.1	5.7

Table 6: Pre-training time of three methods (hours).

Additionally, we provide more statistics of our training process. Specifically, it takes 2.3, 2.4, and 6.2 hours to render RGB, silhouette, and shaded images, respectively. For the 2D self-supervision, we train the model for 80 hours on an NVIDIA Tesla V100 GPU. Knowledge distillation takes 3.8, 5.7, 26 and 62 hours of training for PointNet [7], DGCNN [12], SR-UNet on ModelNet40 [13] and SR-UNet on ScanNet [3], respectively. As for downstream-task training, the PointNet classification model takes 18.5 hours, and the DGCNN classification model takes 75.0 hours. The segmentation models require longer training time, with 32.0 hours and 90.0 hours for PointNet and DGCNN backbone, respectively. For SR-UNet backbone[2], in semantic segmentation task, it consumes 32 and 22 hours for S3DIS [1] and ScanNet [3], respectively. For object detection task on ScanNet [3], it takes 8.5 hours.

6 Future Work

Our method is not without limitations. First, our image encoder is trained from scratch without leveraging existing popular feature extractors such as VGG [9] or ResNet [4]. Further utilizing such pre-trained networks on natural images could potentially improve the performance of the downstream tasks, which could be interesting for future work. Second, the multi-view rendering used in our method could potentially be further explored. While we attempted with position encoding, silhouette, and shaded rendering, there are many other rendering styles that could be experimented, e.g., rendering with colors and textures when applicable, rendering with depth completion, etc. Applying advanced techniques to enhance multi-view rendering is thus a good avenue for future research.

References

1. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1534–1543 (2016)
2. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)
3. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2017)

4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
5. Jing, L., Zhang, L., Tian, Y.: Self-supervised feature learning by cross-modality and cross-view correspondences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1581–1591 (2021)
6. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008), http://nlp.stanford.edu/IR-book/information-retrieval-book.html
7. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
8. Sauder, J., Sievers, B.: Self-supervised deep learning on point clouds by reconstructing space. Advances in Neural Information Processing Systems **32** (2019)
9. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
10. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, D.T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: International Conference on Computer Vision (ICCV) (2019)
11. Wang, H., Liu, Q., Yue, X., Lasenby, J., Kusner, M.J.: Unsupervised point cloud pre-training via occlusion completion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9782–9792 (2021)
12. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5), 1–12 (2019)
13. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
14. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: European Conference on Computer Vision. pp. 574–591. Springer (2020)