Supplementary for "Point Cloud Domain Adaptation via Masked Local 3D Structure Prediction"

Hanxue Liang[1], Hehe Fan[2], Zhiwen Fan[1], Yi Wang[1],
Tianlong Chen[1], Yu Cheng[3], Zhangyang Wang[1]

[1] The University of Texas at Austin
[2] National University of Singapore
[3] Microsoft Research Redmond

1 Implementation Details

1.1 Network architecture

In both classification and segmentation experiments, all the network configurations are the same except for the supervision head. We use DGCNN [4] as the shared feature extractor, which is composed of four point cloud convolution layers of sizes [64, 64, 128, 256] and 1D convolution layer with kernel size 1 and channel number 1024. The output is max-pooled to get global feature. For SSL heads, both normal prediction h_{norm} and position prediction head h_{pos} are implemented similarly with four 1D convolution layers of sizes [256, 256, 128, 3]. The cardinality prediction head is implemented with a 1D convolution layer of size 512 and three fully connected layers of sizes [256, 256, 8] (where 8 is the number of cardinality classes). In PointDA-10 benchmark, the classification head h_{sup} is implemented using three fully connected layers with sizes [512, 256, 10] and takes as input the global feature vector. In PointSegDA benchmark, the segmentation head takes the global feature vector concatenated to feature representations of each point, and it is implemented using four 1D convolution layers of sizes [256, 256, 128, 8]. In all the heads, non-linearity is ReLU and a dropout of 0.5 is applied to the first two hidden layers. Batch normalization is applied after all convolution layers.

1.2 PointDA-10 dataset

Data processing We follow the data preparation procedure used in [1]. Specifically, all object point clouds in all datasets are aligned along the direction of gravity, while arbitrary rotations along the z axis are allowed. Each sample is down-sampled to 1024 points and normalized within a unit ball. A typical 80%/20% data split for training and validation on both source and target domains are employed. During training, jittering with standard deviation is applied and objects are randomly rotated along Z axis.

Table 1: Apply MLSP on source and target datasets on PointDA-10

Domain	M→S	M→S*	S→M	S→S*	S* →M	S* →S	Average
S/T	84.2	56.9	79.2	53.4	76.1	75.4	70.8
T	83.7	55.4	77.1	55.6	78.2	76.1	71.0

Training procedure During training, we alternate between a batch of source samples and a batch of target samples in all methods. We apply a grid search over the SSL loss $\alpha_1, \alpha_2, \alpha_3 \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$ and we empirically set them to be 0.05, 0.5, 0.5 respectively. During entropy-guided self-paced training, we select target samples with pseudo labels based on the entropy of each prediction. After selecting the pseudo labels, these data will be trained with classification loss together with source data.

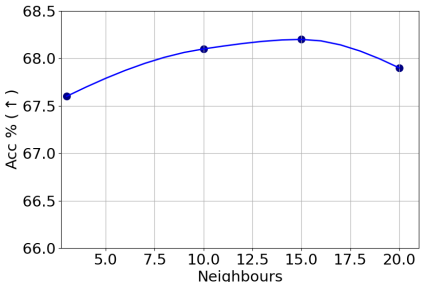

Fig. 1: Classification Accuracy under target domain with respect to different number of neighbouring points.

1.3 PointSegDA dataset

Data processing The data is pre-processed similarly to classification case, while each sample is downsampled by farthest point sampling to 2048 points.

Training procedure The batch size is set to 16 per domain and the number of epochs is set to 200. $\alpha_1, \alpha_2, \alpha_3$ are selected by grid search in $\{0.01, 0.02, 0.05, 0.1, 0.2\}$ and we set them to be 0.01, 0.02, 0.2 empirically.

2 More Quantitative Results

2.1 Apply MLSP on source domain data

Table 1 demonstrates the result when we apply MLSP on both source and target data, and it shows that it degrades the result.

2.2 Evaluation of class-wise classification accuracy (%) on the ModelNet-10 to the ScanNet-10 (M→S*)

We report class-wise classification accuracy on the task M→S* in Table 2. Compared with existing methods, our proposed methods achieve the better performance on most classes, especially on major classes such as Chair and Table. On long-tailed classes like Bathtub and Lamp, we also get competitive results.

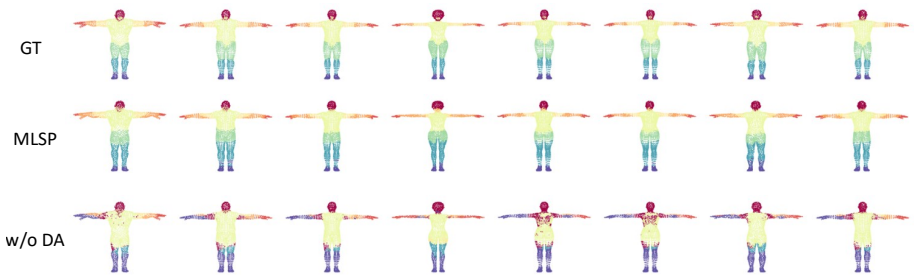

Fig. 2: Visualization of the adaptation result from Faust to Adobe test set.

Table 2: Comparative evaluation in classification accuracy (%) averaged over 3 seeds (± SEM) on ModelNet-10 to ScanNet-10.

Methods	SSL	Pseudo Label	Bathtub	Bed	Bookshelf	Cabinet	Chair	Lamp	Monitor	Plant	Sofa	Table	Avg.
Supervised			76.9	58.8	55.5	73.2	92.5	63.4	70.5	72.0	56.0	85.0	70.4
Baseline (w/o adap.)			61.5	31.8	32.9	0	49.8	36.6	54.1	96.0	30.6	47.5	44.1
DANN [2]			34.6	38.8	34.2	2.7	59.4	12.2	49.2	84.0	**53.0**	57.8	42.6
PointDAN [3]			34.6	36.5	35.6	3.4	61.2	29.3	37.7	76.0	44.8	45.5	40.4
DefRec+PCM [1]	✓		**65.4**	49.4	**49.3**	1.3	61.4	41.4	55.7	88.0	42.5	60.8	51.5
GAST [5]	✓		57.7	38.8	35.6	2.0	74.3	43.9	77.0	**96.0**	45.5	74.1	54.5
MLSP	✓		58.1	**65.6**	43.7	**3.9**	80.0	56.7	59.0	69.7	45.9	71.4	55.4
MLSP+EGA	✓	✓	52.2	39.9	41.8	1.5	**80.3**	**71.7**	**81.5**	85.9	50.6	**85.6**	**59.1**

3 Qualitative Results

3.1 Segmentation

Figure 2 demonstrates the adaptation result from Faust to Adobe test set. Three rows show the result of groudtruth, MLSP adaptation and without adaptation. Images of the same person are shown in each column. Evidently, our method achieves superior performance qualitatively.

3.2 Cardinality, position and normal prediction

Figure 3 demonstrates the cardinality, position and normal prediction on ScanNet-10 dataset. The images of one class are shown in each row and each column shows the ground-truth or prediction of all the classes. Images of the same object are presented in the following order from left to right: initial point cloud, the cardinality ground-truth (different colors shows cardinality classes), masked point cloud ground-truth (the input to the network, red points are masked), the normal ground-truth of masked points, the cardinality prediction, masked points prediction and the normal prediction of masked points. Although it is not accurate and there is some noise, the network can learn to predict the masking area as well as the point cardinality and normals.

Table 3: Effect of number of cardinality classes K.

K	M→S	M→S*	S→M	S→S*	S* →M	S* →S	Average
4	83.4	53.3	71.9	51.7	72.4	75.0	68.0
8	83.0	54.3	74.0	53.5	71.9	75.6	68.7
16	83.7	53.5	72.1	53.6	72.4	74.7	68.3
32	83.2	54.3	72.4	53.3	72.7	73.3	68.2

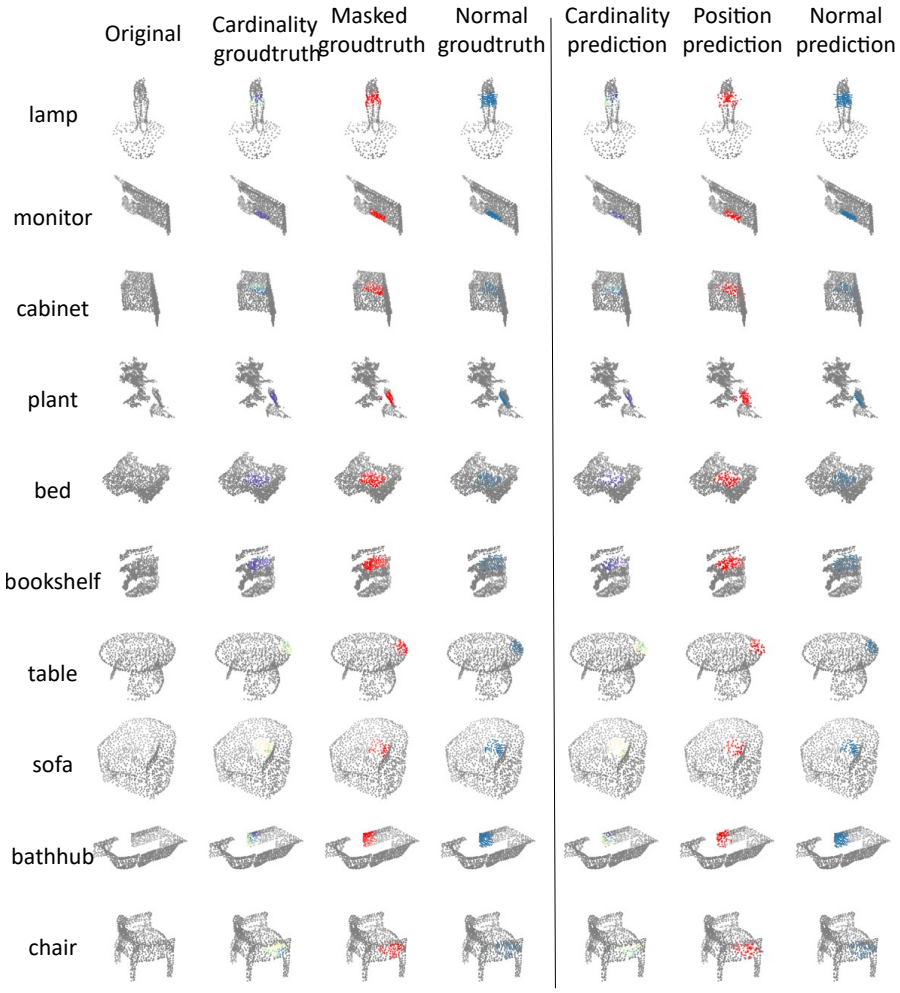

Fig. 3: The visualization of point cardinality, position and normal prediction on ScanNet-10 dataset.

References

1. Achituve, I., Maron, H., Chechik, G.: Self-supervised learning for domain adaptation on point clouds. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 123–133 (2021)
2. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The journal of machine learning research **17**(1), 2096–2030 (2016)
3. Qin, C., You, H., Wang, L., Kuo, C.C.J., Fu, Y.: Pointdan: A multi-scale 3d domain adaption network for point cloud representation. Advances in Neural Information Processing Systems **32** (2019)
4. 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)
5. Zou, L., Tang, H., Chen, K., Jia, K.: Geometry-aware self-training for unsupervised domain adaptation on object point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6403–6412 (2021)