Domain Adaptation on Point Clouds via Geometry-Aware Implicits — Supplementary Material —

Yuefan Shen[1*] Yanchao Yang[2†*] Mi Yan[3] He Wang[3] Youyi Zheng[1†] Leonidas Guibas[2]

[1]Zhejiang Univerisy[‡] [2]Stanford Univerisy [3]Peking Univerisy

1. Ablation on Random Masking

We quantitatively verify the effectiveness of our proposed adaptive unsigned distance (AUD) and our random masking, i.e., the loss term in Eq. (5), by performing an ablation study on the classification adaptation on PointDA-10. We train all six different settings for our method without AUD or random masking and with AUD and random masking, respectively. The comparisons are reported in Tab. 1.

From Tab. 1, we can make three key observations:

- (a): Besides the visual comparison of the reconstructed point clouds using nearest neighbor distance and AUD shown in Fig. 6 in our main paper, the quantitative comparison of adaptation results shows that AUD significantly improves over the non-adaptive one.
- (b): when adaptations involve S* (ScanNet, a dataset where lots of point clouds are in a partial form) in either the source or the target domain, random masking will significantly increase the performance, which confirms that our random masking scheme can improve the effectiveness of our method.
- (c): The scores of adaptations between two synthetic datasets are comparable without and with random masking, which proves that our random masking can be considered as a general augmentation for all the datasets.

These three observations confirm that our proposed AUD and random masking scheme are necessary and effective.

Methods	AUD	Masking	M→S	M→S*	S→M	S→S*	S*→M	S*→S	Avg.
Supervised			93.9 ± 0.2	78.4 ± 0.6	96.2 ± 0.1	78.4 ± 0.6	96.2 ± 0.1	93.9 ± 0.2	89.5
Baseline (w/o adap.)			83.3 ± 0.7	43.8 ± 2.3	75.5 ± 1.8	42.5 ± 1.4	63.8 ± 3.9	64.2 ± 0.8	62.2
Ours			82.3 ± 0.4	48.4 ± 0.5	71.6 ± 0.8	49.8 ± 0.7	67.1 ± 0.3	65.8 ± 0.7	64.2
	✓		85.4 ± 0.5	53.2 ± 0.2	**77.3** ± 0.5	53.6 ± 0.7	71.9 ± 0.6	69.8 ± 0.8	68.5
	✓	✓	**85.8** ± 0.3	**55.3** ± 0.3	77.2 ± 0.4	**55.4** ± 0.5	**73.8** ± 0.6	**72.4** ± 1.0	**70.0**

Table 1. Classification accuracy (%) averaged over 3 seeds (± SEM) on the PointDA-10 dataset. M: ModelNet, S: ShapeNet, S*: ScanNet; → indicates the adaptation direction. Masking: the proposed random masking scheme.

2. Details of GraspNetPC-10

Fig. 1 shows all the selected models from the GraspNet dataset [3]. Most of the models have completely different shapes, and some models have similar shapes, e.g., Pear and Mouse, Can and Shampoo.

[‡]The authors from Zhejiang University are affiliated with the State Key Lab of CAD&CG. *Equal Contributions, †Corresponding Authors.

Figure 1. All of the selected models from the GraspNet dataset [3].

3. Analysis on Geometry-Aware Implicits

Methods	MMD^l	MMD^p	WD	KL	JS
Baseline(w/o adap.)	115.0	6.3e+5	444.66	6.20	0.56
DefRec+PCM [1]	13.91	3012.0	63.123	0.23	0.18
GAST [8]	69.11	4.0e+5	560.53	3.22	0.39
Ours	**0.134**	**0.5306**	**1.0450**	**1e-3**	**3e-4**
Baseline(w/o adap.)	183.2	2.0e+6	1180.4	9.55	0.52
DefRec+PCM [1]	15.09	3.7e+4	156.52	0.79	0.15
GAST [8]	94.92	9.8e+5	962.28	3.58	0.37
Ours	**0.195**	**1.8017**	**2.5340**	**2e-3**	**4e-4**

Table 2. Domain distances between the source and target domains without or with different domain adaptation methods. MMD^l: linear MMD (Maximum Mean Discrepancy), MMD^p: MMD with polynomial kernel, WD: Wasserstein dist., KL and JS divergence. Top: ModelNet to ScanNet in PointDA-10 dataset, bottom: Syn. to Kin. in GraspNetPC-10 dataset. Our method achieves the smallest domain discrepancy.

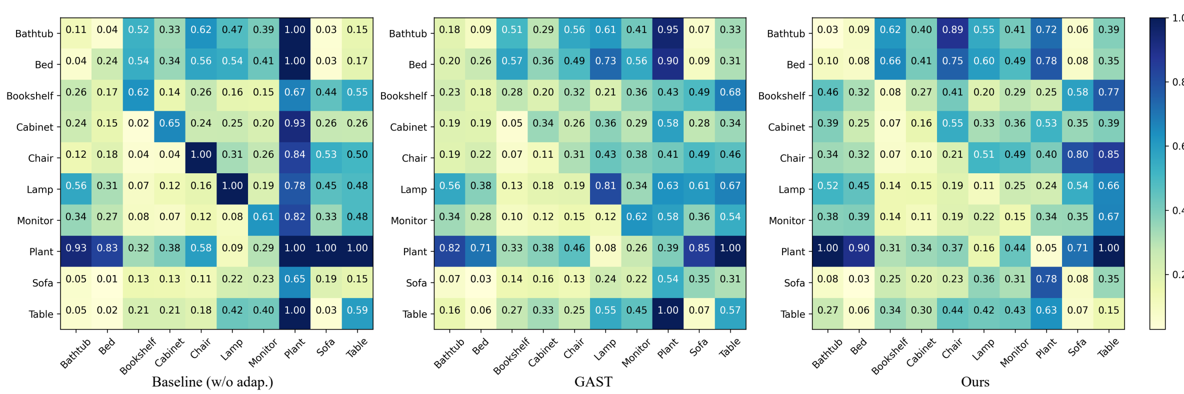


Figure 2. Class-wise MMD^p for the task: ModelNet to ScanNet in PointDA-10 dataset. Diagonal shows source-target distances of the same class. Ours is much smaller. Upper and lower triangular matrices indicate distances between different classes in the source and target domain, respectively. Our method maintains class-wise distances well.

Our high-level insight is that our proposed geometry-aware implicit preserves geometric information of the point clouds, while learning away the domain-specific variations. We add analysis on domain distances in Tab. 2 and Fig. 2. Compared to the Baseline (w/o adapt.) and other domain adaptation methods, our method can significantly decrease the domain distribution gap between the source and target domains. Meanwhile, domain distances between different classes in the target domain are apparent, like in the source domain, which is beneficial for classification using models trained with source domain data.

4. Methods with SPST

In Tab. 3, we further compare our method to the state-of-the-art methods with SPST(self-paced self-training). One can see that our method still outperforms the others due to the efficient initial adaptation with the proposed geometry-aware implicits.

Methods	Adv.	SLT	SPST	Syn.→Kin.	Syn.→RS.	Kin.→RS.	RS.→Kin.	Avg.
Supervised				97.2 ± 0.8	95.6 ± 0.4	95.6 ± 0.3	97.2 ± 0.4	96.4
Baseline (w/o adap.)				61.3 ± 1.0	54.4 ± 0.9	53.4 ± 1.3	68.5 ± 0.5	59.4
PointDAN [4]	✓			77.0 ± 0.2	72.5 ± 0.3	65.9 ± 1.2	82.3 ± 0.5	74.4
	✓		✓	91.8 ± 0.3	79.1 ± 0.6	72.3 ± 0.5	84.7 ± 0.2	82.0
DefRec+PCM [1]		✓		80.7 ± 0.1	70.5 ± 0.4	65.1 ± 0.3	77.7 ± 1.2	73.5
		✓	✓	92.7 ± 0.7	78.3 ± 0.4	73.6 ± 0.2	82.3 ± 0.3	81.7
GAST [8]		✓		69.8 ± 0.4	61.3 ± 0.3	58.7 ± 1.0	70.6 ± 0.3	65.1
		✓	✓	81.3 ± 1.8	72.3 ± 0.8	61.3 ± 0.9	80.1 ± 0.5	73.8
Ours		✓		81.2 ± 0.3	73.1 ± 0.2	66.4 ± 0.5	82.6 ± 0.4	75.8
		✓	✓	**94.6** ± 0.4	**80.5** ± 0.2	**76.8** ± 0.4	**85.9** ± 0.3	**84.4**

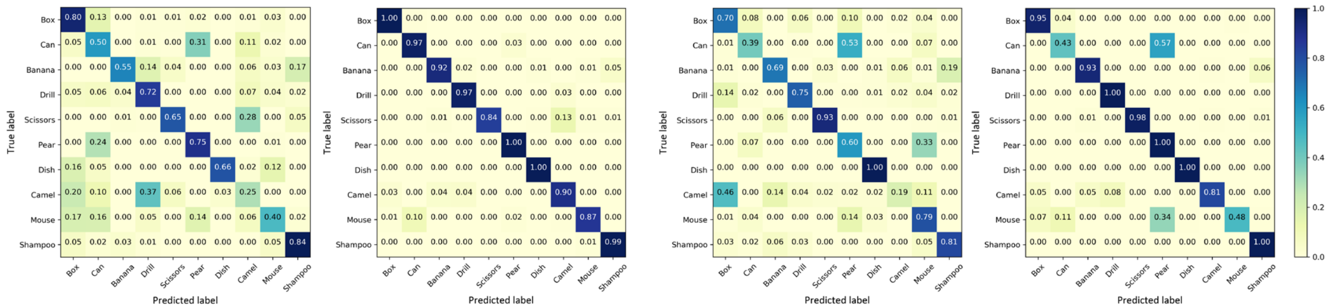
Table 3. Classification accuracy (%) averaged over 3 seeds (± SEM) on GraspNetPC-10. Syn.: Synthetic domain, Kin.: Kinect domain, RS.: Realsense domain. Our models achieve the best performance over all settings.

5. Per-Class Accuracy

Following [8], we also report class-wise classification accuracy in Tab. **??** and the selected experimental setting is Synthetic domain to Kinect domain on the GraspNetPC-10 dataset. We compare our method to the state-of-the-art methods without and with SPST, respectively. Our method reaches the highest scores on most of the objects.

Methods	SPST	Box	Can	Banana	Drill	Scissors	Pear	Dish	Camel	Mouse	Shampoo	Avg.
Supervised		99.6	97.3	94.9	96.9	93.8	100.0	100.0	94.1	96.5	99.2	97.2
Baseline (w/o adap.)		80.1	50.4	55.1	71.9	65.2	75.4	65.6	24.6	40.2	84.4	61.3
PointDAN [4]		97.3	79.7	62.5	86.3	66.0	81.6	100.0	71.1	39.1	85.9	77.0
	✓	99.2	93.4	89.1	**99.2**	80.9	99.2	100.0	91.8	69.1	96.5	91.8
DefRec+PCM [1]		100.0	85.9	64.5	88.7	73.0	81.6	100.0	81.3	32.4	98.8	80.7
	✓	**100.0**	93.8	87.5	98.8	83.2	100.0	100.0	**98.4**	66.4	98.8	92.7
GAST [8]		92.2	46.5	47.3	50.4	71.1	90.2	100.0	44.9	70.7	85.2	69.8
	✓	97.3	52.0	59.4	76.6	74.6	95.7	100.0	90.2	82.0	85.9	81.3
Ours		99.6	83.2	68.4	88.7	79.7	90.2	100.0	35.2	78.5	88.7	81.2
	✓	99.6	**97.3**	**92.2**	96.9	**84.0**	**100.0**	**100.0**	89.8	**86.7**	**99.2**	**94.6**

Moreover, visualizations of confusion matrices in terms of class-wise classification accuracy are shown in Fig. 3. We select results of the baseline without adaptation and our full adaptation method for Synthetic domain to Kinect domain and Realsense domain to Kinect domain. We find that, without adaptation, classifiers cannot distinguish similar objects well, like Can and Pear. For objects with complex geometries, e.g., Camel, classifiers without adaptation cannot recognize them well, neither. With our adaptation, we can significantly increase the classification accuracy for most of the objects. However, for the adaptation from Realsense domain to Kinect domain, adaptation reduces the performance on object Mouse to some degree because the classifier is confused on similar geometries (Pear and Mouse).

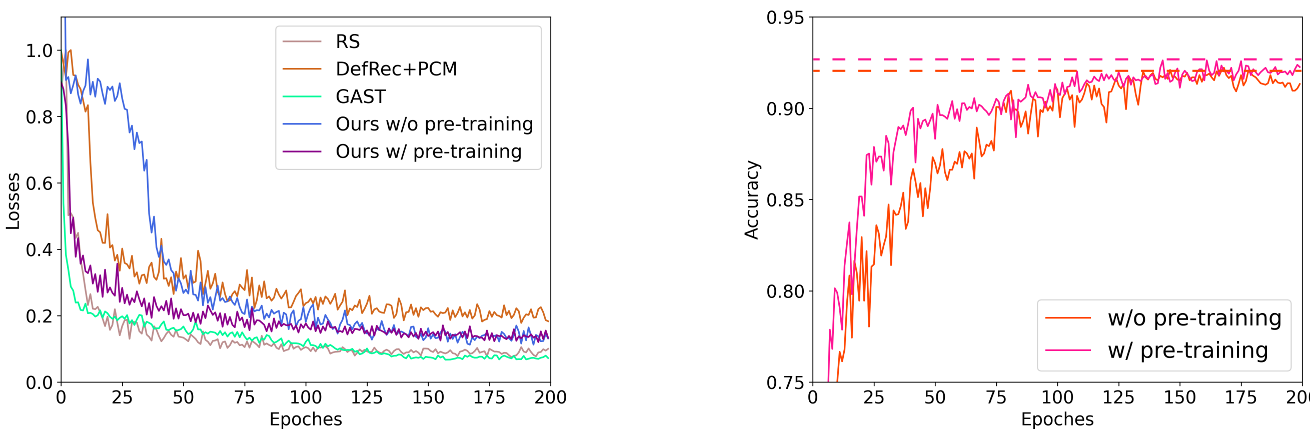


(a) w/o adap. (Syn.→Kin.) (b) Ours with SPST (Syn.→Kin.) (c) w/o adap. (RS.→Kin.) (d) Ours with SPST (RS.→Kin.)

Figure 3. Confusion matrices in terms of class-wise classification accuracy.

6. Analysis on Convergence

As mentioned in our main paper, models are trained for 200 epochs on PointDA-10. We train all models to convergence for fair comparisons. We show the loss decreasing for domain adaptation on PointDA-10 dataset in Fig. 4 (a). One can see that methods except ours all convergence before 150 epochs which means training continually will not bring changes to their performances. For our method without pre-training, the loss is a little difficult to decrease, especially at the beginning. The reason is that our self-supervised task is designed to learn the whole geometry representation and it is difficult to learn than the others. So that, we train our method to 200 epochs to make sure our method convergence. The delay in convergence is a limitation of our method, and more sophisticated sampling could be helpful and hopefully can improve the performance.



(a) (b)

Figure 4. (a) Different loss decreasing curves about domain adaptation. (DefRec+PCM: [1], GAST [8], RS: [5]) (b) Classification accuracy on ModelNet40 [7] with or without pre-training.

7. Standard Self-Supervised Experiment

We also analyze how our self-supervised learning affects the downstream task. Like Fig. 4 (b) shows, we use a pre-trained model on ShapeNet dataset [2] with our method and then train a classification network on ModelNet40 dataset [7] using DGCNN [6]. Our pre-train self-supervised method can speed up the convergence, increase the accuracy slightly and make training more robust.

References

[1] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 123–133, 2021. 2, 3, 4

[2] 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. 4

[3] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020. 1, 2

[4] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. *Advances in Neural Information Processing Systems*, 32:7192–7203, 2019. 3

[5] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *arXiv preprint arXiv:1901.08396*, 2019. 4

[6] 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. 4

[7] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 4

[8] Longkun Zou, Hui Tang, Ke Chen, and Kui Jia. Geometry-aware self-training for unsupervised domain adaptation on object point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6403–6412, 2021. 2, 3, 4