Synergizing Contrastive Learning and Optimal Transport for 3D Point Cloud Domain Adaptation

A. Implementation Details

In this section, we mention our implementation details for reproducibility purposes. For data pre-processing, we follow [1] and align the positive Z axis of all point clouds from the whole PointDA-10 dataset. We use the farthest point sampling algorithm to sample 1024 points uniformly across the object surface. Further, all the point clouds are normalized and scaled to fit in a unit-sphere. For getting renderings of point clouds from multiple views, we place orthographic cameras in a circular rig. We set the number of views to 12 and the image size as 224×224. We set points color to $white$, background color to $black$, points radius to 0.008, and points per pixel to 2. For getting two augmented versions of the original point cloud used in self-supervised contrastive learning, we compose spatial transformations picked from random point cloud scaling, rotation, and translation. The original point cloud which is passed to the source classifier is only transformed with random jittering and random rotation about its Z axis.

For a fair comparison with recent works [1, 4, 6] we use DGCNN [5] as our 3D encoder for extracting global point cloud features. We choose a pre-trained ResNet-50 [3] as our image feature extractor. Both 3D and 2D encoders embed their respective modality into a 256 dimensional feature space for contrastive learning. Whereas, for the classification task, 1024 dimensional feature vector of the original point cloud is used. We use a 3-layer MLP as our classifier, having $(512, 256, 10)$ neurons respectively. Please note that for testing the classification performance, we do not use 2D features and only use the global features given by the 3D encoder. We set the temperature parameter τ used in contrastive losses $\mathcal{L}_{3d}$ and $\mathcal{L}_{mm}$ as 0.1. To solve the optimization problem of the optimal coupling matrix, we use the POT library [2]. We do a grid search to find the best α and β combination from $\mathcal{L}_{ot}$. For most of the dataset combinations, we set the hyperparameters α and β to 0.001 and 0.0001, respectively.

We perform all our experiments on NVIDIA RTX-2080Ti GPUs using the Pytorch framework for implementing our models. We set the batch size to 32, learning rate as 0.001 with cosine annealing as the learning rate scheduler and use Adam optimizer. We set weight decay to 0.00005 and momentum to 0.9. In total, we train our models for 150 epochs on PointDA-10 and 120 epochs on GraspNetPC-10 dataset.

B. Class-wise Performance Analysis

In this section, we analyse the class-wise accuracy of COT on PointDA-10 and and GraspNetPC-10 datasets. Results for PointDA-10 and GraspNetPC-10 are show in Tables 1 and 2, respectively and the confusion matrices are shown in Figure 1 for Point DA-10 dataset and in Figure 2 for GraspNetPC-10 dataset.

	Bathtub	Bed	Bookshelf	Cabinet	Chair	Lamp	Monitor	Plant	Sofa	Table
S*→ M	0.98	0.98	0.99	0	0.98	0.9	0.86	0.9	0.97	0.98
S*→ S	0.86	0	0.98	0.05	0.96	0.67	0.65	0.8	0.36	0.95
M→ S	0.85	0.52	0.98	0	0.94	0.65	0.84	0.97	0.92	0.93
M→ S*	0.46	0.39	0.4	0.05	0.69	0.63	0.74	0.8	0.45	0.69
S→ S*	0.54	0	0.12	0	0.7	0.73	0.77	0.32	0.45	0.58
S→ M	1	0.99	0.62	0.57	0.99	0.95	1	0.91	0.98	1

Table 1. Class-wise accuracies of our COT (with SPST) on the PointDA-10 dataset

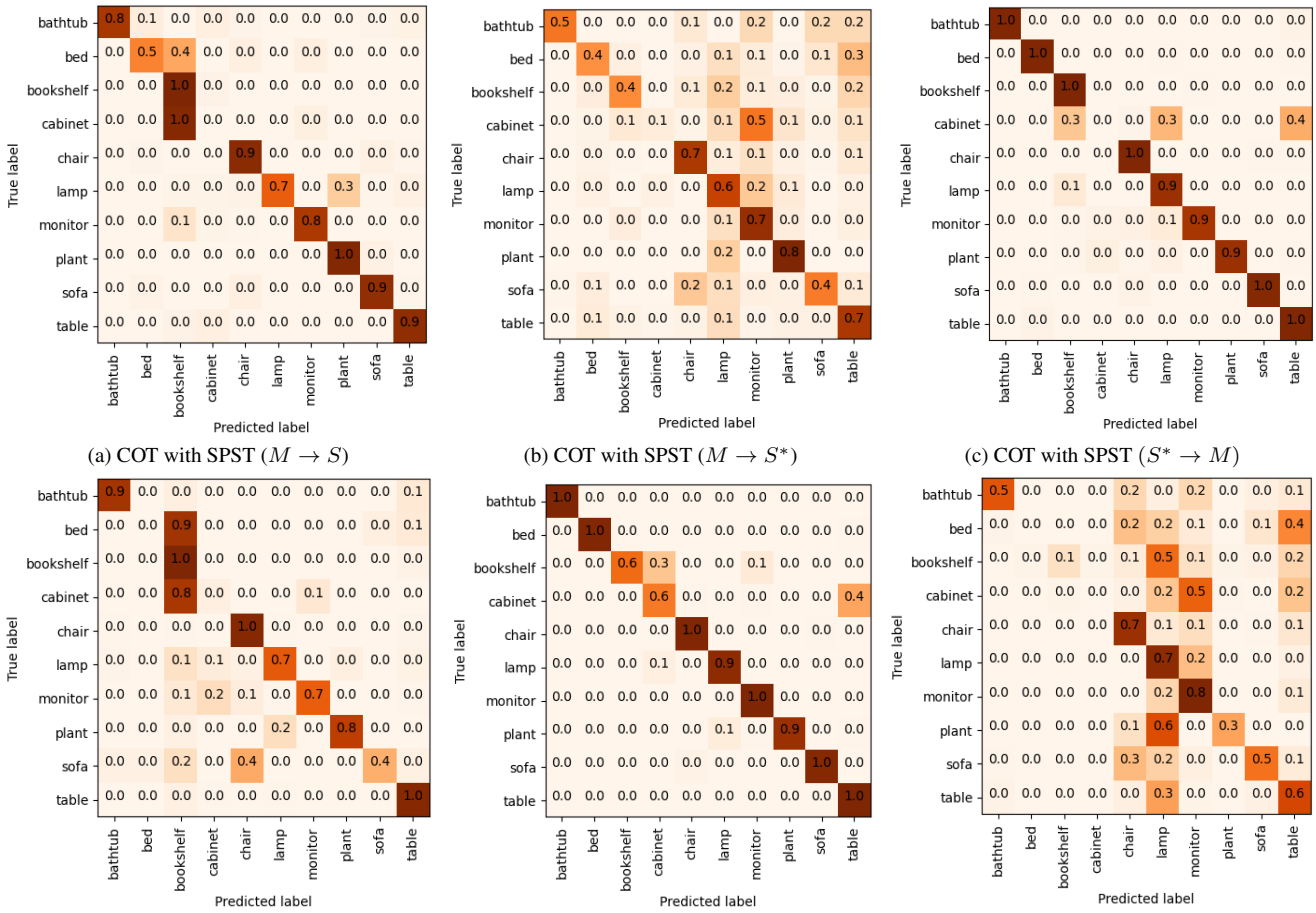
In the case of PointDA-10, in almost all the cases, the cabinet class is the toughest to classify. In some of the combinations even a single example from this class is not classified correctly. In case of GraspNetPC-10 all the samples belonging to the class Dish are always classified correctly.

	Box	Can	Banana	Drill	Scissors	Pear	Dish	Camer	Mouse	Shampoo
Syn→ Kin	1	1	0.98	0.99	0.92	1	1	1	1	1
Syn→ RS	0.95	0.97	0.28	0.84	0.96	0.69	1	1	1	0.65
Kin→ RS	1	0.78	0.63	0.98	0.9	0.31	1	0.83	0.99	0.96
Rs→ Kin	1	1	0.98	0.99	0.98	1	1	1	0.85	1

Table 2. Class-wise accuracies of our COT (with SPST) on the GraspNetPC-10 dataset

C. Domain Alignment Analysis

We compute and plot Maximum-Mean-Discrepancy (MMD) between learned point cloud features for the rest of the dataset combinations. Figures 3, 4, 5, 6, 7 contain MMD plots for all source-target dataset combinations from PointDA-10. Figure 8 contains MMD plots for two



(a) COT with SPST ($M \rightarrow S$)

(b) COT with SPST ($M \rightarrow S^*$)

(c) COT with SPST ($S^* \rightarrow M$)

(d) COT with SPST ($S^* \rightarrow S$)

(e) COT with SPST ($S \rightarrow M$)

(f) COT with SPST ($S \rightarrow S^*$)

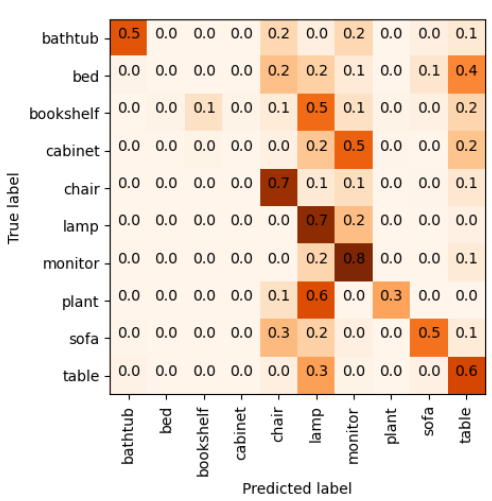
Figure 1. Confusion Matrices of Our COT with SPST on PointDA-10 dataset on all Source→Target experimental settings

experimental settings (Kin.→RS. and RS.→Kin.) from GraspNetPC-10.

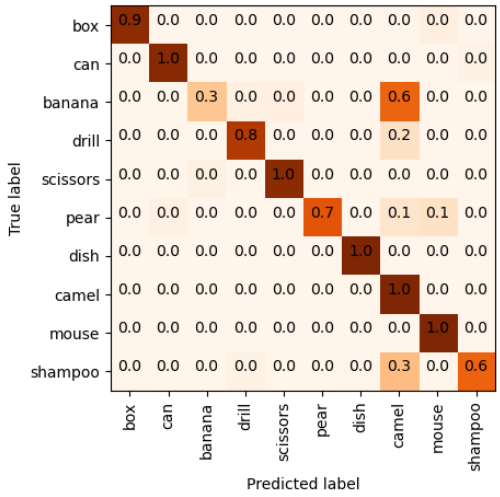
We can observe that the diagonals have lower values for our method, which indicates better class alignment across source and target. Also in the plot from our method, the upper and lower triangular matrices have higher values than without adaptation ones, which indicates better inter-class distance between source and target classes individually.

D. Decision Boundary Analysis

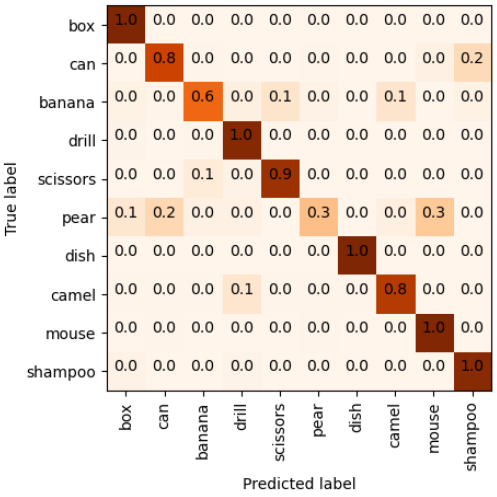
In this section, we analyze the effectiveness of our proposed architectures (COT and COT with SPST) by examining decision boundaries of the baseline variants (PCM and PCM with Contrastive Learning) and our proposed architectures learned on all the experimental settings of PointDA-10 and GraspNetPC-10 datasets. For this analysis, we create decision boundaries by extracting learned representations of the target dataset from 3D encoder and the corresponding predicted target labels. We finally fit an SVM by considering "One-vs-Rest" strategy for a class of interest (here Class: Monitor for PointDA-10 and Class: Dish for GraspNetPC-10). Figure 9 and 10 shows the corresponding decision boundaries of all possible settings of PointDA-10 (except ShapeNet-ModelNet i.e., S→M which is in the main paper) and GraspNetPC-10 dataset, respectively. In both of the datasets, it is clearly visible from the decision boundaries that our proposed COT produces stronger and more robust decision boundaries. For all PointDA-10 combinations, the decision boundaries of our method has compact decision boundary. For the GraspNet-10 combinations, our methods classifier has high confidence (either dark red or dark blue regions) compared to the baselines. These decision boundary plots show the effectiveness of our domain alignment method endowed by contrastive learning and optimal transport.



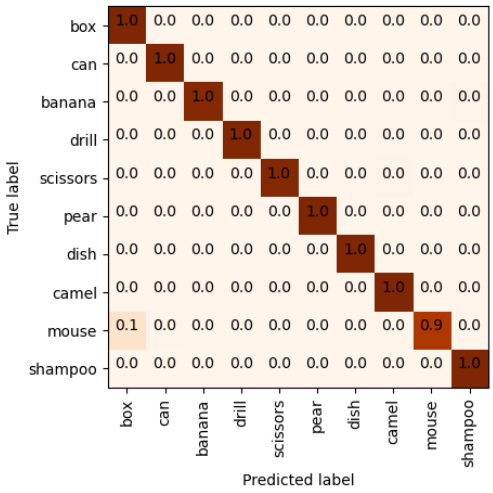
(a) COT with SPST: Syn.→Kin.



(b) COT with SPST: Syn.→RS

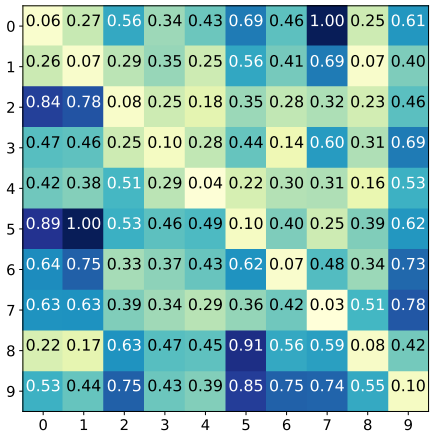


(c) COT with SPST: Kin.→Rs.

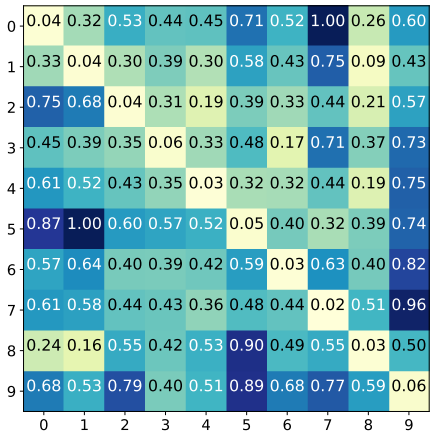


(d) COT with SPST: RS→Kin.

Figure 2. Confusion Matrices of Our COT with SPST on GraspNetPC-10 dataset on all Source→Target experimental settings

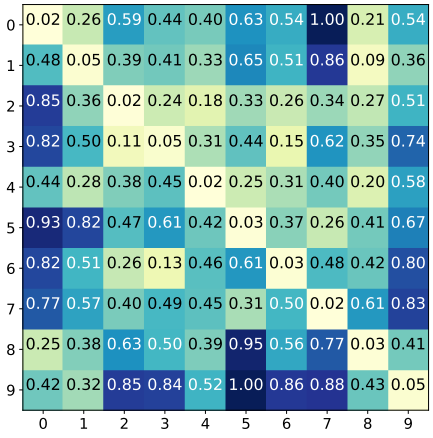


(a) Baseline (PCM without Adaptation): $S^* \rightarrow M$

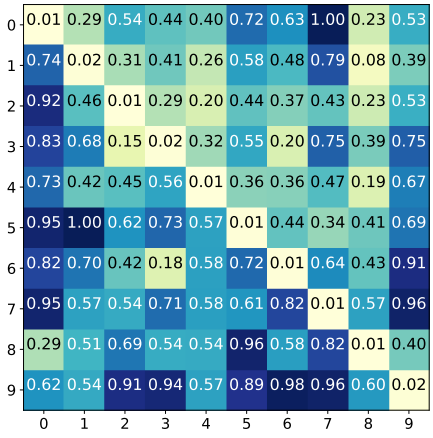


(b) COT with SPST: $S^* \rightarrow M$

Figure 3. Class-wise MMD plots for Baseline (PCM without Adaptation) and Our COT with SPST for $S^* \rightarrow M$.

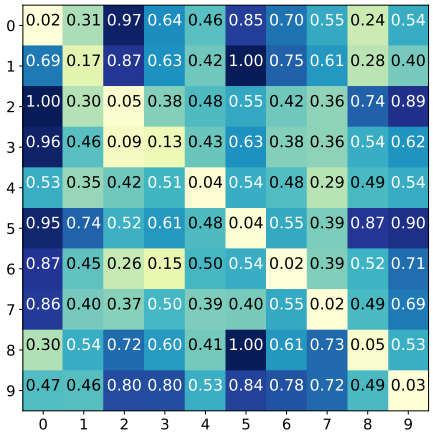


(a) Baseline (PCM without Adaptation): $S^* \rightarrow S$

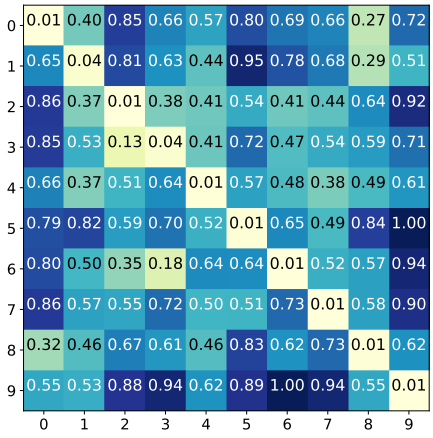


(b) COT with SPST: $S^* \rightarrow S$

Figure 4. Class-wise MMD plots for Baseline (PCM without Adaptation) and Our COT with SPST for $S^* \rightarrow S$.

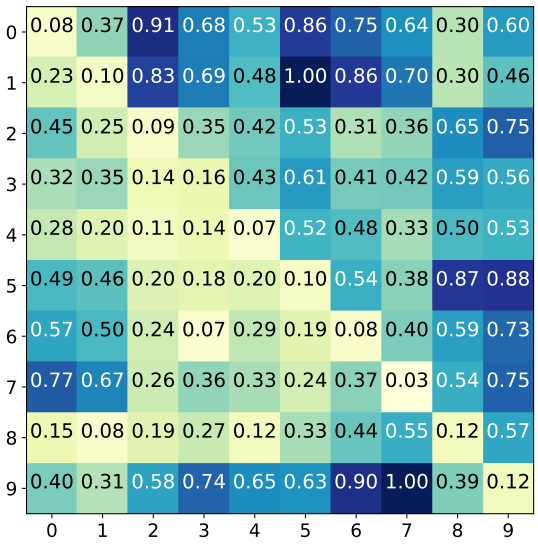


(a) Baseline (PCM without Adaptation): $M \rightarrow S$

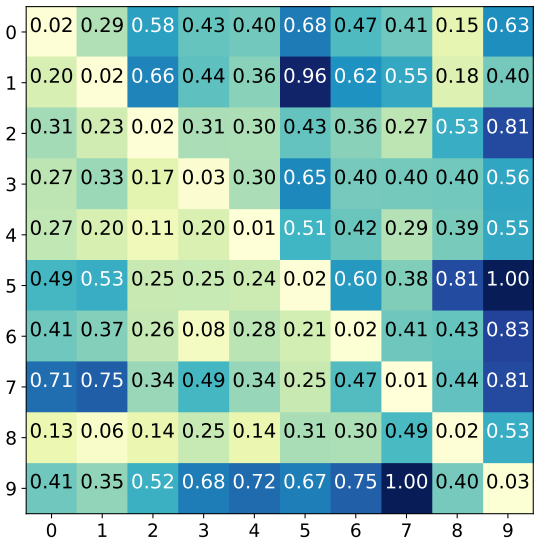


(b) COT with SPST: $M \rightarrow S$

Figure 5. Class-wise MMD plots for Baseline (PCM without Adaptation) and Our COT with SPST for $M \rightarrow S$.

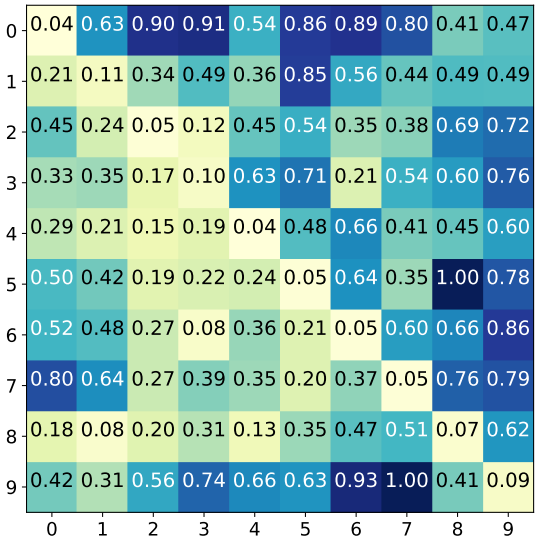


(a) Baseline (PCM without Adaptation): $M \rightarrow S^*$

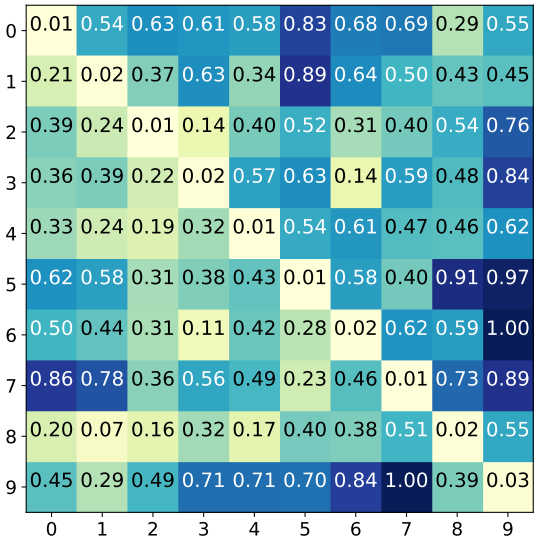


(b) COT with SPST: $M \rightarrow S^*$

Figure 6. Class-wise MMD plots for Baseline (PCM without Adaptation) and Our COT with SPST for $M \rightarrow S^*$.

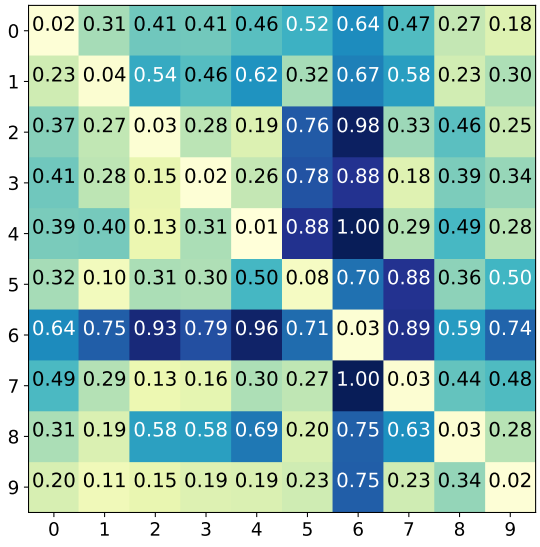


(a) Baseline (PCM without Adaptation): $S \rightarrow S^*$

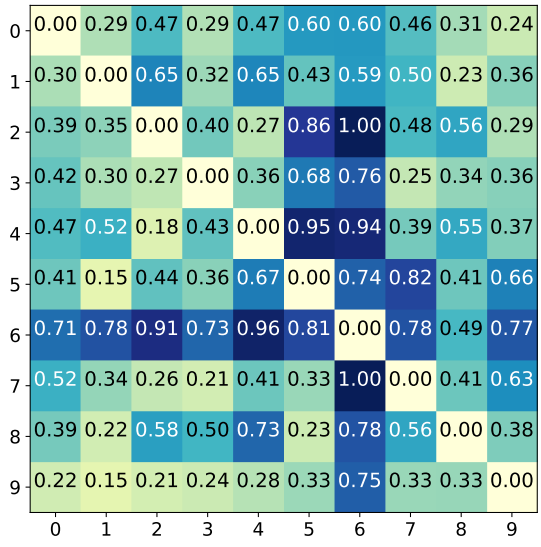


(b) COT with SPST: $S \rightarrow S^*$

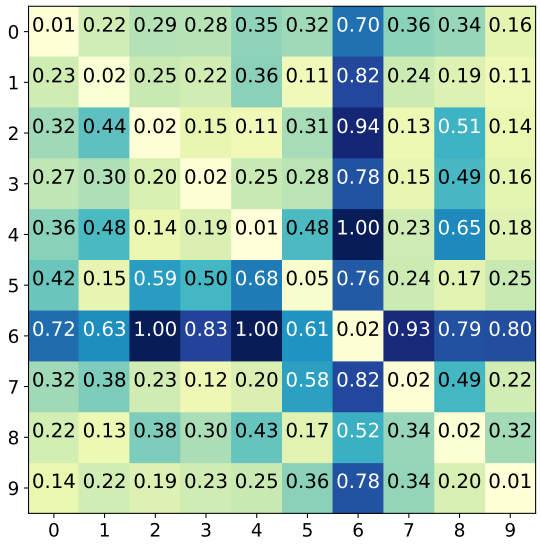
Figure 7. Class-wise MMD plots for Baseline (PCM without Adaptation) and Our COT with SPST for $S \rightarrow S^*$.



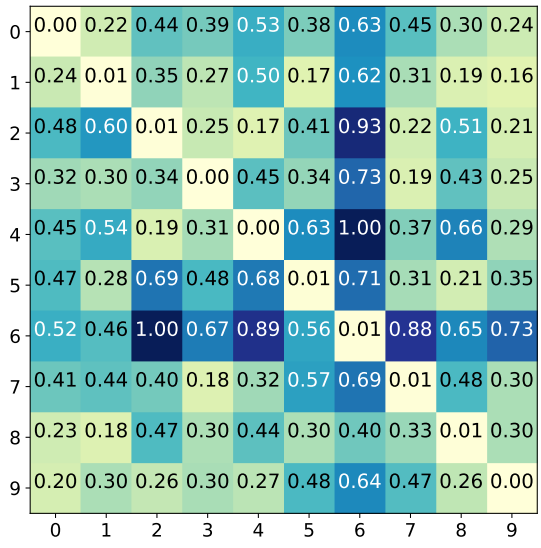
(a) Baseline (PCM without Adaptation): Kin.→ RS



(b) COT with SPST: Kin.→ RS

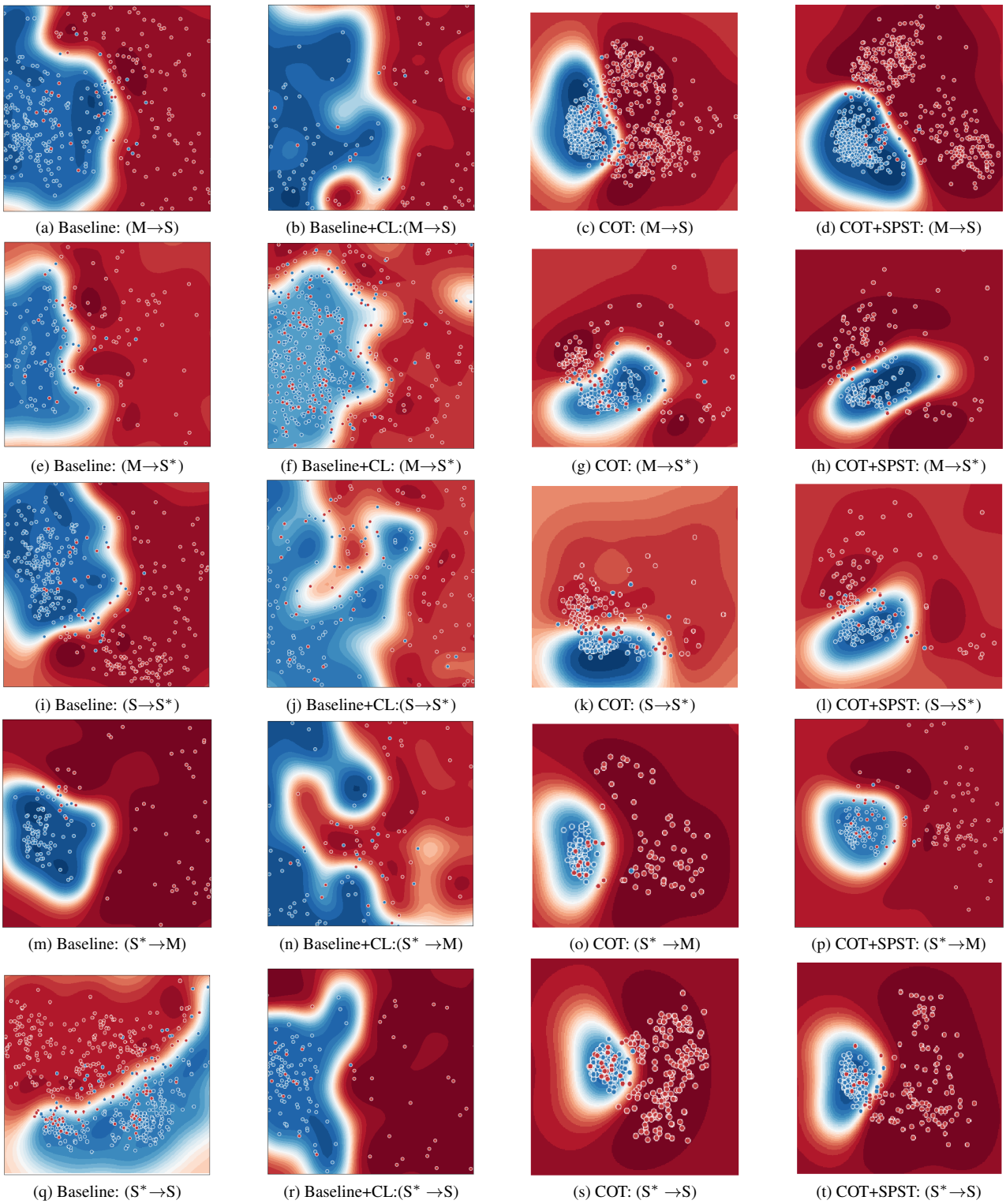


(c) Baseline (PCM without Adaptation): RS.→ Kin



(d) COT with SPST: RS.→ Kin

Figure 8. Class-wise MMD on GraspNetPC-10 dataset for (a), (c) baseline (only PCM without adaptation), and (b), (d) our COT with SPST for two experimental settings (Kin→RS and RS→Kin.)



(a) Baseline: (M→S) (b) Baseline+CL:(M→S) (c) COT: (M→S) (d) COT+SPST: (M→S)

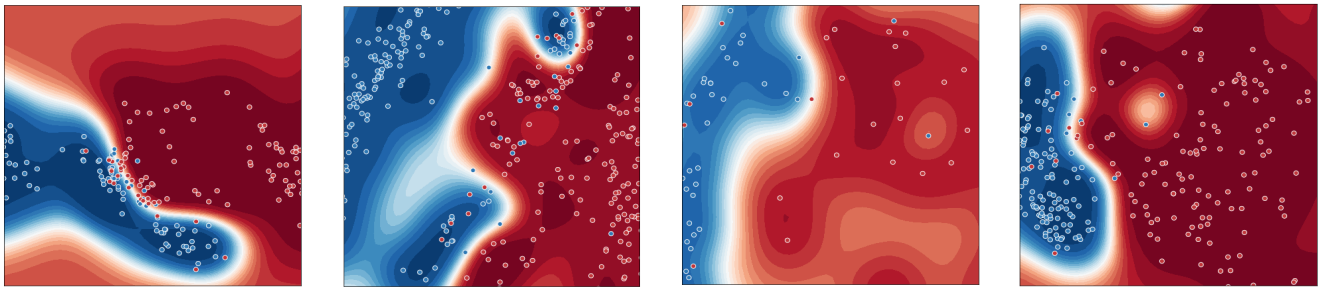
(e) Baseline: (M→S*) (f) Baseline+CL: (M→S*) (g) COT: (M→S*) (h) COT+SPST: (M→S*)

(i) Baseline: (S→S*) (j) Baseline+CL:(S→S*) (k) COT: (S→S*) (l) COT+SPST: (S→S*)

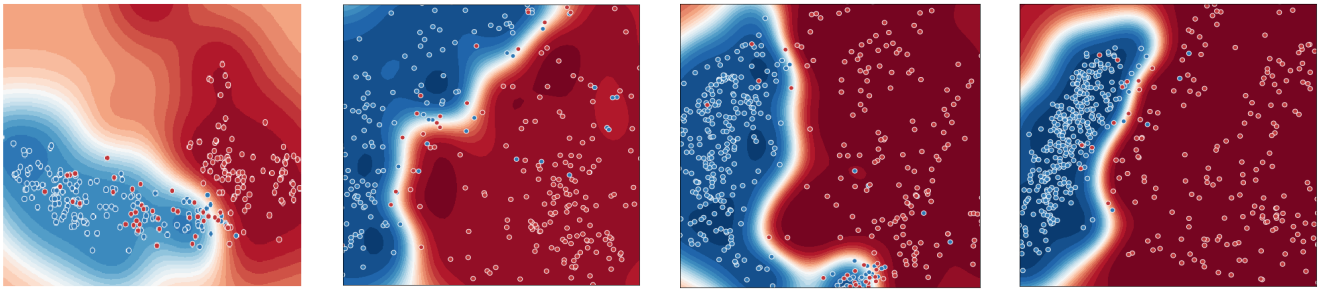
(m) Baseline: (S*→M) (n) Baseline+CL:(S* →M) (o) COT: (S* →M) (p) COT+SPST: (S*→M)

(q) Baseline: (S*→S) (r) Baseline+CL:(S* →S) (s) COT: (S* →S) (t) COT+SPST: (S*→S)

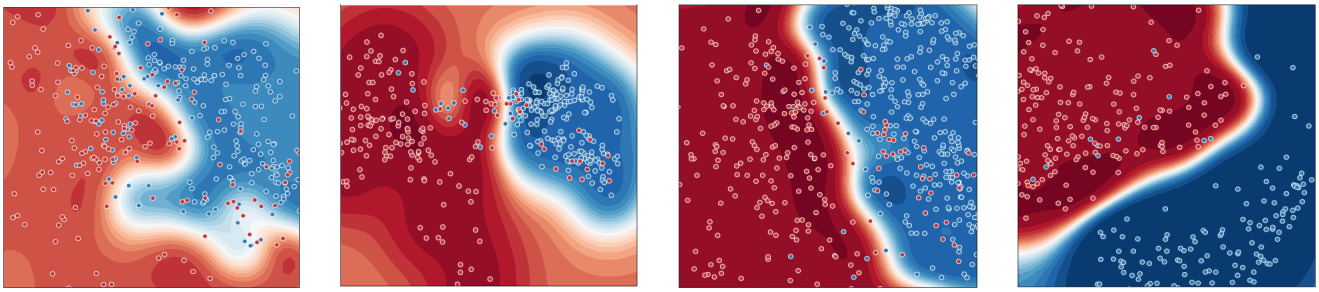
Figure 9. Decision boundaries on target samples for One-vs-Rest (Class: Monitor) for all experiment setups (Row-wise, except S→M which is in the main paper) of PointDA-10 for Baseline, Baseline with Contrastive Learning (CL), Our COT and Our COT with SPST (Column-wise)



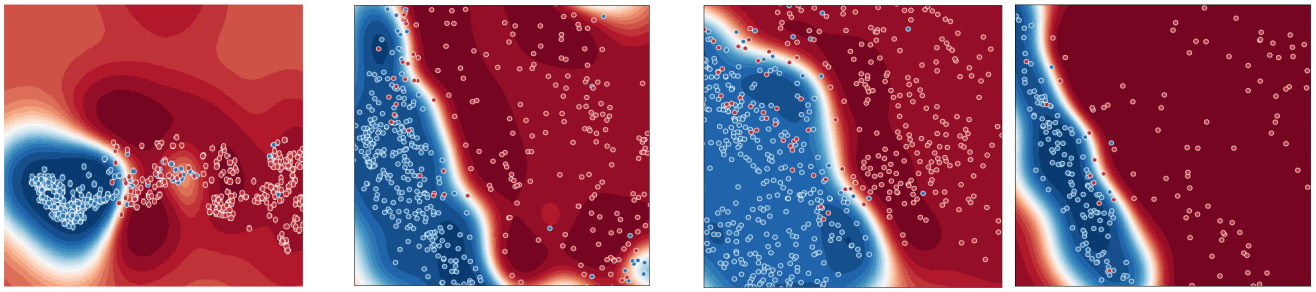
(a) Baseline: (Syn.→Kin.) (b) Baseline+CL: (Syn.→Kin.) (c) COT: (Syn.→Kin.) (d) COT+SPST: (Syn.→Kin.)



(e) Baseline: (Syn.→RS.) (f) Baseline+CL: (Syn.→RS.) (g) COT: (Syn.→RS.) (h) COT+SPST: (Syn.→RS.)

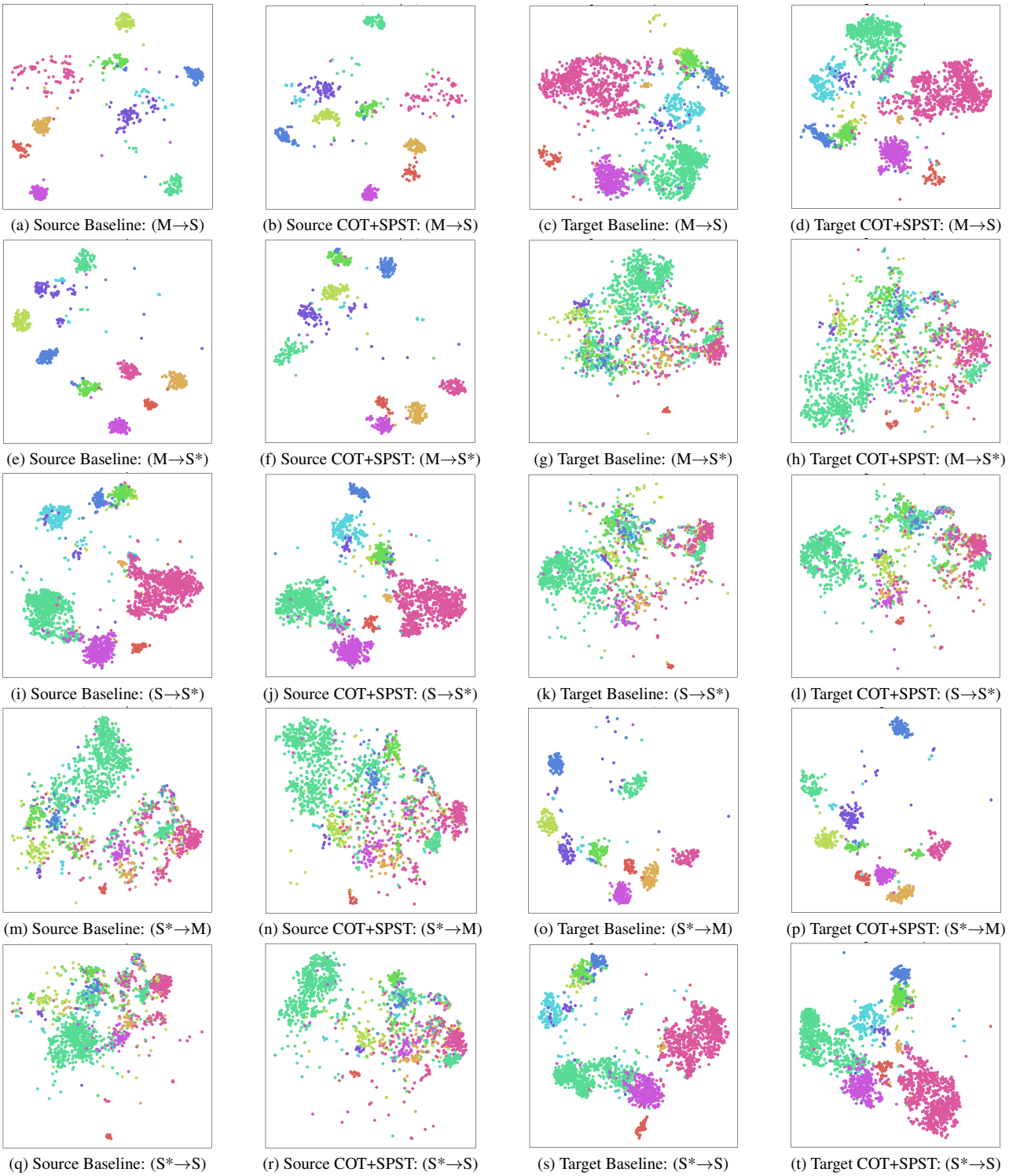


(i) Baseline: (Kin.→RS.) (j) Baseline+CL: (Kin.→RS.) (k) COT: (Kin.→RS.) (l) COT+SPST: (Kin.→RS.)



(m) Baseline: (RS.→Kin) (n) Baseline+CL: (RS.→Kin) (o) COT: (RS.→Kin) (p) COT+SPST: (RS.→Kin)

Figure 10. Decision boundaries on target samples for One-vs-Rest (Class: Dish) for all experiment setups (Row-wise) of GraspNetPC-10 for Baseline, Baseline with Contrastive Learning (CL), Our COT and Our COT with SPST (Column-wise)



(a) Source Baseline: (M→S) (b) Source COT+SPST: (M→S) (c) Target Baseline: (M→S) (d) Target COT+SPST: (M→S)

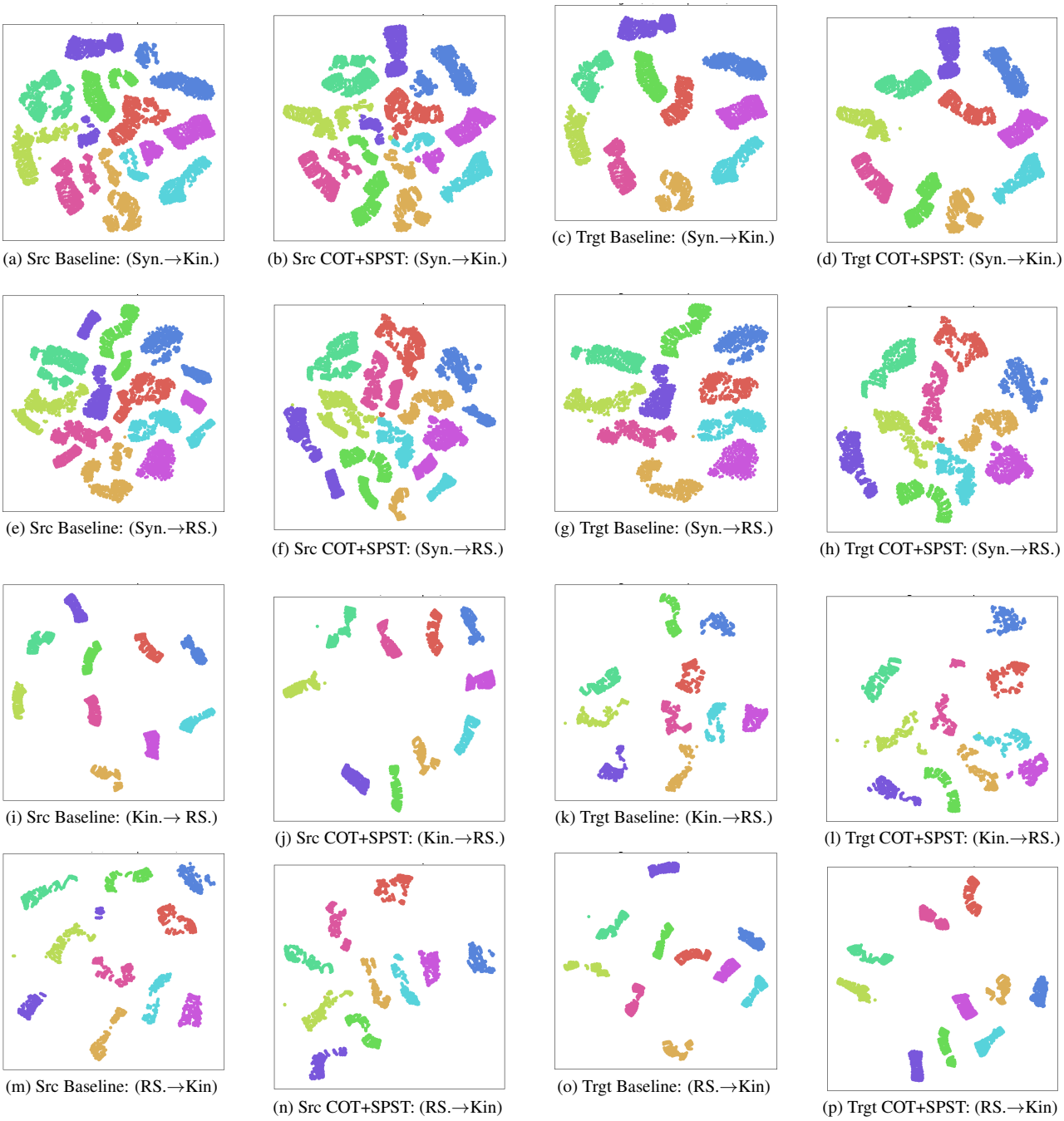
(e) Source Baseline: (M→S*) (f) Source COT+SPST: (M→S*) (g) Target Baseline: (M→S*) (h) Target COT+SPST: (M→S*)

(i) Source Baseline: (S→S*) (j) Source COT+SPST: (S→S*) (k) Target Baseline: (S→S*) (l) Target COT+SPST: (S→S*)

(m) Source Baseline: (S*→M) (n) Source COT+SPST: (S*→M) (o) Target Baseline: (S*→M) (p) Target COT+SPST: (S*→M)

(q) Source Baseline: (S*→S) (r) Source COT+SPST: (S*→S) (s) Target Baseline: (S*→S) (t) Target COT+SPST: (S*→S)

Figure 11. t-SNE visualization of Source (first two columns) and Target (last two columns) test sets (10 classes) for baseline (only PCM w/o adaptation) and our COT with SPST on all experimental setups of PointDA-10.



(a) Src Baseline: (Syn.→Kin.)
(b) Src COT+SPST: (Syn.→Kin.)
(c) Trgt Baseline: (Syn.→Kin.)
(d) Trgt COT+SPST: (Syn.→Kin.)
(e) Src Baseline: (Syn.→RS.)
(f) Src COT+SPST: (Syn.→RS.)
(g) Trgt Baseline: (Syn.→RS.)
(h) Trgt COT+SPST: (Syn.→RS.)
(i) Src Baseline: (Kin.→ RS.)
(j) Src COT+SPST: (Kin.→RS.)
(k) Trgt Baseline: (Kin.→RS.)
(l) Trgt COT+SPST: (Kin.→RS.)
(m) Src Baseline: (RS.→Kin)
(n) Src COT+SPST: (RS.→Kin)
(o) Trgt Baseline: (RS.→Kin)
(p) Trgt COT+SPST: (RS.→Kin)

Figure 12. t-SNE visualization of Source (first two columns) and Target (last two columns) test sets (10 classes) for baseline (only PCM w/o adaptation) and our COT with SPST on all experimental setups of GraspNetPC-10.

E. t-SNE Visualization

We visualize point cloud embeddings of our learned model for PointDA-10 and GraspNetPC-10 datasets using t-SNE. Figure 11 contains t-SNE plots for all source-target experimental settings from PointDA-10. Figure 12 contains t-SNE plots for all source-target experimental settings from Graspnet-10. We use source and target test sets for plotting these t-SNE plots, except for the Syn. dataset from GraspNetPC-10 for which the test set is not available. We consider validation set for Syn. (synthetic) dataset.

Comparing the t-SNE plots between baseline (only PCM without adaptation) and our best performing method (COT with SPST strategy) in both source and target domains, we can observe inter-cluster distances getting increased with well-defined class separations in our proposed method. These plots also show class domain alignment.

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] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *J. Mach. Learn. Res.*, 22(1), jul 2022.

[3] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Ieee Conference On Computer Vision And Pattern Recognition (cvpr)*, 2015.

[4] Yuefan Shen, Yanchao Yang, Mi Yan, He Wang, Youyi Zheng, and Leonidas J. Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7223–7232, June 2022.

[5] 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 Trans. Graph.*, 38(5), oct 2019.

[6] 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.