Self-Supervised Boundary Point Prediction Task for Point Cloud Domain Adaptation

Jintao Chen, Yan Zhang, Kun Huang, Feifan Ma, Zhuangbin Tan, and Zheyu Xu

***Abstract*—Unsupervised domain adaptation (UDA) could significantly improve the cross-domain performance of current supervised 3D deep learning methods and have a widespread application prospect. However, the domain gap between source domain and target domain renders the UDA problem highly challenging. In this letter, we present a novel UDA method for point clouds from the perspective of multi-strategy. First, we explore the effectiveness of state-of-the-art data augmentation methods to point cloud domain adaptation, and introduce a data augmentation procedure to two widely-existed scenarios, i.e., sim-to-sim and sim-to-real. And then, we explore a mask deformation procedure to simulate the missing parts with respect to the real-world point clouds. On one hand, the masked point clouds push network to pay more attention to local features rather than global features; on other hand, we employ a prediction-consistency contrastive loss to improve the prediction robustness of network based on the mask deformation. Moreover, we propose a self-supervised learning task by predicting the boundary points of masked region. Specifically, the network could effectively perceive the occlusion and capture fine-grained features by automatically labeling and predicting the boundary points of the marked region. Extensive experiments conducted on both PointDA-10 and PointSegDA benchmarks for point cloud classification and segmentation, respectively, demonstrate the effectiveness of the proposed method.**

***Index Terms*—3D deep learning, unsupervised domain adaptation, self-supervised learning, point cloud classification, point cloud segmentation.**

I. Introduction

POINT cloud data analysis is becoming a research hotspot in many fields covering autonomous driving [1], robotics [2], [3]. Based on deep learning technology, many supervised deep learning methods [4], [5], [6], [7], [8], [9], [10] have achieved remarkable performance. However, the superiority of these supervised methods relies on large-scale annotated datasets, which limits their applicability to some extent. Because it is difficult and troublesome to equip an individual labeled dataset for each task in a practical perspective. One of the most straightforward and effective solutions is to transfer the learned knowledge from an existing source dataset to an unseen and unlabeled target dataset, the problem is so-called unsupervised domain adaptation (UDA). The main challenge of UDA is to close the gap between source dataset and target dataset. To alleviate this problem, many UDA methods have been achieved, including feature alignment [11], adversarial learning [12], [13], [14], and self-supervised learning (SSL) auxiliary task [15], [16], [17]. Feature alignment attempts to match the embedding of distributions by employing distribution statistic, such as maximum mean discrepancy (MMD) [11], while adversarial learning aims to directly learn unbiased representations through a discriminator to judge whether the learned features are from the target domain or the source domain. Recently, many studies suggested to learn more meaningful representations of point clouds using auxiliary SSL task, thus the way of multiple auxiliary tasks learning has gradually been the standard way to tackle UDA issue. Nevertheless, the performance of those methods relying only on stacking auxiliary tasks is promising but far from perfect regarding UDA problem. Because the performance of a single auxiliary task is usually limited [16], and the superposition of multiple auxiliary tasks usually cannot obtain satisfactory gains but make the algorithm redundant [17]. Therefore, it is still necessary to explore adaptation strategies from new perspectives to further promote the development of this field.

In this letter, we propose to reduce the domain bias by jointly applying data augmentation, contrastive learning and auxiliary SSL task, which tackles the UDA issue from a perspective of multi-strategy. The main concept of our method is illustrated in Fig. 1. Our design motivations lie in: 1) Data augmentation could generally enlarge the variance of source domain in structure which is expected to reduce the gap between source and target domains regarding the two widely-existed scenarios, i.e., sim-to-sim and sim-to-real [18], [19]. 2) We explore a mask deformation procedure to randomly mask a local region of point clouds. On one hand, the masked point clouds enable network to pay more attention to local features rather than global features; on other hand, we employ a prediction-consistency contrastive loss to improve the prediction robustness of network based on the mask deformation mentioned above. We hope that the prediction of network should be as consistent as possible with respect to a point cloud, no matter which local region is masked. 3) The occlusion in real-world point clouds not only leads to the information loss of the missing part but also corrupts the semantic features of boundary points around the occluded

Manuscript received 13 March 2023; accepted 13 July 2023. Date of publication 2 August 2023; date of current version 9 August 2023. This letter was recommended for publication by Associate Editor H. Kasaei and Editor J. Kober upon evaluation of the reviewers' comments. This work was supported by the Shenzhen Science and Technology Program under Grant KQTD20190929172704911. *(Corresponding author: Yan Zhang.)*

The authors are with the School of Aeronautics and Astronautics, Shenzhen Campus of Sun Yat-sen University, Shenzhen 518000, China (e-mail: chenjt66@mail2.sysu.edu.cn; zhangyan25@mail.sysu.edu.cn; huangk77@mail2.sysu.edu.cn; maff@mail2.sysu.edu.cn; tanzhb6@mail2.sysu.edu.cn; xuzhy53@mail2.sysu.edu.cn).

Source code will be available at https://github.com/chenjt66a/MS-UDA-method.

Digital Object Identifier 10.1109/LRA.2023.3301278

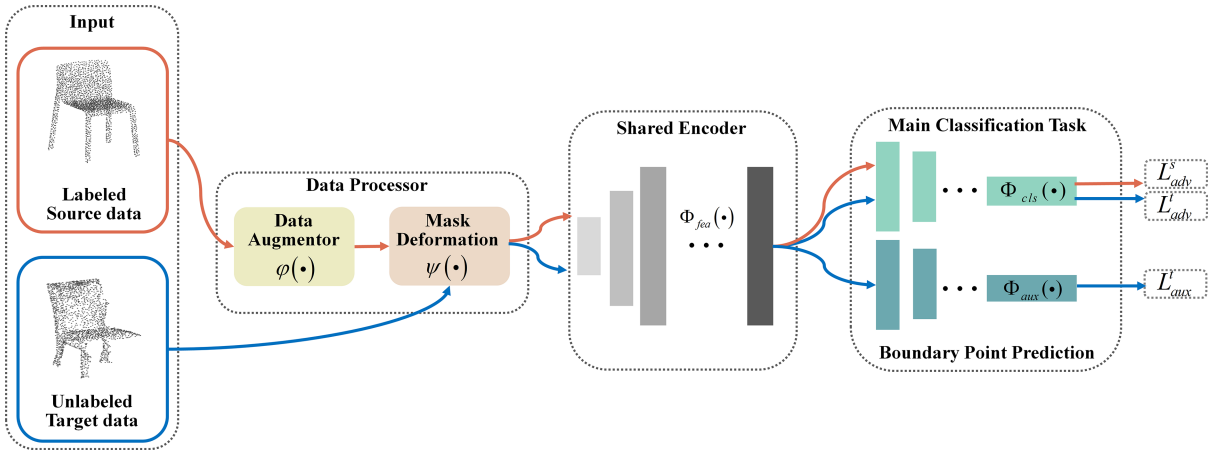

Fig. 1. Illustration of the proposed method. The red/blue arrows represent the data flow of source/target domains. The source data flow takes as input the point clouds from source domain and subsequently goes through the data augmentor, mask deformation, shared encoder, and finally calculates the cross-entropy with the ground-truth labels under the main task head Φ_{cls}. The target data flow takes as input the point clouds from target domain, and spilt into two pathways after mask deformation and shared encoder. One pathway aims to calculate the cross-entropy loss with each other predicted labels under the main task head Φ_{cls}, while another pathway is designed to calculate the self-supervised loss under the auxiliary task head Φ_{aux}.

region, thus fooling the network. Therefore, we propose an auxiliary SSL task by automatically labeling and predicting the boundary points of masked region, so that the network could effectively perceive the occlusion and capture fine-grained semantic features.

In summary, our main contributions could be concluded as follows:

- We explore the adaptation ability of current data augmentation methods to UDA of point clouds and definitely introduce data augmentation to two widely-existed domain adaptation scenarios, i.e., sim-to-sim and sim-to-real.
- We employ a prediction-consistency contrastive loss to improve the prediction robustness of network based on the mask deformation.
- We propose an auxiliary SSL task by automatically labeling and predicting the boundary points around the masked region, which enables the network to effectively perceive the occlusion so as to alleviate the confusion of occlusion to network.
- We validate our method on both PointDA-10 and PointSegDA benchmarks, respectively, for point cloud classification and segmentation, and achieve competitive performances on both benchmarks.

II. Related Works

A. Deep Learning on Point Clouds

Qi et al. [4] proposed PointNet to extract point-wise features by applying multi-layer perceptrons (MLPs), and then a symmetric function, max-pooling (MP), was employed to aggregate the global features of point clouds. But PointNet lacks ability to capture local features of point clouds. To this end, Qi et al. [5] subsequently extended PointNet to PointNet++ which considers PointNet as a local feature encoder in a hierarchical structure. They first use farthest-point sampling (FPS) and k-nearest neighbors (KNN), respectively, to sample and group the point clouds, and then use PointNet to capture the local features. Wang et al. [6] presented DGCNN to semantically group points by dynamically updating a graph of relationships from layer to layer, and EdgeConv operator was introduced to better capture local features. EdgeConv is a lightweight convolution kernel and a plug-and-play which achieves outstanding performance for point cloud classification and segmentation. Subsequently, researchers have proposed many supervised learning methods [20], [21] and achieve state-of-the-art performances on ModelNet40 [22], ShapePartNet [23], and ScanObjectNN [24]. However, these supervised learning methods rely on large-scale annotated datasets and suffer a huge drop of performance across datasets, limiting their application. Therefore, it is necessary to explore suitable domain adaptation methods to extend the application of these algorithms in real life.

B. Unsupervised Domain Adaptation on Point Clouds

Compared with UDA studies of 2D images [25], [26], there are a few studies focusing on point clouds. Qin, et al. [11] proposed the seminal work, PointDAN, which applies a node module with adaptive filed to model the discriminate local structures and minimize an MMD loss to jointly align local and global features. They built a general benchmark, i.e., PointDA-10, extracted from three popular object/scene datasets (i.e., ModelNet, ShapeNet and ScanNet) for cross-domain 3D shape classification evaluation. Recently, many works focus on designing handcrafted auxiliary SSL tasks since it could effectively capture domain-invariant features [27]. These SSL-based point cloud UDA methods typically train the main task on source domain in a supervised manner and then several auxiliary SSL tasks are subsequently trained on both source and target domains to improve the performance on main task. Therein, the main and auxiliary tasks share an encoder but have different heads to perform respective tasks. For example, Achituve, et al. [15] proposed a deformation-reconstruction auxiliary task and then [16]

extended it into a learnable nonlinear deformation task to further improve the performance. Zou, et al. [17] proposed two self-supervised learning tasks, i.e., rotation angle prediction and curvature-aware distortion localization, to learn a domain-shared representation of semantic categories. In addition, they introduced self-paced self-training (SPST) strategy [28], [29] to fine-tune the network and further boost performance. Shen, et al. [30] asked for a latent space that encodes the underlying geometry of the point clouds through implicit functions instead of manually designed classification labels, and achieved state-of-the-art performance on PointDA-10 and GraspNet [31] datasets. Fan, et al. [32] designed two auxiliary tasks, i.e., scaling-up-down prediction and 3D-2D-3D projection-reconstruction, to learn the global and local representation of point clouds, respectively, and conducted experiments on both PointDA-10 and *Sim-to-Real* (i.e., ModelNet11, ScanObjectNN11, ShapeNet9, ScanObjectNN9) [33] datasets. Liang, et al. [34] tackle the UDA problem via three auxiliary SSL tasks including the prediction of cardinality, position and normal of masked points. They also exploited pseudo labels generation strategy and developed a self-paced variant that leverages prediction probability entropy to improve SPST. In this letter, we jointly employ data augmentation and mask deformation, contrastive learning, and auxiliary SSL task to develop a novel multi-strategy UDA method for point clouds.

III. The Proposed Method

In this section, we elaborate the proposed multi-strategy point cloud UDA method. Section III-A reviews the point cloud UDA problem; Section III-B - Section III-D describe the three components of the proposed method, including data augmentation, prediction-consistency contrastive loss and boundary point prediction of masked region; Section III-E concludes the overall loss.

A. Problem Statement

In the point cloud UDA problem, a labeled source domain $\{x_i^s \in X_s, y_i^s \in Y_s\}_{i=1}^{n_s}$ with n_s labeled point clouds and a target domain $T = \{x_i^t \in X_t\}_{i=1}^{n_t}$ with n_t unlabeled point clouds are given, however, $y_i^t \in Y_t$ is not available. As in standard UDA issue, the source domain and target domain are assumed to have the following three settings: 1) they have the same one-hot encoded label space, i.e., $Y_s = Y_t = Y \subset \mathbb{R}^C$ (C denotes the number of categories); 2) they share the same input space, i.e., $X_s, X_t \subset \mathbb{R}^3$; 3) they are with different distributions $P_s(\mathbf{x}) \neq P_t(\mathbf{x})$ (i.e., domain gap), e.g., due to different point scales, object styles, Lidar viewpoints, incompletion, sensor noise, etc. Domain adaptation aims to model a mapping function $\Phi : X \to Y$ that can correctly classify point clouds both on source and target domains. Based on deep learning technology, the mapping function $\Phi(\cdot)$ generally could be formulated into a cascade of a feature encoder Φ_{fea} and a classifier Φ_{cls}, i.e., $\Phi(\cdot) = \Phi_{cls}(\Phi_{fea}(\cdot))$. Therein, $\Phi_{fea} : X \to \mathbb{R}^d$ is designed to capture semantic features of point clouds while $\Phi_{cls} : \mathbb{R}^d \to [0,1]^C$ typically consisting of fully-connected layers. Thus, the category labels could be obtained according to the maximum score criterion. We focus on modelling the $\Phi(\cdot)$ to boost the accuracy of UDA.

B. Data Augmentation

Data augmentation: Data Augmentation enlarging the quantity and diversity of train sets can effectively improve the generalization ability from train sets to test sets, which has been verified at many supervised learning methods [35], [36], [37], [38], but still need to be explored in UDA. Because that source and target domains are with different distribution regarding UDA issue, which is actually distinct from supervised learning, i.e., fully- or weakly-supervised learning. Here we mainly discuss the effectiveness of data augmentation to point cloud domain adaptation from the structure variance perspective. We observe that applying data augmentation on a dataset usually leads to the increasement of structure variance. In this regard, data augmentation could be expected to reduce the domain gap in scenarios where the structure variance of target domain is larger than that of source domain, e.g., sim-to-real scenario. For a point cloud $x_i \in \{x_i^s\} \subset \mathbb{R}^{m\times 3}$, the data augmentation process could be formulated as follows:

$$\tilde{x}_i = \varphi\left(x_i\right), \quad x_i \text{ and } \tilde{x}_i \subset \mathbb{R}^{m\times 3}. \tag{1}$$

where $\varphi(\cdot)$ denotes the data augmentation procedure, and $\tilde{x}_i$ denotes the augmented point cloud. In this work, we explore the adaptation ability of all three state-of-the-art data augmentation methods includng PointAugment [35], PointWOLF [36] and PointCutMix-K [37] to UDA of point clouds. More details about the analyses of these three methods are presented at Section IV-C.

C. Prediction-Consistency Contrastive Loss

Mask deformation: The synthetic point clouds are complete while the real-world point clouds are generally incomplete, which leads to the results that two objects might be weak to align in global features but would still have similar local features. To mitigate this problem, we randomly mask a region of point clouds to ask the network to pay more attention to local features of point clouds. Specifically, given a point cloud, we randomly choose one point which is referred to as the center point, and then KNN algorithm is applied to find out $k-1$ neighbors that belongs to the center point. After that both the center point and their $k-1$ neighbors are dropped out to generate the incomplete point cloud. The illustration of mask deformation is depicted at Fig. 2(b), and this process could be formulated as follows:

$$\hat{x}_i = \psi\left(x_i\right), \quad \hat{x}_i \subset \mathbb{R}^{p\times 3}. \tag{2}$$

where $\psi(\cdot)$ denotes the mask deformation procedure, $x_i \subset \mathbb{R}^{m\times 3}$ and $\hat{x}_i$, respectively, denote the original point cloud i with m points and the corresponding masked point cloud with p points, here $p < m$.

Based on mask deformation mentioned above, we further employ a prediction-consistency contrastive loss to improve the prediction robustness of network. In other words, we expect that the prediction of network should be as consistent as possible with respect to a point cloud, no matter which local region is

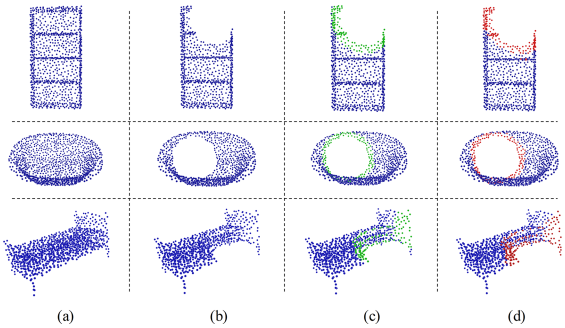

Fig. 2. Illustration of the mask deformation, automatic labeling and prediction of the boundary points. (a) original point clouds; (b) randomly mask 256 points for a point cloud; (c) automatically label 128 boundary points after the mask deformation, and green points denote the labeled boundary points; (d) prediction results of boundary points, and red points denote the predicted boundary points.

masked. Specifically, we randomly mask twice for a point cloud so that a pair of masked point clouds, i.e., $\{(\hat{x}^s_{i,1}, y^s_i), (\hat{x}^s_{i,2}, y^s_i)\}$ or $\{\hat{x}^t_{i,1}, \hat{x}^t_{i,2}\}$ could be obtained regarding source dataset and target dataset, respectively. Once a pair of masked point clouds are obtained, we employ a prediction-consistency contrastive loss for source domain as follows:

$$L^s_{adv} = 0.5 * \begin{pmatrix} -\frac{1}{n^s}\sum_{i=1}^{n^s}\sum_{j=1}^{C} y^s_{i,j} \log\left(\Phi_{cls}(\Phi_{fea}(\hat{x}^s_{i,1}))_j\right) \\ -\frac{1}{n^s}\sum_{i=1}^{n^s}\sum_{j=1}^{C} y^s_{i,j} \log\left(\Phi_{cls}(\Phi_{fea}(\hat{x}^s_{i,2}))_j\right) \end{pmatrix} \tag{3}$$

where $y^s_{i,j}$ represents the ground truth one-hot labels, and $\Phi_{cls}(\Phi_{fea}(\hat{x}^s_{i,1}))_j$, $\Phi_{cls}(\Phi_{fea}(\hat{x}^s_{i,2}))_j$ is the predicted probability of $\{\hat{x}^s_{i,1}, \hat{x}^s_{i,2}\}$ for the jth class.

As for target domain, we supervise each other by the predicted probability of two masked point clouds since we do not have their labels, as follows:

$$L^t_{adv} = 0.5 * \begin{pmatrix} -\frac{1}{n^t}\sum_{i=1}^{n^t}\sum_{j=1}^{C} y^{t,pred1}_{i,j} \log\left(\Phi_{cls}(\Phi_{fea}(\hat{x}^t_{i,1}))_j\right) \\ -\frac{1}{n^t}\sum_{i=1}^{n^t}\sum_{j=1}^{C} y^{t,pred2}_{i,j} \log\left(\Phi_{cls}(\Phi_{fea}(\hat{x}^t_{i,2}))_j\right) \end{pmatrix} \tag{4}$$

where $y^{pred1}_{i,j} = \mathrm{softmax}(\Phi_{cls}(\Phi_{fea}(\hat{x}^t_{i,1})))$ and $y^{pred2}_{i,j} = \mathrm{softmax}(\Phi_{cls}(\Phi_{fea}(\hat{x}^t_{i,2})))$, respectively, are the predicted one-hot labels of $\{\hat{x}^t_{i,1}, \hat{x}^t_{i,2}\}$; $\Phi_{cls}(\Phi_{fea}(\hat{x}^t_{i,1}))_j$ and $\Phi_{cls}(\Phi_{fea}(\hat{x}^t_{i,2}))_j$, respectively, are the predicted probability of $\{\hat{x}^t_{i,1}, \hat{x}^t_{i,2}\}$ for the jth class.

D. Prediction of Boundary Point

We further analyze the impact of point cloud occlusion on network. Occlusion of point clouds not only leads to the information loss of the missing part but also destroys the geometric structures of boundary points around the occluded region, as illustrated at Fig. 3. The destroyed geometric structures might fool the network by the following two reasons: 1) the network may regard the destroyed structures as a special kind of local features since it has no prior knowledge of the occlusion, however, which is wrong actually; 2) many categories could have these special local features due to occlusion, thus decreasing the diversity of

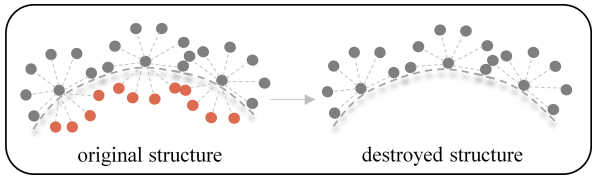

Fig. 3. Analysis of missing part. The missing part could destroy the original structure (the left) of local region and lead to a specical kind of geometric structure (the right).

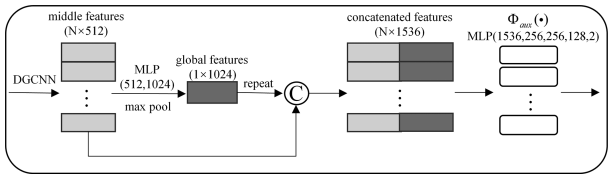

Fig. 4. Implementation of boundary point prediction task.

inter-class. To alleviate this problem, we propose an auxiliary SSL task by automatically labeling and predicting the boundary points around masked region, as presented at Fig. 2(c) and (d), respectively. In practice, the labeling process can be done in conjunction with the aforementioned mask deformation process. Specifically, given a point cloud, we first randomly select a point as the center point, and then the KNN algorithm is applied to find out both $k-1$ and M_p neighbors. Therein, both the center point and the $k-1$ neighbors that most closest to the center point are dropped out to generate the incomplete point cloud, while the rest of M_p neighbors are automatically labeled as the boundary points, so that the labels of boundary points $y^{t,aux}_i \in \{0,1\} \subset \mathbb{R}^2$ for self-supervised learning are generated subsequently. It is worth noting that the labeling process is easy and non-artificial so that it can be performed on target domain. As shown in Fig. 2(c) and (d), the proposed auxiliary SSL task can effectively predict the boundary points. The advantages of the proposed auxiliary task could be concluded as follows: 1) thanks to this auxiliary task, the network could effectively perceive the occlusion of point clouds, so as to alleviate the impact of missing parts on network; 2) the proposed auxiliary task enables network to learn more fine-grained point-wise features under the distribution of target domain, which is actually beneficial to the prediction of target domain.

The implementation of this auxiliary task is illustrated at Fig. 4. We implement this auxiliary task by concatenating the middle features (512 Dims) after the encoder of DGCNN and the max-pooled features (1024 Dims), and then feedind the concatenated features (1536 Dims) into another well designed task head $\Phi_{aux} : \mathbb{R}^d \to [0,1]^2$, which is placed after the shared encoder Φ_{fea}. At last, we train the network by the following loss:

$$L^t_{aux} = -\frac{1}{n^t p}\sum_{i=1}^{n^t}\sum_{k=1}^{p}\sum_{j=1}^{2} \mathbf{I}\left[j = y^{t,aux}_{i,k}\right] \log\left(\Phi_{aux}\left(\Phi_{fea}\left(\tilde{x}_{i,k}\right)\right)_j\right) \tag{5}$$

where p denotes the number of points in a masked point cloud.

E. Overall Loss

The overall loss function is the sum of prediction-consistency contrastive loss on both source domain L^s_{adv} and target domain L^t_{adv}, and the self-supervised learning loss on target domain L^t_{aux}, as follows:

$$L_{overall} = \alpha L^s_{adv} + \beta L^t_{adv} + \omega L^t_{aux} \tag{6}$$

where α, β, ω are hyper-parameters to control the importance of several loss terms, and they are selected empirically.

IV. Evaluation

In this section, we evaluate the proposed method on both PointDA-10 and PointSegDA benchmarks, respectively, for point cloud classification and segmentation.

A. Datasets

PointDA-10 contains three widely-used sub-datasets: ModelNet-10, ShapeNet-10 and ScanNet-10. All of them share the same 10 categories (e.g., bed, table, sofa, chair, etc.). Therein, ModelNet-10 (**M**) contains 4183 train samples and 856 test samples, and ShapeNet-10 (**S**) contains 17378 train samples and 2492 test samples. Both ModelNet-10 and ShapeNet-10 are synthetic point clouds. ScanNet-10 (**S***) is a real-world object dataset with 6110 train samples and 1769 test samples. The objects from ScanNet-10 often lose some parts and get occluded by surroundings. Thus, the domain adaptation on M $\rightarrow$ S* and S $\rightarrow$ S* are more challenging than the others. We follow the data preparation procedure as used in [15], and a typical 80%/20% data split for training and validation on both source and target domains is employed. The point number is 1024 for PointDA-10 dataset.

PointSegDA contains four sub-datasets: ADOBE (**A**), FAUST (**F**), MIT (**M**) and SCAPE (**S**). All four sub-datasets share the same eight categories of human body parts (hand, head, feet, etc.), but with difference in point distribution, pose and human shapes. For data processing, we follow the data processing in [15] and each sample contains 2048 points sampled from mesh vertices and downsampled by farthest point sampling.

B. Implementation Details

Following [15], [17], and [34], we choose DGCNN, a commonly-used point cloud deep learning network as the backbone for the shared encoder Φ_{fea}. The classifier of main task Φ_{cls} is based on a multi-layer perceptron (MLP) with 4 fully-connected (FC) layers (i.e., 1024, 512, 256, 10) in view of 10 categories, while another classifier of auxiliary task Φ_{aux} is implemented by four fully-connected layers (i.e., 1536, 256, 256, 128, 2) in view of two categories (i.e., boundary points for 1 and others for 0), as shown in Fig. 2. The batch size and epochs for PointDA-10 are set to 48 and 150, respectively, while the batch size and epochs for PointSegDA are set to 24 and 250, respectively. We adopt ADAM [39] optimizer with learning rate 0.001 and weight decay 0.00005. A cosine annealing learning rate scheduler implemented via PyTorch is assigned in training. The hyper-parameters α, β, ω are selected to be 1, 0.1, 0.6 empirically. The number of automatically-labeled boundary points M_p is set to 128. We select the best model according to the source-validation strategy since annotations of target domain are not available.

C. Analysis of Data Augmentation

This section analyzes the adaptation ability of three state-of-the-art data augmentation methods, i.e., PointAugment [35], PointWOLF [36], and PointCutMix-K [37], to UDA of point clouds. The configuration parameters associated with these three methods are set the same as the original released codes. The results are recorded in Table I. The baseline DGCNN (w/o) has average accuracy of 62.2%. By employing PointAugment and PointWOLF respectively, +PointAugment and +PointWOLF achieve the average accuracy of 64.6% and 65.5%, increased by 2.4% and 3.3%. However, after adding PointCutMix-K, the average accuracy is only 56.3%, decreased by 5.9%. It indicates that some data augmentation methods could effectively improve the generalization ability of intra-domain, but have less adaptation ability of cross-domain. One possible reason is that both PointAugment and PointWOLF aim to learn smooth and consecutive deformation, while the augmented point clouds of PointCutMix-K are the obvious combination of two object parts under the way of replacement strategy [37]. Inconsecutive deformation of point clouds will make it more difficult to train the network between source and target domains. Furthermore, we observe that both PointAugment and PointWOLF achieve better performance than baseline on sim-to-sim and sim-to-real scenarios, but suffer a huge drop on real-to-sim scenario, i.e., S*$\rightarrow$M. Because that applying data augmentation on a dataset usually leads to the increasement of structure variance, and verifying that data augmentation is more suitable to reduce the domain gap in scenarios where the structure variance of target domain is larger than that of source domain. Based on the above analysis, we select PointAugment and PointWOLF for the following experiments and only perform data augmentation on sim-to-sim and sim-to-real scenarios.

D. Classification Results on PointDA-10

We compare here the results of a series of UDA methods on PointDA-10, including DANN [14], PointDAN [11], RS [27], DefRec (+PCM) [15], GAST [17], LST [16], GAI [30], RefRec [40], MLSP [34]. Table II shows the quantitative results. The proposed method (+PW) achieves competing performance on average accuracy (70.5%), and another Ours (+PA) outperforms all competing methods on average accuracy (71.8%) and 5 out of 6 domain adaptations, without SPST. Compared with those with multiple SSL task, i.e., GAST [17], MLSP [34], the proposed method only applies one SSL task and presents the superiority of multi-strategy. One possible reason that Our (+PA) achieves better performance than Our (+PW) is that the PointAugment is a deep learning-based network, which could be automatically optimized with other loss terms. After adopting the SPST strategy [17], [34], we further boost the performance of our method (+PA) with average accuracy of 74.1%. The

TABLE I
CLASSIFICATION ACCURACIES (%) OF THREE DATA AUGMENTATION METHODS ON THE POINTDA-10 DATASET

Method	M→S	M→S*	S→M	S→S*	S*→M	S*→S	Avg
DGCNN(w/o) [6]	81.7	42.9	72.2	44.2	**67.3**	65.1	62.2
+ PointCutMix-K [37]	83.0	37.1	65.2	33.6	59.9	59.0	56.3
+ PointAugment [35]	**83.8**	52.5	76.0	44.5	64.9	**65.6**	64.6
+ PointWOLF [36]	83.0	**55.7**	**76.3**	**49.3**	63.4	65.3	**65.5**

The bold numbers indicate the highest results in the column.

TABLE II
COMPARATIVE EVALUATION IN CLASSIFICATION ACCURACY (%) ON THE POINTDA-10 DATASET

Method	SPST	M→S	M→S*	S→M	S→S*	S*→M	S*→S	Avg
DGCNN(w/o) [6]		81.7	42.9	72.2	44.2	67.3	65.1	62.2
PointNet++(w/o) [5]		80.1	50.1	76.4	51.8	61.1	62.7	63.7
DANN [14]		75.3	41.5	62.5	46.1	53.3	63.2	57.0
PointDAN [11]		80.1	50.1	76.4	51.8	61.1	62.7	63.7
DefRec [15]		83.3	46.6	79.8	49.9	70.7	64.4	65.8
RS [27]		79.9	46.7	75.2	51.4	71.8	71.2	66.0
DefRec+PCM [15]		81.7	51.8	78.6	54.5	73.7	71.1	68.6
GAST [17]		83.9	56.7	76.4	55.0	73.4	72.2	69.5
LST [16]		82.8	56.3	81.7	54.8	72.9	71.7	70.0
GAI [30]		85.8	55.3	77.2	55.4	73.8	72.4	70.0
RefRec [40]		81.4	56.5	**85.4**	53.3	73.0	73.1	70.5
MLSP [34]		83.7	55.4	77.1	55.6	78.2	76.1	71.0
Ours(+PW)		83.3	54.3	72.4	55.8	79.0	77.9	70.5
Ours(+PA)		86.0	57.5	73.6	56.6	79.0	77.9	71.8
GAST+SPST [17]	✓	84.8	**59.8**	80.8	56.7	81.1	74.9	73.0
GAI+SPST [30]	✓	**86.2**	58.6	81.4	56.9	81.5	74.4	73.2
MLSP+SPST [34]	✓	85.7	59.4	82.3	**57.3**	82.2	76.4	73.8
Ours(+PA)+SPST	✓	85.2	59.1	79.0	56.1	**83.9**	**81.5**	**74.1**

("Ours(+PA)": DGCNN(w/o)+PointAugment+PCCL+AUX; "Ours(+PW)": DGCNN(w/o)+PointWolf+PCCL+AUX; "PCCL": prediction-consistency Contrastive loss; "AUX": auxiliary task of boundary point prediction; "SPST": self-paced self-training.)
The bold numbers indicate the highest results in the column.

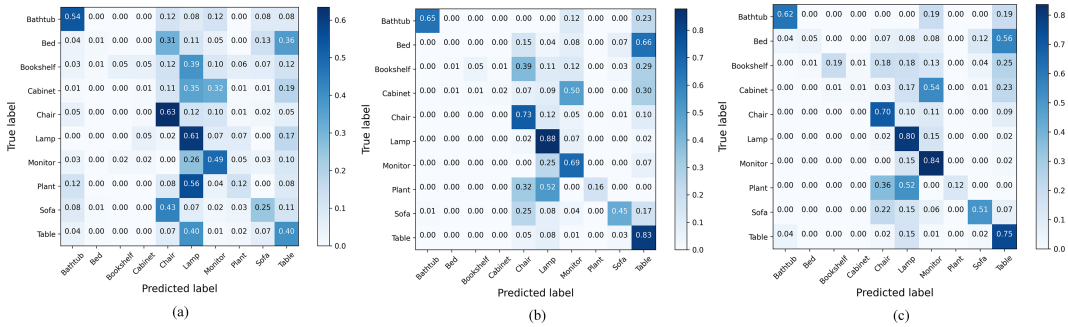

Fig. 5. Confusion matrices of classification results on target domain. (a) DGCNN(w/o): S → S*; (b) Ours(+PA): S → S*; (c) Ours(+PA) + SPST: S → S*.

confusion matrices of class-wise classification accuracy produced by the DGCNN (w/o) and the proposed UDA method (w and w/o SPST) regarding S → S* are presented at Fig. 5. The results show that our method (w/o SPST) is more discriminative among categories than the w/o adapt baseline. We also use t-SNE to visualize the feature distribution on the target domain of the DGCNN (w/o) and the proposed UDA method (w and w/o SPST) on S*→S, as shown in Fig. 6. We can observe that the proposed method (w/o SPST) draws more clear clusters than the w/o adapt baseline.

E. Segmentation Results on PointSegDA

To fully evaluate the effectiveness of the proposed auxiliary SSL task, we embed only the propose SSL task into the w/o adapt baseline, i.e., DGCNN (w/o), and perform experiments on PointSegDA dataset. The evaluation metrices on segmentation task are mean intersection over union (mIoU), and the results are recorded in Table III. By adding the proposed SSL task, our method outperforms other competitive methods on average (63.8%) and 7 out of 12 domain adaptations, which is 3% and 4.3% higher than the state-of-the-art LST [16] and MLSP [34], respectively, and 10.2% higher than the baseline. In addition, Fig. 7 visualizes the qualitative comparison on M → F domain adaptation. These results show that our method could effectively capture fine-grained semantic features and lead to better performance. Both quantitative and qualitative results demonstrate the effectiveness of our method.

More experiments on PointSegDA: We conduct more experiments on PointSegDA to evaluate the sensitivity of the proposed auxiliary task on both the number of masked points (p_1) and labeled boundary points (p_2). The p_1 is set as 128 and 256 respectively, and the p_2 is set as 64, 128, 256 respectively. The results are recorded in Table IV. From the results, most combinations of p_1 and p_2 yield competitive performances, with

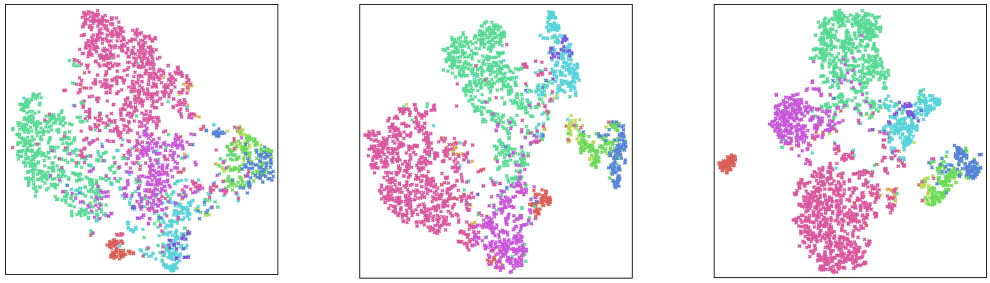

Fig. 6. t-SNE visualization of feature distribution on target domain. Different colors indicate different classes. (a) DGCNN(w/o): S*→S; (b) Ours(+PA): S*→S; (c) Our(+PA) + SPST: S*→S.

TABLE III
COMPARATIVE EVALUATION IN MEAN IOU (%) ON THE POINTSEGDA DATASET

Method	F→M	F→A	F→S	M→F	M→A	M→S	A→F	A→M	A→S	S→F	S→M	S→A	Avg
DGCNN(w/o) [6]	60.9	78.5	66.5	33.6	26.6	69.9	38.5	31.2	30.0	64.5	**74.1**	68.4	53.6
PointNet++(w/o) [5]	54.8	76.0	58.0	48.9	67.2	58.7	41.4	45.8	35.6	58.3	62.9	71.5	56.7
DefRec+PCM [15]	60.9	78.8	63.6	48.6	48.1	70.1	46.9	33.2	37.6	62.6	66.3	66.5	56.9
RS [27]	60.7	78.7	66.9	38.4	59.6	70.4	44.0	30.4	36.6	65.3	70.7	73.0	57.9
DefRec [15]	61.8	79.7	67.4	40.1	67.1	72.6	42.5	28.9	32.3	66.2	66.4	72.2	58.1
MLSP [34]	60.0	**80.9**	65.5	40.4	**67.3**	70.8	45.4	30.1	38.4	72.5	66.6	**74.8**	59.5
LST [16]	61.8	80.3	68.5	56.6	60.8	67.8	**52.3**	38.6	41.1	66.6	67.4	68.0	60.8
DGCNN(w/o)+Aux	**63.3**	76.6	**69.3**	**62.5**	65.1	**72.7**	51.7	**47.4**	**46.8**	**73.7**	63.9	72.3	**63.8**

("AUX": Auxiliary task of boundary point prediction.)
The bold numbers indicate the highest results in the column.

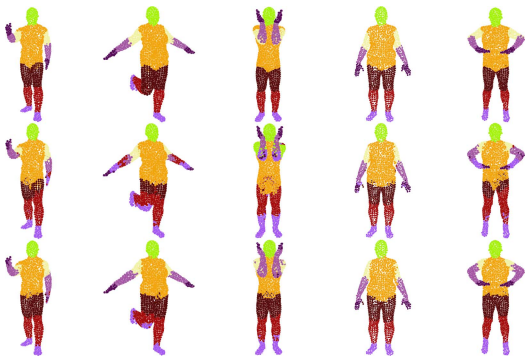

Fig. 7. Qualitative comparison on M → F domain adaptation. The top branch denotes the ground truth (different colors represent different parts); the middle branch denotes the prediction results of DGCNN(w/o); the bottom branch denotes the prediction results of our method.

TABLE IV
AVERAGE MEAN IOU(%) OF TWELVE DOMAIN ADAPTATIONS ON POINTSEGDA DATASET

	p_1=128			p_1=256		
p_2	64	128	256	64	128	256
Avg	63.3	62.7	63.0	63.7	**63.8**	63.2

The bold number indicate the highest result in the row.

an average mIoU above 63.0% across twelve domain adaptations, exhibiting weak sensitivity on these two parameters.

F. Ablation Study

In this section, ablation study is given to analyze the effectiveness of different components of our method. PA, Aux, and PccL, respectively, indicate the data augmentation with PointAugment [35], auxiliary SSL task of boundary point prediction, and prediction-consistency contrastive loss. The results are recorded in Table V. The original baseline DGCNN (w/o) has average accuracy of 62.2%. By seperately applying the PA, Aux, and PccL respectively, the average accuracy are 64.6%, 67.7%, and 68.2%, increased by 2.4%, 5.5%, and 6.0%. These results indicate that all three components can indeed improve the performance of the classifier. In the meanwhile, PA could effectively improve the domain adaptation performance in sim-to-sim and sim-to-real scenarios, but not in real-to-sim scenarios. Both Aux and PccL could improve the performance on all six domain adaptation sceanrios, however, Aux could boost more in sim-to-sim and sim-to-real scenarios while PccL improve more in real-to-sim scenarios. The results also validate that the performance of a single auxiliary task is usually limited, though the performance gains of our proposed auxiliary task could reach 5.5% (from 62.2% to 67.7%). When employing two of the three components, the average accuracy can be further improved. After adopting all three components, the average accuracy is improved to 71.8%, which achieves the best performance and presents the necessity and superiority of multi-strategy.

TABLE V
ABLATION STUDY OF THE PROPOSED METHOD (+POINTAUGMENT) ON POINTDA-10 DATASET

PA	Aux	PccL	M→S	M→S*	S→M	S→S*	S*→M	S*→S	Avg
✗	✗	✗	81.7	42.9	72.2	44.2	67.3	65.1	62.2
✓	✗	✗	83.8	52.5	76.0	44.5	64.9	65.6	64.6
✗	✓	✗	84.3	55.8	**76.1**	51.2	68.8	70.0	67.7
✗	✗	✓	83.8	46.6	72.3	53.8	77.3	**79.1**	68.2
✓	✓	✗	84.1	56.8	75.6	52.6	68.8	70.0	68.0
✓	✗	✓	83.9	54.0	69.6	54.4	77.3	**79.1**	69.7
✗	✓	✓	85.0	54.6	69.4	53.6	**79.0**	77.9	69.9
✓	✓	✓	**86.0**	**57.7**	73.6	**56.6**	**79.0**	77.9	**71.8**

("PA": pointaugment; "AUX": auxiliary task of boundary point prediction; "PCCL": prediction-consistency contrastive loss.)
The bold numbers indicate the highest results in the column.

V. Conclusion

In this letter, we propose a novel UDA method for point clouds from the perspective of multi-strategy, including data augmentation, prediction-consistency contrastive loss and an auxiliary SSL task of prediction of boundary points, respectively. We conduct extensive experiments on both PointDA-10 and PointSegDA datasets for point cloud classification and segmentation. The results prove that our method outperform others competing methods regarding classification and segmentation. We further conduct ablation study on PointDA-10 to validate the effectiveness of these three components. In the future, we expect to explore the domain adaptation problem of large-scale point clouds.

References

[1] Y. Li et al., "Deep learning for LiDAR point clouds in autonomous driving: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3412–3432, Aug. 2021.

[2] Y. Zhu et al., "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 3357–3364.

[3] R. B. Rusu, N. Blodow, Z. Marton, A. Soos, and M. Beetz, "Towards 3D object maps for autonomous household robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 3191–3198.

[4] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[5] R. Q. Charles, L. Yi, H. Su, and L. J. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[6] Y. Wang, Y. B. Sun, Z. W. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[7] H. Zhou, Y. Feng, M. Fang, M. Wei, J. Qin, and T. Lu, "Adaptive graph convolution for point cloud analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 4945–4954.

[8] S. Qiu, S. Anwar, and N. Barnes, "Geometric back-projection network for point cloud classification," *IEEE Trans. Multimedia.*, vol. 24, pp. 1943–1955, 2022.

[9] M. Xu, R. Ding, H. Zhao, and X. Qi, "PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3172–3181.

[10] X.-F. Han, Z.-Y. He, J. Chen, and G.-Q. Xiao, "3CROSSNet: Cross-level cross-scale cross-attention network for point cloud representation," *IEEE Robot Automat. Lett.*, vol. 7, no. 2, pp. 3718–3725, Apr. 2022.

[11] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu, "PointDAN: A multiscale 3D domain adaption network for point cloud representation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 7192–7203.

[12] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1–9.

[13] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2962–2971.

[14] Y. Ganin et al., "Domain adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.

[15] I. Achituve, H. Maron, and G. Chechik, "Self-supervised learning for domain adaptation on point clouds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 123–133.

[16] X. Luo, S. Liu, K. Fu, M. Wang, and Z. Song, "A learnable self-supervised task for unsupervised domain adaptation on point cloud classification and segmentation," *Front. Comput. Sci.*, vol. 17, no. 6, 2023, Art. no. 176708, doi: 10.1007/s11704-022-2435-4.

[17] L. Zou, H. Tang, K. Chen, and K. Jia, "Geometry-aware self-training for unsupervised domain adaptation on object point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6403–6412.

[18] J.-B. Weibel, T. Patten, and M. Vincze, "Addressing the Sim2Real gap in robotic 3-D object classification," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 407–413, Apr. 2020.

[19] J.-B. Weibel, T. Patten, and M. Vincze, "Robust Sim2Real 3D object classification using graph representations and a deep center voting scheme," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8028–8035, Jul. 2022.

[20] J. Chen, Y. Zhang, F. Ma, and Z. Tan, "EB-LG module for 3D point cloud classification and segmentation," *IEEE Robot. Automat. Lett.*, vol. 8, no. 1, pp. 160–167, Jan. 2023.

[21] S. Cheng, X. Chen, X. He, Z. Liu, and X. Bai, "PRA-Net: Point relation-aware network for 3D point cloud analysis," *IEEE Trans. Image Process.*, vol. 30, no. 4, pp. 4436–4448, Apr. 2021.

[22] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.

[23] L. Yi et al., "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 210, pp. 1–12, 2016.

[24] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1588–1597.

[25] M. R. Loghmani, L. Robbiano, M. Planamente, K. Park, B. Caputo, and M. Vincze, "Unsupervised domain adaptation through inter-modal rotation for RGB-D object recognition," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6631–6638, Oct. 2020.

[26] Y. Shao, L. Li, W. Ren, C. Gao, and N. Sang, "Domain adaptation for image dehazing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2808–2817.

[27] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 12962–12972.

[28] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn. Workshops*, 2013, pp. 1–6. [Online]. Available: https://www.researchgate.net/publication/280581078

[29] Y. Zou, Z. Yu, B. V. K. Vijaya Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 297–313.

[30] Y. Shen, Y. Yang, M. Yan, H. Wang, Y. Zheng, and L. Guibas, "Domain adaptation on point clouds via geometry-aware implicits," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7213–7222.

[31] H. -S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1Billion: A large-scale benchmark for general object grasping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11441–11450.

[32] H. Fan, X. Chang, W. Zhang, Y. Cheng, Y. Sun, and M. Kankanhalli, "Self-supervised global-local structure modeling for point cloud domain adaptation with reliable voted pseudo labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6367–6376.

[33] C. Huang, Z. Cao, Y. Wang, J. Wang, and M. Long, "MetaSets: Meta-learning on point sets for generalizable representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8863–8872.

[34] H. Liang et al., "Point cloud domain adaptation via masked local 3D structure prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 156–172.

[35] R. Li, X. Li, P.-A. Heng, and C.-W. Fu, "PointAugment: An auto-augmentation framework for point cloud classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6377–6386.

[36] S. Kim, S. Lee, D. Hwang, J. Lee, S. J. Hwang, and H. J. Kim, "Point cloud augmentation with weighted local transformations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 528–537.

[37] J. Zhang et al., "PointCutMix: Regularization strategy for point cloud classification," *Neurocomputing*, vol. 505, pp. 58–67, 2022.

[38] Y. Chen et al., "PointMixup: Augmentation for point clouds," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 330–345.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15, doi: 10.48550/arXiv.1412.6980.

[40] A. Cardace, R. Spezialetti, P. Z. Ramirez, S. Salti, and L. D. Stefano, "RefRec: Pseudo-labels refinement via shape reconstruction for unsupervised 3D domain adaptation," in *Proc. Int. Conf. 3D Vis.*, 2021, pp. 331–341.