Self-Supervised Global-Local Structure Modeling for Point Cloud Domain Adaptation with Reliable Voted Pseudo Labels

Hehe Fan[1] Xiaojun Chang[2] Wanyue Zhang[3,4*] Yi Cheng[4] Ying Sun[4]

Mohan Kankanhalli[1]

[1]School of Computing, National University of Singapore
[2]ReLER, University of Technology Sydney
[3]Max Planck Institute for Informatics
[4]Institute for Infocomm Research, Agency for Science, Technology, and Research, Singapore

Abstract

In this paper, we propose an unsupervised domain adaptation method for deep point cloud representation learning. To model the internal structures in target point clouds, we first propose to learn the global representations of unlabeled data by scaling up or down point clouds and then predicting the scales. Second, to capture the local structure in a self-supervised manner, we propose to project a 3D local area onto a 2D plane and then learn to reconstruct the squeezed region. Moreover, to effectively transfer the knowledge from source domain, we propose to vote pseudo labels for target samples based on the labels of their nearest source neighbors in the shared feature space. To avoid the noise caused by incorrect pseudo labels, we only select reliable target samples, whose voting consistencies are high enough, for enhancing adaptation. The voting method is able to adaptively select more and more target samples during training, which in return facilitates adaptation because the amount of labeled target data increases. Experiments on PointDA (ModelNet-10, ShapeNet-10 and ScanNet-10) and Sim-to-Real (ModelNet-11, ScanObjectNN-11, ShapeNet-9 and ScanObjectNN-9) demonstrate the effectiveness of our method.

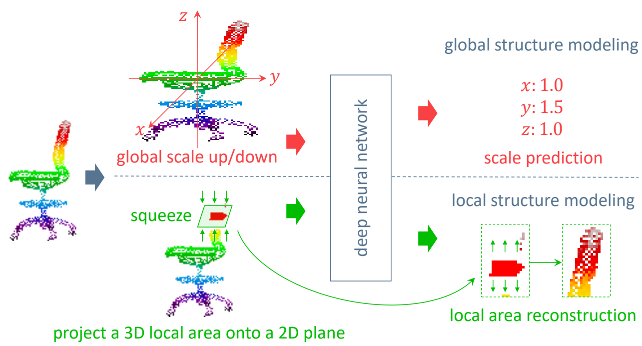


Figure 1. Illustration of the self-supervised global-local structure modeling for point cloud domain adaptation. The global structure is modeled by scaling up/down the point cloud and then predicting the scale. To capture the local structure, a random 3D local area is squeezed onto a 2D plane and then reconstructed by the network.

1. Introduction

Large-scale learning methods based on deep neural networks [7–10, 15, 25, 26, 35, 36, 39] constitute the recent advances in 3D vision, and play an important role for visual perception in intelligent platforms such as robots, drones and self-driving cars. These intelligent platforms often employ real-time depth sensors, such as LiDAR, to capture the accurate geometric information of scenes, which are represented by 3D point clouds. However, deep neural neural networks usually requires massive amounts of labeled point clouds for representation learning, which limits the scalability to the real world. To alleviate this problem, unsupervised point cloud domain adaption is recently attracting increasing attention from the community [1, 27, 31, 43]. Domain adaptation aims to transfer the knowledge from a labeled source domain to a related but unlabeled target domain, in which the source and target domains share the same feature space. However, due to different point scales, object sizes, densities, styles, sensor perspectives, *etc*., point cloud representations in the target domain inevitably deviate from the corresponding representations in the source domain, result-

*Part of this work was done when Wanyue Zhang was an intern at Agency for Science, Technology, and Research, Singapore.

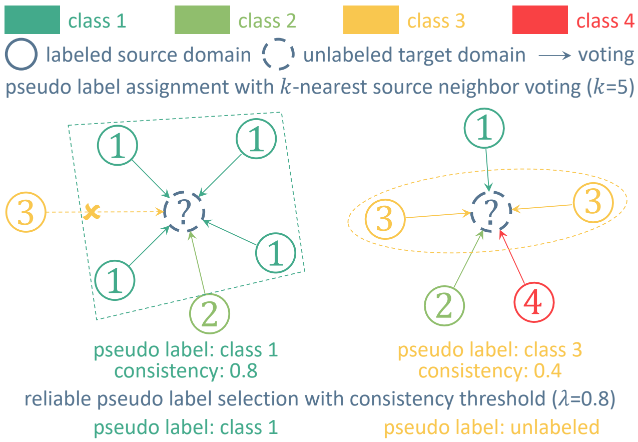


Figure 2. Illustration of reliable voted pseudo label generation. First, target point clouds' pseudo labels are voted by a few of nearest source neighbors in the feature space. Then, the target samples whose nearest source neighbor labels are consistent enough are selected as reliable training data.

ing in the domain shift or distributional shift problem.

To reduce domain shift, one solution is to directly learn from the target domain via self-supervised learning, *i.e*., exploiting the relations or correlations between different input signals. However, most of the existing methods focus on leveraging only one of the global or local structure of unlabeled data, such as predicting global vertical rotation [24], reconstructing point clouds from randomly rearranged object local parts [29], reconstructing a collapsed local region via Chamfer Distance loss [1] or localizing a curvature-changed local area [43]. Another domain adaptation solution is to transfer the knowledge from source to target domain via adversarial training [27] and self -training [43].

This paper is devoted to exploiting the self-supervised and transfer learning for point cloud domain adaptation. First, as shown in Fig. 1, we propose to learn point cloud representations by scaling up or down point coordinates of one dimension and then predicting the scale based on the other two unchanged dimensions. In this way, the network is able to capture the global structure in a self-supervised manner. To model the local structure, we propose to project a 3D local area onto a 2D plane by simply setting point coordinates of a randomly selected dimension to the same value and recover the squeezed area via mean squared error loss. Second, as shown in Fig. 2, to enhance the knowledge transfer from the source domain, we propose a voting method to assign reliable pseudo labels to target samples for self-training. Specifically, pseudo labels are voted based on a few nearest source neighbors in the shared feature space. Then, only the target point clouds whose nearest source neighbor labels are consistent enough are selected as reliable training data. With networks becoming stronger during training, our reliable voting method adaptively selects more target data, which in return facilitates learning because the amount of labeled target data increases.

To evaluate our method, we conduct experiments on the widely-used 3D domain adaptation benchmark PointDA [27], which consists of 10 shared classes from ModelNet40 [37], ShapeNet [2] and ScanNet [3]. Moreover, we also conduct experiments on a Sim-to-Real dataset [16], which consists of 11 shared classes from ModelNet40 and ScanObjectNN [34], and 9 shared classes from ShapeNet and ScanObjectNN, respectively. The contributions of this paper are threefold:

- To model the structure of unlabeled target point clouds, we propose the global scaling-up-down prediction and local 3D-2D-3D projection-reconstruction methods for point cloud domain adaptation.
- To transfer the knowledge from source domain, we propose a voting method to assign reliable pseudo labels to target samples. The method is able to iteratively select more and more target data during training, which in return facilitates learning.
- Extensive experiments on two datasets show that the proposed method effectively improves the accuracy of unsupervised domain adaptation on point clouds.

2. Related Work

Point Cloud Classification. Point cloud classification is one of the fundamental tasks for point cloud processing. Recently, a number of deep neural networks have been proposed to address this problem [25, 26, 32, 35, 36]. Most of these works aim to directly manipulate point clouds without converting irregular points into regular voxel grids, to avoid the quantization errors and high computational cost of voxelization. Since a point cloud is essentially a set of unordered points and invariant to permutations of its points, the key to point cloud processing is to design effective point-based spatial modeling operations that do not rely on point orderings. Our method is independent of these works and employ them to encode point clouds.

Self-supervised Learning. To learn from internal structures in images, self-supervised learning tries to find or exploit the relations or correlations between different input signals [5, 6, 13, 21–23, 38], *e.g*., modifying the input and (1) predict what changed or (2) ensure that the output representation does not change, such as learning spatial context of patches from an image [5], learning to count objects [22], predicting the missing pixels [23], recovering a plausible colored version of a grayscale image [38] and solving Jigsaw puzzles [21]. Self-supervised learning can also be applied to point clouds. Sauder and Sievers proposed to split an input point cloud into several parts and reconstruct the point cloud [29]. Poursaeed *et al*. proposed to rotate a point cloud and then predict the rotation angle [24]. Shen *et al*. proposed to employ geometry-aware implicits in point clouds to reduce domain biases [31]. Achituve *et al*. and Zou *et al*. proposed to first mix up two point clouds and then predict the mixed labels [1] and their angles [43], respectively. Besides, they also proposed to reconstruct and localize deformed local areas, respectively. When modeling local structure, our method is inspired by the two deformation-based methods but achieves better accuracy.

Unsupervised Domain Adaptation. Unsupervised domain adaptation has been well developed on images [11, 19, 28, 30, 33, 40, 41]. These methods can be divided into three categories. 1) Adversarial training [19, 28, 33], which aims to directly learn unbiased representations via a discriminator to judge whether the learned features are from the target domain or the source domain, and a feature generator to confuse the discriminator. 2) Style transfer [40], which employs Generative Adversarial Networks [14] to transfer source images as the target style for training. 3) Self-training with pseudo labels [4, 11, 20, 42]. Networks trained on source domains have a certain ability to recognize target images. Therefore, target data's pseudo labels can be generated for training, in which self-paced learning [18] is usually employed to reduce noisy pseudo labels. The image-based methods can also been used for point-cloud-based domain adaptation. For example, Qin *et al*. [27] employed adversarial training to learn unbiased point cloud representations. Zou *et al*. [43] employed a self-training method, equipped with self-paced learning, for point cloud domain adaptation. Our work is also based on adversarial training and self-training but we propose a novel reliable-voting-based method for pseudo label generation. Different from the existing methods that exploit the centroid label of nearest source neighbors [42] or the prediction probabilities of nearest source neighbors [20], we use the nearest source neighbor labels in a voting manner. Moreover, we integrate voting consistency into our method, so that the unreliable pseudo labels are excluded during training. Besides, point cloud domain adaptation [1, 17] is also explored in 3D segmentation. This paper focuses on representation learning via object or shape classification.

3. Proposed Method

In this section, we first introduce and formulate the point cloud domain adaption problem in Sec. 3.1. Second, we introduce the scaling-up-down method for self-supervised global structure modeling in Sec. 3.2. Third, the 3D-2D-3D project-reconstruction method for local structure learning is described in Sec. 3.3. Fourth, in Sec. 3.4, we briefly introduce the adversarial training used in our paper. Fifth, the reliable voting method for pseudo label assignment is described in details in Sec. 3.5. Finally, the overall training procedure of our method is shown in Sec. 3.6.

3.1. Problem Formulation

The goal of unsupervised domain adaptation on point clouds is to transfer the knowledge for a labeled source domain $\mathcal{S} = \{(\boldsymbol{P}_i^s, y_i^s)\}_{i=1}^{n^s}$ to an unlabeled target domain $\mathcal{T} = \{(\boldsymbol{P}_i^t)\}_{i=1}^{n^t}$, where $\boldsymbol{P} \in \mathbb{R}^{m\times 3}$, $y_i^s \in \mathcal{Y} = \{1, \cdots, c\}$, m is the number of points and c is the number of shared classes. The n^s and n^t denote the number of source and target point clouds, respectively. The key to domain adaptation is to learn a mapping function or point cloud feature generator Φ that projects point clouds from different domains into a shared feature space. The feature generator Φ can be implemented by existing deep neural networks, *e.g*., PointNet [25] and DGCNN [35], which encode a point cloud to a vector, *i.e*., $\boldsymbol{f} = \Phi(\boldsymbol{P})$.

In this paper, we assume that point features, *e.g*., color, norm or other information, are not available. In this case, domain shifts can be caused by different point scales, point densities, object sizes, sensor perspectives, object styles, *etc*. Some of the shifts can be reduced by low-level data preprocessing and augmentation. For example, the point scale and object size problems can be addressed by normalizing object coordinates to a fixed range, *e.g*., $[-1, 1]$. The density problem can be addressed by sampling, *e.g*., Farthest Point Sampling (FPS), point clouds to the same number of points. The perspective shift problem can be mitigated by rotation-based data augmentation. However, the other shifts, *e.g*., object styles, have to be reduced via high-level representations, which is the goal of unsupervised point cloud domain adaption methods.

3.2. Self-Supervised Global (G) Structure Modeling via Scaling-Up-Down Prediction

To enable Φ to capture the global structure of target data without human-annotated class labels, we propose to scale up or down coordinates and then employ a regressor Ω to predict the scale based on the point cloud feature $\boldsymbol{f}$. Specifically, suppose $\boldsymbol{s}_i = (s_i^x, s_i^y, s_i^z) \in {\mathbb{R}^{+}}^{1\times 3}$ is a random scale vector for the i-th target point cloud. Then, the coordinates $\boldsymbol{P}_i^t$ of the point cloud is scaled by $\boldsymbol{s}_i \odot \boldsymbol{P}_i^t$, where $\odot$ is element-wise multiplication and $\boldsymbol{s}_i$ is broadcasted for the multiplication. Finally, regressor Ω is used to predict the scale $\boldsymbol{s}_i$,

$$\min_{\Phi,\Omega} \mathcal{L}_g, \quad \mathcal{L}_g = \frac{1}{n^t}\sum_{i=1}^{n^t} ||\Omega\big(\Phi(\boldsymbol{s}_i \odot \boldsymbol{P}_i^t)\big) - \boldsymbol{s}_i||_2^2. \tag{1}$$

Note that, when predicting the scale, regression Ω is actually based on the relative scales of the three dimensions, instead of the absolute change. For example, when we scale up two dimensions by 2 and leave the last dimension unchanged, regressor Ω would misunderstand that the last dimension is scaled down by 0.5. To avoid this problem, we fix two of three dimensions and only scale

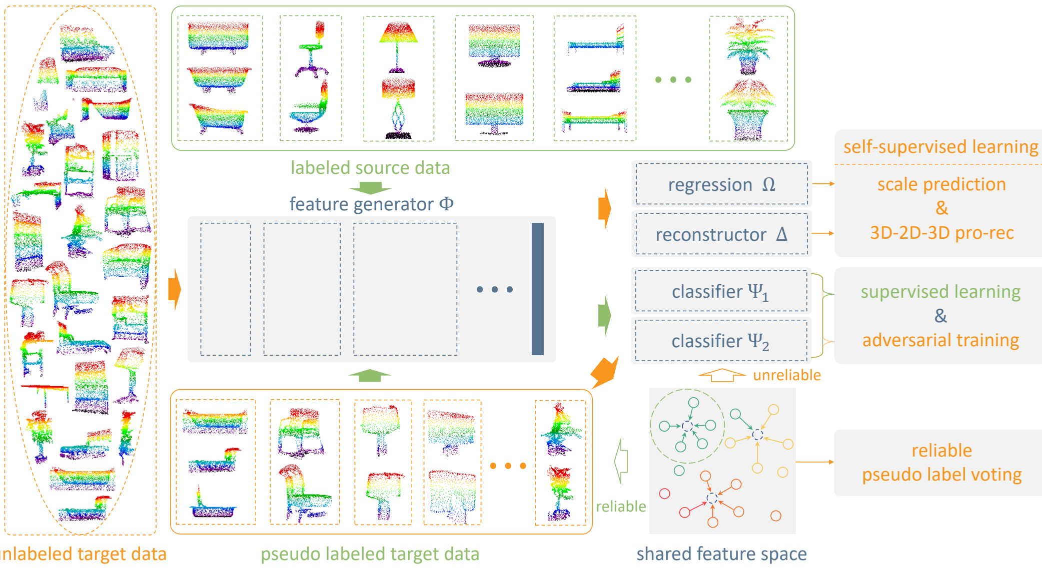


Figure 3. Illustration of the proposed self-supervised Global-Local structure modeling and Reliable Voted pseudo label method (GLRV) for point cloud domain adaption. The framework consists of a feature generator Φ to encode point clouds, a regressor Ω to predict the scaling change for self-supervised global structure modeling, a reconstructor Δ to recover squeezed region for self-supervised local structure modeling and two classifiers Ψ_1 and Ψ_2 for supervised learning and adversarial training. Besides, a reliable voting method is employed to obtain accurate target pseudo labels for enhancing domain adaptation.

up or down one dimension. Therefore, $\boldsymbol{s}_i$ is limited to $\{(s,1,1),(1,s,1),(1,1,s)\}$, where $s \in \mathbb{R}^+$. Moreover, because dramatically scaling up or down point clouds will change their structures, s is sampled from a small range, *e.g.*, $[0.5, 1.5]$ in this paper.

3.3. Self-Supervised Local (L) Structure Modeling via 3D-2D-3D Projection-Reconstruction

To enable Φ to learn the local structure without point-level human annotations, we propose a 3D-2D-3D projection-reconstruction method. Specifically, similar to [1], we first split the normalized 3D space into several regions. In this way, a point cloud is divided into multiple parts. Then, we randomly select a part that contains enough points for projection-reconstruction. Suppose $\boldsymbol{v}_i \in \{0,1\}^{m\times 1}$ is a mask vector to indicate the selected points for the i-th target point cloud. Recall that m is the number of points in the cloud. When $\boldsymbol{v}_i[j]=1$, it indicates the j-point is selected. In this way, the selected local area can be denoted as $\boldsymbol{P}_i^t[\boldsymbol{v}_i] \in \mathbb{R}^{||\boldsymbol{v}_i||_1\times 3}$. Third, we project the selected 3D area $\boldsymbol{P}_i^t[\boldsymbol{v}_i]$ onto a 2D panel. To do so, we randomly select a dimension r. Then, the r-dimensional coordinates of the selected points are squeezed to their mean. In this way, we obtain the locally projected point cloud $\hat{\boldsymbol{P}}_i^t$. Finally, a reconstructor Δ is employed to recover the projected area,

$$\min_{\Phi,\Delta} \mathcal{L}_l, \quad \mathcal{L}_l = \frac{1}{n^t}\sum_{i=1}^{n^t} \boldsymbol{v}_i \odot ||\Delta\big(\Phi(\hat{\boldsymbol{P}}_i^t)\big) - \boldsymbol{P}_i^t||_F^2, \tag{2}$$

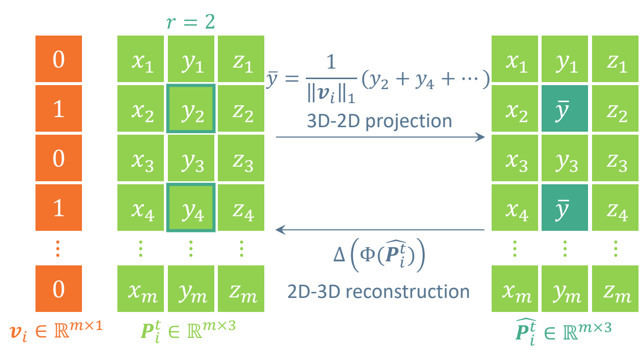


Figure 4. Illustration of the 3D-2D-3D projection-reconstruction process for the i-th target point cloud $\boldsymbol{P}_i^t$. The points of a selected 3D local area is indicated by a mask vector $\boldsymbol{v}_i$. Then, the local area is projected into 2D by squeezing the r-dimensional coordinates to their mean, resulting in the locally projected point cloud $\hat{\boldsymbol{P}}_i^t$. Finally, the feature generator Φ and reconstructor Δ is asked to reconstruct the squeezed area based on $\hat{\boldsymbol{P}}_i^t$.

where $\boldsymbol{v}_i$ is broadcasted for the element-wise multiplication. We illustrate the 3D-2D-3D projection-reconstruction process in Fig. 4.

3.4. Adversarial Training for Unbiased Representation Learning

Like most domain adaptation works, we also employ adversarial training [14] to reduce domain shifts and learn unbiased representations. In this paper, we employ Maximum

Classifier Discrepancy (MCD) [28] for adversarial training. MCD uses two classifiers Ψ_1 and Ψ_2, which map the feature vector $\boldsymbol{f}$ to two probability vectors of length c, *i.e.*, $\Psi_1(\boldsymbol{f}) \in \mathbb{R}^c$ and $\Psi_2(\boldsymbol{f}) \in \mathbb{R}^c$, respectively. Recall that c is the number of classes. For labeled data, MCD performs supervised-learning-based classification,

$$\begin{aligned} &\min_{\Phi,\Psi_1,\Psi_2} \mathcal{L}_s, \\ \mathcal{L}_s = &-\frac{1}{n^s}\sum_{i=1}^{n^s}\sum_{j=1}^{c} \mathbb{1}_{[j=y_i^s]} \cdot \log\big(\Psi_1(\Phi(\boldsymbol{P}_i^s))[j]\big) \\ &-\frac{1}{n^s}\sum_{i=1}^{n^s}\sum_{j=1}^{c} \mathbb{1}_{[j=y_i^s]} \cdot \log\big(\Psi_2(\Phi(\boldsymbol{P}_i^s))[j]\big). \end{aligned} \tag{3}$$

For unlabeled data, MCD first tries to maximize the prediction discrepancy of the two classifiers with fixed Φ,

$$\begin{aligned} &\min_{\Psi_1,\Psi_2} \mathcal{L}_s - \mathcal{L}_{adv}, \\ \mathcal{L}_{adv} = &\frac{1}{n^t c}\sum_{i=1}^{n^t} ||\Psi_1(\Phi(\boldsymbol{P}_i^t)) - \Psi_2(\Phi(\boldsymbol{P}_i^t))||_1. \end{aligned} \tag{4}$$

Then, the generator Φ is trained to minimize the discrepancy with fixed classifiers,

$$\min_{\Phi} \mathcal{L}_{adv}. \tag{5}$$

In this way, Φ is enforced to learn unbiased representations. Note that the adversarial training method is not our contribution.

3.5. Reliable Voted (RV) Target Pseudo Label Generation for Enhancing Domain Adaptation

Although adversarial training provides a way to reduce domain shifts, its effectiveness is usually limited. In this paper, we propose a self-training method to directly transfer the knowledge from source to target domain via target pseudo labels. Specifically, our method employs a voting strategy to assign pseudo labels to target samples. The pseudo labels of target point clouds are voted based on the labels of a few of their nearest source neighbors in the shared feature space. Suppose $\boldsymbol{f}_i^t$ and $\boldsymbol{f}_j^s$ are the features of the i-th target and the j-th source point clouds, respectively. Their similarity is calculated as

$$e_{ij}^{ts} = \frac{\boldsymbol{f}_i^t \cdot \boldsymbol{f}_j^s}{||\boldsymbol{f}_i^t||_2 \times ||\boldsymbol{f}_j^s||_2}. \tag{6}$$

Then, the k-nearest source neighbors are selected as follows,

$$\mathcal{N}(\boldsymbol{P}_i^t, k) = \big\{ j \mid e_{ij}^{ts} \in \text{top}-k\big(\{e_{i1}^{ts}, \cdots, e_{in^s}^{ts}\}\big)\big\}. \tag{7}$$

Third, the pseudo label of the i-th target point cloud is assigned with a voting mechanism,

$$\tilde{y}_i^t = \text{vote}\big(\{y_j^s \mid j \in \mathcal{N}(\boldsymbol{P}_i^t, k)\}\big), \tag{8}$$

where the vote function simply selects the majority as the output.

Although we employ a k-NN based voting method, the pseudo labels can be still unreliable, which may add noise into training data and lead to accuracy drop. To address this problem, we propose to only exploit reliable target point clouds, of which nearest source neighbor labels are consistent enough, to train the model with their voted pseudo labels,

$$h_i = \begin{cases} 0, & \frac{\sum_{j\in\mathcal{N}(\boldsymbol{P}_i^t,k)} \mathbb{1}_{[y_j^s=\tilde{y}_i^t]}}{k} < \lambda, \\ 1, & \frac{\sum_{j\in\mathcal{N}(\boldsymbol{P}_i^t,k)} \mathbb{1}_{[y_j^s=\tilde{y}_i^t]}}{k} \geq \lambda, \end{cases} \tag{9}$$

where $\lambda \in (0, 1]$ is the consistency threshold and h_i indicates where the i-th target point cloud is selected as a reliable training sample. Finally, the selected reliable target data is used to train the feature generator in a supervised manner,

$$\begin{aligned} &\min_{\Phi,\Psi_1,\Psi_2} \mathcal{L}_t, \\ \mathcal{L}_t = &-\frac{1}{||\boldsymbol{h}||_1}\sum_{i=1}^{n^t} h_i \sum_{j=1}^{c} \mathbb{1}_{[j=\tilde{y}_i^t]} \cdot \log\big(\Psi_1(\Phi(\boldsymbol{P}_i^t))[j]\big) \\ &-\frac{1}{||\boldsymbol{h}||_1}\sum_{i=1}^{n^t} h_i \sum_{j=1}^{c} \mathbb{1}_{[j=\tilde{y}_i^t]} \cdot \log\big(\Psi_2(\Phi(\boldsymbol{P}_i^t))[j]\big), \end{aligned} \tag{10}$$

where $\boldsymbol{h} = (h_1, \cdots, h_{n^t})$ is the selection indicator vector.

Note that, k and λ are the only two hyper-parameters of our reliable voting method. Moreover, as shown in the following experiments, we can further simplify this method by fixing λ to 1.0. In this case, k is the only hyper-parameter of this method and a bigger k indicates a higher reliability threshold.

Even though with fixed k and λ, our reliable voting method has the ability to automatically and adaptively select an increasing amount of target data during training. At the early stage of training, because the feature generator Φ is weak and the domain shift is large, only a few target point clouds, which are similar to source samples and easy to be recognized, reach the consistency threshold and are selected as reliable training data. When Φ becomes stronger and domain shift reduces, target representations become more discriminative and the consistency of nearest source neighbors' labels increases. Consequently, more target point clouds are added into the training set. In return, the increasing of labeled target data facilitates training. In this way, the feature generator Φ is improved progressively.

3.6. Overall Training

In summary, our approach includes two self-supervised learning methods, *i.e.*, scaling-up-down prediction and 3D-2D-3D projection-reconstruction, and two transfer learning

ALGORITHM 1: GLRV Training Procedure

Input : labeled source dataset $\mathcal{S} = \{(\boldsymbol{P}_i^s, y_i^s)\}_{i=1}^{n^s}$,
unlabeled target dataset $\mathcal{T} = \{(\boldsymbol{P}_i^t)\}_{i=1}^{n^t}$,
number of source neighbors for voting k,
reliability or consistency threshold λ,
feature generator Φ, classifiers Ψ_1 and Ψ_2,
regression Ω, reconstructor Δ,
number of training rounds R,
number of epochs E for each round.

Output: Φ, Ψ_1, Ψ_2, Ω and Δ.

Initialization: randomly initialize Φ, Ψ_1, Ψ_2, Ω and Δ;
randomly initialize pseudo labels $\{\tilde{y}_i^t\}_{i=1}^{n^t}$;
zero-initialize selection indicator $\boldsymbol{h} = \boldsymbol{0}$.

```
for 1 to R do
    for 1 to E do
        for (P_i^s, y_i^s), (P_i^t, ỹ_i^t) in (S, T) do
            min_{Φ,Ψ1,Ψ2} L_s with (P_i^s, y_i^s);
            min_{Φ,Ω} L_g with P_i^t;
            min_{Φ,Δ} L_l with P_i^t;
            if h_i = 1 then
                min_{Φ,Ψ1,Ψ2} L_t with (P_i^t, ỹ_i^t);
            else
                min_{Ψ1,Ψ2} L_s − L_adv with (P_i^s, y_i^s) and P_i^t;
                min_{Ψ1,Ψ2} L_adv with P_i^t;
            end
        end
    end
    update pseudo labels {ỹ_i^t}_{i=1}^{n^t} and selection indicator
      h based on Φ, k and λ;
end
```

methods, *i.e.*, adversarial training and our reliable voted self-training method. The framework of our approach is illustrated in Fig. 3. The overall training process is shown in Alg. 1. The training contains multiple rounds. After each round, we perform reliable voted pseudo label generation. Each round contains several epochs. In each epoch, we perform supervised learning, scaling-up-down prediction, 3D-2D-3D projection-reconstruction and adversarial training.

4. Experiments

4.1. Datasets

PointDA. The PointDA [27] dataset is a widely-used benchmark for point cloud domain adaptation evaluation, which extracts the samples in 10 shared classes from ModelNet40 [37], ShapeNet [2] and ScanNet [3], respectively. In this way, PointDA consists of three subsets: ModelNet-10 (M10), ShapeNet-10 (S10) and ScanNet-10 (S*10). Given the three subsets, we can conduct six types of adaptation scenarios: M10 → S10, M10 → S*10, S10 → M10, S10 → S*10, S*10 → M10 and S*10 → S10.

Sim-to-Real. The Sim-to-Real [16] dataset is a fairly new benchmark, which consists of 11 shared classes from ModelNet40 and ScanObjectNN [34], and 9 shared classes from ShapeNet and ScanObjectNN, respectively. The dataset is built to evaluate meta-learning on point clouds. In this paper, we also employ the dataset for evaluating point cloud domain adaptation. The dataset consists of four subsets: ModelNet-11 (M11), ScanObjectNN-11 (S*O11), ShapeNet-9 (S9) and ScanObjectNN-9 (S*O9). Different from PointDA, Sim-to-Real asks models to transfer knowledge from simulated ModelNet or ShapeNet to real-world ScanObjectNN. Therefore, there are two types of adaptation scenarios Sim-to-Real: M11 → S*O11 and S9 → S*O9.

Self-Supervised		Transfer			Accuracy
			Pseudo Label		
Scale	3D-2D-3D	Adversarial	Vote	Reliable	
✗	✗	✗	✗	✗	64.2
✓					69.8
	✓				67.9
✓	✓				71.2
		✓			66.7
			✓		67.4
			✓	✓	73.8
		✓	✓	✓	74.2
✓	✓	✓	✓	✓	76.2

Table 1. Ablation study on each component of our method. Experiments are conducted on PointDA with the S*10 → S10 scenario. When none of the components is employed, the model is directly transferred to the target domain without adaptation.

Method	M→S	M→S*	S→M	S→S*	S*→M	S*→S
Rotation [24]	82.8	41.7	74.0	**49.0**	64.7	68.7
Scale (ours)	**84.0**	**46.0**	**76.4**	48.3	**66.2**	**69.8**

Table 2. Comparison of vertical rotation and our scaling-up-down method for self-supervised global structure modeling on PointDA.

Adaptation	Def-Rec [1]	Def-Loc [43]	3D-2D-3D (ours)
M10 → S10	83.3	78.6	**83.5**
M10 → S*10	46.6	52.3	**53.4**
S10 → M10	**79.8**	75.0	75.7
S10 → S*10	49.9	**51.4**	48.7
S*10 → M10	**70.7**	69.3	68.2
S*10 → S10	64.4	63.6	**67.9**

Table 3. Comparison of deformation-reconstruction (Def-Rec), deformation-localization (Def-Loc) and our 3D-2D-3D method for self-supervised local structure modeling on PointDA.

Method	M10 → S10	M10 → S*10	S10 → M10	S10 → S*10	S*10 → M10	S*10 → S10
w/o Adaptation	83.3 ± 0.7	43.8 ± 2.3	75.5 ± 1.8	42.5 ± 1.4	63.8 ± 3.9	64.2 ± 0.8
DANN [12]	74.8 ± 2.8	42.1 ± 0.6	57.5 ± 0.4	50.9 ± 1.0	43.7 ± 2.9	71.6 ± 1.0
PointDAN [27]	83.9 ± 0.3	44.8 ± 1.4	63.3 ± 1.1	45.7 ± 0.7	43.6 ± 2.0	56.4 ± 1.5
RS [29]	79.9 ± 0.8	46.7 ± 4.8	75.2 ± 2.0	51.4 ± 3.9	71.8 ± 2.3	71.2 ± 2.8
DefRec + PCM [1]	81.7 ± 0.6	51.8 ± 0.3	78.6 ± 0.7	54.5 ± 0.3	73.7 ± 1.6	71.1 ± 1.4
GAST [43]	84.8 ± 0.1	59.8 ± 0.2	**80.8 ± 0.6**	56.7 ± 0.2	**81.1 ± 0.8**	74.9 ± 0.5
GLRV (ours)	**85.4 ± 0.4**	**60.4 ± 0.4**	78.8 ± 0.6	**57.7 ± 0.4**	77.8 ± 1.1	**76.2 ± 0.6**

Table 4. Accuracy on the PointDA dataset. Our GLRV method achieves four best accuracies among the six adaptation scenarios.

Method	M11 → S*O11	S9 → S*O9
w/o Adaptation	61.68 ± 1.26	57.42 ± 1.01
PointDAN [27]	63.32 ± 0.85	54.95 ± 0.87
MetaSets [16]	72.42 ± 0.21	60.92 ± 0.76
GLRV (ours)	**75.16 ± 0.34**	**62.46 ± 0.55**

Table 5. Accuracy on the Sim-to-Real dataset. Our GLRV method achieves the best accuracies on the four adaptation scenarios.

The background of ScanObjectNN objects is not used.

4.2. Implementation

Following existing works [1, 16, 43], we employ DGCNN [35] as feature generator. For PointDA, we use the setting of [1,43]. For Sim-to-Real, we follow the setting of [16]. The training contains 20 rounds, with 50 epochs in each round. Batch size is set to 32 and learning rate is set to 0.001. By default, the k and λ of reliable voting is set to 10 and 1.0, respectively.

4.3. Comparison to the State-of-the-art

For PointDA, we compare our method with the state-of-the-art point-based domain adaptation methods, including Domain Adversarial Neural Network (DANN) [12], Point Domain Adaptation Network (PointDAN) [27], Reconstruction Space Network (RS) [29], Deformation Reconstruction Network with Point Cloud Mixup (DefRec + PCM) [1] and Geometry-Aware Self-Training (GAST) [43]. We report the mean accuracy and standard error with three seeds in Table 4. Our GLRV method achieves four best accuracies on the six adaptation scenarios.

We also compare our scaling-up-down and 3D-2D-3D approaches with existing global (vertical rotation classification [24]) and local (deformation-reconstruction [1] and deformation-localization [43]) modeling methods, respectively. Results in Table 2 and Table 3 show the effectiveness of our method.

For Sim-to-Real, we compared our method with a point cloud domain adaptation method, *i.e.*, PointDAN [27], and a meta-learning method, *i.e.*, MetaSets [16]. We perform each adaptation scenarios three times and report the average and the standard deviation of the results in Table 5. Our method outperforms both the domain adaptation and meta-learning methods.

4.4. Ablation Study

A) Influence of scaling-up-down prediction, 3D-2D-3D projection-reconstruction, adversarial training and reliable voted pseudo label.

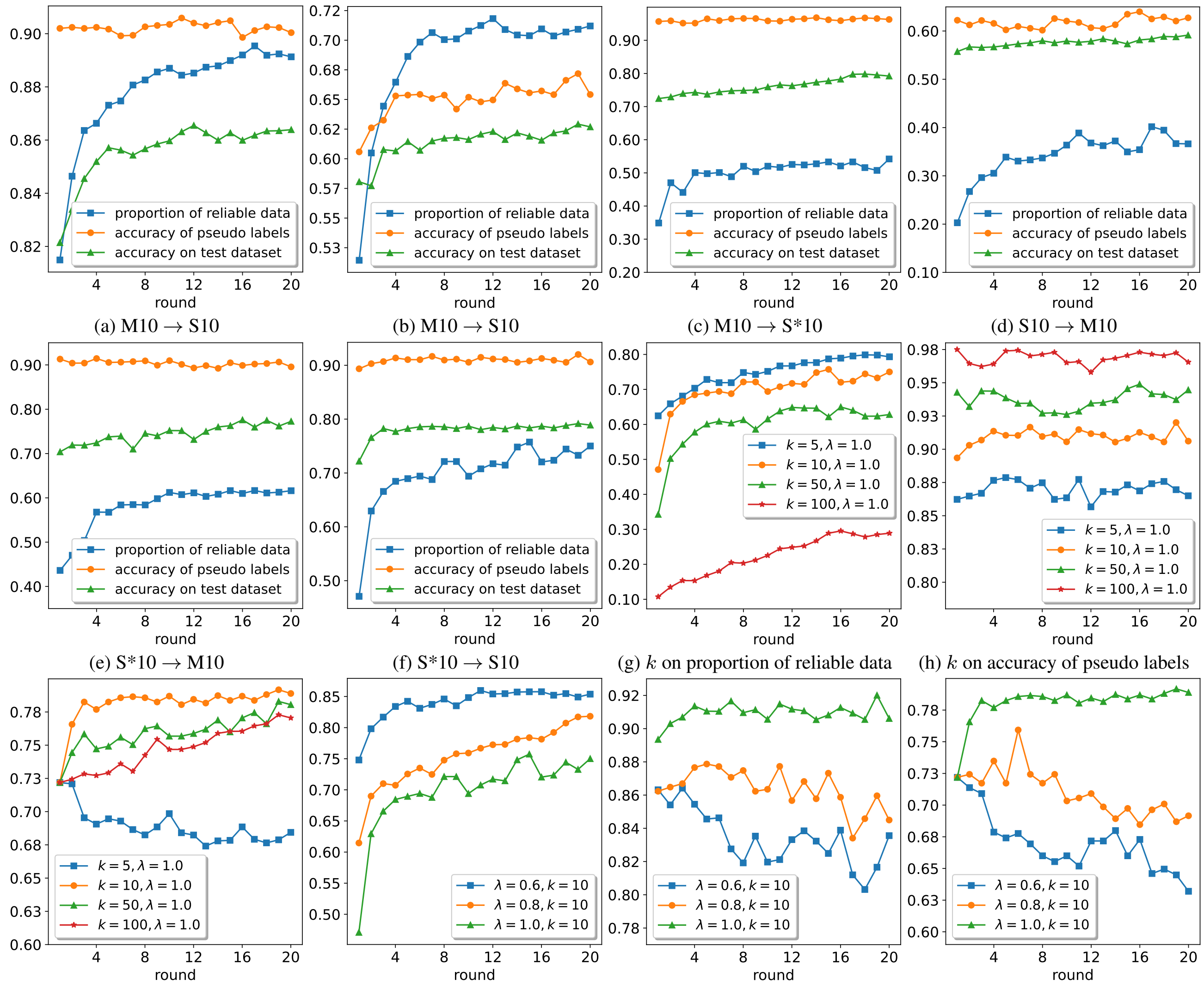
To investigate the influence of each component in our method, we conduct an ablation study on PointDA with the S*10 → S10 scenario. The results are show in Table 1. All the four components effectively improve domain adaption. Among them, the reliable voted pseudo label method (Vote + Reliable) is the most effective, which increases the baseline (64.2%) by 9.6%. Note that, without the consistency-based reliable target sample section, the improvement drops significantly. This is because the single voting method fails to obtain accurate pseudo labels and inevitably adds noise into training data.

B) Progressively selecting more and more target data and gradually improving accuracy.

To verify the ability of the reliable voting method to iteratively and adaptively select more and more target data during training, we show the number of selected reliable target samples, the accuracy of their pseudo labels and the accuracy on the test dataset in Fig. 5a ∼ Fig. 5f. Under the premise of fixed or slightly varying accuracy of pseudo labels, more and more target training data is selected, which means that the number of correctly labeled target point clouds increases. With the correct labeled target data increasing, the feature generator is gradually improved on the target domain.

C) Impact of k and λ on voted pseudo label generation and adaptation performance.

Our reliable voting method contains two hyper-parameters, *i.e.*, k and λ. To investigate the impact of the two hyper-parameters, we conduct the S*10 → S10 adaptation with different k and λ. The results are shown in Fig. 5j ∼ Fig. 5l. To investigate k, we fix λ to 1.0. When $k = 5$, a



(a) M10 → S10 (b) M10 → S10 (c) M10 → S*10 (d) S10 → M10
(e) S*10 → M10 (f) S*10 → S10 (g) k on proportion of reliable data (h) k on accuracy of pseudo labels
(i) k on test accuracy (j) λ on proportion of reliable data (k) λ on accuracy of pseudo labels (l) λ on test accuracy

Figure 5. Influence of reliable voted pseudo labels. Experiments are conducted on PointDA. (a)∼(f): With training, the proposed method adaptively selects more and more target data and test accuracies increase gradually. (g)∼(l): Impact of k and λ on voted pseudo label generation and adaptation performance (S*10 → S10). To show the real trend, test accuracies are evaluated without source validation.

large number of target samples with noisy pseudo labels are selected. In this case, the method plays a negative role to adapting. When k increases, the reliability increases, leading to the accuracy improvement. When k is too large, although the pseudo label accuracy increases, too few target data is selected. Consequently, the advantage of pseudo labels diminishes and the corresponding improvement weakens. To investigate λ, k is fixed to 10. When λ decreases from 1.0 to 0.6, the selected target data becomes less reliable. Because noise is added, the accuracy decreases.

5. Conclusion

In this paper, we propose two self-supervised learning methods, *i.e.*, scaling-up-down prediction and 3D-2D-3D projection-reconstruction, and one reliable voted pseudo label method for point cloud domain adaptation. Experiments on two datasets demonstrate the effectiveness of our approach. However, when selecting target data, our reliable voting method does not take the class balance problem into consideration. A promising improvement is to integrate class diversity into selection, instead of only reliability.

Acknowledgments

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (#A18A2b0046).

References

[1] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point clouds. In *WACV*, pages 123–133, 2021. 1, 2, 3, 4, 6, 7

[2] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *arXiv*, 1512.03012, 2015. 2, 6, 11

[3] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 2432–2443, 2017. 2, 6, 11

[4] Yuhang Ding, Hehe Fan, Mingliang Xu, and Yi Yang. Adaptive exploration for unsupervised person re-identification. *ACM Trans. Multim. Comput. Commun. Appl.*, 16(1):3:1–3:19, 2020. 3

[5] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 2

[6] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(9):1734–1747, 2016. 2

[7] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point spatio-temporal transformer networks for point cloud video modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 1

[8] Hehe Fan, Yi Yang, and Mohan S. Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *CVPR*, pages 14204–14213, 2021. 1

[9] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan S. Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. In *ICLR*, 2021. 1

[10] Hehe Fan, Xin Yu, Yi Yang, and Mohan Kankanhalli. Deep hierarchical representation of point cloud videos via spatio-temporal decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–1, 2021. 1

[11] Hehe Fan, Liang Zheng, Chenggang Yan, and Yi Yang. Unsupervised person re-identification: Clustering and fine-tuning. *ACM Trans. Multim. Comput. Commun. Appl.*, 14(4):83:1–83:18, 2018. 3

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016. 7

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 3, 4

[15] Meng-Hao Guo, Junxiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: point cloud transformer. *Comput. Vis. Media*, 7(2):187–199, 2021. 1

[16] Chao Huang, Zhangjie Cao, Yunbo Wang, Jianmin Wang, and Mingsheng Long. Metasets: Meta-learning on point sets for generalizable representations. In *CVPR*, pages 8863–8872, 2021. 2, 6, 7, 11

[17] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Émilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *CVPR*, pages 12602–12611, 2020. 3

[18] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NeurIPS*, pages 1189–1197, 2010. 3

[19] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. Maximum density divergence for domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(11):3918–3930, 2021. 3

[20] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *CVPR*, pages 16632–16642, 2021. 3

[21] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84, 2016. 2

[22] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, pages 5899–5907, 2017. 2

[23] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 2

[24] Omid Poursaeed, Tianxing Jiang, Han Qiao, Nayun Xu, and Vladimir G. Kim. Self-supervised learning of point clouds via orientation estimation. In *3DV*, pages 1018–1028, 2020. 2, 3, 6, 7

[25] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 77–85, 2017. 1, 2, 3

[26] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. 1, 2

[27] 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. In *NeurIPS*, pages 7190–7201, 2019. 1, 2, 3, 6, 7, 11

[28] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pages 3723–3732, 2018. 3, 5

[29] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *NeurIPS*, pages 12942–12952, 2019. 2, 7

[30] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *NeurIPS*, pages 2110–2118, 2016. 3

[31] Yuefan Shen, Yanchao Yang, Mi Yan, He Wang, Youyi Zheng, and Leonidas Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *CVPR*, 2022. 1, 3

[32] Hugues Thomas, Charles Ruizhongtai Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, pages 6410–6419, 2019. 2

[33] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971, 2017. 3

[34] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *ICCV*, pages 1588–1597, 2019. 2, 6, 11

[35] 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):146:1–146:12, 2019. 1, 2, 3, 7

[36] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, pages 9621–9630, 2019. 1, 2

[37] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920, 2015. 2, 6, 11

[38] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *ECCV*, pages 649–666, 2016. 2

[39] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *ICCV*, pages 16239–16248. IEEE, 2021. 1

[40] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. In *ICCV*, pages 3774–3782, 2017. 3

[41] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Invariance matters: Exemplar memory for domain adaptive person re-identification. In *CVPR*, pages 598–607, 2019. 3

[42] Qiang Zhou, Wen'an Zhou, and Shirui Wang. Semantic adaptation network for unsupervised domain adaptation. *Neurocomputing*, 454:313–323, 2021. 3

[43] Longkun Zou, Hui Tang, Ke Chen, and Kui Jia. Geometry-aware self-training for unsupervised domain adaptation object point clouds. In *ICCV*, pages 6403–6412, 2021. 1, 2, 3, 6, 7, 11