Geometry-Aware Self-Training for Unsupervised Domain Adaptation on Object Point Clouds

Longkun Zou[1,2], Hui Tang[1], Ke Chen[1,3*], and Kui Jia[1,3,4*]

[1]South China University of Technology, [2]DexForce Technology Co., Ltd
[3]Peng Cheng Laboratory, [4]Pazhou Laboratory

{eelongkunzou, eehuitang}@mail.scut.edu.cn, {chenk, kuijia}@scut.edu.cn

Abstract

The point cloud representation of an object can have a large geometric variation in view of inconsistent data acquisition procedure, which thus leads to domain discrepancy due to diverse and uncontrollable shape representation cross datasets. To improve discrimination on unseen distribution of point-based geometries in a practical and feasible perspective, this paper proposes a new method of geometry-aware self-training (GAST) for unsupervised domain adaptation of object point cloud classification. Specifically, this paper aims to learn a domain-shared representation of semantic categories, via two novel self-supervised geometric learning tasks as feature regularization. On one hand, the representation learning is empowered by a linear mixup of point cloud samples with their self-generated rotation labels, to capture a global topological configuration of local geometries. On the other hand, a diverse point distribution across datasets can be normalized with a novel curvature-aware distortion localization. Experiments on the PointDA-10 dataset show that our GAST method can significantly outperform the state-of-the-art methods. Source codes and pre-trained models are available at https://github.com/zou-longkun/GAST.

1. Introduction

The point cloud is a popular shape representation widely adopted in 3D object classification [29, 42, 30, 38], owing to its simple structure and easy acquisition. Specifically, point clouds can be generated via point sampling on the surface of object models, which is the recent typical solution to generate synthetic datasets for point cloud classification, *e.g.* the ShapeNet [5] and ModelNet [41] benchmarks. Beyond synthetic point clouds, a large-scale size of point clouds as a raw output of popular 3D sensors such as LiDAR and depth cameras can be collected in practice. Synthetic point clouds of each dataset typically follow a strategy, *e.g.* uniformly sampling over the whole object surface in the ShapeNet [5], and thus are under the controllable generation procedure. Point clouds existing in real world have a large geometric variation in view of the existence of realistic sensor noises, non-uniform point distribution, and single-view coverage of unclosed surface due to self-occlusion. In view of this, point-based shape representation can have the shifts of distribution, which desires domain adaptation techniques to improve generalization of point cloud classifiers.

On one hand, a large number of synthetic point clouds

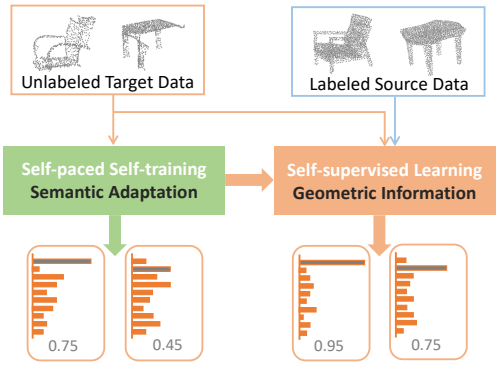


Figure 1: Concept of our geometry-aware self-training UDA on point cloud classification. We compare conventional self-paced self-training semantic adaptation (highlighted in green) and the proposed geometry-aware self-training (with both blocks). Note that the gray numbers in the bottom are predicted classification probabilities for target testing point clouds from the ScanNet [9] (*i.e.* the two target examples on the top left), which can verify the effectiveness of our self-supervised geometry-aware feature regularization on semnatic representation.

*Corresponding authors

can be readily generated based on object CAD models with corresponding semantic labels, which thus leads to sufficient labeled point cloud samples. On the other hand, real point clouds typically demand expensive manual annotations, which therefore causes a limited size of real data. In a practical perspective, a promising setting of domain adaptation on point cloud classification is to leverage the data from a label-rich source domain (*e.g*. synthetic point clouds) with mining certain inter-relation to the target task on a label-scarce target domain (*e.g*. real point clouds). Motivated by the above observation, this paper concerns on unsupervised domain adaptation (UDA) [25] of object point clouds, *i.e*. to cope with the data distribution discrepancy [4, 3] of point-based shape representation. Such a problem aims to learn a model with both training samples from labeled source and unlabeled target point sets that can classify target testing samples into one of the common semantic categories of two domains.

UDA for 2D image classification [23, 15, 13, 34, 44, 26] has been well investigated for years based on domain adaptation theories [4, 3], while very few works [31, 1] explore UDA for point cloud classification. Specifically, these UDA methods on point clouds concern on either semantic feature adaptation via explicit feature alignment across domains as existing image-based UDA [31] or self-supervised feature encoding for domain-invariant geometric patterns without bridging the domain gap of semantic features [1], resulting in a sub-optimal adaptation. Encouraged by the state-of-the-art self-training method [49], this paper adopts a self-paced self-training (SPST) scheme as our baseline to incorporate target discrimination into the semantic representation, via discovering structural similarity of inter-domain semantic patterns. However, such a SPST method is directly adapted from 2D UDA domain, which omits inherent geometric ambiguities of point cloud representations.

In view of recent success of self-supervised learning on point clouds [37, 28, 35, 38] to incorporate local or global geometries to semantic feature representation, this paper proposes a novel *Geometry-Aware Self-Training (GAST)* method for UDA on point clouds, which designs two simple yet effective self-supervised tasks to regularize semantic feature encoding beyond the SPST baseline, whose concept is also illustrated in Figure 1. Specifically, this paper introduces 1) a point cloud mixup for rotation angle classification to discover objects' global topological structure; and 2) curvature-aware distortion localization for feature robustness against inconsistent point distribution. As the source and target point clouds do not have supervision signals for the two pretext tasks, samples from both domains with self-generated rotation/location index labels can be trained jointly in a supervised style. Consequently, geometric patterns captured by self-supervised tasks are shared between both domains, which thus can further boost discrimination of semantic representation to classify target point clouds. Experiments on the 3D UDA benchmarking PointDA-10 [31] show the superiority of our proposed method over the state-of-the-art methods significantly. Our contributions are summarized as follows.

- This paper proposes a novel *Geometry-Aware Self-Training* method for unsupervised domain adaptation on object point sets, which encodes domain-invariant geometrics to semantic representation to mitigate domain discrepancy of point-based representations.
- Technically, based on self-paced self-training on unlabeled target data, our GAST integrates the self-supervised tasks of predicting rotation class and distortion location into representation learning, such that the domain-shared feature space can be constructed.
- Experiments on the public benchmark verify that the proposed GAST achieves the new state-of-the-art performance of unsupervised domain adaption on point cloud classification, especially performs consistently the best for the more important synthetic-to-real tasks.

2. Related Work

Deep Classification on Point Clouds – Most of recent point cloud classification networks [29, 42, 30, 38] concern on coping with sparsity and irregularity of point-based shape representation, which can be categorized into two groups. The first group of algorithms [19, 46, 48, 11, 45] is designed based on multi-layer perceptron (MLP), which densely encode features on each point independently to aggregate a global shape representation. The second group of algorithms [42, 8, 6] aims to encoding each point's local neighborhood into feature representation, either by constructing a spatial/spectral graph [42, 6] or by defining Euclidean convolution operation on irregular points with a continuous space (*e.g*. a sphere [8, 12]) or regular grids (*e.g*. voxels) [39, 20]. These methods for point cloud classification attempt to learn discriminative semantic features from global and/or local geometries, but very few work [31, 1] pay attention to mitigating distribution shifts of point-based representation, which is our main concern in this paper.

2D and 3D Unsupervised Domain Adaptation– Mainstream UDA methods for 2D image classification differ mainly in the strategy of reducing the discrepancy across domains and are accordingly divided into two categories. Methods in the first category minimize a proxy of the domain discrepancy, which is measured by distribution statistics [23, 15, 32] or distance metrics [44, 26, 10]. The second category includes methods aligning source and target feature distributions in an adversarial manner, *i.e*. playing minimax games [14] at the levels of domain [13, 40, 27] or category [34, 33, 17, 7]. Recently, a few works [31, 1] propose the problem of UDA on irregular point-based rep-

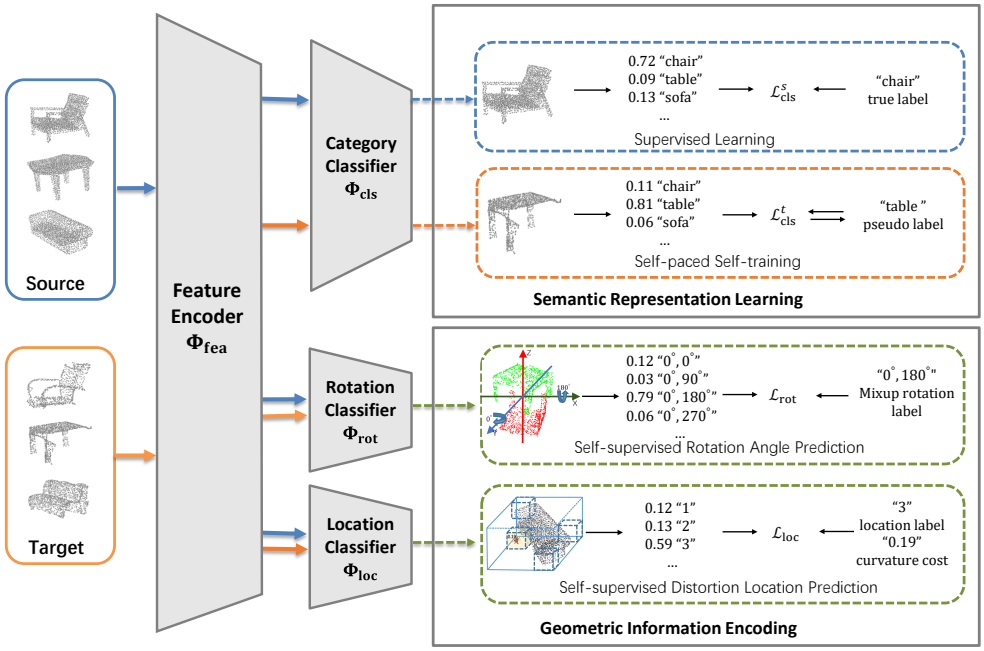


Figure 2: Overview of our *Geometry-Aware Self-Training*, which includes four key components: supervised training on source domain, self-paced self-training on target domain, self-supervised rotation angle prediction, and self-supervised distortion location prediction, corresponding to losses of $\mathcal{L}^s_{\text{cls}}$, $\mathcal{L}^t_{\text{cls}}$, $\mathcal{L}_{\text{rot}}$, and $\mathcal{L}_{\text{loc}}$. The three individual classifiers take the features from the shared feature encoder as their input. Note that the small black arrow indicates the direction of optimization.

resentation, which inherits the challenge of semantic gap as other DA problems and also has its specific challenge of domain-agnostic feature encoding from local geometries of point clouds. Qin *et al*. [31] propose a node module with adaptive receptive field to model the discriminative local structures and minimize an MMD loss to explicitly align local features across domains. Achituve *et al*. [1] learn an informative representation with abundant local geometrics in a self-supervised manner, *i.e*. reconstructing a partially distorted point cloud. Their limitations as discussed in Sec. 1 encourage us to propose our GAST method for a discriminative domain-shared representation. Experiments in Table 1 can verify superior performance of GAST to PointDAN [31] and DefRec [1].

Self-supervised Learning – Self-supervised learning leverages the input itself as supervision for a pretext task, which learns the representation benefiting downstream tasks. A comprehensive summary of existing methods in this direction is provided by [22] and we retrospect those most related ones. For learning geometric feature from point clouds, Sauder *et al*. [35] design to split an input point cloud into several parts with a random permutation on these parts and the goal is to predict the original permutation, while Poursaeed *et al*. [28] propose to rotate the whole input and predict the rotation angle. In this work, our GAST adopts two pretext tasks – the task of rotation angle prediction similar to [1] and the novel distortion localization to distinguish the distorted part from other parts, both of which are formulated into a classification problem. Effects of two pretext tasks are evaluated in Table 1, which can verify our motivation.

3. Methodology

In unsupervised domain adaptation (UDA) on point sets, given a source domain $\mathcal{S} = \{\mathcal{P}^s_i, y^s_i\}_{i=1}^{n_s}$ with n_s labeled samples and a target domain $\mathcal{T} = \{\mathcal{P}^t_i\}_{i=1}^{n_t}$ with n_t unlabeled samples, a semantic label space $\mathcal{Y}$ is shared between $\mathcal{S}$ and $\mathcal{T}$ (*i.e*. $\mathcal{Y}_s = \mathcal{Y}_t$), where point cloud $\mathcal{P} \in \mathcal{X} \subset \mathbb{R}^{m\times 3}$ consisting of m three-dimensional coordinate points (x, y, z) represents one object shape. Let the number of categories $|\mathcal{Y}|$ be C, *i.e*. $y^s \in \{1, 2, \cdots, C\}$ for any source instance $\mathcal{P}^s$. The objective of point-based UDA is to learn a domain-adapted mapping function $\Phi : \mathcal{X} \to \mathcal{Y}$ that can correctly classify point cloud samples into one of C semantic categories. In the context of deep learning, the mapping function Φ can be formulated into a cascade of a feature encoder $\Phi_{\text{fea}} : \mathcal{X} \to \mathbb{R}^d$ for any input $\mathcal{P}$ and a classifier $\Phi_{\text{cls}} : \mathbb{R}^d \to [0, 1]^C$ typically using fully-connected layers as follows:

$$\Phi(\mathcal{P}) = \Phi_{\text{cls}}(\boldsymbol{z}) \circ \Phi_{\text{fea}}(\mathcal{P}) \tag{1}$$

where d denotes the dimension of the feature representation output $\boldsymbol{z} \in \mathcal{Z}$ of $\Phi_{\text{fea}}(\mathcal{P})$. Denote the category probability vector of $\mathcal{P}$ as $\boldsymbol{p} = \Phi_{\text{cls}}(\boldsymbol{z}) = [p_1, \cdots, p_C]$ subject to $\sum_{i=1}^{C} p_i = 1$. Since both $\mathcal{S}$ and $\mathcal{T}$ domains by assumption follow different data distributions, the main challenge in point-based UDA is to reduce domain discrepancy of fea-

ture encoding Φ_{fea} in terms of semantics and geometrics.

This paper introduces a novel geometry-aware self-training (GAST) method for UDA on point set classification, whose pipeline is illustrated in Figure 2. Specifically, the proposed GAST method is made up of two parts – semantic feature adaptation (see Sec. 3.1) and geometry-aware regularization (see Sec. 3.2). Without explicit feature alignment, our GAST applies iterative self-training with self-paced learning on target data to adapt semantic representation, which is extracted from source data in a supervised learning style. To complement the semantic representation learning, the proposed GAST regularizes feature learning via incorporating global and local geometric structures by self-supervision of predicting rotation angle and distortion location.

3.1. Self-Paced Semantic Feature Adaptation

We aim to learn an adaptive classification model for generalizing knowledge induced from a labeled source domain to an unlabeled target one. As discussed before, due to domain discrepancy, the semantic representation generated by learning the classification model $\Phi_{\text{cls}} \circ \Phi_{\text{fea}}$ on source data with category labels y, can lead to classification performance degrading significantly when applied to the instances of unlabeled target domain. In this way, we propose to learn the domain-shared semantic representation via training the same network $\Phi_{\text{cls}} \circ \Phi_{\text{fea}}$ with source and target samples jointly. For training with unlabeled target samples in a supervised learning method, we adopt the self-training scheme in a self-paced learning manner to optimally select confident target samples, which together with pseudo labels are fed into the classification model Φ to refine semantic feature with target discrimination.

Supervised Learning on Source Domain – Denote the labeled source samples $\{\mathcal{P}_i^s, y_i^s\}_{i=1}^{n_s}$ and their category probability vectors $\{\boldsymbol{p}_i^s\}_{i=1}^{n_s}$ predicted by the model Φ, which is trained via minimizing the cross-entropy loss:

$$\min_{\Phi_{\text{fea}},\Phi_{\text{cls}}} \mathcal{L}_{\text{cls}}^s = -\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{c=1}^{C} \mathrm{I}[c = y_i^s] \log p_{i,c}^s, \tag{2}$$

where $p_{i,c}^s$ is the c-th element of category prediction $\boldsymbol{p}_i^s$ of a source point cloud $\mathcal{P}_i^s$, and $\mathrm{I}[\cdot]$ is an indicator function. Supervised learning establishes in feature space $\mathcal{Z}$ a semantic representation $\boldsymbol{z}$ that is discriminative among categories on source domain $\mathcal{S}$.

Target Domain Self-training with Self-paced Learning – As the ground truth labels of target samples are unavailable, we take a direct strategy of self-training [18] that uses pseudo labels to guide the model learning. We have no guarantee in the correctness of the obtained pseudo labels but expect that they are mostly correct. To this end, we employ a self-paced learning in an easy-to-hard learning manner [49], which generates pseudo labels from category predictions at the higher levels of confidence. As a result, the objective of self-paced learning based self-training is depicted as:

$$\begin{aligned} \min_{\Phi_{\text{fea}},\Phi_{\text{cls}},\widehat{\boldsymbol{Y}}^t} \mathcal{L}_{\text{cls}}^t &= -\frac{1}{n_t}\sum_{i=1}^{n_t}\left(\sum_{c=1}^{C} \widehat{y}_{i,c}^t \log p_{i,c}^t + \gamma|\widehat{\boldsymbol{y}}_i^t|_1\right) \\ \text{s.t. } & \widehat{\boldsymbol{y}}_i^t \in \{\{\boldsymbol{e}|\boldsymbol{e}\in\mathbb{R}^C\}\cup\mathbf{0}\}, \forall i \in \{1,2,\cdots,n_t\} \\ & \gamma > 0, \end{aligned} \tag{3}$$

where $\widehat{\boldsymbol{y}}_i^t$ is the assigned pseudo label vector for a target instance $\mathcal{P}_i^t$, $\widehat{y}_{i,c}^t$ is its c-th element, $\widehat{\boldsymbol{Y}}^t$ is the set of all pseudo label vectors $\{\widehat{\boldsymbol{y}}_i^t\}_{i=1}^{n_t}$, $\boldsymbol{e}$ is a one-hot vector, $\mathbf{0}$ is a C-dimensional vector with all zero elements, and γ is a hyper-parameter.

Similar to Eq. (2), the first term in Eq. (3) aims to maximize the mutual information between the selected input $\mathcal{P}^t$ and its assigned label $\widehat{y}^t$ over the same $\Phi_{\text{cls}}\circ\Phi_{\text{fea}}$, giving rise to discriminative features and decision boundaries adapted to the target domain. This indeed makes sense for classification of target samples since the optimal classifiers in individual domains disagree [36]. In the second term, the negative L_1 loss is used to avoid degenerate solutions that assign all $\widehat{\boldsymbol{y}}^t$ as $\mathbf{0}$, *i.e.* ignoring all target samples in network training. γ controls the number of selected target samples. The larger γ, the more samples. More specifically, the optimization of Eq. (3) alternates between the following steps.

- *Updating pseudo labels* – We first fix the model Φ and minimize $\mathcal{L}_{\text{cls}}^t$ in Eq. (3) over the pseudo label vector set $\widehat{\boldsymbol{Y}}^t$. By solving a nonlinear integer programming, we have the optimized solution as follows [49]:

$$\widehat{y}_{i,c}^t = \begin{cases} 1 & \text{if } c = \arg\max_{c'} p_{i,c'}^t,\ p_{i,c}^t > \exp(-\gamma) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

- *Updating the model* Φ – We then fix all pseudo label vectors $\{\widehat{\boldsymbol{y}}_i^t\}_{i=1}^{n_t}$ and minimize $\mathcal{L}_{\text{cls}}^t$ in Eq. (3) over the model Φ.

Denote the maximum of $\boldsymbol{p}^t$ as the measure of easiness for a target instance $\mathcal{P}^t$, *i.e.* $\max_{c'} p_{c'}^t$. During training, the model iteratively adapts the semantic representation with the more confident target samples. With semantic representation increasingly adaptive between the source and target domains, the target samples with less confidence (*i.e.* harder samples) are explored subsequently. Such an easy-to-hard scheme also conforms to the optimization dynamics of supervised learning [2], which first learns easier examples that better fit patterns.

Semantic Representation Learning – By combining source domain supervised learning of Eq. (2) and target

domain self-training with self-paced learning of Eq. (3), we formulate the objective of semantic representation learning:

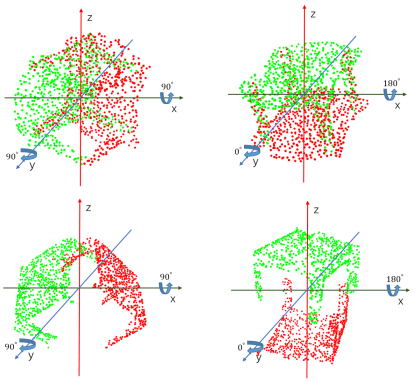
$$\min_{\Phi_{\text{fea}},\Phi_{\text{cls}},\widehat{\boldsymbol{Y}}^t} \mathcal{L}_{\text{sem}} = \mathcal{L}^s_{\text{cls}} + \lambda \mathcal{L}^t_{\text{cls}}, \tag{5}$$

where $\lambda \in [0,1]$ is a penalty parameter to suppress the noisy signal at the early stage of training. Without explicitly aligning feature across domains, the objective (5) enforces feature encoder Φ_{fea} to directly output domain-shared semantic representation in $\mathcal{Z}$ with source and target data jointly. Since the two domains share the same label space $\mathcal{Y}$, their samples corresponding to the same categories would ideally be pushed closer to each other in $\mathcal{Z}$, *naturally* achieving feature alignment across domains.

3.2. Self-Supervised Geometric Feature Encoding

The feature output of Φ_{fea} can be ambiguous due to intra-class shape variation, which is made even more challenging for inconsistent distribution of point cloud representation across domains. As a result, with a common object classifier Φ_{cls} on the source and target domains, learning domain-invariant geometric feature from point clouds is an alternative solution to improve representation discrimination, which is verified in [1]. To this end, we propose to complement the self-training based semantic adaptation with two pretext tasks, *i.e.* the rotation angle prediction and the distortion location prediction, in a self-supervised learning (SSL) fashion, which can model geometric invariance across domains. Note that, the proposed self-supervised geometric feature encoding is utilized in feature encoding on both source and target domains, whose supervision signals are generated automatically from the data as other self-supervised learning methods [35, 1].

Rotation Angle Prediction on Point Cloud Mixup – Given a point cloud $\mathcal{P}$, we first randomly sample a Mixup coefficient $\alpha \in (0,1)$, which is used to sample two shapes $\mathcal{P}_a \in \mathbb{R}^{\lfloor \alpha \cdot m \rfloor \times 3}$ and $\mathcal{P}_b \in \mathbb{R}^{\lfloor (1-\alpha) \cdot m \rfloor \times 3}$ from $\mathcal{P}$ respectively using farthest point sampling (as in [29]), where $\lfloor \alpha \cdot m \rfloor$ and $\lfloor (1-\alpha) \cdot m \rfloor$ are the number of sampled points in $\mathcal{P}_a$ and $\mathcal{P}_b$ and $\lfloor \cdot \rfloor$ denote the floor function to output the integers. Finally, we form a new point cloud mixup $\widetilde{\mathcal{P}}$ by clockwise rotating the $\mathcal{P}_a$ along the x-axis and clockwise rotating the $\mathcal{P}_b$ along the y-axis by two randomly-selected angles within $\{0°, 90°, 180°, 270°\}$ respectively. To implement an auxiliary task for rotation angle prediction of point cloud mixup, we additionally stack a Mixup rotation classifier $\Phi_{\text{rot}} : \mathcal{Z} \rightarrow [0,1]^R$ on top of the feature extractor $\Phi_{\text{fea}}(\cdot)$, where R is the number of rotation angle classes. Two illustrative examples are given in Figure 3.[1] Following other mixup operation in [47], the label mixup as $\alpha \widetilde{y}_a \cup (1-\alpha)\widetilde{y}_b$ is also generated to form a training pair with corresponding point cloud mixup $\widetilde{\mathcal{P}} \in \mathbb{R}^{m \times 3}$, where $\widetilde{y}_a \in \{1,2,3,4\}$ and $\widetilde{y}_b \in \{5,6,7,8\}$ denote the rotation class labels of $\mathcal{P}_a$ and $\mathcal{P}_b$. The rotation classifier $\Phi_{\text{rot}} \circ \Phi_{\text{fea}}(\widetilde{\mathcal{P}})$ takes as input the rotated point cloud mixup $\{\widetilde{\mathcal{P}}_i\}_{i=1}^{n_s+n_t}$ from both the source and target domains. Given rotation labels $\{\widetilde{y}_{a,i}, \widetilde{y}_{b,i}\}_{i=1}^{n_s+n_t}$, we have the following objective for rotation angle classification:



(a) rot 90°, 90° (b) rot 0°, 180°

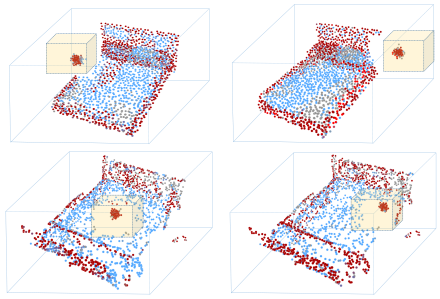
Figure 3: Illustrative examples of generating two Mixup samples of the Table class, by combining sampled point sub-sets for predicting rotation classes along the x-axis and y-axis. Two rows correspond to the ModelNet-10 and the ScanNet-10 of the PointDA-10 [31] respectively.

$$\min_{\Phi_{\text{fea}},\Phi_{\text{rot}}} \mathcal{L}_{\text{rot}} = -\frac{1}{n_s+n_t} \sum_{i=1}^{n_s+n_t} \sum_{r=1}^{R} \Big(\alpha \mathrm{I}[r = \widetilde{y}_{a,i}] \log \widetilde{p}_{i,r} + (1-\alpha) \mathrm{I}[r = \widetilde{y}_{b,i}] \log \widetilde{p}_{i,r} \Big), \tag{6}$$

where $R = 8$ and $\widetilde{p}_{i,r}$ is the r-th element of the predicted rotation probability vectors $\widetilde{\boldsymbol{p}}_i = \Phi_{\text{rot}} \circ \Phi_{\text{fea}}(\widetilde{\mathcal{P}}_i)$. Optimizing the objective (6) enables the model to perceive global and topological configuration of local shape primitives in 3D space as [28].

Curvature-Aware Distortion Localization – To incorporate local geometries into feature representation, the state-of-the-art methods [37, 35, 1] have explored the pretext tasks w.r.t. location or distortion, *e.g.* reconstructing a point cloud with randomly displaced or distorted parts. Inspired by their success, this paper proposes a simple yet effective pretext task, *i.e.* predicting the location distorted point set with explicitly incorporating geometric property – curvature. Intuitively, the higher curvature, the richer geometric information preserve [21]. To this end, we first obtain curvature of each point by direct physical computation based on principle component analysis (PCA) within a local region, *e.g.* seeking an optimal plane that best fitting the central point and its k-nearest neighbours. We then voxelize the point cloud $\mathcal{P}$ into k^3 voxels, from which we randomly select one at equal probability and replace all points within

[1] Note that, following [1, 29] point clouds in our paper are tolerant for arbitrary rotations along the z-axis.



(a) loc 1 (b) loc 2

Figure 4: Illustration of bed examples with different distortion locations and corresponding curvature-based target codes for classification. Two rows correspond to the ModelNet-10 and the ScanNet-10 domains of the PointDA-10 [31] respectively.

such a voxel with an equal number of points sampled from an isotropic Gaussian distribution, where the mean is the center of the sampled voxel and the standard deviation is typically small. Instead of using one-hot target coding, a soft target code revealing local geometries, *e.g.* pointwise geometric property such as curvature, can enforce the network to focus on regions with higher curvature in a cost-sensitive learning manner. In details, the curvature cost is the ratio of the curvature sum of any point subset to be distorted to those of the whole point cloud. By taking these steps, the distorted point clouds $\{\overline{\mathcal{P}}_i\}_{i=1}^{n_s+n_t}$ for both domains are produced, as shown in Figure 4. Such a distortion localization with a curvature-sensitive label can be formulated into a classification problem. Specifically speaking, we stack a location classifier $\Phi_{\text{loc}} : \mathcal{Z} \to [0,1]^L$ on top of $\Phi_{\text{fea}}(\cdot)$ for discovering one distorted geometric cell from the whole point cloud, where $L = k^3$ is the number of voxels to cover all the shape surface. Given the indexes of the chosen cells as location labels $\{\overline{y}_i \in \{1,2,\cdots,L\}\}_{i=1}^{n_s+n_t}$ and the corresponding curvature cost $\{\overline{c}_i \in R\}_{i=1}^{n_s+n_t}$, we train the location prediction model $\Phi_{\text{loc}} \circ \Phi_{\text{fea}}$ by the following objective:

$$\min_{\Phi_{\text{fea}},\Phi_{\text{loc}}} \mathcal{L}_{\text{loc}} = -\frac{1}{n_s+n_t} \sum_{i=1}^{n_s+n_t} \overline{c}_i \sum_{l=1}^{L} \mathrm{I}[l=\overline{y}_i] \log \overline{p}_{i,l} \quad (7)$$

where $\overline{p}_{i,l}$ is the l-th element of the predicted location probability vector $\overline{\boldsymbol{p}}_i = \Phi_{\text{loc}} \circ \Phi_{\text{fea}}(\overline{\mathcal{P}}_i)$. Intuitively, the objective (7) can capture the geometric information from local distributions of spatial points in a point cloud via infering where the distorted points are.

Geometric Feature Encoding – By combining the two self-supervised learning objectives (6) and (7), we form the objective of geometry encoding as:

$$\min_{\Phi_{\text{fea}},\Phi_{\text{rot}},\Phi_{\text{loc}}} \mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{rot}} + \mathcal{L}_{\text{loc}}. \quad (8)$$

Since feature encoding function Φ_{fea} is shared across semantic feature adaption and geometric feature learning, the objective (8) forces Φ_{fea} to learn expressive features capturing global and local geometric invariance across domains via a proxy loss of the two pretext tasks. In other words, by learning global rotation and local deformation of point clouds $\mathcal{P}$ in both $\mathcal{S}$ and $\mathcal{T}$ domains, the domain discrepancy is further reduced in view of construction of common geometric feature space using both source and target data. Specifically, in self-supervised learning tasks, both source and target data will be assigned an automatically generated rotation/distortion index label, and geometric features jointly learned from both source and target data can thus be more robust against point distribution variations in different domains.

3.3. Overall Training and Inference

The overall training objective integrates semantic representation learning (5) and geometry-aware regularization (8), leading to a unified framework of *Geometry-Aware Self-Training (GAST)* for UDA, as follows:

$$\min_{\Phi_{\text{fea}},\Phi_{\text{cls}},\widehat{\boldsymbol{Y}},\Phi_{\text{rot}},\Phi_{\text{loc}}} \mathcal{L}_{\text{GAST}} = \mathcal{L}_{\text{sem}} + \beta \mathcal{L}_{\text{geo}}, \quad (9)$$

where β is a trade-off hyper-parameter and all model parameters of the proposed GAST (cf. Figure 2) are simultaneously learned in an end-to-end manner. Once trained, our model can be simply deployed as a conventional classification model by discarding the classifiers of rotation and distortion location. During testing, we infer the category label for any target test point cloud $\mathcal{P}$ as $\arg\max_c p_c$, where p_c is the c-th element of the predicted category probability vector $\boldsymbol{p} = \Phi_{\text{cls}} \circ \Phi_{\text{fea}}(\mathcal{P})$.

4. Experiments

4.1. Dataset and Settings

Dataset – The PointDA-10 [31] collects object point clouds of 10 shared classes from the ModelNet40 [41], the ShapeNet [5] and the ScanNet [9], leading to the following three distinct domains. **(1)** the ModelNet-10 (**M**) consists of 4,183 training and 856 testing point clouds by sampling 2,048 points from the surface of clean 3D CAD models by following the method [30]. **(2)** ShapeNet-10 (**S**) includes 17,378 training and 2,492 testing point clouds uniformly sampled on the surface of ShapeNet objects, each one also containing 2,048 points. Note that, the ShapeNet-10 is more heterogeneous than the ModelNet-10 since the ShapeNet has more object instances, among which a larger structure variance exists. **(3)** ScanNet-10 (**S***) comprises 6,110 training and 1,769 testing samples which collect 2,048 points from partially visible object point clouds of the ScanNet,

	LocCls	RotCls	SPST	M→S	M→S*	S→M	S→S*	S*→M	S*→S	Avg.
Supervised				93.9 ± 0.2	78.4 ± 0.6	96.2 ± 0.1	78.4 ± 0.6	96.2 ± 0.1	93.9 ± 0.2	89.5 ± 0.3
w/o Adapt				83.3 ± 0.7	43.8 ± 2.3	75.5 ± 1.8	42.5 ± 1.4	63.8 ± 3.9	64.2 ± 0.8	62.2 ± 1.8
DANN [13]				74.8 ± 2.8	42.1 ± 0.6	57.5 ± 0.4	50.9 ± 1.0	43.7 ± 2.9	71.6 ± 1.0	56.8 ± 1.5
PointDAN [31]				83.9 ± 0.3	44.8 ± 1.4	63.3 ± 1.1	45.7 ± 0.7	43.6 ± 2.0	56.4 ± 1.5	56.3 ± 1.2
RS [35]				79.9 ± 0.8	46.7 ± 4.8	75.2 ± 2.0	51.4 ± 3.9	71.8 ± 2.3	71.2 ± 2.8	66.0 ± 1.6
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	68.6 ± 0.8
	✓			78.6 ± 0.3	52.3 ± 0.2	75.0 ± 0.2	51.4 ± 0.3	69.3 ± 0.2	63.6 ± 0.2	65.1 ± 0.2
		✓		84.3 ± 0.2	46.2 ± 0.3	69.8 ± 0.6	49.2 ± 0.3	66.6 ± 0.5	66.1 ± 0.2	63.7 ± 0.4
GAST			✓	84.4 ± 0.4	45.9 ± 0.5	80.5 ± 0.3	48.7 ± 0.4	64.8 ± 0.3	70.4 ± 0.3	65.8 ± 0.4
	✓	✓		83.9 ± 0.2	56.7 ± 0.3	76.4 ± 0.2	55.0 ± 0.2	73.4 ± 0.3	72.2 ± 0.2	69.5 ± 0.2
	✓	✓	✓	**84.8** ± 0.1	**59.8** ± 0.2	**80.8** ± 0.6	**56.7** ± 0.2	**81.1** ± 0.8	**74.9** ± 0.5	**73.0** ± 0.4

Table 1: Comparative evaluation in classification accuracy (%) averaged over 3 seeds (± SEM) on the PointDA-10 dataset.

	LocCls	RotCls	SPST	Bathtub	Bed	Bookshelf	Cabinet	Chair	Lamp	Monitor	Plant	Sofa	Table	Avg.
Supervised				76.9	58.8	55.5	73.2	92.5	63.4	70.5	72.0	56.0	85.0	70.4
w/o Adapt				61.5	31.8	32.9	0	49.8	36.6	54.1	96.0	30.6	47.5	44.1
DANN [13]				34.6	38.8	34.2	2.7	59.4	12.2	49.2	84.0	**53.0**	57.8	42.6
PointDAN [31]				34.6	36.5	35.6	**3.4**	61.2	29.3	37.7	76.0	44.8	45.5	40.4
DefRec + PCM [1]				**65.4**	**49.4**	49.3	1.3	61.4	41.4	55.7	88.0	42.5	60.8	51.5
	✓			61.5	31.8	**51.4**	2.0	61.8	34.1	32.8	76.0	41.0	65.1	45.8
		✓		53.8	28.2	37.7	2.0	54.9	7.3	63.9	84.0	40.3	62.5	43.5
GAST			✓	57.7	35.3	45.2	3.3	54.3	34.1	49.2	76.0	51.5	50.2	45.7
	✓	✓		61.5	44.7	41.8	**3.4**	70.2	39.0	68.9	88.0	38.1	66.8	52.2
	✓	✓	✓	57.7	38.8	35.6	2.0	**74.3**	**43.9**	**77.0**	**96.0**	45.5	**74.1**	**54.5**

Table 2: Evaluation of class-wise classification accuracy (%) on the ModelNet-10 to the ScanNet-10 (M→S*).

within manually annotated bounding boxes. As the ScanNet contains point clouds of scanned and reconstructed real-world scenes, point clouds are usually incomplete in view of occlusion with contextual objects in the scenes in addition to realistic sensor noises. We follow the data preparation and data settings used in [1]. Specifically, all object point clouds in all domains (*i.e.* datasets) are aligned along the x and y axes, only tolerant for arbitrary rotations along the z axis. Moreover, a point subset containing 1,024 points are down-sampled from the original 2,048 point clouds provided by the PointDA-10 and is normalized within a unit ball with random jittering as [29], which is adopted in all the methods for a fair comparison. A typical 80%/20% data split for training and testing on both source and target domains is employed [1].

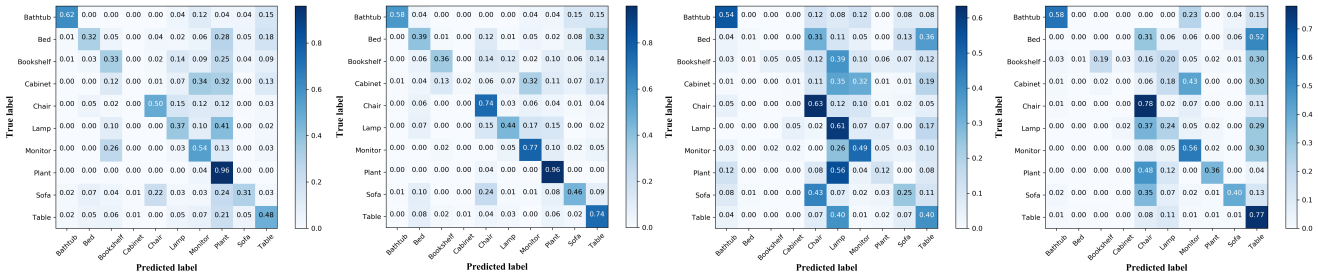
Comparative Methods – We compare our proposed GAST with a serial of representative UDA methods on image classification and the state-of-the-art point-based DA methods including Domain Adversarial Neural Network (**DANN**) [13], Point Domain Adaptation Network (**PointDAN**) [31], Reconstruction Space Network (**RS**) [35], and Deformation Reconstruction Network with Point Cloud Mixup (**DefRec + PCM**) [1]. The **Supervised** method, that trains the same backbone classifier $\Phi_{\text{cls}} \circ \Phi_{\text{fea}}$ with labeled target data only, and the **w/o Adapt** method that trains the identical backbone net with only labeled source samples, are also evaluated as references of the upper and lower performance bounds, respectively. All comparative methods take the same training protocol and the best models are selected according to source-validation based early stopping.

Implementation Details – For our GAST, we adopt the DGCNN [42] as backbone of the *Feature Encoder* Φ_{fea}, while the *Category Classifier* Φ_{cls} is based on a multi-layer perceptron (MLP) with three fully connected (FC) layers (*i.e.* $\{512, 256, 10\}$) in view of 10 semantic classes in the PointDA-10. For self-supervised rotation and distorted part classifiers, the GAST respectively employs two two-layer MLPs (*i.e.* $\{512, 4\}$) in view of Mixup rotation angle classification mentioned in Sec. 3.2, and a three-layer MLP (*i.e.* $\{512, 256, 27\}$) where the whole object surface is voxelized into 3^3 cells. The hyper-parameters of γ and β are empirically set to 0.05 and 1 respectively. During training, on the selected target samples with pseudo labels, we follow [43] to augment data with random rotation along the z-axis. The Adam optimizer [16] is utilized with the initial learning rate 0.001, weight decay 0.00005 and an epoch-wise cosine annealing learning rate scheduler. In total, we train all the methods for 150 epochs with batch size 16 on an NVIDIA GTX-1080 Ti GPU and perform three trials of different random seeds.

4.2. Results

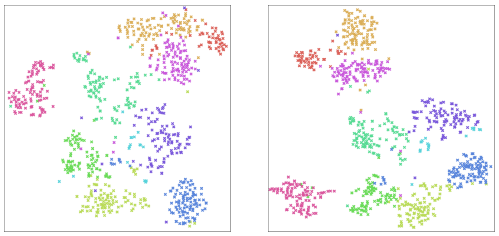
Comparison with the State-of-the-art Methods – Table 1 presents comparisons between our proposed GAST and other competing methods on the PointDA-10 dataset. Evidently, the GAST[2] outperforms all comparative domain adaptation methods with a significant margin, with improving the average accuracy by 4.4% and 16.7% over the state-of-the-art DefRec + PCM [1] and the PointDAN [31] respectively. It is noteworthy that for the challenging yet

[2]The GAST in this paragraph indicates the variant with all components.



(a) w/o Adapt: M → S* (b) GAST: M → S* (c) w/o Adapt: S → S* (d) GAST: S → S*

Figure 5: Confusion matrices of classifying testing samples on target domain.



(a) w/o Adapt: S* → M (b) GAST: S* → M

Figure 6: The t-SNE visualization of feature distribution on the target domain. Colors indicate different classes.

realistically significant synthetic-to-real tasks of **M** → **S*** and **S** → **S***, the GAST acquires a remarkable enhancement over w/o Adapt by 16% and 14.2% respectively. Visualization of confusion matrices in terms of class-wise classification accuracy achieved by the w/o Adapt and our GAST on two synthetic-to-real UDA tasks of M → S* and S → S* shown in Figure 5. As the w/o Adapt baseline and the proposed GAST use the identical DGCNN backbone, significant performance gain (10.8% on average) of our method over the baseline can only be credited to the design of our geometry-aware self-training method to learn a better semantic representation, which is more adaptive across domains and more discriminative among categories, benefiting from integrating self-paced semantic adaptation with self-supervised geometric encoding. More importantly, in comparison with the DefRec + PCM, the proposed GAST achieves superior performance on two synthetic-to-real **M** → **S*** and **S** → **S*** tasks with 8.0% and 2.2% performance gain respectively. Class-wise classification accuracy on the task M → S* is also reported in Table 2. Compared with existing methods, the proposed GAST achieves the better performance on most of the classes, especially those major classes with much more samples such as Chair and Table classes. Results of our method for long-tailed classes such as the Bathtub and Bed classes can be comparable to those of the DefRec + PCM, but our method remains its superiority to the DANN and the PointDAN. The performance gap between head and long-tailed classes of our method can be explained by the representation learning demanding sufficient samples to characterize semantics across domains.

Ablation Studies – We examine the effects of three key components of our GAST, *i.e*. **LocCls** (distortion location prediction), **RotCls** (rotation angle prediction), and **SPST** (self-paced self-training) respectively. Tables 1 and 2 compare the five GAST variants that contain different combinations of these components: **(1)** LocCls only, **(2)** RotCls only, **(3)** SPST only, **(4)** RotCls + LocCls, and **(5)** RotCls + LocCls + SPST. We highlight the main observations below. First, each component has a positive impact and method (5) with all the components achieves the best performance, verifying that all components of our GAST are complementary. Second, self-supervised geometric encoding is effective to handle with distribution shifts of point-based shape representation, whose results (*i.e*. RotCls + LocCls) without the SPST can still outperform the state-of-the-art methods, which can be explained by joint representation learning on source and target data with self-generated labels to capture common geometric patterns across domains.

Feature Visualization – We utilize t-SNE [24] to visualize the feature distribution on the target domain of the UDA task S* → M of the baseline and our GAST in Figure 6. In view of an imbalanced data distribution, features of the head classes (*e.g*. the yellow and green ones) with more samples are emphasized during representation learning, and thus can be more discriminative than those of the baseline.

5. Conclusion

This work aims to learn a domain-shared representation of semantic categories on point clouds via a novel Geometry-Aware Self-Training (GAST) method. Experiments on the PointDA-10 benchmark can verify the effectiveness of key components in our scheme, achieving the new state-of-the-art performance.

Acknowledgements

This work was partially supported by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.: 2017ZT07X183), the National Natural Science Foundation of China (No.: 61771201, 61902131), and the Guangdong R&D key project of China (No.: 2019B010155001).

References

[1] I. Achituve, H. Maron, and G. Chechik. Self-supervised learning for domain adaptation on point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 123–133, 2021. 2, 3, 5, 7

[2] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien. A closer look at memorization in deep networks. In *Int. Conf. Mach. Learn.*, pages 233–242, 2017. 4

[3] S. Ben-David, John B., K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Adv. Neural Inform. Process. Syst.*, pages 137–144, 2007. 2

[4] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79:151–175, 2010. 2

[5] A. X Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *ArXiv*, 1512.03012, 2015. 1, 6

[6] C. Chen, G. Li, R. Xu, T. Chen, M. Wang, and L. Lin. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[7] X. Chen, S. Wang, M. Long, and J. Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *Int. Conf. Mach. Learn.*, volume 97, pages 1081–1090, 2019. 2

[8] T. Cohen, M. Geiger, Jonas Köhler, and M. Welling. Spherical cnns. *ArXiv*, 2018. 2

[9] A. Dai, A. X Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5828–5839, 2017. 1, 6

[10] Z. Deng, Y. Luo, and J. Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *Int. Conf. Comput. Vis.*, pages 9943–9952, 2019. 2

[11] Y. Duan, Y. Zheng, J. Lu, J. Zhou, and Q. Tian. Structural relational reasoning of point clouds. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[12] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis. Learning so(3) equivariant representations with spherical cnns. In *Eur. Conf. Comput. Vis.*, 2018. 2

[13] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:2096–2030, 2016. 2, 7

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Adv. Neural Inform. Process. Syst.*, pages 2672–2680, 2014. 2

[15] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4888–4897, 2019. 2

[16] D. P Kingma and J. Ba. Adam: A method for stochastic optimization. *ArXiv*, 1412.6980, 2014. 7

[17] C. Lee, T. Batra, M. H. Baig, and D. Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10277–10287, 2019. 2

[18] D.-H. Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning (WREPL)*, 2013. 4

[19] J. Li, B. M. Chen, and G. H. Lee. So-net: Self-organizing network for point cloud analysis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2

[20] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: Convolution on x-transformed points. In *Adv. Neural Inform. Process. Syst.*, pages 820–830, 2018. 2

[21] J Lin, X Shi, Y Gao, K Chen, and K Jia. Cad-pu: A curvature-adaptive deep learning solution for point set upsampling. *arXiv preprint arXiv:2009.04660*, 2020. 5

[22] X. Liu, F. Zhang, Z. Hou, Z. Wang, L. Mian, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive. *ArXiv*, 2006.08218, 2020. 3

[23] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan. Transferable representation learning with deep adaptation networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41:3071–3085, 2019. 2

[24] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9:2579–2605, 2008. 8

[25] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22:1345–1359, 2010. 2

[26] Y. Pan, T. Yao, Y. Li, Y. Wang, C. Ngo, and T. Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2234–2242, 2019. 2

[27] P. O. Pinheiro. Unsupervised domain adaptation with similarity learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8004–8013, 2018. 2

[28] O. Poursaeed, T. Jiang, Q. Qiao, N. Xu, and V. Kim. Self-supervised learning of point clouds via orientation estimation. In *3DV*, 2020. 2, 3, 5

[29] C. R Qi, H. Su, K. Mo, and L. J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 652–660, 2017. 1, 2, 5, 7

[30] C. R. Qi, L. Yi, H. Su, and L. J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inform. Process. Syst.*, pages 5099–5108, 2017. 1, 2, 6

[31] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. In *Adv. Neural Inform. Process. Syst.*, pages 7192–7203, 2019. 2, 3, 5, 6, 7

[32] A. Rozantsev, M. Salzmann, and P. Fua. Beyond sharing weights for deep domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41:801–814, 2019. 2

[33] K. Saito, Y. Ushiku, T. Harada, and K. Saenko. Adversarial dropout regularization. In *Int. Conf. Learn. Represent.*, 2018. 2

[34] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3723–3732, 2018. 2

[35] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *Adv. Neural Inform. Process. Syst.*, pages 12962–12972, 2019. 2, 3, 5, 7

[36] R. Shu, H. Bui, H. Narui, and S. Ermon. A DIRT-t approach to unsupervised domain adaptation. In *Int. Conf. Learn. Represent.*, 2018. 4

[37] Y. Sun, E. Tzeng, T. Darrell, and Alexei A. Efros. Unsupervised domain adaptation through self-supervision. *ArXiv*, 1909.11825, 2019. 2, 5

[38] L. Tang, K. Chen, C. Wu, Y. Hong, K. Jia, and Z. Yang. Improving semantic analysis on point clouds via auxiliary supervision of local geometric priors. *ArXiv*, 2001.04803, 2020. 1, 2

[39] H. Thomas, C. R Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Int. Conf. Comput. Vis.*, pages 6411–6420, 2019. 2

[40] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2962–2971, 2017. 2

[41] K. V. Vishwanath, Diwaker Gupta, Amin Vahdat, and Ken Yocum. Modelnet: Towards a datacenter emulation environment. In *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*, pages 81–82, 2009. 1, 6

[42] Y. Wang, Y. Sun, Z. Liu, S. E Sarma, M. M Bronstein, and J. M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1, 2, 7

[43] Q. Xie, M.-T. Luong, E. Hovy, and Q. V Le. Self-training with noisy student improves imagenet classification. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10687–10698, 2020. 7

[44] S. Xie, Z. Zheng, L. Chen, and C. Chen. Learning semantic representations for unsupervised domain adaptation. In *Int. Conf. Mach. Learn.*, pages 5423–5432, 2018. 2

[45] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[46] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian. Modeling point clouds with self-attention and gumbel subset sampling. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[47] H. Zhang, M. Cisse, Y.N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 5

[48] H. Zhao, Li Jiang, C. Fu, and J. Jia. PointWeb: Enhancing local neighborhood features for point cloud processing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[49] Y. Zou, Z. Yu, B.V.K. Vijaya Kumar, and J. Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Eur. Conf. Comput. Vis.*, pages 289–305, 2018. 2, 4