Unsupervised Domain Adaptation on Point Cloud Classification via Imposing Structural Manifolds into Representation Space

Hongchao Zhong, Li Yu✉
Nanjing University of Information Science and Technology
Nanjing, China
{202212200013,li.yu}@nuist.edu.cn

Longkun Zou✉
South China University of Technology
Guangzhou, China
eelongkunzou@mail.scut.edu.cn

Ke Chen
Peng Cheng Laboratory
Shenzhen, China
chenk02@pcl.ac.cn

Abstract

In the context of unsupervised domain adaptation (UDA) for point cloud classification, deep classifiers training on data from source domain (e.g. clean synthetic point clouds) cannot perform well on those from target domain (e.g. noisy real-world ones), which can be caused by a significant domain discrepancy of point representations. For closing domain gap, recent algorithms adopt the popular self-training strategies (e.g. self-paced self-training) but suffer from lack of imposing structural constraints into representation learning. To tackle this issue, we propose a novel dual-augmentation relational learning scheme (i.e. introducing consistency regularization on augmented samples in both observation and feature space) to incorporate low-dimensional manifolds to encourage domain-invariant representations. Moreover, we design a novel filtering mechanism that adaptively adjusts thresholds for each semantic category based on confidence distributions and validates neighborhood consistency to further mitigate feature ambiguities. Comprehensive experiments on the widely-used PointDA-10 dataset demonstrate that our method achieves the state-of-the-art performance.

Keywords: ***Unsupervised domain adaptation point cloud classification low-dimensional manifolds pseudo-label refining.***

1. Introduction

Point cloud classification has garnered significant attention due to its wide range of applications, including autonomous driving, 3D reconstruction, and robotics. However, one of the main challenges in this field arises when deep learning models trained on synthetic data, such as clean point clouds from CAD models [2, 26], are applied to noisy real-world point clouds [5]. The domain discrepancy between synthetic and real-world data often leads to significant performance degradation. This issue is particularly pronounced in unsupervised domain adaptation (UDA) [15], where no labels are available in the target domain. UDA aims to bridge the domain gap by transferring knowledge from a labeled source domain to an unlabeled target domain.

Most existing methods [1, 7, 13, 23, 31] employ self-supervised tasks to pretrain a model, which is then used to generate pseudo-labels for the unlabeled target domain samples. This is followed by a self-training process that refines the model's predictions iteratively. In this process, two critical aspects are the *generation* and *filtering* of pseudo-labels. However, these methods lack the imposition of structural constraints in representation learning, which limits their ability to effectively capture domain-invariant features, thereby affecting the accuracy of pseudo-label generation. Moreover, they often overlook the optimization of the filtering mechanism in later stages, which can cause issues during self-training, such as failing to assign pseudo-labels to categories with lower confidence.

To address these challenges, we propose two key innovations. First, we introduce a low-dimensional manifolds consistency constraint inspired by ReSSL [30], which helps reduce domain discrepancies by capturing the relational structure in the representation space. By comparing the similarity distributions in both feature and prediction spaces between weakly and strongly augmented samples, this strategy reveals the consistency constraint of low-dimensional manifolds and promotes feature alignment, further improving the quality of the initial pseudo-labels. Furthermore, we

adopt a dynamic growth strategy to progressively refine feature representations by aligning classification predictions across augmented versions of the data, thereby enhancing the model's generalization capability. Second, we design a novel pseudo-label filtering mechanism that improves the reliability of the labels generated during self-training. Unlike prior methods that apply a uniform high threshold across all categories, we employ an adaptive thresholding mechanism that adjusts based on the confidence distributions for each semantic category, ensuring pseudo-labels are generated across all classes. To further mitigate ambiguity, we implement neighborhood consistency validation, which evaluates the consistency of pseudo-labels within a local neighborhood, thereby filtering out unreliable samples and selecting more accurate ones for self-training.

Our approach is evaluated on the widely-used PointDA-10 [20] dataset, where it achieves state-of-the-art performance, demonstrating its effectiveness in addressing domain discrepancies in point cloud classification. The key contributions of this paper are as follows:

- We propose a dual-augmentation relational learning scheme to incorporate low-dimensional manifolds to encourage domain-invariant representations, thereby improving the accuracy of pseudo-label generation.
- We design a novel filtering mechanism that improves pseudo-label reliability through adaptive thresholds and neighborhood consistency validation, ensuring the effectiveness and stability of the self-training process.
- Our experimental results on a widely-recognized benchmark demonstrate that our method achieves state-of-the-art performance in unsupervised domain adaptation for point cloud classification.

Source codes and pre-trained models will be released[1].

2. Related Work

2.1. Deep Learning on Point Clouds

Point clouds are sets of points that effectively capture three-dimensional spatial information in a simple and direct way, making classification an essential task in point cloud analysis. However, due to their irregular structure and permutation invariance, traditional 2D deep learning methods cannot be directly applied to point clouds. To address this, several deep neural networks designed specifically for point clouds have been introduced. Qi et al. [18] pioneered deep learning by directly processing raw point clouds, though it lacked the ability to capture local geometric features. To overcome this limitation, Qi et al. [19] integrated both global and local geometric information in a hierarchical structure. Wang et al. [27] created a feature space graph and continuously updates it to aggregate features. Zhao et al. [29] introduced the Transformer architecture for point cloud processing, achieving notable performance across various benchmarks.

2.2. Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) for 2D images has been extensively researched for many years, with two primary strategies emerging: minimizing the domain discrepancy proxy [6, 10, 14, 16] and adversarial training [8, 17, 21]. The first approach focuses on measuring and reducing the statistical differences between domains, while the second uses adversarial techniques to align feature distributions through a minimax game at the domain or class level. Furthermore, pseudo-labeling techniques [4, 9, 12, 24] are used to generate pseudo-labels for target domain data, which helps refine the model and narrow the domain gap. Motivated by advancements in the image domain, UDA has been extended to the point cloud field as well. For example, Qin et al. [20] were the first to apply UDA techniques to point cloud classification, employing adversarial training to align features across different domains. Achituve et al. [1] introduced the deformation-reconstruction task as a self-supervised learning strategy, aimed at extracting informative representations by capturing rich local geometric details. Building on this, Zou et al. [31] introduced a deformation localization task and also predicted the rotation angle of mixed point clouds. Fan et al. [7] identified both global and local patterns in point clouds by predicting scaling factors and reconstructing regions after compression. Shen et al. [23] employed the learning of geometry-aware implicit fields as a self-supervised method. Liang et al. [13] encoded point clouds by predicting three different local properties. Wang et al. [28] improved domain-invariant point cloud representations by progressively concentrating on key points based on geometric consistency. Katageri et al. [11] leveraged multimodal contrastive learning to improve the separation of categories and employed optimal transport to narrow the domain gap. Most of these methods lack structural constraints in representation learning, which is the focus of this paper.

2.3. Pseudo-labels in Self-training on Point Clouds

Self-training is a method which a model learns from its own predictions by generating pseudo-labels for unlabeled data and using them for further training. This process enables the model to adapt to new domains, progressively reducing the domain gap. The success of self-training depends on the accuracy of pseudo-labels, which is particularly crucial in point cloud tasks. To improve its effectiveness, various methods have been proposed. For instance,

[1] https://github.com/Vencoders/PCUDA-MCC

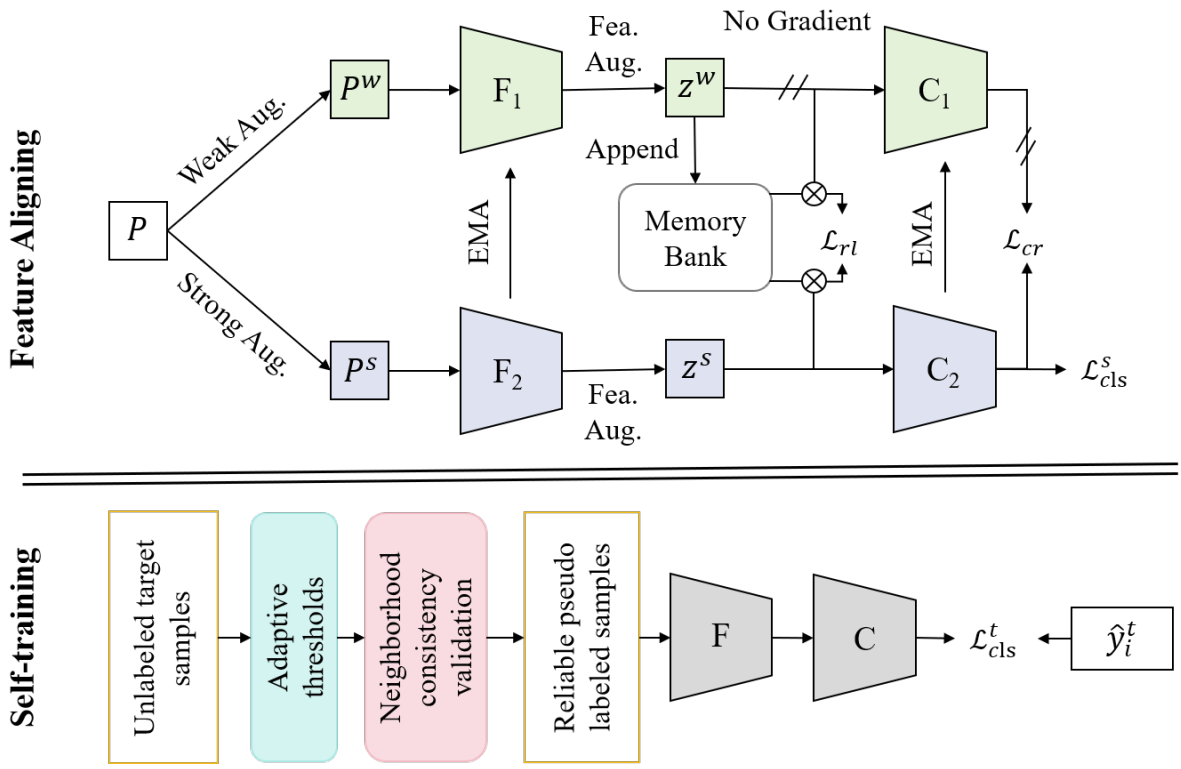


Figure 1. The framework of our proposed method for unsupervised domain adaptation on point clouds.

Zou et al. [31] utilized a self-training approach with self-paced learning to ensure high-quality pseudo-labels are selected for each category. Fan et al. [7] introduced a voting mechanism for pseudo-label generation, effectively improving their reliability. Liang et al. [13] employed an entropy-guided self-paced learning approach, selecting target samples with low prediction probability entropy. Chen et al. [3] introduced a quasi-balanced self-training method that dynamically adjusted the sample ratio to maintain a balanced proportion of pseudo-labeled samples across categories. These methods apply uniform strict criteria across all categories, which can lead to categories with lower overall confidence lacking pseudo-labeled samples. To address this issue, we adaptively set thresholds based on the characteristics of each category and perform additional filtering to select more reliable samples.

3. Method

In this section, we first introduce and formulate the problem of unsupervised domain adaptation for point clouds in Sec. 3.1. Next, we introduce a low-dimensional manifolds consistency constraints to better capture domain-invariant features in 3.2. Meanwhile, we introduce a structural consistency constraint that is gradually incorporated in a dynamic manner in Sec. 3.3. Furthermore, we present a novel pseudo-label filtering mechanism in Sec. 3.4. Finally, the overall loss function and training strategy are described in Sec. 3.5.

3.1. Problem Formulation

In the task of unsupervised domain adaptation (UDA) for point cloud classification, we are given an unlabeled target domain $\mathcal{D}^t = \{(P_i^t)\}_{i=1}^{n_t}$ with n_t unlabeled samples and a labeled source domain $\mathcal{D}^s = \{(P_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labeled samples, where y_i^s denotes the class label of the i-th source sample P_i^s, and its value lies in the shared semantic label space $\mathcal{Y} = \{1, \cdots, C\}$, which is common to both the source and target domains. A point cloud $P \in \mathbb{R}^{N\times 3}$ is composed of N points, where every point is represented by three-dimensional spatial coordinates, and C is the number of categories. Our goal is to learn a domain-invariant mapping function $\Phi : P \rightarrow \mathcal{Y}$ that accurately classifies unlabeled target samples, where $\Phi = \Phi_{Cls} \circ \Phi_{Fea}$. The feature encoder $\Phi_{Fea} : \mathbb{R}^3 \rightarrow \mathbb{R}^D$ maps the input P to a D-dimensional feature representation, and $\Phi_{Cls} : \mathbb{R}^D \rightarrow [0, 1]^C$ is a classifier that maps this feature to a probability distribution across C classes. The framework of our method is presented in Figure 1.

3.2. Low-dimensional Manifolds Consistency Constraint

Low-dimensional manifolds consistency constraints introduced through relational learning encourage the model to capture the structural distribution of features within the representation space. This approach aids in bridging domain gaps and enhances the accuracy of pseudo-labels generated for unlabeled target samples. Building on this idea, we propose a dual-augmentation strategy that operates in both observation and feature spaces. This strategy simulates various disturbances encountered during point cloud acquisition in the observation space while simultaneously enriching the feature representations in the feature space.

Observation Space. In the observation space, we simulate disturbances in point cloud collection by introducing different types of noise: Gaussian noise, salt-and-pepper noise, and distance attenuation noise. Gaussian noise simulates measurement errors by introducing random variations to each point's coordinates. Salt-and-pepper noise randomly selects a proportion of points and sets their coordinates to extreme values, representing occasional extreme errors. Distance attenuation noise reflects the decreasing accuracy of depth sensors with distance by increasing the noise level proportionally to the point's distance from the sensor. Additionally, we apply common data augmentation techniques to distinguish between weak and strong augmentations, such as cropping and scaling.

Feature Space. Observation space augmentation primarily focuses on transformations at the data level, while feature space augmentation operates on the high-level features extracted by the model. Augmenting in the feature space enables direct manipulation of high-dimensional feature representations, generating more diverse features and improving the model's adaptability to different data distributions. After extracting features with the feature extractor, we apply Gaussian noise within the feature space to further enhance these representations. Unlike observation space augmentation, this directly affects high-level representations, allowing the model to capture more abstract variations beyond the data level. This enhances the model's noise robustness, leading to greater stability and reliability in real-world scenarios.

Specifically, two augmented versions, $P_i^w = Aug_w(P_i)$ and $P_i^s = Aug_s(P_i)$, of the input point cloud P_i are first generated through observation space augmentation, where $Aug_w(\cdot)$ denotes weaker augmentation methods, and $Aug_s(\cdot)$ applies stronger augmentation techniques. Subsequently, the corresponding feature embeddings are computed:

$$\boldsymbol{z}_i^w = g(\alpha \cdot \Phi_{Fea}(P_i^w) + \beta), \tag{1}$$

$$\boldsymbol{z}_i^s = g(\alpha \cdot \Phi_{Fea}(P_i^s) + \beta), \tag{2}$$

where $g(\cdot)$ is a linear projection layer. The parameters $\alpha \sim \mathcal{N}(1, \sigma_1)$ and $\beta \sim \mathcal{N}(0, \sigma_2)$ are sampled from two Gaussian distributions, with σ_1 and σ_2 as scalar hyperparameters. We first compute the similarity distribution of weakly and strongly augmented samples relative to the samples in the memory bank:

$$\boldsymbol{r}_i^w = \frac{\exp(Sim(\boldsymbol{z}_i^w, \boldsymbol{z}_k)/\tau_1)}{\sum_{j=1}^{J} \exp(Sim(\boldsymbol{z}_i^w, \boldsymbol{z}_j)/\tau_1)}, \tag{3}$$

$$\boldsymbol{r}_i^s = \frac{\exp(Sim(\boldsymbol{z}_i^s, \boldsymbol{z}_k)/\tau_2)}{\sum_{j=1}^{J} \exp(Sim(\boldsymbol{z}_i^s, \boldsymbol{z}_j)/\tau_2)}, \tag{4}$$

where $\boldsymbol{z}_k$ represents the k-th sample in the memory bank which stores the most recent J samples, similar to ReSSL [30], using a First-In-First-Out (FIFO) strategy. The function $Sim(\cdot)$ computes the similarity between two features. The temperature coefficients, τ_1 and τ_2, with $\tau_1 < \tau_2$, are used to control the sharpness of the target distribution, where a lower τ_1 produces a sharper distribution. Our goal is to enforce structural consistency between the two similarity distributions using a cross-entropy loss function, defined as follows:

$$\mathcal{L}_{rl} = -\frac{1}{n_s + n_t} \sum_{i=1}^{n_s+n_t} \boldsymbol{r}_i^w \log(\boldsymbol{r}_i^s). \tag{5}$$

To further strengthen the structural constraints, we also apply a consistency constraint to align the classification predictions of weakly and strongly augmented samples, thereby enhancing consistency among different augmented versions. This consistency constraint also employs a cross-entropy loss function, defined as:

$$\mathcal{L}_{cr} = -\frac{1}{n_s + n_t} \sum_{i=1}^{n_s+n_t} \boldsymbol{p}_i^w \log(\boldsymbol{p}_i^s), \tag{6}$$

where $\boldsymbol{p}_i^w$ and $\boldsymbol{p}_i^s$ represent the classification predictions of the weakly and strongly augmented versions of the input sample, respectively. The two loss functions reveal the consistency constraints of low-dimensional manifolds, bridging domain gaps and enhancing the accuracy of pseudo-labels generated for unlabeled target samples.

3.3. Structural Constraint under Dynamic Growth

In the early stages of training, the model's performance is suboptimal, leading to significant differences in classification predictions between the two augmentations. Directly applying structural consistency constraint at this stage would introduce considerable noise. Therefore, we adopt a dynamic growth strategy, gradually increasing the weight in proportion to the current training epoch. Specifically, the linear weight is defined as:

$$\lambda_l = \frac{e}{E}, \tag{7}$$

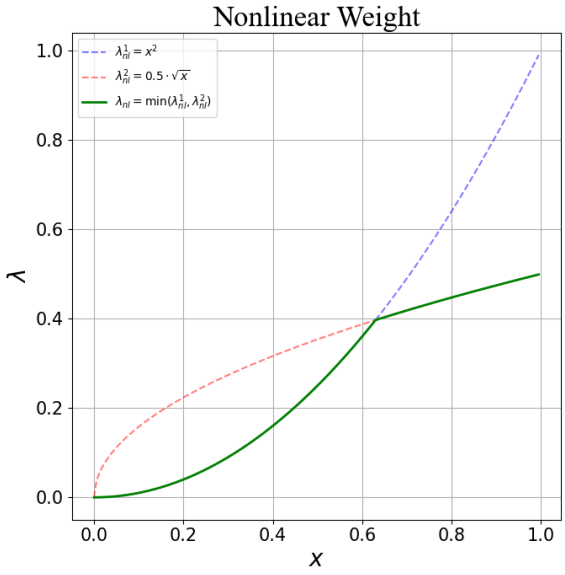


Figure 2. Dynamic growth weight for structural consistency constraint is represented in green.

where e is the current epoch, and E is the total number of epochs. As training progresses, the weight of structural consistency constraint gradually increases, ensuring that the model is not overly constrained in the early stages, allowing for more stable convergence. However, we found that despite the lower initial weight, the noise introduced at the beginning remains high, leading to unstable training. Therefore, we apply a power function to further reduce the weight of consistency constraint during the early stages of training, allowing better control over its intensity and improving training stability. Specifically, the non-linear weight is defined as:

$$\lambda_{nl}^1 = (\lambda_l)^2, \tag{8}$$

the weight can be significantly reduced at the beginning, approaching nearly zero, and gradually increasing as training progresses, leading to more stable training. However, the rapid increase of λ_{nl}^1 in the later stages of training causes the weight to grow too quickly. To address this, we modify the power function to ensure a more gradual increase in the weight towards the end of training, avoiding negative impacts on the model. We introduce a new non-linear weight,

$$\lambda_{nl}^2 = \sqrt{\lambda_l}. \tag{9}$$

A notable aspect of λ_{nl}^2 is that it grows rapidly during the early stages of training but slows down in the later stages. As shown in Figure 2, we then combine λ_{nl}^1 and λ_{nl}^2 to define the final non-linear weight as:

$$\lambda_{nl} = \min(\lambda_{nl}^1, m\lambda_{nl}^2), \tag{10}$$

where m controls when the weight growth slows down, allowing for more precise adjustment of the regularization strength compared to the sigmoid function.

3.4. Dynamic Pseudo-Label Filtering for Self-Training

To improve the accuracy of pseudo-label generation, the model also utilizes labeled samples from the source domain for supervised learning:

$$\mathcal{L}_{\text{cls}}^s = -\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{c=1}^{C}\mathbb{1}[c = y_i^s](\log p_{i,c}^s), \tag{11}$$

where $p_{i,c}^s$ represents the predicted probability of the c-th class, and $\mathbb{1}[\cdot]$ is an indicator function.

Adaptive Thresholds. Self-training is a commonly used strategy that can further reduce the domain gap by assigning pseudo-labels to reliable samples selected from the unlabeled target domain. In traditional self-training methods, pseudo-label assignment typically relies on a uniform high confidence threshold, where only samples with confidence scores above this threshold receive pseudo-labels. While this approach helps to select high-quality pseudo-labels, applying the same threshold across all classes can result in fewer or no pseudo-labels for classes with lower overall confidence, thereby hindering their learning. Therefore, we propose an adaptive thresholding method that adjusts the threshold based on the specific confidence distributions of each category. Specifically, the average confidence of all samples within a class is used as its threshold, allowing classes with higher overall confidence to have higher thresholds, while those with lower confidence are assigned lower thresholds. The threshold for class c is:

$$\theta_c = \frac{1}{|n_c|}\sum_{i=1}^{|n_c|} p_{i,c}^t \quad \text{s.t.} \quad \arg\max(\boldsymbol{p}_i^t) = c, \tag{12}$$

where $p_{i,c}^t$ represents the confidence of the i-th sample in class c, and n_c represents the number of samples in that class.

However, relying solely on the average confidence of each class to set thresholds can introduce new challenges: a threshold that is too low may increase noisy labels, while one that is too high can reduce available samples, both of which hinder effective model training. To address this, we improve the threshold setting by narrowing the gap between thresholds across different classes. Specifically, we calculate the average of all class thresholds and adjust each class's threshold to move closer to this average.

$$\overline{\theta} = \frac{1}{C}\sum_{i=1}^{C}\theta_i, \tag{13}$$

$$\widehat{\theta}_c = (1-\gamma)\cdot\theta_c + \gamma\cdot\overline{\theta}, \tag{14}$$

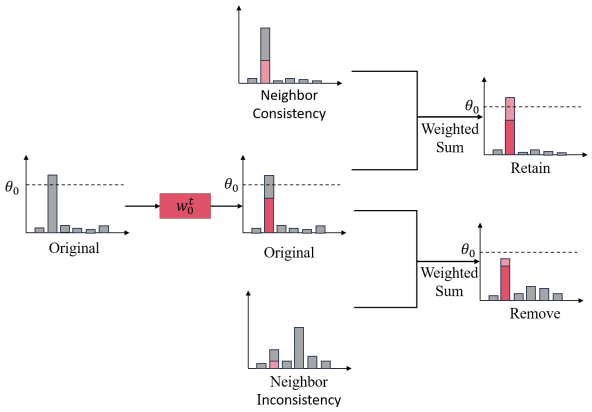


Figure 3. Illustration of Neighborhood Consistency Validation.

where $\overline{\theta}$ represents the average threshold across all classes, and $\gamma \in [0, 1]$ is a hyper-parameter used to control the proximity of each class's threshold to the average. A larger γ pulls the adjusted threshold $\widehat{\theta}_c$ for class c closer to the mean threshold.

Neighborhood Consistency Validation. In addition to the initial filtering based on confidence thresholds, we introduce a further filtering mechanism that utilizes neighboring samples. This approach improves the reliability of pseudo-labels by leveraging the local consistency of samples in the feature space. As shown in Figure 3, if a sample's pseudo-label matches those of its neighboring samples, it is more likely to be correctly labeled. Specifically, we first apply the K-nearest neighbors (KNN) algorithm to find the k nearest neighbors $\{P_n^t | n = 1, ..., k\}$ of the central sample P_0^t in the feature space, with their corresponding similarity scores denoted as $\{s_n^t | n = 1, ..., k\}$. Note that $s_0^t = 1$ represents the similarity between the central sample and itself. The original classification prediction p_0^t for P_0^t is then adjusted using the predictions of its neighboring samples P_n^t. This adjustment is performed through a weighted sum, where the weights are determined by the similarity scores s_n^t:

$$w_n^t = \frac{g(s_n^t)}{\sum_{j=0}^{k} g(s_j^t)}, \tag{15}$$

$$g(s_i^t) = \begin{cases} \rho s_i^t & \text{if } i = 0 \\ s_i^t & \text{if } i \neq 0 \end{cases}. \tag{16}$$

where $\rho > 1$. This ensures that the central sample has a larger weight in the weighted sum, giving it more influence over the final prediction while still considering the neighboring samples' contributions. The more similar a neighboring sample is to P_0^t, the greater influence its prediction has on the adjustment of the central sample's predicted distribution. The adjusted classification prediction for P_0^t is defined as:

$$\boldsymbol{p}_{adj,0}^t = \sum_{n=0}^{k} w_n^t \cdot \boldsymbol{p}_n^t, \tag{17}$$

where $\{\boldsymbol{p}_n^t | n = 1, ..., k\}$ denotes the classification prediction of the neighboring samples. Note that the adjusted classification prediction is only used for filtering pseudo-labels and is not involved in subsequent training. The pseudo-label acquisition strategy can be defined as:

$$\widehat{y}_{i,c}^t = \begin{cases} 1 & \text{if } c = \arg\max_c p_{adj,i,c}^t, p_{adj,i,c}^t > \widehat{\theta}_c \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

After careful filtering, we have ensured that the pseudo-labels used for self-training are sufficiently accurate. To further improve the process, we refine the pseudo-labels to increase the model's confidence in the predicted class while reducing it for other classes. This improves the model's ability to distinguish the target class, reduces uncertainty, and enhances overall performance. The refined classification prediction can be expressed as:

$$\boldsymbol{p}_{ref,i}^t = (1 - \mu) \cdot \boldsymbol{p}_i^t + \mu \cdot \widehat{\boldsymbol{y}}_i^t, \tag{19}$$

where $\widehat{\boldsymbol{y}}_i^t$ is the one-hot vector of the predicted pseudo label for the i-th target sample P_i^t, and $\boldsymbol{p}_i^t$ is its original classification prediction. Finally, the loss function for self-training is defined as follows:

$$\mathcal{L}_{\text{cls}}^t = -\frac{1}{\widehat{n}_t} \sum_{i=1}^{\widehat{n}_t} \sum_{c=1}^{C} \widehat{y}_{i,c}^t \log p_{ref,i,c}^t, \tag{20}$$

where $\widehat{n}_t$ represents the number of pseudo-labeled samples in the target domain.

3.5. Overall Loss

The overall training loss of our method is:

$$\mathcal{L} = \mathcal{L}_{rl} + \lambda_{nl}\mathcal{L}_{cr} + \eta\mathcal{L}_{cls}^s + \mathcal{L}_{cls}^t, \tag{21}$$

where λ_{nl} is the dynamic growing weight proposed in this paper, which controls the influence of the structural consistency constraint during training. Meanwhile, η is hyper-parameter used to balance the weight between methods. Following previous work, we utilize a two-stage optimization process to train the model. In the first stage, the model relies on the first three loss terms to facilitate domain adaptation and generate more accurate pseudo-labels for the unlabeled target samples. In the subsequent stage, more reliable pseudo-labels are selected for self-training.

4. Experiments

4.1. Datasets

PointDA-10 [20] is a widely used dataset for point cloud domain adaptation, consisting of three subsets: Model-

Method	SSL	PS	$M \rightarrow S$	$M \rightarrow S^*$	$S \rightarrow M$	$S \rightarrow S^*$	$S^* \rightarrow M$	$S^* \rightarrow S$	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 [8]			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 [20]			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 [22]	✓		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
GAST [31]	✓		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
GAI [23]	✓		85.8 ± 0.3	55.3 ± 0.3	77.2 ± 0.4	55.4 ± 0.5	73.8 ± 0.6	72.4 ± 1.0	70.0 ± 0.5
MLSP [13]	✓		83.7 ± 0.4	55.4 ± 1.8	77.1 ± 0.9	55.6 ± 0.7	78.2 ± 1.5	76.1 ± 0.5	71.0 ± 0.8
DAPS [28]	✓		84.6 ± 0.9	59.2 ± 0.4	77.1 ± 0.6	56.0 ± 0.8	73.1 ± 0.8	76.2 ± 0.9	70.8 ± 0.7
COT [11]	✓		83.2 ± 0.3	54.6 ± 0.1	78.5 ± 0.4	53.3 ± 1.1	79.4 ± 0.4	77.4 ± 0.5	71.0 ± 0.5
Ours	✓		84.6 ± 0.3	63.6 ± 0.2	82.0 ± 0.8	57.3 ± 0.7	78.3 ± 0.3	75.7 ± 0.2	73.6 ± 0.4
GAST [31]	✓	✓	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
GLRV [7]	✓	✓	85.4 ± 0.4	60.4 ± 0.4	78.8 ± 0.6	57.7 ± 0.4	77.8 ± 1.1	76.2 ± 0.6	72.7 ± 0.6
GAI [23]	✓	✓	86.2 ± 0.2	58.6 ± 0.1	81.4 ± 0.4	56.9 ± 0.2	81.5 ± 0.5	74.4 ± 0.6	73.2 ± 0.3
MLSP [13]	✓	✓	85.7 ± 0.6	59.4 ± 1.3	82.3 ± 0.9	57.3 ± 0.7	82.2 ± 0.5	76.4 ± 0.5	73.8 ± 1.0
DAPS [28]	✓	✓	86.9 ± 0.5	59.7 ± 0.5	78.7 ± 1.2	55.5 ± 1.1	82.0 ± 2.0	80.5 ± 0.7	73.9 ± 1.0
DAS [28]	✓	✓	87.2 ± 0.9	60.5 ± 0.2	82.4 ± 0.7	58.1 ± 0.8	84.8 ± 2.3	82.3 ± 1.5	75.9 ± 1.1
COT [11]	✓	✓	84.7 ± 0.2	57.6 ± 0.2	89.6 ± 1.4	51.6 ± 0.8	**85.5** ± 2.2	77.6 ± 0.5	74.4 ± 0.9
Ours (SPST)	✓	✓	87.6 ± 0.3	64.1 ± 0.3	86.9 ± 0.6	58.5 ± 0.3	82.9 ± 0.5	81.7 ± 0.7	77.0 ± 0.5
Ours	✓	✓	**87.8** ± 0.1	**64.5** ± 0.2	**90.2** ± 1.0	**59.0** ± 0.5	85.2 ± 0.2	**86.4** ± 1.2	**78.9** ± 0.5

Table 1. Comparative evaluation in classification accuracy (%) averaged over 3 seeds (± SEM) on the PointDA-10 dataset. The best results in each column are in bold.

	$M \rightarrow S$	$M \rightarrow S^*$	$S \rightarrow M$	$S \rightarrow S^*$	$S^* \rightarrow M$	$S^* \rightarrow S$	Avg.
ObsS	83.7	59.4	81.2	55.3	74.7	74.1	71.3
+FeaS	84.3	59.6	81.7	56.9	75.7	75.0	72.2
+StrC	**84.6**	**63.6**	**82.0**	**57.3**	**78.3**	**75.7**	**73.6**

Table 2. Ablation Study on Different Components of the Self-Supervised Module: ObsS denotes augmentation in the observation space only, +FeaS adds feature space augmentation, and +StrC further incorporates structural consistency constraint under dynamic growth. The best results in each column are in bold.

$M \rightarrow S^*$	Non	λ_l	λ_{nl}^1	λ_{nl}
Accuracy	59.6	55.3	61.1	63.6

Table 3. The Impact of Different Approaches for Introducing Structural Consistency Constraint.

Net40 [26], ShapeNet [2], and ScanNet [5]. For the experiments, 10 common categories such as sofa, lamp, and chair are selected from these datasets, forming ModelNet-10 (*M*), ShapeNet-10 (*S*), and ScanNet-10 (*S**). Both *M* and *S* are synthetic datasets generated from CAD models. *M* contains $4,183$ training samples and 856 test samples, while *S* includes $17,378$ training samples and $2,492$ test samples. In contrast, *S** consists of real-world point clouds collected from indoor scenes, with $6,110$ training samples and $2,048$ test samples, which are incomplete due to occlusions from nearby objects.

4.2. Implementation

In this work, we employ DGCNN as the feature extractor. During training, we apply the Adam optimizer with an initial learning rate of 0.001 and a weight decay of 0.00005. Additionally, we use a cosine annealing scheduler to adjust the learning rate over epochs. The hyper-parameters m, γ, μ, and η are empirically set to 0.5, 0.4, 0.2, and 0.5, respectively. The number of neighboring samples k is set to 1, and ρ is assigned a value of 4. All methods are trained for 200 epochs with a batch size of 32, using three different random seeds, on an NVIDIA RTX 4090 GPU.

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

We compare our method with several state-of-the-art point cloud domain adaptation methods on the PointDA-10 dataset, including Domain Adversarial Neural Network (DANN) [8], Point Domain Adaptation Network (PointDAN) [20], Reconstruction Space Network (RS) [22], Deformation Reconstruction Network with Point Cloud

	$M \to S$	$M \to S^*$	$S \to M$	$S \to S^*$	$S^* \to M$	$S^* \to S$	Avg.
SSL	84.6	63.6	82.0	57.3	78.3	75.7	73.6
w/o Source	84.0	58.9	81.1	52.9	74.4	72.0	70.6

Table 4. Ablation Study Across Different Domains, where w/o Source means only target domain data is used for self-supervised learning.

	DymT	NeiC	Refine	$M \to S$	$M \to S^*$	$S \to M$	$S \to S^*$	$S^* \to M$	$S^* \to S$	Avg.
SPST				87.6	64.1	86.9	58.5	82.9	81.7	77.0
Ours	✓			87.7	64.3	87.4	58.8	83.2	83.8	77.5
	✓	✓		**87.8**	64.3	88.6	58.7	83.8	85.9	78.2
	✓	✓	✓	**87.8**	**64.5**	**90.2**	**59.0**	**85.2**	**86.4**	**78.9**

Table 5. Ablation Study on Different Components of Self-training.

Mixup (DefRec+PCM) [1], Geometry-aware self-training (GAST) [31], Global-Local Structure Modeling with Reliable Voted Pseudo Labels (GLRV) [7], Geometry-Aware Implicits (ImplicitPCDA) [23], Masked Local Structure Prediction (MLSP) [13], Domain Adaptive Point Sampling (DAPS) [28] and Contrastive Learning and Optimal Transport (COT) [11]. Note that DAPS uses the traditional pseudo-labeling (PS) method, while DAS [28] employs a more complex network architecture. The supervised method trains the model using only labeled target samples, while the w/o adapt method is trained exclusively on labeled source samples.

As shown in Table 1, our proposed method outperforms all baselines, surpassing the current SOTA method DAS by 3%. Specifically, in the sim-to-real settings ($M \to S^*$ and $S \to S^*$), we achieve improvements of 4% and 0.9% respectively, demonstrating its superior ability to reduce the domain gap between significantly different distributions. Our method achieves the best results in 5 out of 6 settings. Although it falls behind COT by 0.3% in the $S^* \to M$ setting, the standard deviation is 2% lower, indicating greater stability. Even when applying the standard self-training strategy (SPST), our method achieves significant gains, outperforming COT by 2.6%. Overall, these results demonstrate the effectiveness and robustness of our method across various domain adaptation tasks.

4.4. Ablation Study

The Impact of Self-supervised Task Components. To investigate the impact of feature space augmentation and structural consistency constraint under dynamic growth, we conduct an ablation study on the PointDA-10 dataset. As shown in Table 2, both components have a positive impact, with consistency constraint contributing notably to improvements in the $M \to S^*$ and $S^* \to M$ settings, where performance increases by 4% and 2.6%, respectively.

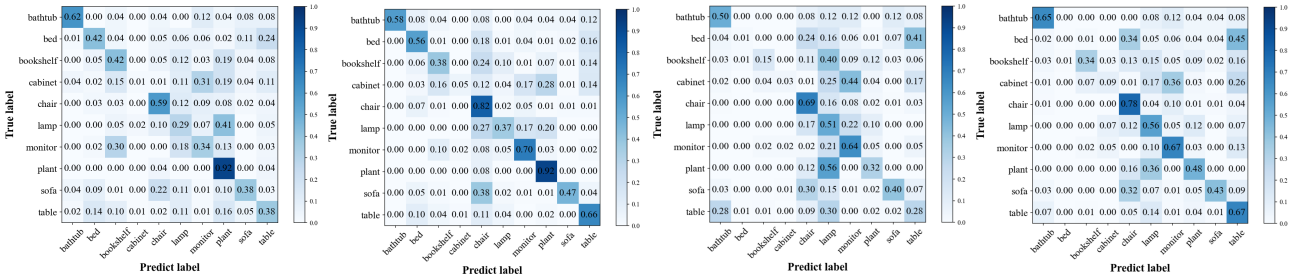
The Impact of Different Approaches for Introducing structural consistency constraint. To investigate the impact of different approaches for introducing structural consistency constraint, we apply it using three methods: linear, concave, and the proposed dynamic growth approach. Table 3 presents the results, where "Non" indicates the absence of consistency constraint. The linear approach λ_l results in a sharp 4.3% performance drop due to the excessive introduction of noise early in training. The concave approach λ_{nl}^1 improves performance by 1.5% as it reduces early noise, but the rapid weight increase in the later stages limits further gains. In contrast, our proposed dynamic growth approach mitigates the weight surge in the later training phase, allowing the model to learn in a more gradual manner, ultimately leading to a 4% improvement.

	k	ρ	R/A (acc)	Accuracy
$S \to S^*$	1		7730/8476 (91.20%)	**86.4**
	2	4	6406/6806 (94.12%)	85.7
	3		5293/5505 (96.15%)	84.9
		3	7443/8029 (92.70%)	86.1
	1	4	7730/8476 (91.20%)	**86.4**
		5	7988/8829 (90.47%)	86.3

Table 6. Ablation Study of Neighborhood Consistency in Experimental Setting $S \to S^*$, where R/A represents the accuracy of the predicted pseudo-labels.

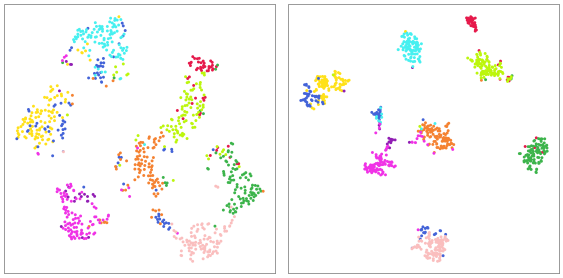
The Impact of Domain Usage. To assess whether the knowledge from the source domain is effectively transferred to the target domain, we perform self-supervised learning using data from both domains as well as using only target domain data. As shown in Table 4, the results indicate that the proposed self-supervised method effectively reduces the domain gap, particularly in the two sim-to-real settings ($M \to S^*$ and $S \to S^*$), where it achieves improvements of 4.7% and 4.4%, respectively.

The Impact of Self-training Task Components. To explore the impact of the proposed pseudo-label filtering mechanism on self-training, we conduct an ablation study on the PointDA-10 dataset. As shown in Table 5, the results indicate that our dynamic thresholding method outperforms



(a) w/o Adapt: $M \rightarrow S^*$ (b) Our: $M \rightarrow S^*$ (c) w/o Adapt: $S \rightarrow S^*$ (d) Our: $S \rightarrow S^*$

Figure 4. Confusion matrices for the classification of test samples on the target domain. Darker colors within the visualization reflect higher levels of accuracy.



(a) w/o Adapt: $S^* \rightarrow M$ (b) Our: $S^* \rightarrow M$

Figure 5. The t-SNE visualization of feature distribution on the target domain. Different colors represent different classes.

SPST, which applies the same threshold for all categories, with a 0.5% improvement. Further improvement of 0.7% is achieved by incorporating neighbor consistency validation to filter more reliable pseudo-labels. Finally, refining the pseudo-label predictions leads to an additional 0.7% gain.

The Impact of k and ρ on Neighbor Consistency Verification. The proposed pseudo-label filtering mechanism based on neighborhood consistency involves two key hyper-parameters, k and ρ. To investigate the influence of these two hyper-parameters, we conducted an ablation study under experimental setting $S \rightarrow S^*$ with different values of k and ρ. As shown in Table 6, to investigate the effect of k, we fix ρ at 4. The results show that as the number of neighbors increases, the accuracy of predicted pseudo-labels improves, but fewer samples are assigned pseudo-labels. To explore the influence of ρ, we fix k at 1. As the weight of the central sample decreases, meaning the relative weight of neighboring samples increases, the accuracy of pseudo-label predictions also improves. This suggests that relying on neighboring samples can effectively enhance the reliability of pseudo-labels. Considering both the number of assigned pseudo-labels and prediction accuracy, the best results are achieved when $k = 1$ and $\rho = 4$.

Class-Wise Accuracy Visualization. We use confusion matrices to visualize the model's accuracy across categories, with rows as actual categories and columns as predictions. This method shows overall accuracy and highlights categories prone to misclassification. As shown in Figure 4, the confusion matrix displays the class-wise classification accuracy for both the baseline (w/o Adapt) and our method on $M \rightarrow S^*$ and $S \rightarrow S^*$. Figure 4a and 4c show the results without adaptation, while Figure 4b and 4d show the results using our proposed adaptation method, where the darker diagonal lines in the confusion matrices indicate better overall accuracy.

Feature Visualization. We employ t-SNE [25] to visualize the feature distribution on the target domains for the UDA task $S^* \rightarrow M$. Figure 5a shows the distribution without adaptation, while Figure 5b shows the result using the methods proposed in this paper. Without domain adaptation, features from different classes in the target domain tend to overlap. However, with domain adaptation, the feature distribution in the target domain begins to converge, forming distinct clusters and significantly reducing the overlap between features from different classes.

5. Conclusions

In this paper, we propose a novel approach to bridge the domain gap in unsupervised domain adaptation for point cloud classification. We employs a dual-augmentation relational learning scheme to incorporate low-dimensional manifolds to encourage domain-invariant representations, thereby improving the accuracy of pseudo-label generation. Additionally, we design a filtering mechanism that adaptively adjusts thresholds for each semantic category based on confidence distributions and validates neighborhood consistency to further mitigate feature ambiguities, resulting in more reliable pseudo-label selection for self-training. Experiments on the widely-recognized PointDA-10 dataset show that our approach achieves state-of-the-art performance, demonstrating its effectiveness in addressing the domain gap.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62002172; and in part by The Startup Foundation for Introducing Talent of NUIST under Grant 2023r131.

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. 1, 2, 7, 8

[2] 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 preprint arXiv:1512.03012*, 2015. 1, 7

[3] Y. Chen, Z. Wang, L. Zou, K. Chen, and K. Jia. Quasi-balanced self-training on noise-aware synthesis of object point clouds for closing domain gap. In *European Conference on Computer Vision*, pages 728–745. Springer, 2022. 3

[4] Y. Chen, C. Wei, A. Kumar, and T. Ma. Self-training avoids using spurious features under domain shift. *Advances in Neural Information Processing Systems*, 33:21061–21071, 2020. 2

[5] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 7

[6] Z. Deng, Y. Luo, and J. Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9944–9953, 2019. 2

[7] H. Fan, X. Chang, W. Zhang, Y. Cheng, Y. Sun, and M. Kankanhalli. Self-supervised global-local structure modeling for point cloud domain adaptation with reliable voted pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6377–6386, 2022. 1, 2, 3, 7, 8

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 2, 7

[9] X. Gu, J. Sun, and Z. Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9101–9110, 2020. 2

[10] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4893–4902, 2019. 2

[11] S. Katageri, A. De, C. Devaguptapu, V. Prasad, C. Sharma, and M. Kaul. Synergizing contrastive learning and optimal transport for 3d point cloud domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2942–2951, 2024. 2, 7, 8

[12] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013. 2

[13] H. Liang, H. Fan, Z. Fan, Y. Wang, T. Chen, Y. Cheng, and Z. Wang. Point cloud domain adaptation via masked local 3d structure prediction. In *European Conference on Computer Vision*, pages 156–172. Springer, 2022. 1, 2, 3, 7, 8

[14] H. Liang, Q. Zhang, P. Dai, and J. Lu. Boosting the generalization capability in cross-domain few-shot learning via noise-enhanced supervised autoencoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9424–9434, 2021. 2

[15] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 1

[16] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2239–2247, 2019. 2

[17] P. O. Pinheiro. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8004–8013, 2018. 2

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[20] 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. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 6, 7

[21] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 2

[22] J. Sauder and B. Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32, 2019. 7

[23] Y. Shen, Y. Yang, M. Yan, H. Wang, Y. Zheng, and L. J. Guibas. Domain adaptation on point clouds via geometry-aware implicits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7223–7232, 2022. 1, 2, 7, 8

[24] I. Shin, S. Woo, F. Pan, and I. S. Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 532–548. Springer, 2020. 2

[25] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 9

[26] K. V. Vishwanath, D. Gupta, A. Vahdat, and K. Yocum. Modelnet: Towards a datacenter emulation environment. In

2009 IEEE Ninth International Conference on Peer-to-Peer Computing, pages 81–82. IEEE, 2009. 1, 7

[27] 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. 2

[28] Z. Wang, W. Li, and D. Xu. Domain adaptive sampling for cross-domain point cloud recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12):7604–7615, 2023. 2, 7, 8

[29] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2

[30] M. Zheng, S. You, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu. Ressl: Relational self-supervised learning with weak augmentation. *Advances in Neural Information Processing Systems*, 34:2543–2555, 2021. 1, 4

[31] L. Zou, H. Tang, K. Chen, and K. Jia. Geometry-aware self-training for unsupervised domain adaptation on object point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6403–6412, 2021. 1, 2, 3, 7, 8