2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

# Masked Local-Global Representation Learning for 3D Point Cloud Domain Adaptation

Bowei Xing, Xianghua Ying*, and Ruibin Wang

***Abstract*— Point cloud is a popular and widely used geometric representation, which has attracted significant attention in 3D vision. However, the geometric variability of point cloud representations across different datasets can cause domain discrepancies, which hinder knowledge transfer and model generalization, resulting in degraded performance in target domain. In this paper, we present a novel approach to improve point cloud domain adaptation by employing masked representation learning in a self-supervised manner. Specifically, our method combines masked feature prediction and masked sample consistency to encode both local structure and global semantic information for learning invariant point cloud representation across domains. Moreover, to learn domain-specific representation and transfer knowledge from source to target, we propose prototype-calibrated self-training. By exploiting class-wise prototypes in the shared feature space, the soft pseudo labels can be adaptively denoised, which benefits the decision boundary learning in target domain. We conduct experiments on PointDA-10 and PointSegDA for 3D point cloud shape classification and semantic segmentation, respectively. The results demonstrate the effectiveness of our method and show that we can achieve the new state-of-the-art performance on point cloud domain adaptation.**

## I. Introduction

Point cloud is an important data format for representing 3D objects and 3D scenes. With the development of deep learning, various point cloud models [1]–[4] have been proposed, bringing great achievements for 3D computer vision. Despite the success of point cloud models, they always have to rely on a large scale of labeled data, where the annotation is expensive and time-consuming. Recent advances in unsupervised domain adaptation (UDA) help to alleviate the labeling efforts required for training fully-supervised models, which is especially helpful for point cloud analysis. UDA aims to transfer the knowledge from the model trained on labeled source domain, *e.g.*, synthetic point clouds, to unlabeled target domains, *e.g.*, real-world objects, thus saving plenty of annotation expenses, which can be beneficial to be deployed on robotic systems for transferable 3D scene and object understanding.

Recently, an increasing number of methods have been proposed to address UDA on 3D point cloud. For example, PointDAN [5] explicitly achieves feature alignment across domains with a feature extractor and a domain discriminator. Several works [6]–[10] design self-supervised learning tasks to learn the internal structure of unlabeled target data, enabling the model to embed source and target data in a shared feature space. Besides, self-training is also proved helpful for point cloud domain adaptation [8], [11], [12]. However, simply generating pseudo labels based on confidence threshold will introduce much noise in self-training, thus affecting the performance of the model on target domain.

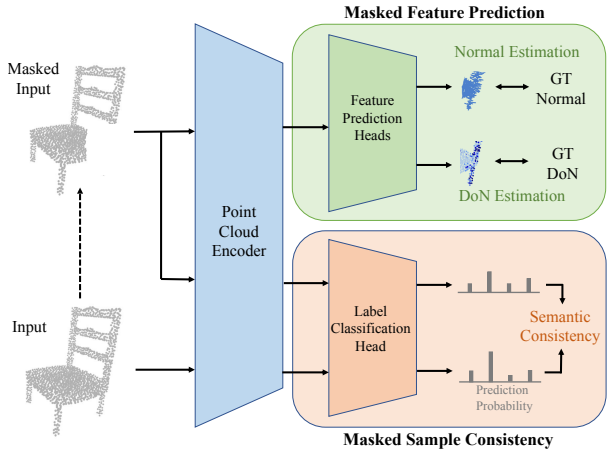


Fig. 1. Illustration of the proposed masked local-global representation learning. With the input point cloud, we randomly mask it to obtain a masked sample, then define two self-supervised tasks: 1) Masked feature prediction, which predicts point normal and difference of normals (DoN) in mask region to model local structure; 2) Masked sample consistency, which enforces semantic consistency of the inputs to encode global information.

In this paper, in order to improve point cloud domain adaptation, we exploit the effective self-supervised learning method and innovatively propose masked local-global representation learning to encode point cloud characters. Moreover, we introduce class-wise prototypes into the self-training process, which calibrates the distribution and alleviates the noise of pseudo labels, bringing better adaptation effect.

To align source and target domains and learn domain-invariant features, we leverage self-supervised learning for 3D representation learning. Specifically, we randomly mask a region of the input point cloud and introduce 1) masked feature prediction to encode local structure information, and 2) masked sample consistency for global semantic information learning, as shown in Figure 1. With the masked input, the model is encouraged to predict the normal and Difference of Normals (DoN) of the points for missing part. Since these two feature descriptors represent the structure of local region and are invariant across domains, they are effective for modelling local information. Moreover, we

This work was supported by the National Natural Science Foundation of China (NSFC) (No. 62371009 and No. 61971008).

B. Xing, X. Ying and R. Wang are with National Key Laboratory of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University, Beijing 100871, China.

*Corresponding author: Xianghua Ying (e-mail: xhying@pku.edu.cn).

additionally enforce the semantic consistency between the masked sample and the original sample, so as to encode global semantic information. Since the proposed masked local-global representation learning does not need category supervision labels, it can be conducted on both source and target domains. Therefore, the geometric structure and semantic information captured by self-supervised tasks are shared between domains, which can boost the discrimination of feature representations to classify target point clouds.

To learn domain-specific features and transfer knowledge from source to target, we conduct prototype-calibrated self-training to exploit the specific information on target domain. Concretely, we leverage the overall domain representations to calculate class-wise prototypes, with which we can adaptively refine the soft pseudo labels. In this way, the pseudo-labels are obtained by considering the distance to global prototypes rather than relying on single samples. It facilitates dataset-level reasoning and allows to correct the misclassified samples, leading to the refinement of pseudo-labels. Moreover, the shared feature space is well clustered thanks to the self-supervised learning, thus further benefiting the prototype learning and pseudo label refining.

We conduct experiments on PointDA-10 [5] for classification and PointSegDA [7] for segmentation. The results demonstrate the effectiveness of our method, leading to improved performance compared with previous methods. Our main contributions are threefold:

1. We propose masked local-global representation learning to learn invariant features across domains, which simultaneously encodes local structure and global semantic information for point cloud domain adaptation.

2. We propose prototype-calibrated self-training to learn domain specific knowledge, which effectively refines the pseudo labels and transfers knowledge from source to target.

3. Extensive experiments on two datasets validate the effectiveness of the proposed method and significantly outperform previous methods.

## II. Related Work

### A. Point Cloud Domain Adaptation

Different types of UDA methods have been proposed in 2D vision, including adversarial training [13]–[15], discrepancy alignment [16]–[19], self-training [20], [21] and so on. In recent years, there has been a growing interest in developing domain adaptation methods for point clouds [5], [7], [8], [11], [12], [22]. Specifically, PointDAN [5] is the pioneering work that addresses point cloud classification in UDA setting, which adopts Maximum Classifier Discrepancy to align the features across different domains. Achituve et.al [7] combine deformation reconstruction and point cloud mixup to improve adaptation effect. [8] and [10] utilize self-supervised learning to obtain representative features in both domains, implemented with different additional tasks like rotation angle prediction or 2D-3D projection reconstruction. In this paper, we employ self-supervised learning to learn domain-invariant features and develop self-training to learn domain-specific features.

### B. Self-Supervised Learning on Point Cloud

Self-supervised learning aims to learn from internal structures by exploiting the input sample itself without label supervision, thus benefiting the downstream tasks. Several methods have applied self-supervised learning on point clouds [6], [8], [10], [23]–[28]. Specifically, Sauder and Sievers [6] develop an architecture-agnostic self-supervised learning method by reconstructing a randomly displaced point cloud. PointContrast [25] utilizes contrastive learning to pull distance between different views of the same example. [9] leverages implicit function learning as a self-supervised task, so as to exploit the geometry information of point cloud. Cardace et.al [12] utilize peculiar 3D data augmentations and self-distillation to bridge the gap between source and target. Inspired by MAE [29] and MaskFeature [30] for image-based self-supervised learning, we intend to explore masked representation learning method to learn local geometry and global semantic information for 3D point cloud.

## III. Method

### A. Problem Formulation

For point cloud domain adaptation, a labelled source dataset $\mathcal{S} = \{\mathcal{P}_{i,s}, y_{i,s}\}_{i=1}^{n_s}$ and an unlabeled target dataset $\mathcal{T} = \{\mathcal{P}_{i,t}\}_{i=1}^{n_t}$ are given, where $\mathcal{P} \in \mathbb{R}^{m\times 3}$ is the point cloud consisting of $m$ three-dimensional coordinate points $p = (x, y, z)$, $n_s$ and $n_t$ are the respective sample numbers in each dataset. The label set $\mathcal{Y} = \{1, 2, \cdots, K\}$ is shared across the domains, where $K$ is the number of object categories. The goal is to learn an adaptive model with the labelled $\mathcal{S}$ and unlabelled $\mathcal{T}$, enabling to achieve good performance on target domain without label supervision. Generally, the point cloud model $\Phi = f \circ g$ can be regarded as the combination of a feature extractor $f$ and a classifier $g$. In this paper, we propose masked local-global representation learning and prototype-calibrated self-training for point cloud domain adaptation, as illustrated in Fig.2.

### B. Masked Local-Global Representation Learning

To minimize domain discrepancy across source and target domains, we intend to exploit self-supervised learning to learn a shared feature extractor $f$. Specifically, we propose masked local-global representation learning to obtain domain-invariant features, consisting of masked feature prediction and masked sample consistency. With the predefined self-supervised tasks, the model is encouraged to capture local structure and global semantic information of the input, thus benefiting the following label classification. Note that, the proposed representation learning strategy is implemented on both domains, where the supervision signals can be generated automatically from input without category label.

Concretely, for a given point cloud $\mathcal{P}$, We denote the masked part as $\mathcal{P}_m$, and the visible part after mask as $\mathcal{P}_v$. As for the mask strategy, we voxelize the input point cloud $\mathcal{P}$ into the $3\times3\times3$ voxels, from which we randomly select one at equal probability and remove all the points in this region, leading to the masked point cloud. With the masked point cloud, we define the following two tasks.

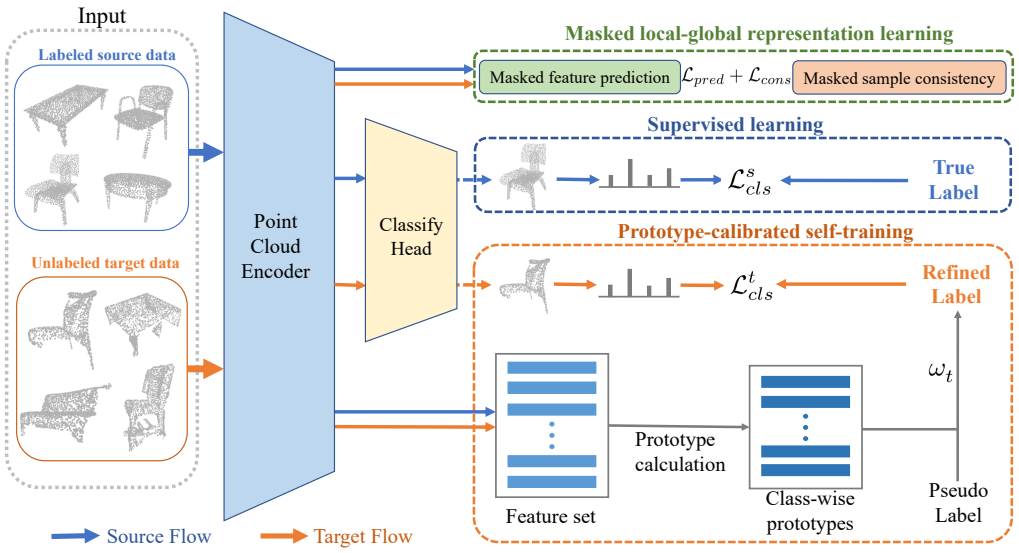


Fig. 2. Overview of the proposed framework, which consists of three main components, *i.e.*, masked local-global representation learning on both domains, supervised learning on source domain and prototype-calibrated representation learning on target domain.

**Masked Feature Prediction.** We intend to enforce the model to predict certain feature descriptors, $i.e.$, normal and Difference of Normals (DoN) for the masked points, so as to capture local geometry structure of the point cloud. The normal of a given point is generated from several neighbors around it, which provides local structure information for point cloud understanding [31]. However, the normal can only reflect the orientation of the point cloud surface, but lacks the gradient and variance information of the surface. To this end, we further considers DoN [32] to capture local gradient details, which contains multi-scale information and reflects the variation of surface gradient. Detailedly, as shown in Fig.3, DoN algorithm first computes the normal at each point by fitting a specified number of neighbors. Then, it computes the difference of normals that from different scales. The resulting DoN is effective for detecting sharp edges or fine details in the geometry structure of a point cloud.

The normal and DoN jointly represent the underlying geometry orientation and gradient of the local surface, and these basic structures are shared across different domains and objects. Therefore, we would like to facilitate the model to regress these feature descriptors for the masked content, so as to acquire an adequate understanding of the geometry structures within point cloud, and obtain domain-invariant representations across datasets.

Concretely, for a point $p$ in point cloud $\mathcal{P}$, the normal $\boldsymbol{n}(p)$ and DoN operator $\boldsymbol{\Delta}_{\mathbf{n}}(p)$ can be calculated from the input point cloud using the following formulations, which serve as the ground truth for masked feature prediction task:

$$\begin{aligned} \boldsymbol{n}(p,r) &= \underset{n\in\mathbb{R}^3}{\operatorname{argmin}} \sum_{q\in\mathcal{N}(p,r)} (n\cdot(q-p))^2 \\ \boldsymbol{\Delta}_{\mathbf{n}}(p,r_l,r_s) &= \frac{\boldsymbol{n}(p,r_s)-\boldsymbol{n}(p,r_l)}{2} \end{aligned} \tag{1}$$

where $\mathcal{N}(p,r)$ refers to the nearby region around $p$ that consists of $r$ adjacent points, $r_l$ and $r_s$ refer to the number of adjacent points, respectively leading to the large or small neighbor region.

After calculating the above ground truth feature descriptors, we introduce two feature prediction heads $h_{norm}$ and $h_{DoN}$ after the feature extractor $f$, which are respectively responsible for predicting the normal and DoN of the masked points with the visible part $\mathcal{P}_v$. For the point $p \in \mathcal{P}_m$, we utilize the prediction heads to obtain the predicted normal $\hat{n}(p) = h_{norm}(f(\mathcal{P}_v))$ and the predicted DoN $\hat{\Delta}_n(p) = h_{DoN}(f(\mathcal{P}_v))$. After that, we employ cosine distance and euclidean distance as the measurement to minimize the gap between the predicted feature discriptors and pre-calculated ground-truth, leading to the following masked feature prediction loss:

$$\mathcal{L}_{pred} = \sum_{p\in\mathcal{P}_m} (cos(\boldsymbol{n}(p),\hat{n}(p)) + \|\boldsymbol{\Delta}_{\mathbf{n}}(p) - \hat{\Delta}_n(p)\|_2) \tag{2}$$

**Masked Sample Consistency.** In addition to the masked feature prediction task to capture local geometry structure, we further propose to utilize the masked sample consistency to learn global semantic information of the point cloud. Intuitively, the masked sample still preserve the same semantic class label as the original example, which provides an effective way to enforce the representation discriminability and learn a clustered feature space. Based on this observation, we propose to enforce the semantic consistency between the original input $\mathcal{P}$ and the masked input $\mathcal{P}_v$.

Specifically, for $\mathcal{P}$ and $\mathcal{P}_v$, since the two samples can be seen as the different variations of the same object, we propose to minimize the Kullback–Leibler (KL) divergence between the prediction probabilities under the two views:

$$\mathcal{L}_{cons} = KL(\Phi(P_v)||\Phi(P)) \tag{3}$$

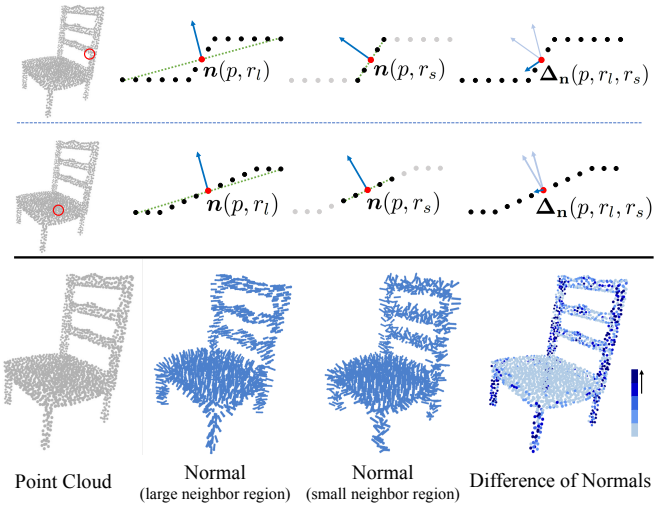


Fig. 3. **Top**: The calculation of point cloud normal $\boldsymbol{n}$ and Difference of Normals (DoN) $\boldsymbol{\Delta}_{\mathbf{n}}$. The point in sharp area has larger DoN (first row), while the point in smooth area has smaller DoN (second row). **Bottom**: The visualization of normal and DoN for the given point cloud. Specifically, we visualize DoN using its $l_2$ norm, where the darker color refers to larger value and vice versa.

Intuitively, the above formulation enforces the network to give consistent predictions for the similar point clouds, resulting in more compact representations in the shared feature space. What's more, the masked sample can also imitate the point cloud with occlusion that seen in real-world objects, helping to narrowing the domain discrepancy, and the learned representation tends to be invariant to the domain gap between synthetic data and real-world data.

### C. Prototype-Calibrated Self-Training

To better improve the domain adaptation effect, we adopt self-training, which can explicitly transfer knowledge from source to target. The generation of pseudo labels for target samples is important for self-training. Previous works [8], [9] simply use confidence-based methods to select pseudo labels based on the single example's prediction score. However, the high scores are not necessarily correct, making the network fail to learn reliable knowledge in the target domain. To address this problem, we intend to utilize prototypes to refine the pseudo label generation, thus to boost the performance of self-training. The proposed method considers the features from both domains to maintain class-wise prototypes, and adopts the feature distance to prototypes to progressively reweight the pseudo labels. With the update in accordance with the freshly learned knowledge, we then use the refined pseudo labels for self-training in target domain.

To obtain the prototypes for each class, at first, we extract the feature representation and give the predicted probability vector $p = \Phi(\mathcal{P})$ for the input source or target domain sample $\mathcal{P}$, and initialize the prototypes $c^k$ for the $k^{th}$ class:

$$c^k = \frac{\sum_{\mathcal{P}\in\mathcal{S}\cup\mathcal{T}} \mathbf{1}\left(p^k > \lambda\right) * p^k * f\left(\mathcal{P}\right)}{\sum_{\mathcal{P}\in\mathcal{S}\cup\mathcal{T}} \mathbf{1}\left(p^k > \lambda\right)} \tag{4}$$

where $\mathbf{1}(\cdot)$ is the indicator function, $p^k$ is the predicted probability that the sample belongs to class $k$, $\lambda$ is the pre-defined threshold to select reliable samples for calculating prototypes. Specifically, we select the high-confidence predictions to increase the reliability of the prototype calculation. Besides, we also multiply with the prediction probability so that the highly confident samples are weighted more.

In the training process, we estimate the prototypes as the moving average of the cluster centroids in mini-batches. Specifically, in each iteration, the prototype is updated as: $c^k \leftarrow \mu c^k + (1-\mu)c'^k$ where $c'^k$ is the newly calculated prototype of class $k$ within the current training batch, and $\mu$ is the momentum coefficient for the updating process.

With the obtained prototypes, we can adaptively refine target sample's pseudo label based on the feature distance to class-wise prototypes. Intuitively, when the learned feature $f(\mathcal{P}_t)$ is far from $c^k$, it is likely to be an outlier and we will reduce its probability of being classified to the $k^{th}$ category:

$$\begin{aligned} \omega_t^k &= \frac{\exp\left(-\left\|f\left(\mathcal{P}_t\right) - c^k\right\|/\tau\right)}{\sum_{k'}\exp\left(-\left\|f\left(\mathcal{P}_t\right) - c^{k'}\right\|/\tau\right)} \\ \hat{y}_t^k &= \omega_t^k p_t^k \end{aligned} \tag{5}$$

where $\tau$ is a temperature coefficient.

After the refinement, we obtain the newly refined soft pseudo label $\hat{y}_t$ for the target sample $\mathcal{P}_t$. It is worth noting that we use the soft label instead of the hard label to improve the performance. With the refined pseudo label, we are able to conduct self-training on target domain, in the meanwhile combined with the supervised training in source domain.

$$\begin{aligned} \mathcal{L}_{\text{cls}}^{\text{t}} &= -\frac{1}{n_{\text{t}}}\sum_{i=1}^{n_{\text{t}}}\sum_{k=1}^{K}\hat{y}_{i,t}^k \log(g(f(\mathcal{P}_{i,t}))^k) \\ \mathcal{L}_{\text{cls}}^{\text{s}} &= -\frac{1}{n_{\text{s}}}\sum_{i=1}^{n_{\text{s}}}\sum_{k=1}^{K} y_{i,s}^k \log(g\left(f\left(\mathcal{P}_{i,s}\right)\right)^k) \end{aligned} \tag{6}$$

where $\hat{y}_{i,t}$ is the refined soft pseudo label for the $i^{th}$ sample in target domain, $y_{i,s}$ is the ground-truth one-hot label in source domain, and $g(f(\mathcal{P}_{i,t}))^k$ represents the model's predicted probability for the $k^{th}$ class.

Since the prototypes are calculated by utilizing the whole domain data, we can refine pseudo labels from a global perspective of the entire datasets, which is more effective than previous strategy that predicts for each sample isolatedly. By facilitating the sample's interaction with class-wise prototypes, it has more potential to select the correct but low confidential pseudo labels, in the meanwhile alleviating the wrong predictions. Besides, thanks to the self-supervised learning, the learned representations are more clustered, which is help for the pseudo-label refinement and self-training in target domain.

### D. Overall Training Objective

The overall objective is the sum of self-supervised learning loss on $\mathcal{S}$ and $\mathcal{T}$, the supervised learning loss on $\mathcal{S}$ and the self-training loss on $\mathcal{T}$, using the labelled source data, the

TABLE I
THE COMPARISON RESULTS OF ACCURACY ON POINTDA-10 [5].

| Method | M→S | M→S* | S→M | S→S* | S*→M | S*→S | Avg |
|---|---|---|---|---|---|---|---|
| Oracle | 93.9 | 78.4 | 96.2 | 78.4 | 96.2 | 93.9 | 89.5 |
| w/o Adapt | 83.3 | 43.8 | 75.5 | 42.5 | 63.8 | 64.2 | 62.2 |
| DANN [14] | 74.8 | 42.1 | 57.5 | 50.9 | 43.7 | 71.6 | 56.8 |
| PointDAN [5] | 83.9 | 44.8 | 63.3 | 45.7 | 43.6 | 56.4 | 56.3 |
| RS [6] | 79.9 | 46.7 | 75.2 | 51.4 | 71.8 | 71.2 | 66.0 |
| DefRec [7] | 81.7 | 51.8 | 78.6 | 54.5 | 73.7 | 71.1 | 68.6 |
| RefRec [12] | 81.4 | 56.5 | 83.4 | 53.3 | 73.0 | 73.1 | 70.5 |
| GAST [8] | 84.8 | 59.8 | 80.8 | 56.7 | 81.1 | 74.9 | 73.0 |
| GLRV [10] | 85.4 | 60.4 | 78.8 | 57.7 | 77.8 | 76.2 | 72.7 |
| ImplicitDA [9] | 86.2 | 58.6 | 81.4 | 56.9 | 81.5 | 74.4 | 73.2 |
| MLSP [31] | 86.2 | 59.1 | 83.5 | 57.6 | 81.2 | 76.4 | 74.0 |
| SD [22] | 83.9 | 61.1 | 80.3 | 58.9 | **85.5** | **80.9** | 75.1 |
| Ours | **86.9** | **61.2** | **85.7** | **59.1** | 81.7 | 77.3 | **75.3** |

unlabelled target data and the generated pseudo labels:

$$\mathcal{L} = \mathcal{L}^{s,t}_{pred} + \mathcal{L}^{s,t}_{cons} + \mathcal{L}^{s}_{cls} + \mathcal{L}^{t}_{cls} \tag{7}$$

The training contains multiple rounds. At each round, we first train the model and update model parameters. With the newly obtained model, we update prototypes and generate pseudo labels on target domain, preparing for next round.

## IV. EXPERIMENT

### A. Datasets

PointDA-10 [5] is a popular benchmark for point cloud domain adaptation evaluation. It contains point cloud samples of 10 shared classes from ModelNet [33], ShapeNet [34] and ScanNet [35], where the samples in ModelNet (M) and ShapeNet (S) are extracted from synthetic 3D CAD models, while ScanNet (S*) is obtained from multiple real RGB-D scans and thus exhibits several forms of noise and occlusions.

PointSegDA [7] is based on a dataset of human models and comprises four subsets: ADOBE (A), FAUST (F), MIT (M), and SCAPE (S). These subsets cover eight classes of human body parts, such as hands, heads, and feet, and exhibit variations in point distribution, pose, and human shapes.

### B. Implementation

Following previous works [8], [9], [31], we adopt DGCNN [2] as point cloud feature extractor, which produces a 1024-dimensional global feature for each input. In the meanwhile, the classifier is implemented based on a multi-layer perceptron (MLP). As for the normal prediction head and DoN prediction head, we employ two-layer MLPs to predict the feature descriptors respectively. During training, we use Adam optimizer [37], which is adopted with the initial learning rate 0.001, weight decay 0.00005 and an epoch-wise cosine annealing learning rate scheduler. The batch size is set as 16 for each method. The neighboring points number $r$ for normal calculation is empirically set as 8. As for the DoN calculation, $r_s$ is set equal to $r$, and $r_l$ is set as $2r$. To obtain proper prototypes, we set 0.8 for threshold $\lambda$ to select reliable samples, and set $\mu$ as 0.999 for prototype update.

### C. Comparison Results

We compare our proposed method with a list of recent state-of-the-art point-based domain adaptation methods, covering a variety of types. Additionally, we give the results of oracle method that trains the point cloud model with labeled target data supervisedly, and the w/o Adapt method that trains model with only labeled source samples, serving as the upper and lower performance bounds, respectively.

We report the experimental results in Table I. Compared to the baseline model, a great improvement on all six benchmarks can be observed with our method. Thanks to the masked representation learning and prototype-calibrated self-training, the model learns better representation across domains and obtains more discrimative decision boundaries in target domain. Besides, compared to the existing state-of-the-art methods, our proposed method achieves superior performance on average. Besides, in the challenging and difficult synthetic-to-real settings, such as $M \rightarrow S^*$ and $S \rightarrow S^*$, we make a great improvement than baseline and also surpass previous methods.

Furthermore, to demonstrate the generalizability of our approach, we extend it to point cloud segmentation on PointSegDA and present the comparision results in Table II. Compared to the baseline, a significant performance improvement can be observed in almost all adaptation settings, which validates the effectiveness of our approach for point cloud segmentation. Besides, we also achieve the best performance on average, surpassing previous methods.

### D. Ablation Study

**The Effectiveness of Each Component.** We detail the performance of each design as well as their improvement by progressively intergrating them to the training pipeline. As illustrated in Table III, the self-supervised learning and self-training consistently improve the performance over the source-only baseline. The highest classification accuracy is achieved when combined the two strategies. The results of self-supervised learning validate our claim that joint learning of masked feature prediction and masked sample consistency lead to more generic and transferable point cloud representations. An interesting observation is that, directly introducing prototypes to the framework does not bring much performance gain. However, when combined with the self-supervised learning, a remarkable improvement is achieved, which can be attributed to the well learnt clustered features. Thanks to proposed representation learning technique, the features are more effectively clustered within the shared feature space, leading to improved performance in both prototype-learning and self-training.

**Masked Local-Global Representation Learning.** To validate the effectiveness of domain-invariant feature learning, we investigate the contribution of masked local-global representation learning, especially the usage of normal and DoN estimation, in the meanwhile compare with several other representation learning methods [7], [8], [10] proposed for point cloud domain adaptation. The results in Table IV show the effectiveness of masked learning strategy. Since

TABLE II

THE COMPARISON RESULTS OF SEGMENTATION MIOU ON POINTSEGDA [7].

| Method | F→A | F→M | F→S | M→A | M→F | M→S | A→F | A→M | A→S | S→A | S→F | S→M | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Oracle | 84.0 | 81.8 | 82.4 | 84.0 | 80.9 | 82.4 | 80.9 | 81.8 | 82.4 | 84.0 | 80.9 | 81.8 | 82.3 |
| w/o Adaptation | 78.5 | 60.9 | 66.5 | 26.6 | 33.6 | 69.9 | 38.5 | 31.2 | 30.0 | 74.1 | 68.4 | 65.5 | 53.6 |
| Adapt-SegMap [36] | 70.5 | 60.1 | 65.3 | 49.1 | **54.0** | 62.8 | 44.2 | **35.4** | 35.1 | 70.1 | 67.7 | 63.8 | 56.5 |
| RS [6] | 78.7 | 60.7 | 66.9 | 59.6 | 38.4 | 70.4 | 44.0 | 30.4 | 36.6 | 70.7 | **73.0** | 65.3 | 57.9 |
| DefRec [7] | 79.7 | 61.8 | **67.4** | 67.1 | 40.1 | **72.6** | 42.5 | 28.9 | 32.2 | 66.4 | 72.2 | 66.2 | 58.1 |
| MLSP [31] | **80.9** | 60.0 | 65.5 | 67.3 | 40.4 | 70.8 | 45.4 | 31.1 | 38.4 | 74.8 | 72.5 | 66.6 | 59.5 |
| Ours | 80.5 | **62.7** | 67.0 | **67.9** | 41.7 | 70.8 | **46.8** | 31.5 | **39.8** | **75.9** | 72.1 | **67.2** | **60.2** |

TABLE III

ABLATION STUDY ON EACH COMPONENT OF OUR METHOD. THE EXPERIMENTS ARE CARRIED ON S → S* OF POINTDA-10.

| self-supervised | | self-training | | Accuracy | Gain |
|---|---|---|---|---|---|
| local | global | pseudo-label | prototype | | |
| | | | | 42.5 | |
| ✓ | | | | 54.5 | +12.0 |
| | ✓ | | | 50.2 | +7.7 |
| ✓ | ✓ | | | 55.9 | +13.4 |
| | | ✓ | | 48.9 | +6.4 |
| | | ✓ | ✓ | 49.3 | +6.8 |
| ✓ | ✓ | ✓ | | 57.0 | +14.5 |
| ✓ | ✓ | ✓ | ✓ | **59.1** | **+16.6** |

TABLE IV

COMPARISON OF POINT CLOUD REPRESENTATION LEARNING.

| Method | Avg Accuracy |
|---|---|
| mixup rotation classification [8] | 49.0 |
| scaling-up-down [10] | 48.3 |
| deformation-reconstruction [7] | 49.9 |
| 3D-2D-3D projection [10] | 48.7 |
| deformation-localization [8] | 51.4 |
| masked normal prediction | 53.8 |
| masked DoN prediction | 53.0 |
| Normal & DoN prediction | **54.5** |

point normal and DoN are both obtained by considering the neighborhood, it has superiority in modelling point cloud geometry structures, resulting in better performance than the existing both global and local methods.

**Prototype-Calibrated Self-Training.** Several works [8], [10], [31] also apply self-training for point cloud domain adaptation. In this section, we specifically compare the self-training results under different strategies. We start by applying the conventional self-training (first row), $i.e.$, using target pseudo-labels whose prediction confidences are above the threshold to train classifier. And we also implement the entropy-based [31] and reliable vote [10] pseudo label selection strategy. Since other configurations are set the same for comparison, from Table V, we can see the improvements brought by our prototype-calibrated self-training.

### *E. Hyper-Parameter Sensitivity Analysis*

In this section, we analysis the sensitivity of the hyper-parameters in our method, $e.g.$, update momentum $\mu$ and threshold $\lambda$ for prototype calculation. The parameter sensitivity for our proposed framework is illustrated in Table VI. It is observed that our prototype-calibrated technique is robust to the update momentum $\mu$ in a wide numerical range when refining pseudo labels. For $\lambda$ that controls the confidence reliability of the sampled features, too low threshold will introduce much wrong predictions and noise for prototypes, while too high threshold will affect the selected feature numbers, especially for those long tailed classes. Finally, we empirically set the momentum as 0.999 and the threshold as 0.8. In addition, we empirically set the neighboring points number $r$ as 8 to achieve the best performance.

TABLE V

COMPARISON OF DIFFERENT SELF-TRAINING STRATEGIES.

| Method | M→S | M→S* | S→M | S→S* | S*→M | S*→S |
|---|---|---|---|---|---|---|
| Confidence [8] | 85.7 | 59.4 | 82.3 | 57.0 | **82.2** | 76.4 |
| Entropy [31] | 86.0 | 59.9 | 83.5 | 58.0 | 81.2 | 76.7 |
| Vote [10] | 85.4 | 60.4 | 82.6 | 58.5 | 79.8 | 75.9 |
| Ours | **86.9** | **61.2** | **85.7** | **59.1** | 81.6 | **77.3** |

TABLE VI

HYPER-PARAMETER SENSITIVITY ANALYSIS FOR THE MOMENTUM $\mu$, THRESHOLD $\lambda$ AND NEIGHBORING POINTS NUMBER $r$.

| Momentum ($\mu$) | 0.9 | 0.95 | 0.99 | 0.999 | 0.9999 |
|---|---|---|---|---|---|
| Accuracy | 58.5 | 58.5 | 58.9 | **59.1** | 59.0 |
| Threshold ($\lambda$) | 0 | 0.3 | 0.6 | 0.8 | 0.95 |
| Accuracy | 56.7 | 57.2 | 58.5 | **59.1** | 58.7 |
| Neighbor Points ($r$) | 4 | 6 | 8 | 10 | 12 |
| Accuracy | 58.0 | 58.8 | **59.1** | 58.9 | 58.5 |

## V. CONCLUSION

In this paper, we propose masked local-global representation learning and prototype-calibrated self-training to improve point cloud domain adaptation. The normal estimation and DoN estimation facilitate the model to understand the geometry structure of point cloud, while the masked sample consistency helps to capture global semantic information. With the learned representative features, we are able to maintain class-wise prototypes by leveraging the overall domain data, allowing for the refinement of pseudo labels from a broader perspective, which is beneficial and effective for the self-training in the target domain. Combining the proposed strategies, we establish new state-of-the-art results on the two widely used point cloud domain adaptation datasets.

## REFERENCES


[1] 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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 77–85, 2017.

[2] 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)*, vol. 38, no. 5, pp. 1–12, 2019.

[3] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, 2021.

[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 30, pp. 5099–5108, 2017.

[5] C. Qin, H. You, L. Wang, C. J. Kuo, and Y. Fu, "Pointdan: A multi-scale 3d domain adaption network for point cloud representation," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 7190–7201, 2019.

[6] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 12942–12952, 2019.

[7] 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*, pp. 123–133, 2021.

[8] 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*, pp. 6383–6392, 2021.

[9] 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*, pp. 7213–7222, 2022.

[10] H. Fan, X. Chang, W. Zhang, Y. Cheng, Y. Sun, and M. S. 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*, pp. 6367–6376, 2022.

[11] 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 *Proceedings of the European Conference on Computer Vision*, pp. 728–745, 2022.

[12] A. Cardace, R. Spezialetti, P. Z. Ramirez, S. Salti, and L. D. Stefano, "Refrec: Pseudo-labels refinement via shape reconstruction for unsupervised 3d domain adaptation," in *Proceedings of the International Conference on 3D Vision*, pp. 331–341, 2021.

[13] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2962–2971, 2017.

[14] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, pp. 59:1–59:35, 2016.

[15] L. Chen, H. Chen, Z. Wei, X. Jin, X. Tan, Y. Jin, and E. Chen, "Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7171–7180, 2022.

[16] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732, 2018.

[17] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 513–520, 2006.

[18] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," *arXiv preprint arXiv:1612.01939*, 2016.

[19] P. Morerio, J. Cavazza, and V. Murino, "Minimal-entropy correlation alignment for unsupervised deep domain adaptation," in *Proceedings of the International Conference on Learning Representations*, 2018.

[20] J. Liang, D. Hu, and J. Feng, "Domain adaptation with auxiliary target domain-oriented classifier," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16632–16642, 2021.

[21] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12414–12424, 2021.

[22] A. Cardace, R. Spezialetti, P. Z. Ramirez, S. Salti, and L. D. Stefano, "Self-distillation for unsupervised 3d domain adaptation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4155–4166, 2023.

[23] A. K. Thabet, H. Alwassel, and B. Ghanem, "Mortonnet: Self-supervised learning of local features in 3d point clouds," *arXiv preprint arXiv:1904.00230*, 2019.

[24] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via occlusion completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9762–9772, 2021.

[25] S. Xie, J. Gu, D. Guo, C. R. Qi, L. J. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *Proceedings of the European Conference on Computer Vision*, pp. 574–591, 2020.

[26] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked autoencoders for point cloud self-supervised learning," in *Proceedings of the European Conference on Computer Vision*, vol. 13662, pp. 604–621, 2022.

[27] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5376–5385, 2020.

[28] D. Liu, C. Chen, C. Xu, R. C. Qiu, and L. Chu, "Self-supervised point cloud registration with deep versatile descriptors for intelligent driving," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[29] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15979–15988, 2022.

[30] C. Wei, H. Fan, S. Xie, C. Wu, A. L. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14648–14658, 2022.

[31] 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 *Proceedings of the European Conference on Computer Vision*, pp. 156–172, 2022.

[32] Y. Ioannou, B. Taati, R. Harrap, and M. A. Greenspan, "Difference of normals as a multi-scale operator in unorganized point clouds," in *Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pp. 501–508, 2012.

[33] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, 2015.

[34] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[35] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2432–2443, 2017.

[36] Y. Tsai, W. Hung, S. Schulter, K. Sohn, M. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7472–7481, 2018.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015.