PointClustering: Unsupervised Point Cloud Pre-training using Transformation Invariance in Clustering

Fuchen Long, Ting Yao, Zhaofan Qiu, Lusong Li and Tao Mei
HiDream.ai Inc.
JD Explore Academy, Beijing, China
{longfc.ustc, tingyao.ustc, zhaofanqiu, lilusong}@gmail.com
tmei@hidream.ai

Abstract

Feature invariance under different data transformations, i.e., transformation invariance, can be regarded as a type of self-supervision for representation learning. In this paper, we present PointClustering, a new unsupervised representation learning scheme that leverages transformation invariance for point cloud pre-training. PointClustering formulates the pretext task as deep clustering and employs transformation invariance as an inductive bias, following the philosophy that common point cloud transformation will not change the geometric properties and semantics. Technically, PointClustering iteratively optimizes the feature clusters and backbone, and delves into the transformation invariance as learning regularization from two perspectives: point level and instance level. Point-level invariance learning maintains local geometric properties through gathering point features of one instance across transformations, while instance-level invariance learning further measures clusters over the entire dataset to explore semantics of instances. Our PointClustering is architecture-agnostic and readily applicable to MLP-based, CNN-based and Transformer-based backbones. We empirically demonstrate that the models pre-learnt on the ScanNet dataset by PointClustering provide superior performances on six benchmarks, across downstream tasks of classification and segmentation. More remarkably, PointClustering achieves an accuracy of 94.5% on ModelNet40 with Transformer backbone. Source code is available at https://github.com/FuchenUSTC/PointClustering.

1. Introduction

3D point cloud analysis has seen tremendous progress and made great success in industrial applications, e.g., autonomous driving, augmented reality and robotics. The achievements heavily rely on large quantities of human annotations for supervised learning. However, acquiring and manual labeling 3D point cloud data is very expensive and time-consuming, while the underlying rich data structure is also not yet fully leveraged. In contrast, unsupervised learning leaves it on its own to characterize the underlying feature distribution completely on data itself and is therefore an appealing way towards more generic model pre-training.

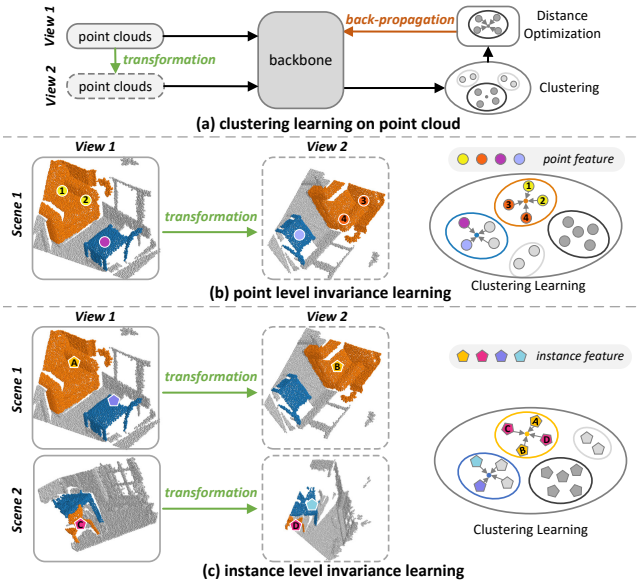

Figure 1. Illustration of (a) clustering learning on point cloud by using feature invariance at (b) point level and (c) instance level.

The research in unsupervised point cloud pre-training has mainly proceeded along two dimensions with respect to the formulation of pretext task: contrastive learning [23, 65, 74] and reconstruction [34, 52, 60]. Early works of contrastive learning generally suggest to leverage point or scene discrimination across different views [65] or modalities [1, 74] for similarity learning. Instead, the direction of point cloud reconstruction [34, 60] formulates the learning target as shape completion from the partial points. Unlike existing discrimination or reconstruction paradigm in a

sample-specific manner, clustering technique estimates the data distribution *holistically for class level*. We rely on such recipe and shape a new unsupervised point cloud pre-training scheme that capitalizes on deep clustering as the pretext task. Technically, we iteratively optimize feature clusters and backbone as shown in Figure 1(a), and utilize transformation invariance as an inductive bias. We look into the feature invariance learning across data transformations from two aspects: point level and instance level. The rationale behind point level feature invariance is that the point features of an identical object (e.g., points of the chair in Figure 1(b)) should be invariant across different transformations since the geometric properties will not change with transformations. Similar in spirit, the high-level semantics of instances across 3D scenes (e.g., the instances of chair in Figure 1(c)) do not vary along with the transformations. As such, we delve into both point-level and instance-level transformation invariance to regulate deep clustering.

By materializing the idea of transformation invariance as regularization for deep clustering, we present a novel PointClustering approach for unsupervised point cloud pre-training. Specifically, we first obtain the instance masks of objects in each 3D scene via Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [12] algorithm. Based on the instance masks, the point features of an identical object under different transformations are clustered together to characterize geometric properties of points. The instance-level feature of one object is then computed by globally pooling all point features of that object. Given all instance features over the entire dataset, PointClustering further seeks the feature consistency across transformations at instance level. We employ InfoNCE loss to optimize the similarity between points or instances and their corresponding clustering centroids (i.e., prototypes).

The main contribution of this work is a new paradigm that leverages feature invariance under different data transformations for unsupervised point cloud pre-training. The solution also leads to the elegant view of how to explore self-supervision from the standpoint of transformation invariance, and how to indicate geometric properties and semantics of point cloud for unsupervised pre-training. Extensive experiments on six benchmarks over three downstream tasks verify that PointClustering outperforms the state-of-the-art unsupervised pre-training models.

2. Related Work

3D Architecture Design. Rapid development of deep architectures to analyze 3D point cloud [10, 11, 16, 17, 22, 25, 26, 42, 43, 45, 54] has been witnessed in recent years. Related works can be grouped into three categories: CNN-based [9, 16], MLP-based [39, 43, 44, 48] and Transformer-based [29, 75] architectures. CNN-based methods [9, 16] transform the irregular point clouds to regular 3D voxels for the operation of sparse convolution. Nevertheless, the quantization in voxelization may result in the loss of geometric information and limit the model capacity. By directly processing the irregular points without quantization, MLP-based networks [43,44] obtain fairly well accuracy on point cloud classification task. Inspired by the success of self-attention in NLP and visual understanding, Transformer-based point cloud backbones [29, 75] start to emerge. The attention mechanism that is invariant to input permutation is applicable to point cloud modeling. In this paper, we conduct experiments on all the three kinds of backbones to verify our unsupervised pre-training scheme.

Unsupervised Model Pre-training. Model pre-training on image [4, 7, 20, 21, 31] or video [13, 32, 35–38, 46, 47, 55, 61, 68] data without human annotations is a fundamental research topic. The key point is how to formulate a pretext task for model optimization. There are various directions to mine self-supervision from images or videos, e.g., rotation prediction [15], reconstruction [20, 55], colorization [30, 59, 72] and contrastive learning [7, 8, 21, 32, 57, 63]. Great success of unsupervised pre-training has been achieved by even showing better downstream performances [7, 20] against supervised pre-training.

More recently, unsupervised model pre-training for 3D point cloud data [18, 19, 23, 49, 50, 52, 60, 65, 67] begins to be investigated. Xie *et al.* [65] first demonstrate the effectiveness of the point-level contrastive learning among different views. To alleviate the cost of view alignment, DepthContrast [74] is further designed for single view but multi-modality 3D contrastive learning. Getting inspiration from the masked image modeling [20], Liu *et al.* [34] train the point Transformer through the pretext task of shape completion. Despite having these innovations, the feature invariance of point cloud across different data transformations is seldom explored for unsupervised model pre-training.

Learning to Cluster. Employing deep models to learn clustering-friendly feature embedding [5, 6, 64, 71] has been widely studied in image domain. During the cluster learning, feature vectors of images from the entire dataset are first grouped by K-means to assign pseudo label to each sample, and then the networks are trained on the pseudo labels. Such iterative optimization between clustering and networks training enhances feature embedding to account for high-level visual similarity. In our work, we capitalize on deep clustering and formulate it as the pretext task for unsupervised point cloud pre-training.

In short, our work mainly focuses on a new unsupervised point cloud pre-training scheme that exploits deep clustering as the pretext task. The proposal of PointClustering contributes by studying not only modeling feature invariance under different transformations in clustering learning, but also how geometric properties and semantics of point cloud can be leveraged to improve representation.

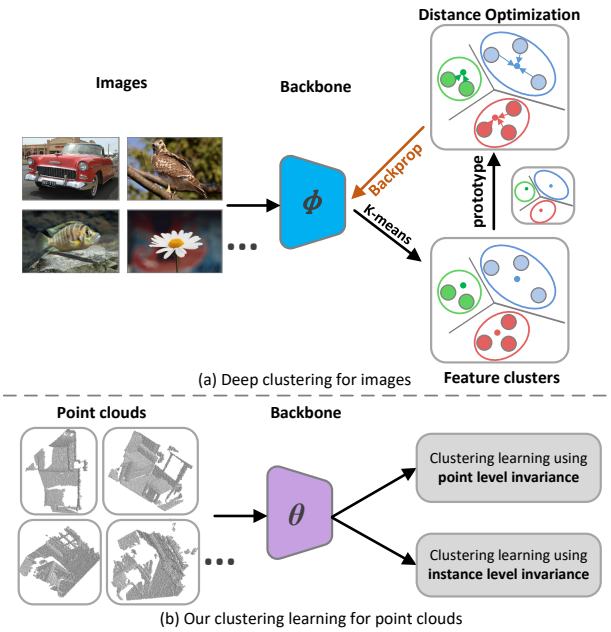

Figure 2. Illustration of deep clustering learning for (a) images and (b) point clouds in our proposal.

3. PointClustering

In this section, we introduce our newly-minted PointClustering for unsupervised point cloud pre-training. The main idea is to formulate the pretext task as deep clustering and regulate the learning procedure by transformation invariance. We investigate the transformation invariance learning from two aspects, i.e., point level and instance level, which aims to characterize the geometric properties and semantics of point clouds, respectively. Point-level invariance learning emphasizes the point feature consistency across different data transformations, and instance-level feature invariance learning explores the semantics of instances over the entire dataset to strengthen representation.

3.1. Preliminaries: Deep Clustering

Prior works [5, 6] on image unsupervised representation learning formulate the pretext task as deep feature clustering. The goal is to train deep models for clustering images into different groups that could partition semantics. The problem is not trivial due to the lack of semantic supervision. To alleviate this issue, Caron *et al.* [5] propose an iterative learning paradigm where feature clusters and networks are jointly optimized to estimate class-level data distribution. We proceed along this direction and formulate the pretext task of point cloud pre-training as deep clustering. The transformation invariance is further leveraged as an inductive bias to regulate the clustering learning.

Here we introduce the preliminary knowledge of image deep clustering as depicted in Figure 2(a) and then extend this method for point cloud representation learning. Formally, given a set of unlabeled images $\{x_i\}_{i=0}^{N-1}$, we first extract the image features $\{f_i\}_{i=0}^{N-1}$ from the deep model ϕ. Next, K-means algorithm is employed to group the image features into K clusters. We denote the corresponding clustering centroids (prototypes) as $\mathbf{u} = \{u_j\}_{j=0}^{K-1}$ and cluster assignments (labels) as $\mathbf{y} = \{y_i\}_{i=0}^{N-1}$, respectively. Instead of projecting the image features via a learnable classifier [5] for classification, we propose a clustering loss L_c to optimize the similarity between images and the clustering prototypes via InfoNCE [57]. That avoids involving additional parameters in training. For the i-th image with label y_i, the image feature f_i is optimized by L_c as follows:

$$L_c(f_i, \mathbf{u}, \mathbf{y}) = -\log \frac{\exp\left(f_i \cdot u_{y_i}/\tau\right)}{\sum_{j=0}^{K-1} \exp\left(f_i \cdot u_j/\tau\right)}, \tag{1}$$

where u_{y_i} is the prototype with clustering label y_i, and τ denotes the temperature hyper-parameter. The extracted feature clustering and similarity optimization are executed iteratively in each training epoch to explore image semantics.

We exploit the recipe and extend the image deep clustering for point cloud feature learning as shown in Figure 2(b). One natural extension is to directly replace the image feature f_i in Eq.(1) with the global point cloud feature. However, such solution ignores the inherent geometry of point cloud thus may limit the model capacity. To better consolidate deep clustering on point cloud data, we propose to employ the feature invariance under different data transformations (e.g., rotation) as an inductive bias in optimization. The learning objective is devised to maintain the feature invariance regardless of data transformation during clustering. Compared to solely relying on data itself, involving this kind of regularization will facilitate clustering construction. Moreover, we integrate the exploitation of geometric properties and semantics on point cloud data into deep clustering by considering feature invariance from two aspects, i.e., point level and instance level, respectively.

3.2. Point-level Invariance Learning

Different from the image data, point clouds typically contain plenty of geometric details. Several advances [28, 33, 66] demonstrate that involving transformation invariance (e.g., rotation invariance) into the design of local point descriptor is helpful to build a robust system for point cloud understanding. The common data transformations do not influence the geometric properties of the objects or 3D scenes. Therefore, we introduce to leverage the feature invariance at point level as a regularization term for clustering learning. The features of points from the same instance under different data transformations are expected to group together to reflect the geometric properties .

Figure 3 details the pipeline of point cloud feature clustering learning by using feature invariance at point level. Technically, given a single 3D scene, we first cluster

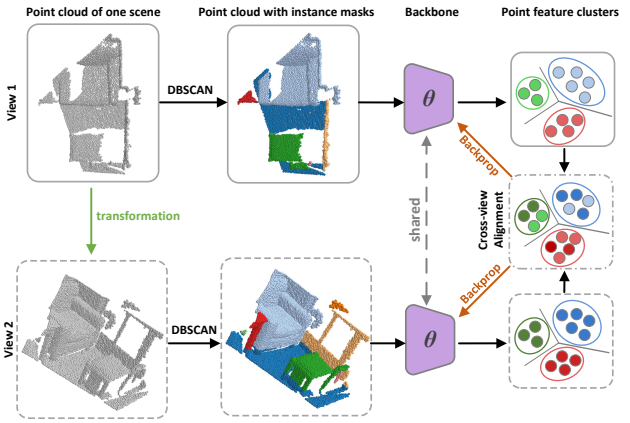

Figure 3. Pipeline of point-level invariance learning in clustering for a single 3D scene.

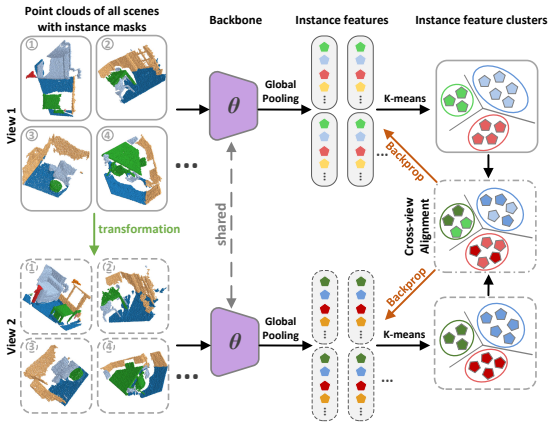

Figure 4. Pipeline of instance-level invariance learning in clustering for all 3D scenes over the entire dataset.

the point cloud coordinates into different groups through the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [12] algorithm, which has been validated for point cloud analysis [2]. We take the clustering results as the instance masks of objects. Meanwhile, we feed the pair of the 3D scene with and without data transformation into the backbone θ for point feature extraction. We regard such two inputs as the data of two views, and the instance masks are same in each view. Conditioned on the instance masks, the feature of one instance is obtained by globally pooling all point features of that instance. Next, we collect all instance features in the scene as clustering prototypes and form the point prototype sets $\mathbf{u}^{P_1}$ and $\mathbf{u}^{P_2}$ for the two views. The corresponding cluster assignments are taken as point label sets $\mathbf{y}^{P_1}$ and $\mathbf{y}^{P_2}$ for each view. Given the extracted point features $f_i^{P_1}$ and $f_i^{P_2}$ of the i-th point in the 3D scene across two views, the inner-view point clustering loss L_{ine}^{p} for point level feature clustering optimization is computed by:

$$L_{ine}^{P} = L_c(f_i^{P_1}, \mathbf{u}^{P_1}, \mathbf{y}^{P_1}) + L_c(f_i^{P_2}, \mathbf{u}^{P_2}, \mathbf{y}^{P_2}), \quad (2)$$

where $L_c(\cdot,\cdot,\cdot)$ denotes the clustering loss defined by Eq.(1). In addition to the inner-view clustering learning which solely depends on point cloud data in its own view, the point features should be invariant across different data transformations. Therefore, we make an alignment of the learnt clusters across the two views to achieve the feature invariance during clustering, and adopt the cross-view point clustering loss L_{cro}^{p} as:

$$L_{cro}^{P} = L_c(f_i^{P_1}, \mathbf{u}^{P_2}, \mathbf{y}^{P_2}) + L_c(f_i^{P_2}, \mathbf{u}^{P_1}, \mathbf{y}^{P_1}). \quad (3)$$

By doing so, the feature of the point in one view can be learnt close to the prototype of the same point in the other view. Thus, the corresponding two clusters will be in close proximity. Finally, the objective for clustering learning by using feature invariance at point level is the combination of inner-view and cross-view point clustering losses:

$$L^{P} = L_{ine}^{P} + L_{cro}^{P}. \quad (4)$$

3.3. Instance-level Invariance Learning

Point-level invariance learning for clustering probes geometric properties of points to discriminate different instances in a 3D scene. The robust learning of point cloud feature also necessitates exploring high-level semantics of instances across different scenes. In image analysis, semantic mining through clustering [5, 6] is the core idea to boost unsupervised representation learning. To further estimate category level distributions of point clouds through clustering, we propose to employ the feature invariance at instance level to regulate clustering optimization, which is executed on all instances across different 3D scenes over the entire dataset for similarity learning.

Similarly, the design of instance-level invariance learning in clustering is to group the features of similar instances together irrespective of data transformation. Figure 4 illustrates the learning procedure of clustering across scenes under the constraint of feature invariance at instance level. As mentioned in Section 3.2, we first obtain the instance features through globally pooling point features based on the instance masks. For the input data of two views (original and transformed data), we collect the instance features of all scenes over the entire dataset for each view, respectively. After that, K-means algorithm is utilized to cluster all instance features in each view into K^I groups, respectively, and two instance prototype sets, $\mathbf{u}^{I_1}$ and $\mathbf{u}^{I_2}$, are generated. We take the corresponding cluster assignments of each view as instance label sets $\mathbf{y}^{I_1}$ and $\mathbf{y}^{I_2}$. Given the i-th instance of the dataset, the obtained instance features $f_i^{I_1}$ and $f_i^{I_2}$ across two views will be optimized via the inner-view instance clustering loss L_{ine}^{I} and cross-view instance clustering loss L_{cro}^{I}. The inner-view instance clustering objective L_{ine}^{I} is formulated as follows:

$$L_{ine}^{I} = L_c(f_i^{I_1}, \mathbf{u}^{I_1}, \mathbf{y}^{I_1}) + L_c(f_i^{I_2}, \mathbf{u}^{I_2}, \mathbf{y}^{I_2}). \quad (5)$$

Meanwhile, the alignment of instance clusters across two views is to maintain feature invariance under different data

Table 1. The statistics of six datasets for downstream tasks, and the performance gains by using PointClustering against scratch training with PointViT [34] backbone for model pre-training.

Dataset	Statistic	Task	Gain
ModelNet40 [62]	9.8K train, 2.5K val	Object Cls.	**+3.0% Acc**
ScanObjectNN [56]	11.4K train, 2.9K val	Object Cls.	**+10.1% Acc**
ShapeNetPart [69]	14.0K train, 2.9K val	Part Seg.	**+1.6% mIoU**
PartNet [40]	17.1K train, 2.5K val	Part Seg.	**+4.3% mIoU**
S3DIS [3]	199 train, 67 val	Semantic Seg.	**+6.7% mIoU**
ScanNetV2 [10]	1.2K train, 312 val	Semantic Seg.	**+5.7% mIoU**

transformations. Similar to the rationale behind point-level invariance learning, L^I_{cro} is calculated by

$$L^I_{cro} = L_c(f^{I_1}_i, \mathbf{u}^{I_2}, \mathbf{y}^{I_2}) + L_c(f^{I_2}_i, \mathbf{u}^{I_1}, \mathbf{y}^{I_1}). \quad (6)$$

The objective for clustering learning by employing feature invariance at instance level is

$$L^I = L^I_{ine} + L^I_{cro}. \quad (7)$$

In the training stage, we iteratively measure feature clusters and optimize backbone as in DeepCluster [5] from the aspects of both point and instance level. The overall training objective of PointClustering integrates L^P and L^I as

$$L_{ov} = L^P + L^I. \quad (8)$$

Here, we empirically treat each loss term equally.

4. Experiments

We adopt the standard *unsupervised pre-training + supervised fine-tuning* protocol [60, 65, 74] to verify the merit of our PointClustering. The backbone is first pre-trained by PointClustering on the **ScanNet** [10] dataset and then evaluated on a variety of downstream tasks, including object classification, part segmentation and semantic segmentation.

4.1. Datasets and Implementation Details

Datasets. The **ScanNet** [10] dataset for pre-training contains 2.5 million RGB-D scaning frames from more than $1,500$ indoor scenes. We employ similar data pre processing as in DepthContrast [74] and extract around 190K 3D scans from about $1,200$ depth video sequences in the training set. For each scan, we sample $8,192$ points for point cloud pre-training without any 3D registration operation.

We conduct experiments on downstream tasks over six datasets, including two point cloud object classification datasets of **ModelNet40** [62] and **ScanObjectNN** [56], two part segmentation datasets of **ShapeNetPart** [69] and **PartNet** [40], and two semantic segmentation datasets of **S3DIS** [3] and **ScanNetV2** [10]. Table 1 summarizes the statistics of the six datasets and performance improvements by using PointClustering over scratch training with PointViT [34] backbone for unsupervised point cloud pre-training.

Backbones. For the fair comparisons with the recent unsupervised learning approaches, we experiment on three kinds of backbones, i.e., MLP-based (PointNet++ [44]), CNN-based (SR-UNet [65]) and Transformer-based (PointViT [34]) networks. PointNet++ consists of three layers for feature extraction and three layers for feature aggregation. The network takes irregular 3D points as the input and exploits the multi-scale grouping strategy. SR-UNet is a 34-layer U-Net [51] architecture that includes an encoder of 21 convolution layers and a decoder of 13 convolution/deconvolution layers. The input to SR-UNet is 3D voxel by setting the voxel size as $2cm$. PointViT is a standard point Transformer by taking points as the inputs and has 12-layer encoder and a single layer decoder. In the attention block, the hidden dimension is set as 384 and the number of heads is 6. The expansion ratio of feed forward layer is 4 with the 0.1 drop rate of stochastic depth.

Table 2. Top-1 accuracy on ModelNet40 and ScanObjectNN. Gains over scratch training are indicated in the bracket.

Approach	Backbone	ModelNet40	ScanObjectNN
Scratch	PointNet++	90.7	77.9
DepthContrast [74]	PointNet++	91.3	-
GLR [50]	PointNet++	93.0	-
ReSp [52]	DGCNN	92.4	-
OcCo [60]	DGCNN	93.0	-
PointClustering	PointNet++	**94.1** (+3.4)	**84.5** (+6.6)
Scratch	SR-UNet	90.1	76.2
PointContrast [65]	SR-UNet	91.2	-
PointClustering	SR-UNet	**93.6** (+3.5)	**83.7** (+7.5)
Scratch	PointViT	91.5	77.2
Point-BERT [70]	PointViT	93.2	83.1
MaskPoint [34]	PointViT	93.8	84.3
Point-MAE [41]	PointViT	93.8	85.2
MaskSurf [73]	PointViT	93.4	85.8
PointClustering	PointViT	**94.5** (+3.0)	**87.3** (+10.1)

Network Training. We implement our PointClustering approach on PyTorch framework. The data transformations include random rotation and scaling. In each training epoch, the clusters are calculated with mini-batch K-means [53] accelerated by GPUs using FAISS [27] library. The point feature dimension for clustering learning is 32. We employ max pooling on point features to generate instance feature, and set the instance clustering number K^I as 32 by cross validation. The mini-batch Stochastic Gradient Descent (SGD) algorithm is employed for optimization. We set the base learning rate as 0.001 for PointNet++ and PointViT, and 0.01 for SR-UNet. The maximum training epoch number is 128. The mini-batch size is 32 and the weight decay parameter is set as 0.0001. For the training stage of supervised fine-tuning, we provide full implementation details and settings in the supplementary material.

4.2. Evaluation on Object Classification

We first conduct supervised fine-tuning of object classification on ModelNet40 and ScanObjectNN. Note that we

Table 3. Few-shot evaluation on ModelNet40. Average top-1 accuracy and standard deviation of 10 independent runs are reported.

Approach	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
Scratch (PointViT)	87.8 ± 5.3	93.3 ± 4.5	84.6 ± 5.5	89.4 ± 6.3
Point-BERT [70]	94.6 ± 3.1	96.3 ± 2.7	91.0 ± 5.4	92.7 ± 5.1
MaskPoint [34]	95.0 ± 3.7	97.2 ± 1.7	91.4 ± 4.0	93.4 ± 3.5
PointClustering	$\mathbf{95.8 \pm 3.0}$	$\mathbf{97.6 \pm 0.9}$	$\mathbf{92.8 \pm 3.2}$	$\mathbf{93.8 \pm 3.4}$

sample $1,024$ points of each sample in ModelNet40 for training and validation. For the ScanObjectNN dataset, there are three commonly adopted [34,41,56,70] data splits: OBJ-ONLY (object only), OBJ-BG (with background) and PB-T50-RS (with background and manual perturbations). Here, we take the most challenging setting PB-T50-RS as the evaluation. The top-1 accuracy is reported on the validation set of the two datasets for performance comparison.

Table 2 summarizes the performances across different unsupervised pre-training approaches. Scratch is the run, where the model is trained from scratch with parameter random initialization. Overall, PointClustering consistently achieves better performances than other baselines across three backbones and two datasets. In particular, PointClustering with SR-UNet leads to the highest performance gain of 3.5% over Scratch among the three backbones on ModelNet40. Though voxelization may result in the loss of geometric information in CNN-based SR-UNet backbone, PointClustering alleviates this downside via characterizing geometric properties by regulating point-level invariance in clustering. Similar performance improvements are also observed on the more challenging ScanObjectNN which contains more data noise. PointClustering with PointViT backbone obtains 87.3% top-1 accuracy, surpassing the best competitor MaskSurf by 1.5%. The result basically indicates the advantage of exploring transformation invariance in clustering for unsupervised point cloud pre-training.

To better validate model generalization ability, we follow the previous works [34, 70] and evaluate PointClustering (PointViT as backbone) under the setting of few-shot learning on ModelNet40. The typical setting is the "K-way N-shot" which randomly chooses K classes with N+20 samples per class. The N samples of each class are utilized for training, and the rest 20 samples are used for testing. Table 3 lists the performances under the settings of $K \in \{5, 10\}$ and $N \in \{10, 20\}$. PointClustering constantly exhibits better accuracy than other models across the four settings and shows the smallest standard deviations. Even fine-tuning with 10 samples, PointClustering still manifests the strong transferability for point cloud understanding.

4.3. Evaluation on Part Segmentation

Part segmentation is a fine-grained classification task that classifies each point of one known object into part label (e.g., the leg of chair). We adopt the standard data split [44] of ShapeNetPart and the fine-grained level annotations [40] of PartNet for evaluation. Table 4 details the instance mean Intersection over Union (mIoU) on two datasets. As indicated by the results, PointClustering outperforms all baselines and particularly attains 86.7% instance mIoU with PointViT backbone on ShapeNetPart. Despite having strong backbone of Transformer, PointClustering still leads to 1.6% mIoU gain and such improvement again verifies the powerful generalization ability of our proposal. Similar performance trend is also shown on the PartNet dataset.

Table 4. Instance mIoU on ShapeNetPart and PartNet datasets. Performance gain over scratch training is shown in the bracket.

Approach	Backbone	ShapeNetPart	PartNet
Scratch	PointNet++	84.9	42.5
OcCo [60]	DGCNN	85.0	-
ReSp [52]	DGCNN	85.3	-
PointClustering	PointNet++	**85.9** (+1.0)	**47.0** (+4.5)
Scratch	SR-UNet	84.7	38.9
PointContrast [65]	SR-UNet	85.1	41.5
PointClustering	SR-UNet	**86.0** (+1.3)	**42.1** (+3.2)
Scratch	PointViT	85.1	45.8
Point-BERT [70]	PointViT	85.6	-
MaskPoint [34]	PointViT	86.0	-
MaskSurf [73]	PointViT	86.1	-
Point-MAE [41]	PointViT	86.1	-
PointClustering	PointViT	**86.7** (+1.6)	**50.1** (+4.3)

Table 5. Performance of mIoU on S3DIS and ScanNetV2 datasets. Performance gain over scratch training is shown in the bracket.

Approach	Backbone	S3DIS	ScanNetV2
Scratch	PointNet++	55.3	57.9
OcCo [60]	DGCNN	58.0	-
PointClustering	PointNet++	**61.2** (+5.9)	**62.6** (+4.7)
Scratch	SR-UNet	68.2	70.3
DepthContrast [74]	SR-UNet	71.5	71.2
CSC [23]	SR-UNet	72.2	73.8
PointContrast [65]	SR-UNet	70.9	74.1
PointClustering	SR-UNet	**73.2** (+5.0)	**75.5** (+5.2)
Scratch	PointViT	58.9	60.1
Point-MAE [41]	PointViT	60.0	-
MaskSurf [73]	PointViT	61.6	-
PointClustering	PointViT	**65.6** (+6.7)	**65.8** (+5.7)

4.4. Evaluation on Semantic Segmentation

We further evaluate PointClustering on point cloud semantic segmentation task which is to categorize the points in the 3D scenes into different classes. Here, we experiment with model fine-tuning on the S3DIS and ScanNetV2 datasets based on the standard settings [9, 65]. The mIoU performances across different approaches are summarized in Table 5. PointClustering shows substantial performance boosts (4.7%∼6.7%) compared to scratch training. The results verify that PointClustering benefits from instance level

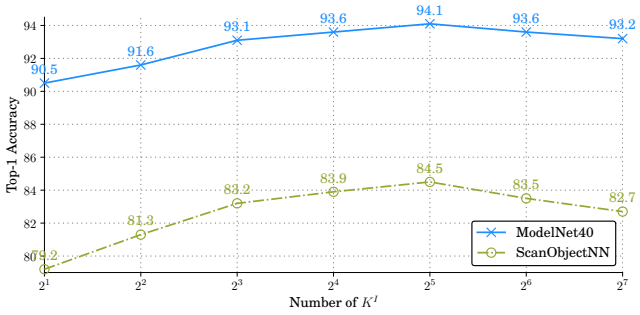

Figure 5. Performances of PointClustering on ModelNet40 and ScanObjectNN by varying instance clustering number K^I.

Table 6. Performance contribution of each kind of invariance learning in PointClustering on ModelNet40 and ScanObjectNN.

Model		ModelNet40	ScanObjectNN
point-level inv.	instance-level inv.		
Scratch		90.7	77.9
SceneClustering		91.0	78.1
PointClustering$^-$		91.5	80.1
✓		93.0	82.6
	✓	93.4	83.1
✓	✓	**94.1**	**84.5**

invariance learning to leverage semantics in point cloud data and naturally endows the networks more power for semantic segmentation. Notably, with SR-UNet backbone, PointClustering achieves the highest 75.5% mIoU on ScanNetV2 and outperforms the recent deliberately designed CNN based architecture BPNet [24] by 0.6%.

4.5. Analysis of PointClustering

In this section, we perform a series of studies to delve into the point cloud representation learning of our proposed PointClustering. Note that all the experiments here are conducted with the backbone of PointNet++.

Invariance Learning. We first investigate how each kind of invariance learning in our PointClustering influences the model generalization ability. Table 6 summarizes the top-1 accuracy of different variants of PointClustering by fine-tuning on ModelNet40 and ScanObjectNN datasets. SceneClustering is the run that measures clustering on global scene level features, and PointClustering$^-$ is a degraded version of PointClustering without using any invariance learning. SceneClustering obtains comparable performances with scratch training. This somewhat reveals the weakness of SceneClustering, where directly applying deep clustering on 3D scenes will not lead to apparent improvement when not taking the inherent geometry of point cloud into account. Through deriving the spatial density of point cloud from DBSCAN, PointClustering$^-$ is superior to SceneClustering. By further considering point level invariance learning in clustering, the performance is increased from 91.5% to 93.0% on ModelNet40. Similarly, solely learning clusters with the regularization of instance level feature invariance improves the accuracy to 93.4%. PointClustering by simultaneously leveraging the invariance from both perspectives to explore geometric properties and semantics finally reaches 94.1% and 84.5% on the two datasets, respectively.

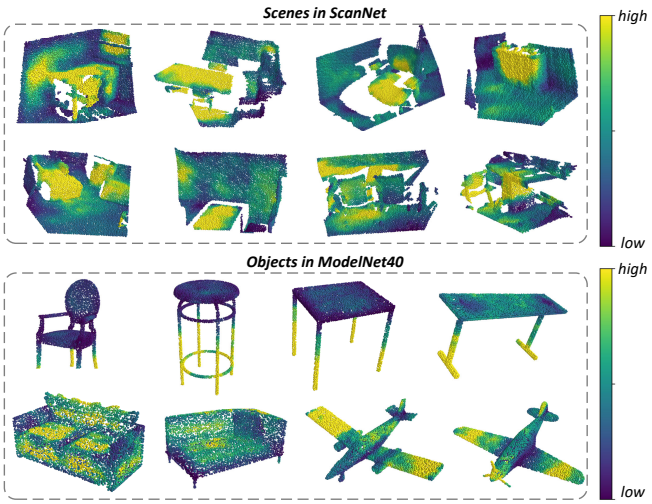

Figure 6. Visualization of point features of 3D scenes in ScanNet and objects in ModelNet40. We colour the points based on the channel activation value of the features learnt by PointClustering.

Instance Clustering Number K^I. Next, we study the impact of K^I for the feature clustering at instance level. The top-1 accuracies of the fine-tuned models on ModelNet40 and ScanObjectNN datasets are reported. We vary K^I on a logarithmic scale and Figure 5 shows the performances. On both of the two datasets, the best performances are obtained when $K^I = 32$. Given the fact that the backbone is pre-trained by PointClustering on the ScanNet dataset which contains 20 object categories, it is expected that setting K^I to around 20 would yield the best performances, but seemingly some amount for over segmentation is potentially more helpful.

Point-level Feature Discrimination. The point level invariance learning of our PointClustering aims for grouping point features of an identical object together across different data transformations. Such objective could be interpreted as two learning dimensions. One is to gather point features of one object in the scene to enhance the capability for instance discrimination. The other is the feature consistency learning on the same points across different transformations, facilitating the model to characterize geometric properties of objects. As a result, PointClustering is expected to be able to distinguish points of different instances in the scene or different geometric parts of the object. To verify our claim, Figure 6 visualizes the point features of 3D scenes from ScanNet and objects from ModelNet40. We colour the points based on the channel activation value of the features learnt by PointClustering. As depicted in the figure, PointClustering nicely differentiates the instances (e.g., chairs) in one

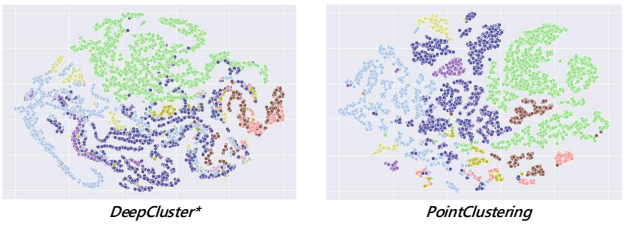

Figure 7. Visualization of the learnt point feature distribution on ScanNetV2. We depict the t-SNE [58] visualization of point features learnt by DeepCluster* and PointClustering.

Table 7. Evaluation on unsupervised point cloud semantic segmentation. Performances on the ScanNetV2 dataset are reported.

Approach	Accuracy	mIoU
supervised model	66.3	57.9
K-means	10.6	5.7
DeepCluster*	19.8	10.1
PointClustering (w/o tuning)	**42.2**	**18.3**

scene and some specific parts (e.g., the legs of the chair) of one object. For the extreme cases of objects which do not appear in the pre-training ScanNet dataset, such as airplane, PointClustering also describes the patterns of wings well. The results again confirm the impact of exploring geometric properties of point cloud in clustering to obtain good point level feature discrimination.

Instance-level Semantic Exploration. The invariance learning at instance level in our PointClustering concentrates on semantic exploration for unsupervised model pre-training. The goal is to cluster the features of instances which share similar semantics together. By aligning the learnt clusters with object labels, we additionally probe and examine the learnt instance level semantics on the unsupervised point cloud semantic segmentation task. Empirically, we experiment with PointClustering on ScanNetV2 dataset.

Following the unsupervised image semantic segmentation protocol [14], we adopt Hungarian matching algorithm to align the clusters on instance features with the ground-truth object categories. Note that we choose the instance-level clustering number $K^I = 20$ to match the number of semantic labels in this setting. Two more runs of K-means and DeepCluster* are devised and included for comparisons. K-means directly clusters point cloud coordinates into 20 categories and DeepCluster* is a variant of DeepCluster [5] by learning clusters on point level features of 3D scenes. Table 7 compares the accuracy and mean IoU performances on the ScanNetV2 dataset. Note that because K-means solely exploits coordinates for clustering and completely disregards the semantics of point cloud data, it is not surprising that K-means performs the worst. Instead, DeepCluster* probes into the semantic distributions of points by using deep models for point clustering and achieves 10.1% mIoU. Our PointClustering also derives the spirit of deep clustering, but further regulates the training procedure with transformation invariance, enhancing the learning of high-level semantics. The results verify the effectiveness of our PointClustering, as evidenced by a performance boost of mIoU from 10.1% to 18.3%. We visualize the distribution of the point features learnt by DeepCluster* and PointClustering in Figure 7 through t-SNE [58]. Compared to DeepCluster*, PointClustering apparently separates the point features from different categories better. Figure 8 also showcases four unsupervised semantic segmentation results from ScanNetV2 by the two approaches. As illustrated in the figure, PointClustering successfully segments the major objects (e.g., floor) and performs well on splitting several small objects (e.g., chair) in 3D scenes, validating the exploration of rich semantics by instance level clustering.

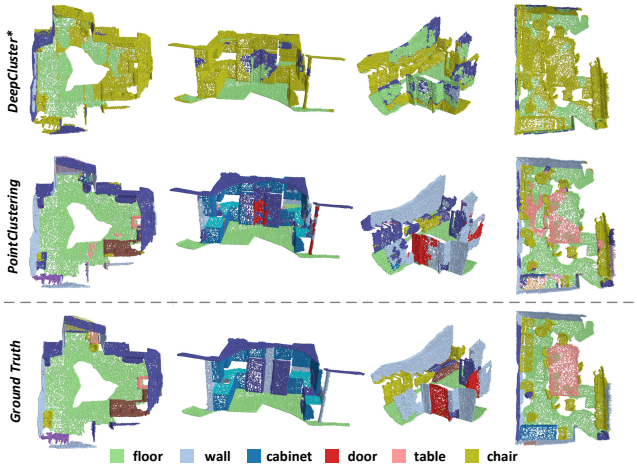

Figure 8. Four visual examples of unsupervised semantic segmentation by DeepCluster* and our PointClustering on ScanNetV2.

5. Conclusions and Discussions

This paper explores deep clustering for unsupervised point cloud pre-training. Particularly, we study the problem from a novel viewpoint of leveraging feature invariance under different data transformations as an inductive bias for clustering learning. To materialize our idea, we have devised PointClustering, which characterizes the geometric properties and semantics of point cloud data by considering feature invariance learning from two perspectives: point level and instance level. The point-level features of an identical object across different transformations are expected to group together, and the feature consistency at instance level is further maintained during clustering optimization. Experiments conducted on six datasets over three downstream tasks demonstrate the superiority of PointClustering. Furthermore, our work indicates that clustering is potentially a new paradigm for unsupervised point cloud pre-training.

Broader Impact. The unsupervised model pre-training scheme of our work can require storage of huge datasets or energy-consuming training of large models. Associated resources can have a negative environmental impact.

References

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. CrossPoint: Self-Supervised Cross-Modal Contrastive Learning for 3D Point Cloud Understanding. In *CVPR*, 2022. 1

[2] Syeda Mariam Ahmed and Chew Chee Meng. Density Based Clustering for 3D Object Detection in Point Clouds. In *CVPR*, 2020. 4

[3] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D Semantic Parsing of Large-Scale Indoor Space. In *CVPR*, 2016. 5

[4] Qi Cai, Yu Wang, Yingwei Pan, Ting Yao, and Tao Mei. Joint Contrastive Learning with Infinite Possibilities. In *NeurIPS*, 2020. 2

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*, 2018. 2, 3, 4, 5, 8

[6] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised Pre-Training of Image Features on Non-Curated Data. In *ICCV*, 2019. 2, 3, 4

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 2020. 2

[8] Xinlei Chen and Kaiming He. Exploring Simple Siamese Representation Learning. In *CVPR*, 2021. 2

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-temporal Convnets: Minkowski Convolutional Neural Networks. In *CVPR*, 2019. 2, 6

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017. 2, 5

[11] Francis Engelmann, Martin Bokenloh, Alireza Fathi, Bastian Leibe, and Matthias Niessner. 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In *CVPR*, 2020. 2

[12] Martin Ester, Hans-Peter Kriegel, Jrg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *SIGKDD*, 1996. 2, 4

[13] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning. In *CVPR*, 2021. 2

[14] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised Semantic Segmentation by Contrasting Object Mask Proposals. In *ICCV*, 2021. 8

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*, 2018. 2

[16] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolution Networks. In *CVPR*, 2018. 2

[17] JunYoung Gwak, Christopher Choy, and Silvio Savarese. Generative Sparse Detection Networks for 3D Single-Shot Object Detection. In *ECCV*, 2020. 2

[18] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Multi-Angle Point Cloud-VAE: Unsupervised Feature Learning for 3D Point Clouds from Multiple Angles by Joint Self-Reconstruction and Half-to-Half Prediction. In *ICCV*, 2019. 2

[19] Kaveh Hassani and Mike Haley. Unsupervised Multi-Task Feature Learning on Point Clouds. In *ICCV*, 2019. 2

[20] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick. Masked Autoencoders are Scalable Vision Learners. In *CVPR*, 2022. 2

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*, 2020. 2

[22] Ji Hou, Angela Dai, and Matthias Niessner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *CVPR*, 2019. 2

[23] Ji Hou, Benjamin Graham, Matthias Niessner, and Saining Xie. Exploring Data-Efficient 3D Scene Understanding with Contrastive Scene Contexts. In *CVPR*, 2021. 1, 2, 6

[24] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional Projection Network for Cross Dimension Scene Understanding. In *CVPR*, 2021. 7

[25] Haiyong Jiang, Feilong Yan, Jianfei Cai, Jianmin Zheng, and Jun Xiao. End-to-End 3D Point Cloud Instance Segmentation Without Detection. In *CVPR*, 2020. 2

[26] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In *CVPR*, 2020. 2

[27] Jeff Johnson, Matthijs Douze, and Herve Jegou. Billion-Scale Similarity Search with GPUs. *IEEE Trans. on Big Data*, 2019. 5

[28] Seohyun Kim, Jaeyoo Park, and Bohyung Han. Rotation-Invariant Local-to-Global Representation Learning for 3D Point Cloud. In *NeurIPS*, 2020. 3

[29] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified Transformer for 3D Point Cloud Segmentation. In *CVPR*, 2022. 2

[30] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a Proxy Task for Visual Understanding. In *CVPR*, 2017. 2

[31] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical Contrastive Learning of Unsupervised Representations. In *ICLR*, 2021. 2

[32] Rui Li, Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Motion-Focused Contrastive Learning of Video Representations. In *ICCV*, 2021. 2

[33] Xianzhi Li, Ruihui Li, Guangyong Chen, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. A Rotation-invariant Framework for Deep Point Cloud Analysis. *IEEE Trans. on Visualization and Computer Graphics*, 2021. 3

[34] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked Discrimination for Self-Supervised Learning on Point Clouds. In *ECCV*, 2022. 1, 2, 5, 6

[35] Fuchen Long, Zhaofan Qiu, Yingwei Pan, Ting Yao, Jiebo Luo, and Tao Mei. Stand-Alone Inter-Frame Attention in Video Models. In *CVPR*, 2022. 2

[36] Fuchen Long, Zhaofan Qiu, Yingwei Pan, Ting Yao, Chong-Wah Ngo, and Tao Mei. Dynamic Temporal Filtering in Video Models. In *ECCV*, 2022. 2

[37] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Bi-Calibration Networks for Weakly-Supervised Video Representation Learning. *arXiv preprint arXiv:2206.10491*, 2022. 2

[38] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised Learning of Long-Term Motion Dynamics for Videos. In *CVPR*, 2017. 2

[39] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. In *ICLR*, 2022. 2

[40] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A Large-scale Benchmark for Fine-grained and Hierarchical Part-level 3D Object Understanding. In *CVPR*, 2019. 5, 6

[41] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked Autoencoders for Point Cloud Self-supervised Learning. In *ECCV*, 2022. 5, 6

[42] Charles R. Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. ImVoteNet: Boosting 3D Object Detection in Point Clouds with Image Votes. In *CVPR*, 2020. 2

[43] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, 2017. 2

[44] Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NIPS*, 2017. 2, 5, 6

[45] Zhaofan Qiu, Yehao Li, Yu Wang, Yingwei Pan, Ting Yao, and Tao Mei. SPE-Net: Boosting Point Cloud Analysis via Rotation Robustness Enhancement. In *ECCV*, 2022. 2

[46] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, and Tao Mei. MLP-3D: A MLP-like 3D Architecture with Grouped Time Mixing. In *CVPR*, 2022. 2

[47] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xiao-Ping Zhang, Dong Wu, and Tao Mei. Boosting Video Representation Learning with Multi-Faceted Integration. In *CVPR*, 2021. 2

[48] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface Representation for Point Clouds. In *CVPR*, 2022. 2

[49] Yongming Rao, Benlin Liu, Yi Wei, Jiwen Lu, Cho-Jui Hsieh, and Jie Zhou. RandomRooms: Unsupervised Pre-training from Synthetic Shapes and Randomized Layouts for 3D Object Detection. In *ICCV*, 2021. 2

[50] Yongming Rao, Jiwen Lu, and Jie Zhou. Global-Local Bidirectional Reasoning for Unsupervised Representation Learning of 3D Point Clouds. In *CVPR*, 2020. 2, 5

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, 2015. 5

[52] Jonathan Sauder and Bjarne Sievers. Self-Supervised Deep Learning on Point Clouds by Reconstructing Space. In *NeurIPS*, 2019. 1, 2, 5, 6

[53] David Sculley. Web-Scale K-Means Clustering. In *WWW*, 2010. 5

[54] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Segmentation of 3D Point Clouds. In *3DV*, 2017. 2

[55] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. In *NeurIPS*, 2022. 2

[56] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In *ICCV*, 2019. 5, 6

[57] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. In *NeurIPS*, 2018. 2, 3

[58] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *JMLR*, 2008. 8

[59] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking Emerges by Colorizing Videos. In *ECCV*, 2018. 2

[60] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J. Kusner. Unsupervised Point Cloud Pre-training via Occlusion Completion. In *ICCV*, 2021. 1, 2, 5, 6

[61] Xiaolong Wang and Abhinav Gupta. Unsupervised Learning of Visual Representations using Videos. In *ICCV*, 2015. 2

[62] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shape Modeling. In *CVPR*, 2015. 5

[63] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised Feature Learning via Non-parametric Instance Discrimination. In *CVPR*, 2018. 2

[64] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised Deep Embedding for Clustering Analysis. In *ICML*, 2016. 2

[65] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas J Guibas, and Or Litany. PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding. In *ECCV*, 2020. 1, 2, 5, 6

[66] Jianyun Xu, Xin Tang, Yushi Zhu, Jie Sun, and Shiliang Pu. SGMNet: Learning Rotation-Invariant Point Cloud Representations via Sorted Gram Matrix. In *ICCV*, 2021. 3

[67] Juyoung Yang, Pyunghwan Ahn, Doyeon Kim, Haeil Lee, and Junmo Kim. Progressive Seed Generation Auto-encoder for Unsupervised Point Cloud Learning. In *ICCV*, 2021. 2

[68] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring Sequence Supervision for Unsupervised Representation Learning. In *AAAI*, 2021. 2

[69] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyuan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A Scalable Active Framework for Region Annotation in 3D Shape Collections. *ACM Trans. on Graphics*, 2016. 5

[70] Xumin Yu, LuLu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-training 3D Point Cloud Transformers with Masked Point Modeling. In *CVPR*, 2022. 5, 6

[71] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Online Deep Clustering for Unsupervised Representation Learning. In *CVPR*, 2020. 2

[72] Richard Zhang, Phillip Isola, and Alexei A. Efros. Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction. In *CVPR*, 2017. 2

[73] Yabin Zhang, Jiehong Lin, Chenhang He, Yongwei Chen, Kui Jia, and Lei Zhang. Masked Surfel Prediction for Self-Supervised Point Cloud Learning. *arXiv preprint arXiv:2207.03111*, 2022. 5, 6

[74] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-Supervised Pretraining of 3D Features on any Point-Cloud. In *ICCV*, 2021. 1, 2, 5, 6

[75] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point Transformer. In *ICCV*, 2021. 2