Unsupervised Learning of Geometric Sampling Invariant Representations for 3D Point Clouds

Haolan Chen, Shitong Luo, Xiang Gao and Wei Hu*
Wangxuan Institute of Computer Technology
Peking University
{chenhl99, luost, gyshgx88, forhuwei}@pku.edu.cn

Abstract

Point clouds consist of a discrete set of points irregularly sampled from continuous 3D objects. Most existing approaches for point cloud learning are in (semi)-supervised fashions, which nevertheless require costly human annotations. To this end, we propose a novel unsupervised learning of geometric sampling invariant representations, aiming to learn intrinsic feature representations of point clouds on graphs based on that the geometry of one object can be sampled in various patterns and densities into different forms of point clouds. In particular, we represent point clouds on graphs and exploit invariant representations at multiple hierarchies: the low-resolution invariance and original-resolution invariance. To learn invariance at a lower resolution, we subsample the input point cloud in distinct patterns, and maximize the mutual information among the subsampled variants. Further, to learn invariance at the original resolution, we increase the resolution of the subsampled point clouds to the original resolution of the input based on the learned features, and minimize the distance between the input and each of the upsampled versions. In experiments, we apply the learned representations to representative downstream tasks of point clouds, and results on point cloud classification, segmentation and upsampling demonstrate the superiority of the proposed model.

1. Introduction

3D geometric data such as point clouds provides a natural representation for the 3D world, which is crucial to a variety of applications such as autonomous driving, robotics and augmented/virtual reality. Recent advances in neural networks tailored for irregular point clouds have achieved great success in point cloud analysis [36, 37, 56, 52, 27, 57, 4]. However, these methods are mostly trained in a (semi-)supervised fashion, which requires costly human annotations. This limits the wide applicability of point clouds, especially for large-scale data. Hence, it is in demand to learn feature representations of point clouds in an unsupervised fashion.

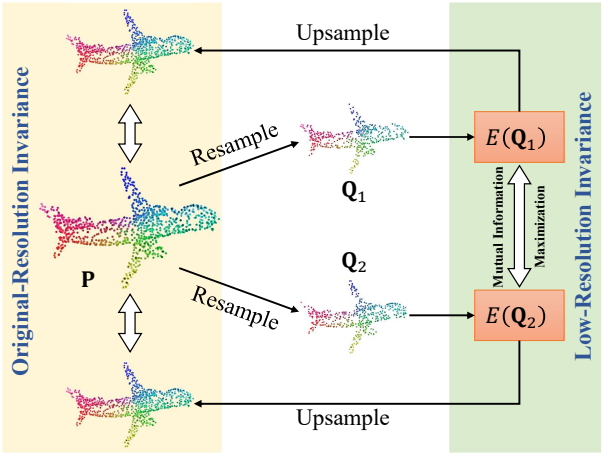


Figure 1. **Illustration of the proposed unsupervised learning of geometric sampling invariant representations for point clouds.** We exploit such invariant representations under sampling at multiple hierarchies: the *low-resolution* invariance and the *original-resolution* invariance.

The main purpose of unsupervised point cloud learning is to learn discriminative and generic representations. Several attempts have been made for unsupervised representation learning on point clouds. These approaches are mainly based on reconstruction [10, 60, 28, 17, 65] or generation [1, 51, 14, 24, 48]. The former aims to train an encoder to learn feature representations by reconstructing the input data via a decoder, while the latter attempts to learn feature representations by training Generative Adversarial Networks (GANs) [15] or Variational Auto-Encoders (VAEs) [22] to generate 3D point clouds. These methods have demonstrated to be effective in capturing structural and low-

*Corresponding author: Wei Hu (forhuwei@pku.edu.cn) This work was supported by National Natural Science Foundation of China (61972009).

level information of point clouds, but usually fail to learn high-level semantic information.

Recently, many approaches have sought to explore semantic information for point cloud learning. Zhang *et al.* [63] propose to learn unsupervised semantic features for point clouds by solving the pretext tasks of part contrasting and object clustering. Sauder *et al.* [45] attempt to reconstruct point clouds whose parts have been randomly rearranged so as to capture semantic properties of the point cloud. These representations are learned by inferring the relationship among parts, while the global information is not fully exploited.

To this end, we propose unsupervised point cloud learning based on *Geometric Sampling Invariant Representations*, aiming to characterize both local and global geometric and semantic information at various scales. This takes inspiration from the principle that the continuous surface of one 3D object can be sampled in various patterns and densities into different forms of point clouds, *i.e.*, these distinctly sampled point clouds describe the underlying structure of the *same* 3D object and thus share the *invariant* representation. Hence, we propose a novel unsupervised learning of geometric sampling invariant representations, which learns the intrinsic features of point clouds. In particular, as demonstrated in Figure 1, we exploit such invariant representations at multiple hierarchies: the low-resolution invariance and original-resolution invariance, so as to capture the geometric structures at various scales.

To learn invariance at a lower resolution, we subsample the input point cloud in distinct patterns randomly, encode the features of these subsampled variants over graphs respectively via a Siamese graph-based encoder network with shared weights [56], and maximize the mutual information among them so as to ensure the invariance. Meanwhile, to learn more discriminative representations, we enforce features of point clouds subsampled from different objects to be distinguishable. We achieve the mutual information maximization and discriminative representations among different objects by optimizing the InfoNCE loss [34], which approaches the lower bound of mutual information.

Further, to learn invariance at the original resolution, we upsample the subsampled point clouds to the original resolution of the input based on the learned features via the patch manifold reconstruction method in [31], and minimize the distance between the input and each of the upsampled versions to learn the invariance. By jointly optimizing the invariant representation learning at both resolutions, the learned representations convey structural and semantic information at various scales.

To evaluate the proposed model, we apply the learned representation to representative downstream tasks of point clouds, including point cloud segmentation, point cloud classification and point cloud upsampling over several benchmark datasets. Experimental results show that we achieve the state-of-the-art performance in unsupervised point cloud classification and segmentation, and improve the upsampling performance by pre-training with our model.

In summary, our main contributions include

- We propose a novel unsupervised learning paradigm of geometric sampling invariant representations, aiming to learn intrinsic features of point clouds that are invariant under various sampling patterns and densities.
- We explore geometric sampling invariant representations at multiple hierarchies: the low-resolution invariance and original-resolution invariance, which captures geometric structures and semantic information at various scales.
- Experimental results demonstrate the superiority of our model over several representative point cloud downstream tasks.

2. Related Work

2.1. Deep Learning on Point Clouds

Deep learning on 3D point clouds has attracted increasing attention in recent years. Unlike 2D images, 3D point clouds are irregularly sampled and have some special properties such as permutation invariance. Previous works [66, 54] try to apply 3D convolutions by dividing point clouds into regular voxels or 3D grids. However, these methods suffer from information loss due to the approximation. Recently, various techniques [36, 37, 16, 27, 57, 29, 49, 64, 55, 56] have been designed to directly consume the unordered point clouds.

One pioneer method PointNet [36] proposes to learn the features of each point independently, while PointNet++ [37] introduces a hierarchical architecture that applies PointNet on a nested partitioning of the input point set to extract local structures. Local structures have also been exploited by methods such as PCPNet [16], PointCNN [27], PointConv [57], and Relation-Shape CNN [29] to further improve the quality of point cloud representation learning. In addition, Graph Convolutional Neural Networks (GCNNs) have also been applied to point clouds by constructing a K-nearest-neighbor (KNN) graph or complete graph to learn feature representations [49, 64, 55, 56]. Among them, Dynamic Graph Convolutional Neural Network (DGCNN) [56] proposes to construct graphs in the feature space and dynamically update them at each layer of the network. Edge Convolution is also proposed to aggregate features from the neighborhood for each node.

2.2. Self-supervised Representation Learning

Self-supervised representation learning has attracted increasing attention since the cost of human annotations is quite expensive. Auto-Encoders (AEs) [22] and Generative Adversarial Networks (GANs) [15] are two representative unsupervised approaches. AEs aim to train an encoder to learn feature representations by reconstructing the input data via a decoder. The idea is based on that feature representations should contain sufficient information to reconstruct the input data. FoldingNet [60] proposes a novel folding-based decoder that deforms a canonical 2D grid onto the underlying surface of a point cloud, achieving low reconstruction errors even for objects with delicate structures. [3, 62, 26, 38] employ point cloud upsampling or reconstruction as their pretext tasks to learn the representations of point clouds. MAP-VAE [17] introduces an innovative multi-angle analysis to effectively learn the local geometry and structure on point clouds from semantic local self-supervision. PointOE [35] proposed to rotate a point cloud with various angles and predict rotation angles to mine the intrinsic features. In contrast, GANs extract feature representations in an unsupervised fashion by generating data from input noises via a pair of generator and discriminator. LGAN [1] introduces the first deep generative models for 3D point clouds. RL-GAN-Net [44] presents a point cloud generation model that is robust to low-availability data and requires no prior knowledge.

In addition to AEs and GANs, another important paradigm called contrastive learning aims to train an encoder to be *contrastive* between the representations of positive samples and negative samples [19, 8, 7, 59, 43]. PointGLR [39] learns point cloud representations by the bidirectional reasoning between global and local features. Deep Infomax (DIM) [20] also proposes to learn image feature representations by maximizing the mutual information between local patches and the corresponding global representation through a contrastive learning task. AMDIM [5] enhances the positive association between a local feature and its context by randomly sampling two different views of an image to generate the local feature vector and context vector. Deep Graph Infomax [53] and InfoGraph [47] extend the framework of DIM to non-Euclidean data.

3. Method

We first formulate the unsupervised learning problem of geometric sampling invariant representations in Sec. 3.1, and then present an algorithm to implement the formulation in Sec. 3.2.

3.1. The Formulation

Point clouds are discrete samples of functions on Riemannian manifolds (surfaces), which represent the geometry of objects [21]. Geometric properties are invariant under sampling, *i.e.*, point clouds sampled in various patterns and densities from the same manifold correspond to the same shape, which share the invariant representations.

Formally, given a point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$ with N points, we consider two sampling operators $\mathcal{S}_1$ and $\mathcal{S}_2$ for simplicity. A function $E(\cdot)$ is *geometric sampling invariant* if it satisfies

$$E(\mathcal{S}_1\mathbf{P}) = E(\mathcal{S}_2\mathbf{P}) = E(\mathbf{P}), \tag{1}$$

which can be extended to more sampled point clouds.

Our goal is to learn a function $E : \mathbf{P} \mapsto E(\mathbf{P})$, which extracts invariant feature representations of $\mathbf{P}$. In particular, in order to reveal geometric structures at various scales, we learn geometric sampling invariant representations at *multiple resolutions*: invariance at a low resolution and invariance at the original resolution. That is, we ensure the representations to be invariant between 1) subsampled point clouds $\mathbf{Q}_1 = \mathcal{S}_1\mathbf{P}$ and $\mathbf{Q}_2 = \mathcal{S}_2\mathbf{P}$ at a low resolution, where $\mathcal{S}_1$ and $\mathcal{S}_1$ are downsampling operators; 2) the input point cloud $\mathbf{P}$ and each of the upsampled point clouds from $\mathbf{Q}_1$ and $\mathbf{Q}_2$ respectively at the original high resolution. We formulate the two hierarchies of representation invariance respectively as follows.

Invariance at a Low Resolution. To ensure the geometric sampling invariance of representations at a low resolution, *i.e.*, $E(\mathcal{S}_1\mathbf{P}) = E(\mathcal{S}_2\mathbf{P})$, we propose to maximize the mutual information between learned features of $\mathbf{Q}_1 = \mathcal{S}_1\mathbf{P}$ and $\mathbf{Q}_2 = \mathcal{S}_2\mathbf{P}$ that correspond to the same object:

$$\max_{E} \; I(E(\mathbf{Q}_1), E(\mathbf{Q}_2)), \tag{2}$$

where $I(\cdot)$ denotes the mutual information.

Since the mutual information between learned features is difficult to compute directly, we instead maximize the lower bound of the mutual information. Among existing methods [6, 12, 41] that are able to approach the lower bound of mutual information, we choose the InfoNCE loss [32, 34]. The reason is that the InfoNCE does not require an additional network to approximate the mutual information as in other methods, which avoids complicated training procedures. Also, minimizing the InfoNCE loss has been proven to be equivalent to maximizing the mutual information [34].

Specifically, apart from maximizing the mutual information between subsampled point clouds that correspond to the same object (positive samples), we also attempt to maximize the discrepancy among point clouds corresponding to different objects (negative samples), leading to discriminative feature learning. Mathematically, given a set of point clouds $\mathcal{P} = \{\mathbf{P}_1, ..., \mathbf{P}_M\}$ with M random samples containing one positive sample and $M-1$ negative samples, the InfoNCE loss aims to minimize

$$\mathcal{L}_{\mathrm{N}} = -\mathop{\mathbb{E}}_{\mathbf{P}_i\in\mathcal{P}}\left[\log \frac{E(\mathcal{S}_1\mathbf{P}_i)\odot E(\mathcal{S}_2\mathbf{P}_i)}{\Sigma_{\mathbf{P}_k\in\mathcal{P}} E(\mathcal{S}_1\mathbf{P}_k)\odot E(\mathcal{S}_2\mathbf{P}_i)}\right], \tag{3}$$

where $\mathbf{P}_i$ is the i-th positive point cloud sample, while $\mathbf{P}_k$ is the k-th point cloud in a mini batch $\mathcal{P}$. $\odot$ denotes the Hadamard product between two feature vectors. Here, we adopt the Hadamard product to measure the similarity between features as it is beneficial to discriminate samples from different point clouds.

Invariance at the Original Resolution. In addition to exploiting invariant representation learning at a low resolution, we also explore geometric invariance at the original resolution. Compared to subsampled point clouds, original-resolution point clouds often contain more geometric details and semantic information that are invariant under sampling, which is complementary to the low-resolution invariant representation learning. If the low-resolution data shares invariant features with the original-resolution point cloud, we are able to reconstruct the point cloud using features learned from low-resolution point clouds.

Hence, in order to ensure that features learned at low resolution contains geometric details in the original point cloud, we first upsample the low-resolution point clouds $\mathbf{Q}_1$ and $\mathbf{Q}_2$ to $\hat{\mathbf{P}}_1$ and $\hat{\mathbf{P}}_2$ respectively with the same resolution as $\mathbf{P}$ using the learned features. Then, we minimize the difference between $\mathbf{P}$ and each of the upsampled point clouds $\hat{\mathbf{P}}_1$ and $\hat{\mathbf{P}}_2$ respectively by the Chamfer Distance $\mathcal{L}_{\text{CD}}$ [11]. Taking $\hat{\mathbf{P}}_1$ as an example, the Chamfer Distance is defined as

$$\mathcal{L}_{\text{CD}}(\hat{\mathbf{P}}_1, \mathbf{P}) = \frac{1}{\left|\hat{\mathbf{P}}_1\right|} \sum_{\mathbf{p}\in\hat{\mathbf{P}}_1} \min_{\mathbf{q}\in\mathbf{P}} \|\mathbf{p}-\mathbf{q}\|_2^2 + \frac{1}{|\mathbf{P}|} \sum_{\mathbf{q}\in\mathbf{P}} \min_{\mathbf{p}\in\hat{\mathbf{P}}_1} \|\mathbf{q}-\mathbf{p}\|_2^2. \tag{4}$$

The same can be defined for $\mathcal{L}_{\text{CD}}(\hat{\mathbf{P}}_2, \mathbf{P})$. By averaging $\mathcal{L}_{\text{CD}}(\hat{\mathbf{P}}_1, \mathbf{P})$ and $\mathcal{L}_{\text{CD}}(\hat{\mathbf{P}}_2, \mathbf{P})$, we acquire the loss function for invariance at the original resolution.

Final Formulation. Taking both the InfoNCE loss between subsampled point clouds and the Chamfer Distance loss between the input and each of the upsampled point clouds into consideration, the entire network is trained end-to-end by minimizing the loss

$$\min_{E} \ \mathcal{L}_{\text{N}} + \alpha \cdot \mathop{\mathbb{E}}_{\hat{\mathbf{P}}} \left[\mathcal{L}_{\text{CD}}(\hat{\mathbf{P}}_i, \mathbf{P})\right], \tag{5}$$

where α is the hyper-parameter that strikes a balance between the invariance at a low resolution and that at the original resolution. The expectation $\mathbb{E}$ is taken over the upsampled point clouds $\hat{\mathbf{P}}$.

We update the parameters in the feature extractor E iteratively by backward propagation of the loss in Eq. (5). Next, we elaborate on the proposed algorithm.

3.2. The Algorithm

Based on the formulation, the proposed framework consists of three modules: point cloud resampling, low-resolution invariant representation learning and original-resolution invariant representation learning. The architecture of the network is demonstrated in Figure 2.

Algorithm 1 Unsupervised Learning of Geometric Sampling Invariance Representations

Input: Point cloud dataset P
Input: Randomly initialized model weights $\mathbf{\Theta}$ of the feature extractor E and model weights $\mathbf{\Phi}$ of the PMR network F
Output: Learned model weights $\mathbf{\Theta}$ of the feature extractor E

for each mini-batch $\mathcal{P} = \{\mathbf{P}_1, ..., \mathbf{P}_M\}$ in P **do**
  Subsample $\mathcal{Q}_1 = \{\mathcal{S}_1\mathbf{P}_1, ..., \mathcal{S}_1\mathbf{P}_M\}$
  Subsample $\mathcal{Q}_2 = \{\mathcal{S}_2\mathbf{P}_1, ..., \mathcal{S}_2\mathbf{P}_M\}$
  Extract the features $E(\mathcal{S}_1\mathbf{P}_i)$ and $E(\mathcal{S}_2\mathbf{P}_i)$, $i = 1, ..., M$
  Calculate the InfoNCE loss in Eq. (3)
  Upsample the point clouds in $\mathcal{Q}_1$ and $\mathcal{Q}_2$ via the PMR
  Calculate the Chamfer Distance loss in Eq. (4)
  Calculate the final loss in Eq. (5)
  Update the model weights $\mathbf{\Theta}$ and $\mathbf{\Phi}$ through backpropagation of the final loss
end for

Point Cloud Resampling. Given an input point cloud $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N\}^\top, \mathbf{p}_i \in \mathbb{R}^3$, we first resample $\mathbf{P}$ into different patterns. In order to explore low-resolution invariant representation learning, we choose to subsample $\mathbf{P}$ into a lower resolution. While there exist a variety of sampling methods, such as uniform sampling, farthest point sampling [33], and local sampling, we randomly sample a subset of points uniformly for simplicity. By performing uniform random subsampling on $\mathbf{P}$ twice respectively, we acquire two sub-clouds $\mathbf{Q}_1$ and $\mathbf{Q}_2$, where $|\mathbf{Q}_1| = |\mathbf{Q}_2| = N/2$.

Low-resolution Invariant Representation Learning. The feature extractor E takes the coordinates of the subsampled point clouds $\mathbf{Q}_1$ and $\mathbf{Q}_2$ as input. We encode point-wise features of irregular sub-clouds $\mathbf{Q}_1$ and $\mathbf{Q}_2$ through a Siamese network with shared weights, where Edge Convolution (EdgeConv) layers in DGCNN [56] are adopted as basic feature extraction blocks. Specifically, given an input sub-cloud $\mathbf{Q} = \{\mathbf{q}_1, ..., \mathbf{q}_{N/2}\}^\top, \mathbf{q}_i \in \mathbb{R}^3$, we first construct a K-nearest-neighbor graph, where each point is connected to its K nearest neighbors in terms of the Euclidean distance. Then, the encoded feature of a point $\mathbf{q}_i$ is

$$\mathbf{f}_i = E(\mathbf{Q})_i = \max_{j\in\mathcal{N}(i)} \text{ReLU}(\theta(\mathbf{q}_j - \mathbf{q}_i) + \phi\mathbf{q}_i), \tag{6}$$

where $j \in \mathcal{N}(i)$ denotes point j is in the neighborhood of point i. θ and ϕ are learnable network parameters.

The edge convolution in Eq. (6) over each node essentially aggregates features from neighboring nodes via edge weights that capture the distance between points. By stacking several edge convolution layers, each node is aggregated with features of neighbors hops away. Further, as the subsampling operation enlarges the distance between neighboring points, edge convolution for the low-resolution point

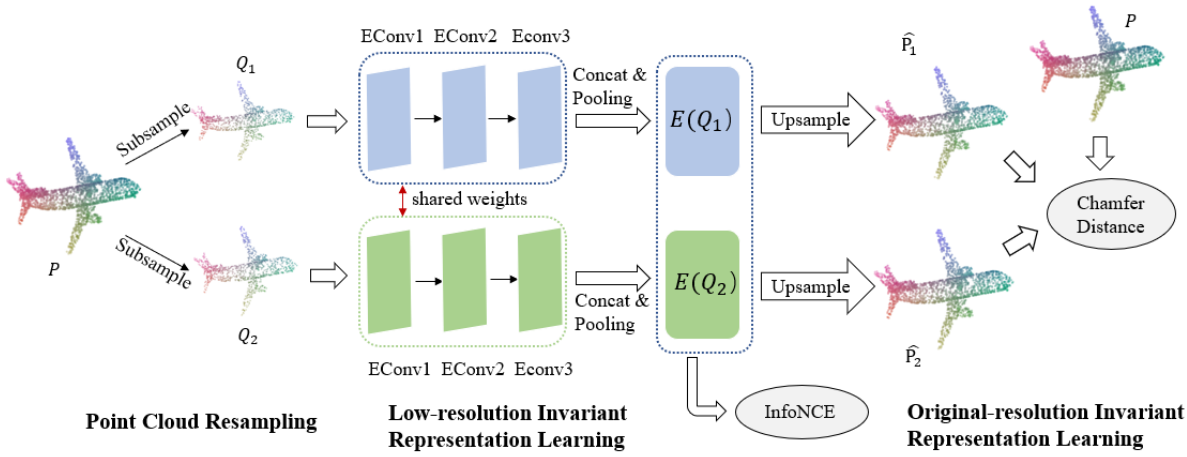


Figure 2. **The architecture of the proposed network.** We subsample the input point cloud into two sub-clouds, and encode them through a Siamese network with shared weights based on edge convolution. The InfoNCE loss is employed to maximize the mutual information between distributions of the global features to perform low-resolution invariant representation learning. We then upsample each sub-cloud to the original resolution, and minimize the distance from the input point cloud for the original-resolution invariant representation learning.

cloud captures longer-range dependencies and thus extracts higher-level features that are invariant under various sampling. Hence, the learned representation is able to capture intrinsic geometric structures at low resolution.

Meanwhile, in order to learn discriminative features for each object, we treat the features extracted from resampled point clouds corresponding to the same object as *positive* samples, while the features of point clouds resampled from different objects as *negative* samples. We ensure features of *positive* pairs to be as similar as possible, which satisfies geometric invariance under various sampling patterns. In contrast, features of the negative pairs are enforced to be as distinctive as possible so as to distinguish among different objects. This is achieved by optimizing the InfoNCE loss in Eq. (3).

Original-resolution Invariant Representation Learning. To ensure the geometric sampling invariant representations at the original resolution, we upsample the subsampled point clouds $\mathbf{Q}_1$ and $\mathbf{Q}_2$ to the resolution of the input point cloud $\mathbf{P}$, and minimize the Chamfer Distance between $\mathbf{P}$ and each of the upsampled point clouds as described in Eq. (4).

Among various point cloud upsampling approaches [60, 62, 31], we choose the upsampling method based on patch manifold reconstruction (PMR) in [31] for its simplicity and effectiveness. Given a subsampled point cloud $\mathbf{Q}$ and its encoded point-wise features $E(\mathbf{Q})$, the idea of PMR-based upsampling is to transform each point $\mathbf{q}_i$ in $\mathbf{Q}$ along with its embedded neighborhood feature $\mathbf{f}_i$ into a local surface centering at $\mathbf{q}_i$—referred to as a *patch manifold*. By sampling r times on these patch manifolds ($r = 2$ in our setting), we are able to acquire the upsampled point cloud.

Specifically, we first construct a patch manifold around each point in $\mathbf{Q}$. A 2D manifold $\mathcal{M}$ embedded in the 3D space parameterized by a feature vector $\mathbf{f}$ is defined as:

$$\mathcal{M}(u, v; \mathbf{f}) : [-1, 1] \times [-1, 1] \rightarrow \mathbb{R}^3, \tag{7}$$

where (u, v) is some point in the 2D rectangular area $[-1, 1]^2$. Eq. (7) maps the 2D rectangle to an arbitrarily shaped patch manifold parameterized by $\mathbf{f}$. Hence, we draw samples from the uniform distribution over $[-1, 1]^2$ and then transform them into the 3D space via the MLP-based mapping:

$$\mathcal{M}_i(u, v; \mathbf{f}_i) = \text{MLP}_{\mathcal{M}}([u, v, \mathbf{f}_i]), \tag{8}$$

which approximates arbitrarily-shaped manifolds.

Then, we sample two points from each patch manifold $\mathcal{M}_i([u, v, \mathbf{f}_i])$, leading to an upsampled point cloud with N points:

$$\hat{\mathbf{P}} = F(\mathbf{Q}) = \begin{pmatrix} q_1 + \text{MLP}_{\mathcal{M}}([u_{11}, v_{11}, \mathbf{f}_1]) \\ q_1 + \text{MLP}_{\mathcal{M}}([u_{12}, v_{12}, \mathbf{f}_1]) \\ \vdots \\ q_M + \text{MLP}_{\mathcal{M}}([u_{M1}, v_{M1}, \mathbf{f}_M]) \\ q_M + \text{MLP}_{\mathcal{M}}([u_{M2}, v_{M2}, \mathbf{f}_M]) \end{pmatrix}, \tag{9}$$

where F denotes the PMR network, $M = |\mathbf{Q}| = N/2$.

Based on this approach, we upsample $\mathbf{Q}_1$ and $\mathbf{Q}_2$ to $\hat{\mathbf{P}}_1$ and $\hat{\mathbf{P}}_2$ respectively, and minimize the Chamfer Distance in Eq. (4) between the input point cloud $\mathbf{P}$ and each of the upsampled point clouds to ensure the original-resolution invariant representation learning.

Method	Year	Unsupervised	Accuracy
PointNet[36]	2017	No	89.2
KD-Net [23]	2017	No	90.6
PointNet++[37]	2017	No	90.7
PointCNN [27]	2018	No	92.2
PCNN [4]	2018	No	92.3
DGCNN [56]	2019	No	92.9
RS-CNN [29]	2019	No	93.6
LGAN [1]	2018	Yes	85.7
MRTNet [14]	2018	Yes	86.4
PCGAN [24]	2018	Yes	87.8
FoldingNet [60]	2018	Yes	88.4
ContrastNet [63]	2019	Yes	86.7
NSampler [40]	2019	Yes	88.7
3D-PointCapsNet [65]	2019	Yes	88.9
Multi-task [18]	2019	Yes	89.1
MAP-VAE [17]	2019	Yes	90.15
PointDist [46]	2020	Yes	84.7
PointGrow [48]	2020	Yes	85.8
ACD [13]	2020	Yes	89.8
PointOE [35] (1.83M para.)	2020	Yes	90.75
Ours (89.28K para.)		Yes	**90.36**

Table 1. Classification accuracy (%) on ModelNet40 dataset.

Finally, we jointly optimize the low-resolution and original-resolution invariant representation learning by minimizing the loss function in Eq. (5) end-to-end. A summary of the proposed algorithm is presented in Algorithm 1.

4. Experiments

In this section, we evaluate the proposed model by applying it to point clouds on three representative downstream tasks: point cloud classification, point cloud segmentation and point cloud upsampling. We compare the proposed approach with the state-of-the-art supervised and unsupervised methods.

Note that, in all the tasks, we set the parameter α in Eq. (5) in the same way. We initialize $\alpha = 1$, and divide it by 2 at the end of each epoch if $\mathcal{L}_{\text{CD}} > \mathcal{L}_{\text{N}}$. This aims to keep $\mathcal{L}_{\text{CD}}$ and $\mathcal{L}_{\text{N}}$ at the same magnitude. During training, We adopt NVIDIA GeForce RTX 2080Ti in our experiments.

4.1. Point Cloud Classification

Dataset We adopt the commonly used **ModelNet40** dataset [58] for point cloud classification. This dataset consists of 12,311 CAD models from 40 categories. The dataset is divided into a training set containing 9,843 models and a testing set containing 2,468 models. We sample 1,024 points for training and testing our model on the classification task.

Implementation Details In this task, We train our feature extractor via the Adam optimizer with a batch size of 32. As for the hyper-parameters of the Adam optimizer, we set the initial learning rate as 0.01, β_1 as 0.9 and β_2 as 0.999. We use the cosine annealing scheduler [30] to decay the learning rate. The setting is the same for the network of learning the original-resolution invariant representations.

Specifically, The feature extractor contains five EdgeConv [56] blocks whose output feature dimensions are 32, 32, 64, 128 and 256, respectively. We concatenate the output features of these layers to acquire a 512-dimensional feature vector for each node. Then, the global max pooling and average pooling layer are deployed to acquire a 1,024-dimensional global feature vector. To evaluate the quality of the extracted features, we train a linear Support Vector Machine (SVM) [9] using the extracted global feature vector to produce predictions. We train the feature extractor on the training set in an unsupervised fashion, whose weights are frozen during training the SVM classifier. After that, we test our method on the testing set for evaluation.

Experimental Results We compare our method with unsupervised and supervised point cloud classification approaches. As shown in Table 1, our method outperforms almost all the unsupervised methods by a large margin, and is also competitive to the state-of-the-art unsupervised method PointOE [35]. However, it is worth noting that our model is much more *lightweight*, with much fewer parameters (89.28K) than PointOE (1.83M). Also, as we adopt DGCNN [56] as the backbone and employ a linear SVM classifier *without any pretraining*, we mainly compare our method with networks using DGCNN as their backbone and testing on ModelNet40 without pretraining for fair comparison. Thus, approaches with a different backbone and multiple tasks [39] or a different dataset [59] and approaches using larger datasets like ShapeNet to pretrain the model [45, 43] are not considered here. Moreover, we compare our method with well-known *supervised* models including PointNet [36], PointNet++ [37] and DGCNN [56]. Though all parameters are optimized under supervision in their models, some of the methods are comparable to our unsupervised model.

Ablation Studies We conduct two ablation studies: 1) invariant representation learning at a single resolution; 2) different resampling strategies. Firstly, we study the effectiveness of the multi-resolution invariant representation learning. We keep one resolution invariance learning phase while removing the other, and compare the results with that of the complete model in Table 2. While the performance at a single low resolution or the original resolution is reasonable, the multi-resolution invariance learning significantly outperforms only learning geometric invariant features at a single resolution. As we discussed in Sec. 3, the original-resolution invariant representation learning helps extract more geometric details and semantic information of the original point cloud, and the low-resolution invariant representation learning is able to maximize the discrepancy among point clouds corresponding to different objects. Both phases are critical to downstream tasks.

Resolution	Low	Original	Low + Original
Accuracy (%)	84.89	84.68	**90.36**

Table 2. Ablation study on learning resolution.

Sampling Method	URS	LS	FPS
Accuracy (%)	**90.36**	88.70	89.63

Table 3. Ablation study on sampling methods. URS: Uniform Random Sampling; LS: Local Sampling; FPS: Farthest Point Sampling.

Secondly, we explore the influence of resampling methods. We choose three different sampling strategies for comparison: 1) Uniform Random Sampling (URS), where the probability of sampling each point in the point cloud follows a uniform distribution; 2) Farthest Point Sampling (FPS), where each point to be sampled is as far away as possible from points in the sampled set; 3) Local Sampling (LS), where the points are sampled from a local part of the point cloud. As presented in Table 3, the classification accuracy is comparable under all the three sampling methods. This shows that the performance of the proposed model is insensitive to the sampling strategies, *i.e.*, employing the simplest sampling method is sufficient to learn discriminative invariant feature representations. This complies with the principle of geometric sampling invariant representations, where various sampling patterns lead to invariant representations of the same 3D object.

4.2. Point Cloud Segmentation

Dataset Point cloud segmentation is a fine-grained task, aiming at predicting the category of each point in a given point cloud. We employ the commonly used **ShapeNet Part** dataset [61] as a benchmark for point cloud segmentation, which consists of 16,881 point clouds from 16 categories. Each point cloud is annotated with fewer than six parts and there are altogether 50 parts among all categories. We employ 12,137 models for training and 2,874 models for testing. In point cloud segmentation, we sample 2,048 points for this task.

Implementation Details We also use Adam optimizer to train the network, and cosine annealing scheduler to decay the learning rate. Hyper-parameters are set all the same as they are in the classification task. The architecture of the network is slightly different from above. Four EdgeConv blocks compose the feature extractor, and their dimensions are all 64. We then train a 4-layer MLP whose dimensions are [1024, 256, 256, 128] following the settings of DGCNN [56] to classify every point.

As in previous works [18, 65, 25], we evaluate our method in a semi-supervised fashion. Firstly, we pre-train the feature extractor unsupervisedly. Then, we randomly sample a tiny fraction of data (1% and 5%) from the training set to fine-tune the model of DGCNN.

Method	Year	1% labeled data IoU	5% labeled data IoU
SO-Net [25]	2018	64.0	69.0
3D-PointCapsNet [65]	2019	67.0	70.0
Multi-task [18]	2019	68.2	77.7
MortonNet [50]	2019	-	77.1
JointSSL [2]	2020	71.9	77.4
Ours		**71.6**	**78.2**

Table 4. Comparison with other semi-supervised segmentation methods on ShapeNet Part dataset. Metric is mIoU (%) on points.

Experimental Results We adopt the commonly used Intersection-over-Union (IoU) as the metric of point cloud segmentation. We follow the same evaluation protocol as in the PointNet [36]: the IoU of a point cloud is calculated by averaging the IoUs of different parts occurring in that point cloud, and the IoU of a category is obtained by averaging the IoUs of all the point clouds belonging to that category. The mean IoU (mIoU) is finally calculated by averaging the IoUs of all the test point clouds.

We compare our method with other semi-supervised methods in Table 4. Results show that our model achieves the state-of-the-art accuracy with 5% labeled data, and achieves competitive performance compared with the state-of-the-art with 1% labeled data. Note that, we didn't compare with PointContrast [59], since training from scratch with their backbone (SR-UNet [42]) already achieves 71.8% with 1% labeled data and 79.3% with 5% labeled data. Also, we didn't compare with [45] as they have no test under 1% or 5% labeled data.

Further, we present the per category comparison with supervised methods and one state-of-the-art semi-supervised method Multi-task [18] with 5% labeled data in Table 5, while the per category accuracy is not reported in other semi-supervised methods listed in Table 4. Results show that our semi-supervised model achieves 78.2% when trained with only 5% of data, which pushes closer toward the fully-supervised methods.

We also visualize our segmentation results in Figure 3. Compared with the state-of-the-art unsupervised point cloud representation learning method MAP-VAE [17], our model is able to distinguish geometric details better, such as the transition between the leg and surface of a chair.

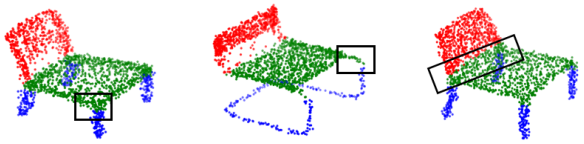
4.3. Point Cloud Upsampling

Point clouds acquired from LiDAR scanners or depth sensors are often sparse, which hinders shape analysis and reconstruction. Point cloud upsampling is thus crucial to the subsequent 3D vision applications. We evaluate the proposed unsupervised model on the point cloud upsampling task over the ShapeNet Part dataset.

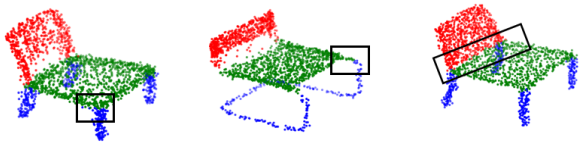
Implementation Details We demonstrate a *pre-training* strategy to evaluate if the unsupervised pre-training with our model helps improve the performance. We still choose the PMR-based upsampling network [31] described in Sec. 3.2

Model	train data	Mean	Aero	Bag	Cap	Car	Chair	Ear Phone	Guitar	Knife	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate Board	Table
Samples			2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
PointNet [36]		83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++ [37]		85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
KD-Net [23]		82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3
PCNN [4]	**100%**	85.1	82.4	80.1	85.5	79.5	90.8	73.2	91.3	86.0	85.0	95.7	73.2	94.8	83.3	51.0	75.0	81.8
PointCNN [27]		86.1	84.1	86.5	86.0	80.8	90.6	79.7	92.3	88.4	85.3	96.1	77.2	95.3	84.2	64.2	80.0	83.0
DGCNN [56]		85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
RS-CNN [29]		86.2	83.5	84.8	88.8	79.6	91.2	81.1	91.6	88.4	86.0	96.0	73.7	94.1	83.4	60.5	77.7	83.6
Multi-task [18]	**5%**	77.7	**78.4**	**67.7**	78.2	**66.2**	85.5	52.6	**87.7**	**81.6**	76.3	93.7	**56.1**	**80.1**	**70.9**	**44.7**	60.7	73.0
Ours	**5%**	**78.2**	76.1	44.8	**79.0**	64.5	**87.5**	**65.1**	86.8	80.7	**76.5**	**94.5**	26.0	71.9	64.6	24.1	**61.2**	**79.4**

Table 5. Segmentation results on ShapeNet part dataset. We adopt both mIoU (%) on points and instance-averaged IoU (%).



(a) MAP-VAE



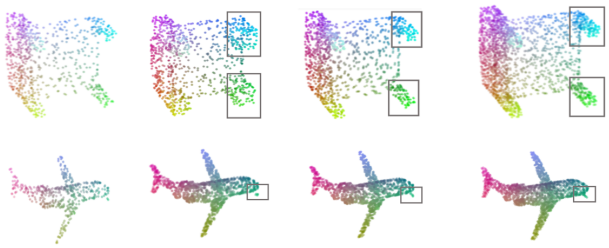
(b) Ours

Figure 3. **Visual comparison of point cloud part segmentation with the state-of-the-art unsupervised method MAP-VAE.** Our method achieves more accurate results in tiny parts like junction.

for evaluation. We first pre-train the feature extractor with our approach in an unsupervised fashion, and then employ the learned representation from the pre-trained feature extractor as an initialization. We evaluate the effectiveness of our model by comparing the results of PMR-based upsampling with our initialization and those with random initialization in a supervised fashion. The architecture of the feature extractor and the settings of hyper-parameters are all the same as in the segmentation task. Hyper-parameters including the learning rate, training epochs and weight decay are identical in training the network with and without pre-training.

We randomly sample 1,024 and 512 points from each point cloud in the ShapeNet Part dataset, and upsample them to 2,048 points respectively, denoted as 2x and 4x. The quality of the upsampled point cloud is measured by the Chamfer Distance between the original and upsampled point clouds.

Experimental Results As shown in Table 6, self-supervised pre-training with our method outperforms the randomly initialized PMR model at both upsampling rates. The gain is much more significant in 4x upsampling which requires more semantic features. Also, our results admit much smaller standard deviation over multiple training,



(a) Sparse Input (b) PMR (c) PMR+Pre-train (d) Ground Truth

Figure 4. **Visualization of the upsampling results at the upsampling rate 4x.** The results with our pre-training preserve geometric details better than those without pre-training.

Method	PMR	PMR (Pre-training)
2x	4.82 ± 0.12	**4.63 ± 0.013**
4x	8.80 ± 0.31	**8.30 ± 0.084**

Table 6. Comparison of point cloud upsampling results with and without pre-training at the upsampling rates 2x and 4x, respectively. The evaluation metric is the Chamfer distance.

which validates the stability of our method.

Further, we visualize the upsampling results from both methods by the upsampling rate 4x in Figure 4. As marked in rectangles, geometric details such as the legs of chairs and tiny parts in airplanes are well preserved by our pre-training method, while those in the random-initialization method are significantly deformed.

5. Conclusion

In this paper, we propose a novel unsupervised learning of geometric sampling invariant representations over graphs, aiming to learn discriminative and generic representations that are invariant under various sampling patterns and densities. To capture geometric structures and semantic information at various scales, we exploit invariant representations at both a low resolution and the original resolution, which enforces the encoder to learn intrinsic representations. We apply the proposed model to downstream tasks of point clouds including classification, segmentation and upsampling, and experimental results demonstrate the superiority of our model.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 1, 3, 6

[2] Antonio Alliegro, Davide Boscaini, and Tatiana Tommasi. Joint supervised and self-supervised learning for 3d real-world challenges. *arXiv preprint arXiv:2004.07392*, 2020. 7

[3] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 3

[4] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, July 2018. 1, 6, 8

[5] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems (NIPS)*, pages 15535–15545, 2019. 3

[6] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540, 2018. 3

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 3

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[9] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 6

[10] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018. 1

[11] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 4

[12] Marylou Gabrié, Andre Manoel, Clément Luneau, Nicolas Macris, Florent Krzakala, Lenka Zdeborová, et al. Entropy and mutual information in models of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 1821–1831, 2018. 3

[13] Matheus Gadelha, Aruni RoyChowdhury, Gopal Sharma, Evangelos Kalogerakis, Liangliang Cao, Erik Learned-Miller, Rui Wang, and Subhransu Maji. Label-efficient learning on point clouds using approximate convex decompositions. In *European Conference on Computer Vision*, pages 473–491. Springer, 2020. 6

[14] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018. 1, 6

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 3

[16] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. PCPnet: learning local shape properties from raw point clouds. In *Computer Graphics Forum*, volume 37, pages 75–85. Wiley Online Library, 2018. 2

[17] 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 *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10441–10450. IEEE, 2019. 1, 3, 6, 7

[18] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8160–8171, 2019. 6, 7, 8

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 3

[20] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 3

[21] Wei Hu, Jiahao Pang, Xianming Liu, Dong Tian, Chia-Wen Lin, and Anthony Vetro. Graph Signal Processing for geometric data and beyond: Theory and applications. *arXiv preprint arXiv:2008.01918*, 2020. 3

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 3

[23] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872, 2017. 6, 8

[24] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018. 1, 6

[25] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406, 2018. 7

[26] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7203–7212, 2019. 3

[27] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018. 1, 2, 6, 8

[28] Xinhai Liu, Zhizhong Han, Xin Wen, Yu-Shen Liu, and Matthias Zwicker. L2g auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 989–997, 2019. 1

[29] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8895–8904, 2019. 2, 6, 8

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[31] Shitong Luo and Wei Hu. Differentiable manifold reconstruction for point cloud denoising. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1330–1338, 2020. 2, 5, 7

[32] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems*, pages 2265–2273, 2013. 3

[33] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003. 4

[34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 3

[35] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028, 2020. 3, 6

[36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2, 6, 7, 8

[37] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1, 2, 6, 8

[38] Yue Qian, Junhui Hou, Sam Kwong, and Ying He. Pugeonet: A geometry-centric network for 3d point cloud upsampling. In *European Conference on Computer Vision*, pages 752–769. Springer, 2020. 3

[39] Yongming Rao, Jiwen Lu, and Jie 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*, pages 5376–5385, 2020. 3, 6

[40] Edoardo Remelli, Pierre Baque, and Pascal Fua. Neuralsampler: Euclidean point cloud auto-encoder and sampler. *arXiv preprint arXiv:1901.09394*, 2019. 6

[41] Gerhard Rigoll. Maximum mutual information neural networks for hybrid connectionist-hmm speech recognition systems. *IEEE Transactions on Speech and Audio Processing*, 2(1):175–184, 1994. 3

[42] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 7

[43] Aditya Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *European Conference on Computer Vision*, pages 626–642. Springer, 2020. 3, 6

[44] Muhammad Sarmad, Hyunjoo Jenny Lee, and Young Min Kim. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5898–5907, 2019. 3

[45] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *Advances in Neural Information Processing Systems*, pages 12962–12972, 2019. 2, 6, 7

[46] Yi Shi, Mengchen Xu, Shuaihang Yuan, and Yi Fang. Unsupervised deep shape descriptor with point distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9353–9362, 2020. 6

[47] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*, 2019. 3

[48] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 61–70, 2020. 1, 6

[49] Gusi Te, Wei Hu, Amin Zheng, and Zongming Guo. RGCNN: regularized graph cnn for point cloud segmentation. In *Proceedings of the 26th ACM International Conference on Multimedia*, pages 746–754, 2018. 2

[50] Ali Thabet, Humam Alwassel, and Bernard Ghanem. Mortonnet: Self-supervised learning of local features in 3d point clouds. *arXiv preprint arXiv:1904.00230*, 2019. 7

[51] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3d point clouds via graph convolution. In *International Conference on Learning Representations (ICLR)*, 2018. 1

[52] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 1

[53] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *ICLR (Poster)*, 2019. 3

[54] Cheng Wang, Ming Cheng, Ferdous Sohel, Mohammed Bennamoun, and Jonathan Li. Normalnet: A voxel-based cnn for 3d object classification and retrieval. *Neurocomputing*, 323:139–147, 2019. 2

[55] Chu Wang, Babak Samari, and Kaleem Siddiqi. Local spectral graph convolution for point set feature learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–66, 2018. 2

[56] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1, 2, 4, 6, 7, 8

[57] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9621–9630, 2019. 1, 2

[58] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. 6

[59] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *arXiv preprint arXiv:2007.10985*, 2020. 3, 6, 7

[60] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018. 1, 3, 5, 6

[61] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, Leonidas Guibas, et al. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)*, 35(6):210, 2016. 7

[62] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2018. 3, 5

[63] Ling Zhang and Zhigang Zhu. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. In *2019 International Conference on 3D Vision (3DV)*, pages 395–404. IEEE, 2019. 2, 6

[64] Yingxue Zhang and Michael Rabbat. A graph-cnn for 3d point cloud classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6279–6283. IEEE, 2018. 2

[65] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1009–1018, 2019. 1, 6, 7

[66] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2