

# Unsupervised 3D Learning for Shape Analysis via Multiresolution Instance Discrimination

PENG-SHUAI WANG, Microsoft Research Asia

YU-QI YANG, Tsinghua University and Microsoft Research Asia

QIAN-FANG ZOU, University of Science and Technology of China and Microsoft Research Asia

ZHIRONG WU, YANG LIU, and XIN TONG, Microsoft Research Asia

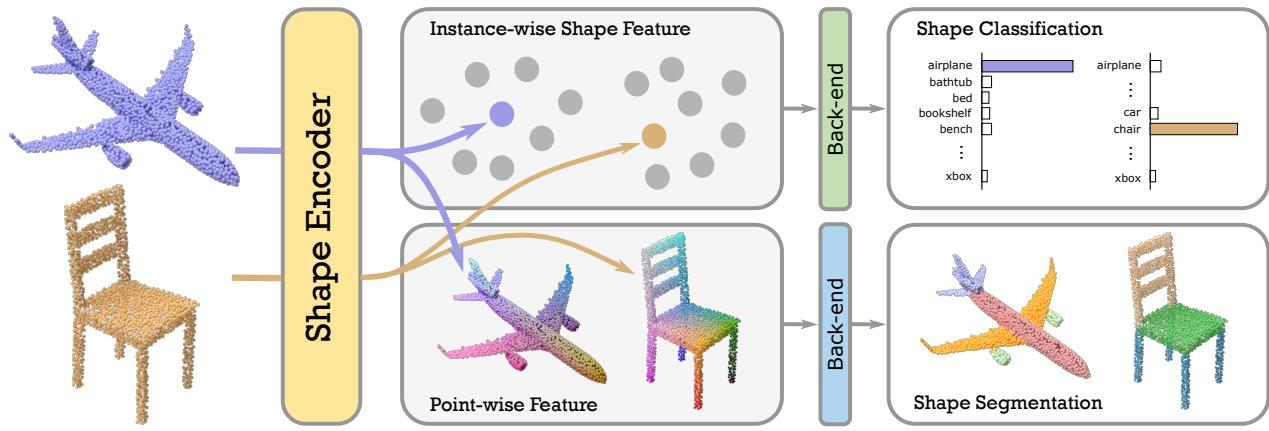


Fig. 1. Overview of our multiresolution-instance-discrimination-based learning approach. Our MID encoder is learned from a collection of unlabeled 3D shapes. It takes 3D point cloud as inputs and generates both instance-wise shape features and point-wise features (shown in the middle), which can be used in different downstream shape analysis tasks, via concatenated simple back-ends with or without fine-tuning.

Although unsupervised feature learning has demonstrated its advantages to reducing the workload of data labeling and network design in many fields, existing unsupervised 3D learning methods still cannot offer a generic network for various shape analysis tasks with competitive performance to supervised methods. In this paper, we propose an unsupervised method for learning a generic and efficient shape encoding network for different shape analysis tasks. The key idea of our method is to jointly encode and learn shape and point features from unlabeled 3D point clouds. For this purpose, we adapt HR-Net to octree-based convolutional neural networks for jointly encoding shape and point features with fused multiresolution subnetworks and design a simple-yet-efficient *Multiresolution Instance Discrimination* (MID) loss for jointly learning the shape and point features. Our network takes a 3D point cloud as input and output both shape and point features. After training, the network is concatenated with simple task-specific back-end layers and fine-tuned for different shape analysis tasks.

We evaluate the efficacy and generality of our method and validate our network and loss design with a set of shape analysis tasks, including shape classification, semantic shape segmentation, as well as shape registration tasks. With simple back-ends, our network demonstrates the best performance among all unsupervised methods and achieves competitive performance to supervised methods, especially in tasks with a small labeled dataset.

Authors' addresses: Peng-Shuai Wang, penwan@microsoft.com, Microsoft Research Asia; Yu-Qi Yang, yangyq18@mails.tsinghua.edu.cn, Tsinghua University, Microsoft Research Asia; Qian-Fang Zou, zou@mail.ustc.edu.cn, University of Science and Technology of China, Microsoft Research Asia; Zhirong Wu, wuzhiron@microsoft.com; Yang Liu, yangliu@microsoft.com; Xin Tong, xtong@microsoft.com, Microsoft Research Asia.

For fine-grained shape segmentation, our method even surpasses existing supervised methods by a large margin.

CCS Concepts: • Computing methodologies → Shape analysis; Neural networks;

Additional Key Words and Phrases: 3D unsupervised learning, multiresolution instance discrimination, 3D shape analysis

## 1 INTRODUCTION

3D shape analysis plays an important role in many graphics and vision applications. A key step in all shape analysis tasks is to extract representative features (or called descriptors) in different levels from 3D shapes. In particular, distinguishable shape instance features are preferred for shape classification, while per-point features are essential to fine-level analysis tasks, like semantic shape segmentation and registration.

Early methods compute handcrafted features of 3D shapes. Although these manually-designed features can preserve some good properties such as transformation invariant, they are difficult to be tailored to specific shape analysis applications. State-of-the-art methods integrate the feature extraction with specific shape analysis task and learn an end-to-end deep neural network with the supervision of labeled shape analysis results. The success of these supervised learning methods is built upon the large-scale labeled dataset, and the networks optimized for one task are difficult to adapt to others.

Unsupervised pre-training methods first learn a feature extraction backbone network from an unlabeled dataset via carefully designed unsupervised pretext task losses. After that, the pre-trained backbone network is concatenated with task-specific back-end networks and refined for different downstream tasks via transfer learning. In computer vision and natural language processing tasks, unsupervised pre-training have demonstrated their advantages for reducing the workload of data labeling and network design [Deng et al. 2018; Hassani and Haley 2019; Wu et al. 2016; Yang et al. 2018; Zhao et al. 2019]. However, these networks and training schemes cannot be easily adapted for 3D shape analysis due to irregular representation of 3D point clouds and multi-level shape features required by different shape analysis tasks. A set of unsupervised 3D learning methods [Deng et al. 2018; Wu et al. 2016; Yang et al. 2018; Zhao et al. 2019] have been proposed for extracting shape features from 3D point clouds, none of them offers a generic backbone network for different shape analysis tasks with competitive performance to the supervised methods.

In this paper, we present an unsupervised pre-training method for learning a generic 3D shape encoding network for 3D shape analysis. Our key observation is that a 3D shape is composed of its local parts and thus the feature for shape and points are coherent and should be encoded and trained jointly. Based on this observation, our shape encoding backbone network adapts HRNet [Wang et al. 2019a] to an octree-based convolutional network [Wang et al. 2017] for extracting and fusing features from both points and shapes via parallel multiresolution subnetworks and connections across subnetworks. It takes 3D point cloud as input and outputs an instance-wise feature of the whole 3D shape as well as point-wise features. Inspired by the instance discrimination designed for 2D image classification [Wu et al. 2018], we design a simple-yet-efficient Multi-resolution Instance Discrimination (MID) losses for supervision of extracted shape and point features, in which a shape instance discrimination loss classifies augmented copies of each shape instance of a 3D dataset in one class, while a point instance discrimination loss classifies the same points on the augmented copies of a shape instance in a class.

A naïve implementation of point instance discrimination is prohibitively expensive due to the huge number of classes for all point instances in a 3D dataset. To resolve this issue, we first exploit the shape instance discrimination loss to classify the point of each shape into a class and then apply the point instance discrimination to classify the points on each shape separately. To further reduce the computational cost and memory footprint of point discrimination loss, we exploit the geometric similarity of nearby points and cluster local points on a shape as one class.

We train our backbone shape encoding network (denoted as MID-Net) with ShapeNetCore55 [Chang et al. 2015] and evaluate its performance with simple back-ends in various shape analysis tasks, including shape classification, two shape segmentation tasks, and 3D shape registration. Our experiments demonstrate that in all these tasks, our pre-trained backbone offers better performance than the same network trained with the labeled data of downstream tasks, especially as the amount of labeled data in the downstream tasks becomes small. Among all unsupervised 3D learning methods, our method achieves the best performance in all shape analysis tasks.

Moreover, it achieves a competitive performance to the state-of-the-art supervised methods in all tasks. In fine-grained PartNet segmentation, our method surpasses state-of-the-art supervised methods by a large margin. We also conduct a detailed ablation study to validate the advantages of our network and loss design. To stimulate more research in this direction and ensure reproducibility, we will release our code, network model, as well as the benchmark results to the public.

## 2 RELATED WORK

### 2.1 Handcrafted 3D shape descriptors

Classical 3D shape descriptors compute transformation invariant features of 3D shapes. They can be roughly categorized into global shape descriptors e.g., shape size, shape distribution, and shape histogram, and local shape descriptors e.g., mean curvature, spin images, and heat kernel signatures. A comprehensive survey of 3D descriptors can be found in [Laga et al. 2018]. These features and the bags of these features are suitable for many low-level tasks like key-point matching, but they perform poorly on high-level analysis tasks such as semantic segmentation and fall behind learning-based approaches.

### 2.2 Supervised 3D feature learning for shape analysis

Discriminative shape features can be learned with the supervision of the labeled data in a task-specific manner. Many 3D learning frameworks take full advantage of label information, like shape category labels existing in ModelNet [Wu et al. 2015] or segmented patch labels of ShapeNet [Chang et al. 2015], to learn the most representative features for the corresponding shape analysis tasks. Among them, supervised deep learning approaches achieve the best performance on the benchmark datasets of shape classification and semantic segmentation [Mo et al. 2019; Song 2019; Yi et al. 2017a; Yu et al. 2019]. According to the underlying 3D representations, existing 3D deep learning methods can be classified into several categories: multi-view based CNNs [Choy et al. 2016; Kalogerakis et al. 2017; Maturana and Scherer 2015; Su et al. 2015], volumetric and sparse-voxel-based CNNs [Graham 2015; Graham et al. 2018; Riegler et al. 2017; Wang et al. 2017; Wu et al. 2015], point-based networks [Li et al. 2018b,a; Qi et al. 2017a,b], manifold-based CNNs [Boscaini et al. 2015, 2016; Hanocka et al. 2019] and graph-based approaches [Monti et al. 2017; Wang et al. 2019b; Yi et al. 2017b]. The learned features are bound with the employed representation, like points, or the whole shape.

Despite the good performance of supervised 3D learning, the learned features are task-specific and thus difficult to be adapted to other tasks. Moreover, labeling a large amount of 3D data is a time-consuming job.

### 2.3 Unsupervised 3D feature learning for shape analysis

unsupervised learning methods aim to directly learn generic 3D shape representations from the unlabeled data via carefully designed pretext tasks. Most prior works use shape reconstruction as a pretext task by modeling shapes as latent vectors of autoencoders [Goodfellow et al. 2016a] or generative networks [Goodfellow et al. 2016b]. They can be classified according to the level of learned features:

shape-level feature learning, patch-level feature learning, and point-level feature learning.

*Shape-level feature learning.* 3D-GANs [Wu et al. 2016] train a generative adversarial network (GAN) on volumetric data, and its discriminator is used for extracting shape-level features. L-GANs learn deep shape representations by combining an autoencoder network and a GAN [Achlioptas et al. 2018]. FoldingNets [Yang et al. 2018] and AtlasNets [Groueix et al. 2018] optimize an autoencoder by reconstructing shapes from deformed point clouds. Zhang and Zhu [2019] cascade two pretext tasks – part contrasting and object clustering to learn the shape feature space. The unsupervised shape-level features are mainly used for shape classification and retrieval.

*Patch-level feature learning.* Shu et al. [Shu et al. 2016] train an autoencoder to recover the low-level hand-crafted geometry features of over-segmented patches and use high-level latent vectors for clustering patches into large and possibly meaningful parts. It is unclear whether the learned patch-level features can help with other shape analysis tasks.

*Point-level feature learning.* PPF-FoldNets [Deng et al. 2018] first extract hand-crafted point-pair features (PPF) from geometric patches, then uses a FoldingNet-style autoencoder to compress PPF features. As the learned point features are rotation invariant, they are suitable for shape matching. SO-Nets [Li et al. 2018b] extract hierarchical features from individual points and nodes of a self-organizing map, and represent the input point cloud by a single feature vector. The network can be pre-trained via unsupervised reconstruction. Point-CapsNets [Zhao et al. 2019] extend capsule networks [Sabour et al. 2017] to 3D point clouds and are trained by the shape reconstruction loss. Point features are directly extracted from the network and the latent capsules can be reshaped into a one-dimensional feature as a shape-level feature. Hassani and Haley [2019] propose to train a multiscale graph-based encoder with multiple pretext tasks including shape reconstruction, self-supervised classification, and feature clustering. Li et al. [2020] propose a unsupervised clustering task to learn point features for detecting distinctive shape regions. These tasks eventually add losses on the shape-level feature, while point-level features are not self-supervised explicitly. The recent work [Sauder and Sievers 2019] proposes a novel pretext task that reconstructs the voxel indices of randomly arranged points of a point cloud, thus it can supervise point features directly.

Although the above works can generate both point-level and shape-level features, their pretext tasks do not impose explicit self-supervision on both levels. We fill this gap by discriminating different-level shape features simultaneously and achieve significant improvements.

## 2.4 Unsupervised pre-training

Unsupervised pre-training is actively studied due to its success in various fields [Bengio et al. 2013; Erhan et al. 2010; Paine et al. 2015] due to the generality of learned features and performance improvements by fine-tuning on less labeled data. One of its key ideas is to design a set of pretext tasks (or called surrogate tasks) that use supervision signals from the data itself for unsupervised feature learning. In the natural language processing field, BERT models [Devlin et al.

2018] use masked context prediction and next sentence prediction as the pretext tasks, and GPT models [Radford et al. 2018] prefer the language modeling task. Both of them achieve outstanding performance on a variety of language inference problems. In computer vision field, various pretext tasks or dedicated loss functions like colorization [Zhang et al. 2016], context prediction [Doersch et al. 2015], motion segmentation [Pathak et al. 2017], iterative feature clustering [Caron et al. 2018], image instance discrimination [Wu et al. 2018] and contrastive losses [Hadsell et al. 2006; He et al. 2020], have also shown their strengths in learning effective image features.

Inspired by the idea of instance discrimination [Wu et al. 2018] which operates on a single image-level, we design a pretext task to discriminate multiresolution instances of shapes for learning representative shape features and develop effective training schemes to reduce the computational and memory cost caused by the large number of instances which cannot be handled by Wu et al.’s approach [2018].

## 3 UNSUPERVISED FEATURE LEARNING VIA MULTIRESOLUTION INSTANCE DISCRIMINATION

### 3.1 Overview

Given a large unlabeled 3D shape collection that consists of  $N$  3D models, we assume each 3D shape  $X_i$  is represented with a point cloud with  $M_i$  points, and denote the  $j$ -th point of  $X_i$  by  $\mathbf{p}_{i,j}$ . Here the term “unlabeled” means that the data has no shape category information or other human-annotated labels like shape parts, shape structures, and semantic labels. The goal of our method is to train a shape encoder network  $f_\theta$  that takes the point cloud of  $X_i$  as input and generate the representative and shape-level feature  $\mathbf{s}_i$  and point-wise features  $\{\mathbf{v}_{i,j}, j = 1, \dots, M_i\}$  for  $X_i$ :

$$f_\theta : X_i \longmapsto [\mathbf{s}_i, \mathbf{v}_{i,1}, \dots, \mathbf{v}_{i,M_i}], \quad (1)$$

where  $\theta$  denotes the set of network parameters.

To train the above feature space in an unsupervised manner, we first augment shapes via various transformations and create multiresolution class labels for later training (see Section 3.2), then feed them into a deep neural network which maintains and fuses multi-scale resolution feature maps efficiently (see Section 3.3). We use the multi-resolution instance discrimination pretext task (see Section 3.4) to self-supervise the feature learning progress (see Section 3.5). Figure 2 shows an overview of our method.

### 3.2 Input data processing

*Preprocessing and data augmentation.* The input to our pre-training network is the point cloud of a shape. We pre-process the point cloud to assign a normal vector for each point via principal component analysis if the accurate normal information is not available in the dataset, which will be used for constructing input features. The following data augmentation scheme is employed for generating shape and point instances for our unsupervised instance discrimination pretext task: we first normalize each point cloud into a unit sphere, then generate shape instances with a transformation composed by random rotations, random translations within [-0.25, 0.25], and random scaling along each coordinate axis with the ratio within [0.75, 1.25]. For the dataset in which the up-right directions

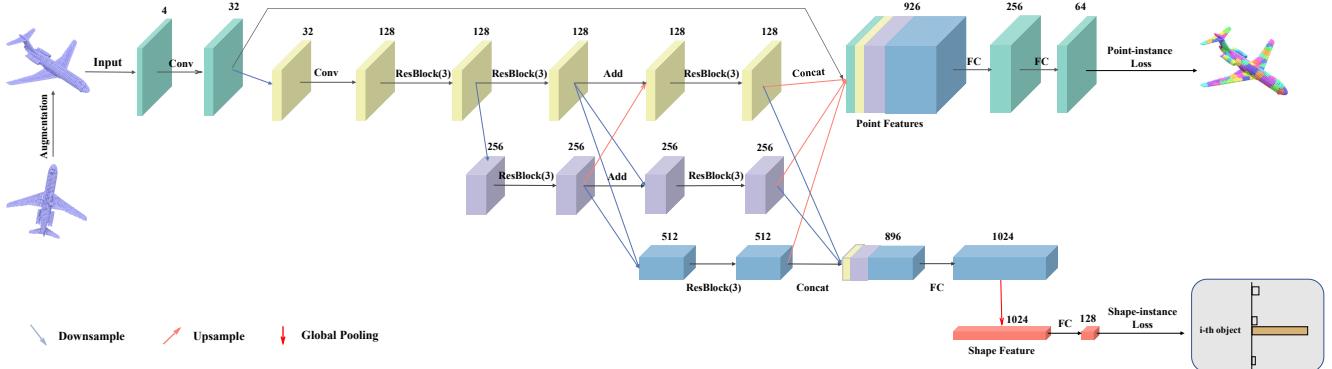


Fig. 2. Overview of our multiresolution-instance-discrimination (MID) unsupervised pre-training pipeline. An augmented input point cloud via transformations is fed into the deep neural network which maintains and fuses multi-scale resolution feature maps. The shape-level features and point-wise features are extracted from the network, and they are encouraged to be discriminative and transformation-invariant under the supervision of the MID loss on both shape instance and point levels.



Fig. 3. Visualization of patch-instance classes. An airplane model under different transformations is shown here, the over-segmented patches are color-coded according to their corresponding IDs. These transformed planes also belong to one shape-instance class.

of models are pre-aligned, our random rotation is restricted on the rotation along the upright axis. The small number of points outside the unit sphere after transformations is truncated.

*Multiresolution instance class creation.* We label the transformed instances of the same shape with its index in the input dataset. Also, each point on the generated instance is labeled by the same index of the corresponding point in the input shape in the dataset. So in total, We create  $N$  shape-instance classes and  $M_i$  point-instance classes for each shape-instance class. These multiresolution class labels serve as self-supervision signals for our network training. Note that these labels can be acquired by free and there is no human annotations here. As the total number of point-level classes of a large-scale shape dataset could be huge:  $\sum_{i=1}^N M_i$ , it will lead to huge memory consumption in network training. To overcome this issue, we introduce the concept of patch-instance class. On each shape  $X_i$  in the dataset, we over-segment it into  $K_i$  patches ( $K_i \ll M_i$ ), and on each shape-instance class, we can create  $K_i$  patch-instance classes, similar to the construction of point-instance classes. In this way, the point-instance classes can be approximated by the patch-instance classes, which have a less total number. For simplicity, we use the K-Means algorithm to compute over-segmented patches and choose the same  $K$  for all the shapes,  $K = 100$ , by default. In Figure 3, we visualize the segmented patches of an airplane shape and its augmented copies under different transformations and colorize patches according to their patch-instance IDs.

### 3.3 Network design

We adopt HRNet [Wang et al. 2019a] to extract multiresolution instance features. Different from the conventional U-Net [Ronneberger et al. 2015] architecture that cascades sequential convolutional layers from high to low resolutions and then recover high resolution features from low-resolution ones in an inverted order, HRNet maintains parallel multiresolution subnetworks and simultaneously outputs multi-resolution features. The features extracted by subnetworks in different resolutions are fused in different intermediate stages of HRNet. For 3D shape encoding, HRNet can simultaneously output low-resolution shape-level features and high-resolution point-wise features with one network. This property well matches our observation about multiresolution instance features as introduced in Section 1. Also, we can easily apply loss functions for outputs in different resolutions and each loss function will contribute to the training of all subnetworks.

We built HRNet upon an octree-based CNN framework [Wang et al. 2017] due to its efficiency in both computational cost and memory consumption and its natural multiresolution representation for building multiresolution subnetworks. A 3D point cloud is first converted to an octree representation, by default, in  $64^3$  resolution. For each non-empty octant at the finest level, we fit a plane to the points inside it, where the plane normal is fixed to the average normal of these points. We take the unit normal of the plane and the plane offset to the octant center as the raw feature vector. To deal with the possible inconsistent normal orientation issue across the shape, we convert each normal component to its absolute value, and find this trick does not hurt network performance.

As shown in Figure 2, our network takes the features at the finest level octants as input and the high-to-low resolution subnetworks are gradually added one by one and the second-highest resolution subnetwork is always kept. The features extracted from each subnetwork are downsampled and fed into low-resolution networks, and they are also upsampled and fed into the high-resolution subnetwork. In the last stage, the network outputs the low-resolution shape-level feature by concatenating the feature of the low-resolution

network and the feature downsampled from the output of the high-resolution networks. For point-wise features, we interpolate the high-resolution features defined at the second finest-level octants according to the point position via tri-linear interpolation. Note that the high-resolution features at each octant are also the concatenation of features obtained from the high-resolution network and the upsampled feature from the low-resolution subnetwork.

The details of the HRNet structure used in our method are illustrated in Figure 2. The numbers of feature channels are listed on the top of each feature map. The convolution kernel size is  $3 \times 3 \times 3$ . ResBlock(3) represents three cascaded ResNet blocks with a bottleneck structure [He et al. 2016]. The Downsample operation is implemented by max-pooling, and the Upsample operation is simply tri-linear up-sampling.

### 3.4 Multi-resolution instance discrimination

As shown in Figure 2, our network training is supervised by two loss functions: the shape-instance discrimination loss and the point-instance discrimination loss. Both loss functions are based on input 3D models and the multiresolution instance class labels created in Section 3.2.

*Shape-instance discrimination loss.* The shape-level feature is supervised by a linear classifier that classifies shape instances into  $N$  classes, where each 3D shape  $X_i$  and its augmented instances are classified into the  $i$ -th class. The cross-entropy loss for the  $i$ -th class is defined as:

$$L_s(s_i) = -\log \frac{\exp((s_i \cdot \tilde{s}_i)/\tau_s)}{\sum_{k=1}^N \exp((s_i \cdot \tilde{s}_k)/\tau_s)}, \quad (2)$$

where  $\tilde{s}_i$  is the weight vector in the linear classifier for the  $i$ -th class. We follow the approach of [Wang et al. 2018; Wu et al. 2018] to normalize  $s_i$  and  $\tilde{s}_i$  to be unit-length and measure their difference by the cosine distance.  $\tau_s$  is a parameter controlling the concentration level of the extracted features and is set to 0.1 empirically.

To obtain a good classification, an optimal  $\tilde{s}_i$  expects to be the average center of features of all shape instances in this class [Liu et al. 2018]. By optimizing the loss function Equation (2), the features of shape instances under different transformations converge to  $\tilde{s}_i$  and thus they are invariant to the imposed transformations. Meanwhile, the loss function of Equation (2) tends to maximize the distance between  $s_i$  and other  $\tilde{s}_k (k \neq i)$ , which makes the shape feature  $s_i$  of shape  $X_i$  is discriminative against the features of other 3D models.

*Point-instance discrimination loss.* To optimize point-wise features, we can also use a linear classifier to classify the points of a shape and its augmented copies into  $M_j$  classes, with the created supervision signals (Section 3.2). For points in the  $j$ -th point class, the cross-entropy loss is defined as:

$$L_p(v_{i,j}) = -\log \frac{\exp((v_{i,j} \cdot \tilde{v}_{i,j})/\tau_p)}{\sum_{k=1}^{M_j} \exp((v_{i,j} \cdot \tilde{v}_{i,k})/\tau_p)}, \quad (3)$$

where  $v_{i,j}$  is the point-wise feature of  $j$ -th point of an augmented shape instance of  $X_i$ ,  $\tilde{v}_{i,j}$  is the weight vector in the linear classifier for the  $j$ -th point class. All the point feature vectors are also unit-length and the control parameter  $\tau_p$  is set to 0.1.

As discussed in Section 3.2, treating each point of a 3D shape as an individual class is impractical as it leads to a large number of classes for each object and results in huge memory consumption and computational cost for storing and updating the weights of linear classifiers. Therefore, we propose to approximate point-instance discrimination by patch-instance discrimination, and revise the loss function as:

$$L_p(v_{i,j}) = -\log \frac{\exp((v_{i,j} \cdot \tilde{v}_{i,c(i,j)})/\tau_p)}{\sum_{c=1}^K \exp((v_{i,j} \cdot \tilde{v}_{i,c})/\tau_p)}, \quad (4)$$

where  $c(i,j)$  is the index of the patch containing the  $j$ -th point on shape  $X_i$ .

*MID Loss.* By combining the above two multiresolution instance discrimination loss functions, we define the MID loss for an instance of shape  $X_i$  as:

$$L(X_i) = L_s(s_i) + \frac{1}{M_i} \sum_{j=1}^{M_i} L_p(v_{i,j}). \quad (5)$$

Note that although our method applies different linear classifiers for points of different 3D shapes, the shape encoder is shared by all 3D shapes.

### 3.5 Network training

Given the MID loss function defined above, we optimize the network parameters of  $f_\theta$  with an input 3D shape collection by a stochastic gradient descent (SGD) algorithm. In each mini-batch, we randomly pick a set of 3D shapes from the dataset and generate an augmented shape for each original shape in runtime.

A naïve approach to train the network is to update the weights of the shape encoder network and the linear classifier together according to the back-propagated gradient of the MID loss function. However, we found that updating the linear classifiers in each mini-batch often makes the classifier training unstable and hinders the optimization of the shape encoder. A similar observation is also reported in a concurrent work for visual representation learning [He et al. 2020]. Inspired by the temporal ensembling scheme in [Laine and Aila 2017], we update the weights of the linear shape-instance classifier slowly by

$$\tilde{s}_i = (1 - \lambda_s) \cdot \tilde{s}_i + \lambda_s \cdot s_i, \quad (6)$$

---

#### Algorithm 1: Network training procedure

---

**Input:** A set of shapes  $\{X_i\}_{i=1}^N$ ;  
**Output:** Network  $f_\theta$ ,  $\{\tilde{s}_i\}_{i=1}^N$  and  $\{\{\tilde{v}_{i,c}\}_{c=1}^K\}_{i=1}^N$ ;

- 1 Assign multiresolution instance labels to  $\{X_i\}_{i=1}^N$  according to Section 3.2
- 2 **for**  $l$  in  $[0, max\_iteration]$  **do**
- 3     Randomly sample a batch of  $B$  samples  $\{X_{b_i}\}_{i=1}^B$ ;
- 4     Compute loss  $L = \frac{1}{B} \sum_{i=1}^B L(X_{b_i})$  (Equation (5));
- 5     Compute gradient  $\nabla L(\theta)$  and update  $\theta$  with SGD;
- 6     Update  $\{\tilde{s}_{b_i}\}_{i=1}^B$  and  $\{\{\tilde{v}_{b_i,c}\}_{c=1}^K\}_{i=1}^B$  with Equation (6) & Equation (7).

---

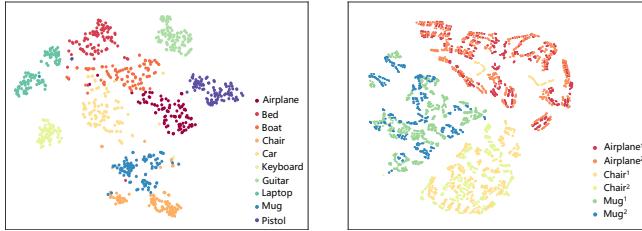


Fig. 4. Visualization of t-SNE mapping of shape-level features and point-wise features of 3D shapes. The left figure displays the T-SNE map of shape-level features 100 3D models selected from 10 shape categories. The right figure exhibits the T-SNE map of point-wise features of 6 shapes.

where  $s_i$  is the shape-level feature of the current shape instance for the  $i$ -th shape-instance class,  $\lambda_s$  is a momentum parameter and is set to 0.5 in our implementation. For point-instance classifier, we use a similar update rule:

$$\tilde{v}_{i,c} = (1 - \lambda_p) \cdot \tilde{v}_{i,c} + \lambda_p \cdot v_{i,c}, \quad (7)$$

where  $v_{i,c}$  is the average of the point-wise features of all points that belong to the patch class  $c$ ,  $\lambda_p$  is a momentum parameter and is set to 0.5 too.

A detailed training procedure is summarized in Algorithm 1. We have implemented our network using Tensorflow and trained the MID-Net on ShapeNetCore55 dataset [Chang et al. 2015] that consists of 57,449 3D shapes. For each shape, we average the points inside each non-empty finest-level octants and use these average points as the sample points of this shape. We set the batch size as 32, momentum as 0.9, and weight decay as 0.0005. The initial learning rate is 0.03 and it decays by a factor of 10 after 200 epochs and 300 epochs, respectively. The whole training process is finished after 400 epochs, which takes about 60 hours on an NVIDIA 2080 Ti graphics card. The overall network parameter size is 1.5M, which is comparable to PointNet++ [Qi et al. 2017b] (1.4 M), DGCNN [Wang et al. 2019b] (1.5 M), and much smaller than PointCNN [Li et al. 2018a] (8.2 M).

### 3.6 Feature visualization

The quality of learned shape-level features and point-wise features can be assessed intuitively via visualization. In Figure 4 we show the t-SNE mapping [Maaten and Hinton 2008] of shape-level features of 1000 3D shapes randomly sampled from 10 shape categories, as well as the t-SNE mapping of point-wise features of 6 shapes (2 airplanes, 2 chairs, and 2 mugs). Note that the shape-level features and point-wise features of 3D shapes have similar distributions in their feature space, where the features of 3D models with similar overall shapes are clustered together and can be easily discriminated from the features of different 3D models.

## 4 MID-NETS FOR SHAPE ANALYSIS

As a pre-trained 3D shape-encoder, our MID-Net can be applied in various shape analysis tasks. To this end, we evaluate our method on one shape classification task, two semantic shape segmentation tasks, and one shape registration task. In this section, we first present our back-end network design and its training scheme and then

discuss the performance of our method in each downstream task. Finally, we summarize and discuss the overall performance of our method in all three tasks. We will release our code and models to the public.

### 4.1 Back-end design and training

For each downstream task, we concatenate MID-Net with simple back-end layers. We use a one-layer fully-connected (FC) network for shape classification; for shape segmentation, two-layer FC for shape segmentation. We denote the MID-Net with the FC back-end as **MID-FC** and optimize the concatenated networks with two training schemes:

- **MID-FC(Fix)**, in which we fix the pre-trained MID-Net and only train the back-end with the labeled training data in each shape analysis task.
- **MID-FC(Finetune)**, in which we fine-tune both MID-Net and the FC back-end with the labeled training data in each shape analysis task. The MID-Net is initialized with the pre-trained weights, and the FC back-end is randomly initialized.

In the training process, our method first constructs the octree of input 3D point cloud and compute input features of the finest-level octants with the same process used in MID-Net pre-training. Then the MID-Net computes the shape and point features of 3D input, where the feature of each point is computed from the output features of nearby eight leave octants via tri-linear interpolation. After that, the output shape or point features are fed into the following back end layers for computing the shape analysis results. The hyper-parameter of network training for each shape analysis task is described in the following subsections respectively.

To evaluate the impact of pre-training to downstream tasks, we also discard the pre-training and train the concatenated networks in a supervised way. In particular, we initialize both MID-Net and FC layers with random initialization and train the network from scratch with the labeled training data of each shape analysis task. We denote the resulting networks as **MID-FC(NoPre)**.

### 4.2 Shape classification

*DataSet.* For shape classification, we use the ModelNet40 benchmark [Wu et al. 2015], in which there are 13,834 3D models across 40 categories: 9,843 models are used for training and 3,991 models are used for testing. The input is the point cloud of 3D shapes and

Table 1. The accuracy of MID-Net in ModelNet40 shape classification task. For comparison, we also list the performance of the state-of-the-art supervised methods (second column) and unsupervised methods (the third column). The best accuracy is highlighted. All the unsupervised methods are pre-trained with the ShapeNet dataset and use a linear classifier for shape classification. There is no voting or orientation pooling trick used by any method.

Our Method	Accuracy	Supervised	Accuracy	Unsupervised	Accuracy
MID-FC(NoPre)	92.9	PointNet++	90.7	L-GAN [Zhang and Zhu 2019]	84.5
MID-FC(Fix)	90.3	PointCNN	92.2	FoldingNet	86.8
<b>MID-FC(Finetune)</b>	<b>93.1</b>	PointConv	92.3	Multi-Task	88.4
		DGCNN	92.9	PointCapsNets	89.1
		KPCConv	92.9	[Sauder and Sievers 2019]	89.3
		RS-CNN	92.9		90.6

Table 2. The performance comparison of MID-FC(Fix) and FoldingNet for shape classification, where the linear classifier back-end in both methods are trained by different ratios of training data. Note that in all settings, our method achieves better accuracy than FoldingNet. The last two rows contain the performance of MID-FC(NoPre) and MID-FC(Finetune).

Data	1%	2%	5%	10%	20%
FoldingNet	56.4	66.9	75.6	81.2	83.6
MID-FC(Fix)	<b>61.5</b>	<b>73.1</b>	<b>80.2</b>	<b>84.2</b>	<b>86.9</b>
MID-FC(NoPre)	58.5	71.2	80.1	85.4	88.7
MID-FC(Finetune)	<b>68.1</b>	<b>77.2</b>	<b>83.6</b>	<b>88.4</b>	<b>90.2</b>

the output is the category ID that each shape belongs to. We follow the method of [Wang et al. 2017] to convert the triangle mesh of each 3D shape into a point cloud but without using accurate point normals. We use the instance classification accuracy to measure the shape classification quality.

*Results and comparisons.* We optimize the networks via SGD with 240 epochs in all three training schemes. The batch size is 32. The initial learning rate is set as 0.1 for MID-FC(NoPre) and the FC back-end, and 0.01 for MID-FC(Finetune). The learning rate decays by a factor of 1/10 after 120 and 180 epochs. For a fair comparison, we do not use any strategy like voting or orientation pooling [Atzmon et al. 2018; Li et al. 2018a; Qi et al. 2017b; Wang et al. 2017] to improve the results during the testing phase.

Table 1 lists the accuracy of our MID-FC models and other state-of-the-art unsupervised and supervised shape classification approaches whose inputs are point clouds. All methods in the table are trained with 100% labeled data by default. As shown in the first column, our pre-trained MID-Net provides a good initialization to MID-FC(Finetune) and thus results in better accuracy (93.1%) than MID-FC(NoPre) (92.9%) that is trained from random initialization. Compared to the state-of-the-art supervised methods including O-CNN [Wang et al. 2017], PointNet++ [Qi et al. 2017b], PointCNN [Li et al. 2018a], PointConv [Wu et al. 2019], SpiderCNN [Xu et al. 2018], DGCNN [Wang et al. 2019b], KPCConv [Thomas et al. 2019], and RS-CNN [Liu et al. 2019], our MID-FC(Finetune) also exhibits superior accuracy.

We also compare our method with other unsupervised methods listed in the third column of Table 1, including L-GAN [Achlioptas et al. 2018], [Zhang and Zhu 2019], FoldingNet [Yang et al. 2018], PointCapsNet [Zhao et al. 2019], Multi-Task [Hassani and Haley 2019], and the work of [Sauder and Sievers 2019]. Here all the unsupervised methods are pre-trained with the ShapeNet dataset and use a linear classifier. Our MID-FC(Fix) achieves the second-best performance (90.3%) among all unsupervised approaches listed in the table, and is slightly worse than [Sauder and Sievers 2019].

In Table 2, we further evaluate the accuracy of MID-FC learned from different ratios of labeled training data: 1%, 2%, 5%, 10%, and 20%, where our method can still attain good performance. Compared with FoldingNet, our MID-FC(Fix) achieves better accuracies. By comparing the performance of MID-FC(Fix) and MID-FC(Finetune), it can be seen that MID-FC(Finetune) is also better than MID-FC(Fix), which verified the effectiveness of our pretraining.

#### 4.3 Fine-grained PartNet segmentation

*Dataset.* We evaluate our method with fine-grained segmentation on the PartNet dataset [Mo et al. 2019], which is a challenging dataset that consists of 24,506 3D shapes in 17 categories with fine-grained (*i.e.*, Level-3) labels. The part numbers in each shape category vary from 3 to 50. Each shape contains 10,000 points and each point has a part ID label. We use these data and their Level-3 labels in our experiments, and follow the train-validation-test split setup in [Mo et al. 2019]. Each algorithm takes a 3D point cloud as input and predicts the part ID for each point. We measure the accuracy of the segmentation results by mean IOU across all 17 categories. Here, we note that the ShapeNet dataset used for pre-training does not contain Door, Fridge, and Storage categories of PartNet.

*Results and comparisons.* We optimize our networks via SGD with batch size 32. The learning rate of MID-FC(Fix) and MID-FC(NoPre) starts from  $10^{-1}$  and decreases to  $10^{-3}$ , and the learning rate of MID-FC(Finetune) decays from  $10^{-2}$  to  $10^{-4}$ .

Table 3 shows the performance of MID-FC and other state-of-the-art supervised approaches, including PointNet [Qi et al. 2017a], PointNet++ [Qi et al. 2017b], SpiderCNN [Xu et al. 2018], and PointCNN [Li et al. 2018a]. MID-FC(Fix) outperforms all existing supervised approaches, which is significant considering the fact that MID-FC(Fix) only contains 2 trainable FC layers and optimizes less than 0.3M parameters. With fine-tuning, MID-FC(Finetune) achieves more improvements and the accuracy increases 14.3 points at least, compared to supervised approaches, and 2 point improvement over MID-FC(NoPre). The results clearly show the advantage of our pre-training and the efficacy of our network structure. For the three categories that are not in ShapeNet, MID-FC(Finetune) and MID-FC(Fix) still achieve better performance, which demonstrates the good generality of our pre-trained features.

The segmentation performance of all our three networks learned from different ratios of training data: 1%, 5%, 10%, and 20% is reported in Table 4. MID-FC(Finetune) and MID-FC(Fix) have better performance than the supervised MID-FC(NoPre), and the accuracy of MID-FC(Fix) and MID-FC(Finetune) trained with 20% labeled data is comparable to other supervised methods trained with 100% labeled data.

#### 4.4 ShapeNet-Part segmentation

*DataSet.* In this task, we use the ShapeNet-Part dataset [Yi et al. 2017a] which contains 16,881 3D point clouds, collected from 16 categories of ShapeNet [Chang et al. 2015]. Each point cloud has 2 to 6 part segmentation labels. Shapes in each category share the same part segmentation (*i.e.*, the same label set), while different shape categories have their own segmentation. We follow the scheme in [Yi et al. 2017a] exactly to split the data into training, validation, and test datasets. For shape segmentation, each algorithm takes 3D point cloud as input and predicts the part label for each point. The number of points of a shape varies from 1000 to 3000, which is sparser than the point cloud used in our pre-training, where the point number varies from 6,000 to 8,000. The accuracy of the segmentation results is measured by the mean-IoU across all categories and the mean-IoU across all instances in each category.

Table 3. Category-specific fine-grained semantic segmentation results on the PartNet dataset. The number of shapes in the training and testing datasets are reported, per category. mIoU means the averaged IoU over categories.

Model	mIoU	Bed	Bottle	Chair	Clock	Dish	Display	Door	Ear	Faucet	Knife	Lamp	Micro	Frid.	Stora.	Table	Trash	Vase
#Train		133	315	4489	406	111	633	149	147	435	221	1554	133	136	1588	5707	221	741
#Test		37	84	1217	98	51	191	51	53	132	77	419	39	31	451	1668	63	233
PointNet	35.6	13.4	29.5	27.8	28.4	48.9	76.5	30.4	33.4	47.6	32.9	18.9	37.2	33.5	38.0	29.0	34.8	44.4
PointNet++	42.5	30.3	41.4	39.2	41.6	50.1	80.7	32.6	38.4	52.4	34.1	25.3	48.5	36.4	40.5	33.9	46.7	49.8
SpiderCNN	37.0	36.2	32.2	30.0	24.8	50.0	80.1	30.5	37.2	44.1	22.2	19.6	43.9	39.1	44.6	20.1	42.4	32.4
PointCNN	46.5	41.9	41.8	43.9	36.3	58.7	82.5	37.8	48.9	60.5	34.1	20.1	58.2	42.9	49.4	21.3	53.1	58.9
MID-FC(NoPre)	58.4	45.9	56.1	54.6	52.9	68.1	90.6	51.1	52.1	64.9	54.4	29.9	75.9	58.9	62.8	43.7	62.4	68.3
MID-FC(Fix)	49.4	42.9	35.0	36.2	44.4	70.0	87.5	48.0	49.2	54.1	46.6	23.0	69.0	50.3	41.9	24.7	55.9	60.9
MID-FC(Finetune)	<b>60.8</b>	<b>51.6</b>	<b>56.5</b>	<b>55.7</b>	<b>55.3</b>	<b>75.6</b>	<b>91.3</b>	<b>56.6</b>	<b>53.8</b>	<b>64.6</b>	<b>55.4</b>	<b>31.2</b>	<b>78.7</b>	<b>63.1</b>	<b>62.8</b>	<b>45.7</b>	<b>65.8</b>	<b>69.3</b>

Table 4. Category-specific fine-grained semantic segmentation results on the PartNet dataset trained with different ratios of labeled data.

Model	1%	5%	10%	20%
MID-FC(NoPre)	14.5	27.7	29.5	41.6
MID-FC(Fix)	21.7	31.3	34.4	38.9
MID-FC(Finetune)	<b>21.8</b>	<b>35.4</b>	<b>40.2</b>	<b>46.3</b>

*Results and Comparisons.* We train the networks via SGD with batch size 32. For MID-FC(Fix) and MID-FC(NoPre), the learning rate decays from  $10^{-1}$  to  $10^{-3}$ . For MID-FC(Finetune), the learning rate decays from  $10^{-2}$  to  $10^{-4}$ , respectively.

Table 5 shows the accuracy of our methods and other state-of-the-art methods, including supervised methods such as PointNet++ [Qi et al. 2017b], SynSpecCNN [Yi et al. 2017b] and DGCNN [Wang et al. 2019b] and SPLATNet [Su et al. 2018], and fine-tuned unsupervised methods like Multi-Task [Hassani and Haley 2019], PointCapsNet [Zhao et al. 2019] and [Sauder and Sievers 2019]. The accuracy gap between MID-FC(Finetune) and MID-FC(NoPre) clearly demonstrates the advantage of the pre-trained MID-Net. Compared with supervised methods, both MID-FC(Fix) and MID-FC(Finetune) have a better performance. MID-FC(Fix) and MID-FC(Finetune) also surpass other unsupervised methods when all the methods are fine-tuned with a small amount of labeled data (*i.e.*, 1%, and 5%) in the downstream task.

In Table 6, we also compare with other state-of-the-art supervised methods Submanifold [Graham et al. 2018], KPConv [Thomas et al. 2019], PointCNN [Li et al. 2018a], RS-CNN [Liu et al. 2019], that employ the voting strategy to improve the segmentation accuracy. We use a similar strategy in our method too. We can see that all the methods have improved accuracy, while MID-FC(Fix) and MID-FC(Finetune) have better the average category IOUs.

#### 4.5 Shape registration

The process of point cloud registration is to align a point cloud with its transformed version, which is a fundamental problem in computer graphics. To handle arbitrary rotations of shapes, we trained a new MID-Net with shapes augmented with arbitrary rotations for obtaining rotation-invariant multiresolution instance features. We regard the point-wise features extracted from our pre-trained MID backbone network as the point descriptors and use them to find closest point pairs from two input point clouds for computing the initial rigid transformation for the standard ICP algorithm.

*Dataset.* We conduct a comparison on the test set of ModelNet40 [Wu et al. 2015], which contains 2,468 man-made models across 40 categories. We normalize each shape inside a unit sphere and apply a random rigid transformation to the shape. In particular, the rotation angle along each coordinate axis is randomly sampled from  $[0^\circ, 360^\circ]$ , while the translation along each coordinate axis is randomly sampled from  $[-0.25, 0.25]$ .

We uniformly sample 2,048 points from the original shape and the same number of points from the transformed shapes and take these two point clouds as input for the shape registration task. In total, we create 2468 paired data for testing. As the transformation is known during the data creation, we can compute the ground-truth registration error by transforming the two paired shapes together and computing the Hausdorff distance of these two point clouds.

*Results and comparisons.* On the test dataset, we compare our method with four state-of-the-art methods, including Go-ICP [Yang et al. 2015], FGR [Zhou et al. 2016], PointLK [Aoki et al. 2019], and DCP [Wang and Solomon 2019]. Since PointLK and DCP are learning-based methods, we trained their models with the ShapeNet dataset, the training data creation strategies are the same as their works. Since DCP and PointLK were originally designed to support transformations with rotation angles sampled within  $[0^\circ, 45^\circ]$  only, we use the code provided by the authors to support arbitrary rotations. However, we found that the trained networks had a poor performance in dealing with large rotations.

For all the methods except FGR which performs nonlinear optimization, we run the ICP algorithm to refine the initial registration computed by them. We evaluate the registration results by computing the Hausdorff distance between two registered shapes by different algorithms with ICP refinement. Figure 6 shows the accuracy curve within a given Hausdorff distance bound. The ground-truth curve and the curve by applying the standard ICP algorithm serve as the upper and lower bound of all the algorithms. Our method outperforms the other four methods. Figure 5 show registration results of two examples, where our method registers the input correctly.

#### 4.6 Summary and Discussion

From the results and comparisons in the above experiments, it is clear that our unsupervised pre-trained MID-Net model adapts to both shape-level and point-level analysis tasks easily and demonstrates its generality, efficiency, and robustness in all the downstream tasks. In particular,

Table 5. Category-specific semantic segmentation on the ShapeNet-Part dataset. For each method, the averaged IoU over instances (I.mIOU%) and categories (C.mIOU%), as well as the IOU of each category are reported. ‘-’ means that the data is not available from the author’s paper.

Model	%train	C.mIOU	I.mIOU	Aero	Bag	Cap	Car	Chair	Ear.	Guitar	Knife	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate.	Table
PointNet	100%	80.4	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++		81.9	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
DGCNN		82.3	85.1	<b>84.2</b>	83.7	84.4	77.1	90.9	78.5	91.5	87.3	82.9	96.0	67.8	93.3	82.6	59.7	75.5	82.0
SynSpecCNN		82.0	84.7	81.6	81.7	81.9	75.2	90.2	74.9	<b>93.0</b>	86.1	<b>84.7</b>	95.6	66.7	92.7	81.6	60.6	<b>82.9</b>	82.1
SPLATNet		83.7	85.4	83.2	84.3	89.1	80.3	<b>90.7</b>	75.5	92.1	<b>87.1</b>	83.9	<b>96.3</b>	75.6	95.8	<b>83.8</b>	64.0	75.5	81.8
Sauder et al. [2019]	100%	83.0	85.3	84.1	84.0	85.8	77.0	90.9	80.0	91.5	87.0	83.2	95.8	71.6	94.0	82.6	60.0	77.9	81.8
MID-FC(NoPre)		84.2	85.2	83.3	<b>85.4</b>	88.1	81.9	90.4	79.0	91.8	85.6	78.9	<b>95.9</b>	<b>79.0</b>	<b>96.0</b>	83.3	64.2	82.4	82.7
MID-FC(Fix)		82.7	84.2	80.4	82.5	89.0	80.0	89.9	80.7	90.5	85.7	77.8	95.9	73.4	94.8	81.1	56.7	81.8	82.4
MID-FC(Finetune)		<b>84.8</b>	<b>85.5</b>	83.6	82.9	<b>91.3</b>	<b>81.6</b>	90.4	<b>81.5</b>	91.8	<b>87.1</b>	79.3	95.7	78.7	95.2	83.6	<b>68.3</b>	82.7	<b>83.2</b>
Multi-Task	1%	-	68.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
PointCapsNets		-	67.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
MID-FC(Fix)		59.7	72.4	73.4	44.9	66.7	51.5	86.5	27.9	42.7	70.7	60.1	93.7	48.1	74.5	75.1	22.7	40.6	76.4
MID-FC(Finetune)		63.6	76.7	77.5	50.4	36.6	60.9	86.7	39.1	88.6	81.0	65.5	95.0	33.9	93.3	61.1	22.7	48.5	76.6
Multi-Task	5%	73.0	80.7	80.0	47.3	64.8	76.0	87.2	43.8	88.9	80.0	74.4	94.6	58.3	93.5	79.5	42.1	78.2	78.9
PointCapsNets		-	70.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
MID-FC(Fix)		77.9	81.1	77.9	79.5	80.1	75.9	87.9	68.9	89.5	82.5	72.0	95.0	62.7	93.7	76.5	46.5	78.3	79.3
MID-FC(Finetune)		79.7	82.1	80.3	79.8	86.8	77.7	88.0	66.9	90.0	84.7	73.3	94.3	67.2	94.1	79.6	53.3	79.6	80.2

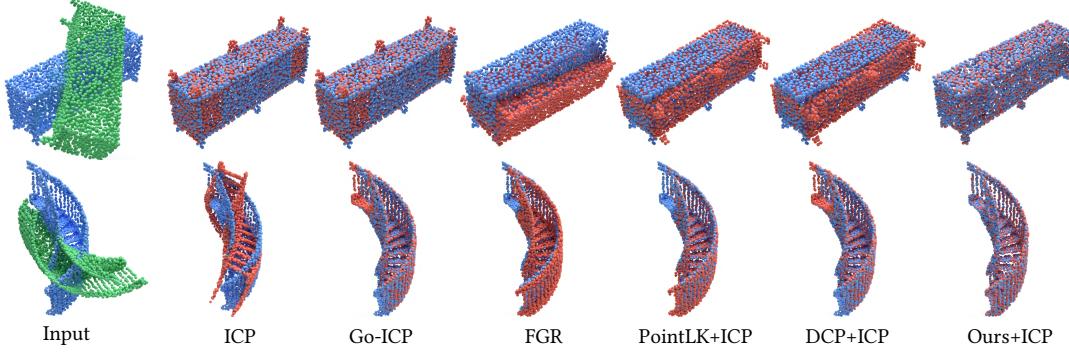


Fig. 5. Two examples of point cloud registration. The blue shape is fixed, the green one is transformed by different methods and its transformed version is rendered in orange color. The stair model is not in the ShapeNet dataset. For both examples, only our method registers the inputs correctly.

Table 6. Category-specific semantic segmentation on the ShapeNet-Part dataset. During the testing phase, all the methods use the voting strategy to get the best result.

Model	C.mIOU	I.mIOU
Submanifold	83.3	86.0
RS-CNN	84.0	86.2
PointCNN	84.6	86.1
KPCConv	85.1	<b>86.4</b>
MID-FC(NoPre)	84.6	85.5
MID-FC(Fix)	83.4	84.6
MID-FC(Finetune)	<b>85.2</b>	85.8

- The MID-Net model outperforms most state-of-the-art unsupervised methods in both classification and segmentation tasks, especially when the amounts of labeled training data in the downstream tasks are small.
- With simple back-end and fine-tuning, our pre-trained MID-Net model can surpass all the state-of-the-art supervised methods that are designed and optimized for specific downstream tasks, especially in the challenging PartNet segmentation benchmark. Furthermore, the fine-tuning pre-trained MID-Net always

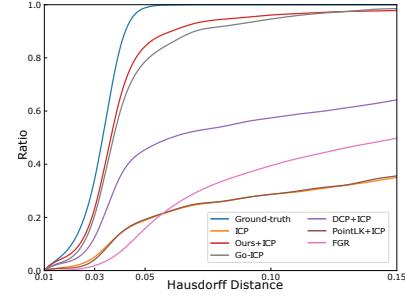


Fig. 6. Point cloud registration benchmark. The curve represents the ratio of registration results under a specific Hausdorff distance value. Obviously, our method achieves the best performance.

- achieves better performance than the MID net trained with random initialization.
- Our pre-trained MID-Net model is robust for 3D point clouds whose density is different to the pre-training data (e.g., point clouds in ShapeNet-Part segmentation test), and the ones that are out of shape collections used in pre-training (e.g., three categories in the PartNet segmentation test).

**Remark:** it is non-trivial for an unsupervised pre-trained model to achieve superior performance than supervised methods designed and optimized for specific downstream tasks due to the gap between the generic pretext task and the downstream tasks. For 3D shape analysis, our finding here is that as the amount of the labeled data in the downstream tasks decreases, a pre-trained model learned from large unlabeled shape collections via a well-designed pretext task could resolve this gap and outperform supervised approaches. Considering the difficulty of 3D shape labeling and the unsupervised nature of our method, we believe that an unsupervised pre-trained model can provide a more generic and efficient solution than supervised methods for various shape analysis tasks.

## 5 METHOD VALIDATION

In this section, we first evaluate the contributions of HR-Net and the MID loss to the performance of our pre-trained model in downstream tasks, as well as the generality of our MID scheme for other network architectures. After that, we conduct a set of ablation studies to validate design decisions and parameter setting of our method.

### 5.1 Contributions of HR-Net and MID

*Contribution of HR-Net.* To evaluate the contribution of individual HR-Net to the final results, we compare the performance of MID-FC(NoPre) with the existing supervised methods that can achieve the best performance in each downstream task. Because the existing supervised methods and MID-FC(NoPre) have different network architectures but share similar supervised training schemes with random initializations, the difference between their performance can illustrate the contribution of HR-Net.

As shown in Table 7, our octree-based HR-Net achieves very similar performance to the best supervised methods in shape classification and ShapeNet-Part segmentation (with 0.0 and +0.5 differences, respectively). For fine-grained shape segmentation, the performance of MID-FC(NoPre) is significantly better (with +11.9 difference). The good performance on different downstream tasks illustrates that our octree-based HR-Net offers an efficient network architecture for our generic pre-trained model.

*Contribution of MID.* For the contribution of the MID loss, we evaluate its contribution in each task by comparing the performance of MID-FC(NoPre) and MID-FC(Finetune). The two models share the same network architecture but are trained with different learning schemes. The MID-FC(NoPre) is trained from scratch with the task-specific labeled data by a supervised scheme, while the MID-FC(Finetune) is initialized with the weights of MID-Net pre-trained via our MID scheme and then refined with the task-specific labeled data.

As shown in Table 7, the performance gain of MID-FC(Finetune) over the supervised MID-FC(NoPre) counterparts in all three tasks (+0.2, +0.6, +2.4, respectively) clearly demonstrates the contribution of our MID scheme for different downstream tasks.

*Generality of the MID scheme.* To validate the generality and advantage of the MID scheme for pre-training, we apply the MID scheme to PointNet++ with the default network in [Qi et al. 2017b] that provides both instance level and point-level features within

Table 7. Evaluation of octree-based HR-Net and the MID scheme. The performance difference between the models in the first two rows illustrates the contribution of individual HR-Net to the final results. The last two rows list the performance of pre-trained PointNet++ with the MID scheme and the one without pre-training. For both network architectures, the MID-based pre-training scheme demonstrates its advantage in all three downstream tasks. The performance difference between the two pre-trained models (MID and PointNet++) further demonstrates the contribution of our octree-based HR-Net to 3D pre-training.

Model	Shape Classification	ShapeNet-Part Segmentation (C.mIoU)	PartNet Segmentation
Prior Art.	92.9 (DGCNN)	83.7 (SPLATNet)	46.5 (PointCNN)
MID-FC(NoPre)	92.9	84.2	58.4
MID-FC(Finetune)	93.1	84.8	60.8
Point-FC(NoPre)	90.1	81.9	43.8
Point-FC(Finetune)	90.1	83.4	49.4

one unified network structure. Similar to MID-Net, we concatenate PointNet++ with a two-layered FC layer and fine-tune the networks (denoted as Point-FC(Finetune)) for all the tasks. Specifically, we initialize PointNet++ and FC layers with a pre-trained model and random initialization, respectively, and fine-tune the whole network with the task-specific labeled data. For comparison, we also train the same concatenated PointNet++ networks from scratch with the task-specific training data (denote as Point-FC(NoPre)).

Table 7 shows that the performance gain between Point-FC(Finetune) and Point-FC(NoPre) in all three tasks exhibits the efficiency of our MID pre-training scheme to other network architectures, and the inferior performance of Point-FC(Finetune) to MID-FC(Finetune) also illustrates the importance of network architecture to the pre-trained model and the advantage of our octree-based HR-Net.

As a generic pre-training scheme, our MID scheme could also be applied to other network architectures, such as DGCNN and PointCNN. To this end, we have to extend these network architectures designed for specific tasks to unified ones that are efficient for different downstream tasks, which is a non-trivial task according to our experiments. We thus leave it as future work.

### 5.2 Ablation Study

To further validate the design choices in our method, we pre-train our networks with one modified component and keep all other settings unchanged. After pre-training, we test the modified models with shape classification and ShapeNet-Part segmentation tasks described in the last section. We still take one FC as a back-end for shape classification and two-layered FC (MID-FC) as the back-end for 3D segmentation. In both downstream tasks, we fix the pre-trained networks and optimize the back-ends with a full labeled training dataset and test the performance of resulting networks.

To better evaluate the efficacy of our pre-training in ablation studies, we use a denser version of point clouds provided by [Wang et al. 2017] for the ShapeNet-Part segmentation task, which has a similar point density to the data used in the pre-training. Table 8 lists all results of our ablation studies, where MID-FC is the network trained with our default setting.

*HR-Net versus other network structures.* Compared with the U-Net [Ronneberger et al. 2015] that is widely used in many graphics and vision tasks, the key advantages of HR-Net [Wang et al. 2019a] are the parallel multi-resolution subnetworks and feature fusion in

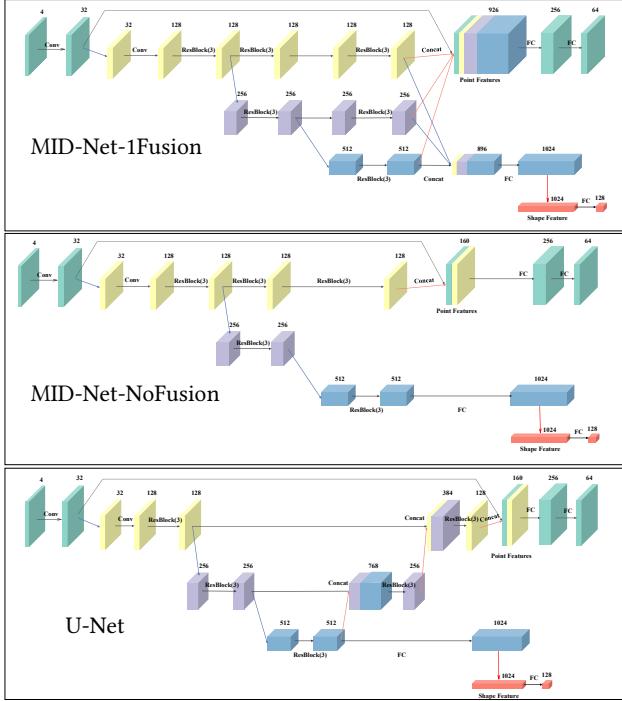


Fig. 7. Alternative MID network structures for ablation study.

different resolutions. To validate the advantages of HRNet in 3D pre-training, we develop two alternative networks that gradually remove the fusion layers between subnetworks and a traditional U-Net. As shown in Figure 7, the MID-Net-1Fusion removes the first fusion layer from the MID-Net, while the MID-Net-NoFusion removes all two fusion layers in the MID-Net. The U-Net can be regarded as a variation of HR-Net by replacing some convolutions in each subnetwork with skip connections and removing the part of fusions from two fusion layers. It extracts shape-level features from the bottleneck layer and outputs point-wise features from the last decoding layer.

As shown in Table 8, the accuracy of the network decreases as we drop more fusion layers from the baseline network (e.g., 0.4% for MID-Net-1Fusion and 2.1% for MID-Net-NoFusion in shape classification), which clearly demonstrates the importance of the feature fusion in MID-Net. Compared with the baseline, the accuracy of U-Net also drops 1.4% in shape classification and 0.5% in shape segmentation, which also validates the advantage of HR-Net in 3D pre-training.

*MID Loss versus single-level Loss.* To demonstrate the advantages of multi-resolution instance loss functions presented in the paper, we train two networks, each of which is trained with one loss function only. Compared with the MID-Net trained with two loss functions, the performance of these two networks (denoted as Shape-Loss-Only and Point-Loss-Only in Table 8) drops, which indicates that each loss function makes its own contribution to the full network training and affects the performance of shape encoding in both shape and point levels.

Table 8. Ablation study of different design choices of MID-Net.

Model	Shape Classification	3D Segmentation (C.mIoU)
MID-FC	90.3	85.5
MID-Net-1Fusion	89.9	85.3
MID-Net-NoFusion	88.2	83.1
U-Net	89.3	84.5
Point-Loss-Only	89.5	85.3
Shape-Loss-Only	88.9	83.3
NoRot-Augment	89.9	85.3
NoScale-Augment	89.2	84.3
NoTrans-Augment	89.7	85.1
No-Augment	82.3	83.9
50-Patch	90.1	84.9
200-Patch	90.0	85.5
400-Patch	90.2	85.4

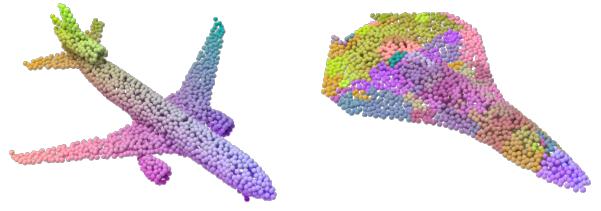


Fig. 8. A failure case of point-wise feature encoding. The heads of two airplanes have similar local geometry. However, our MID-Net does not offer similar point-wise features in these regions. Here the color map over 3D models encodes point-wise features.

And when the training data is limited, the performance gap is further enlarged: with only 1% of the training data, the testing accuracy on the classification task and category IoU on the segmentation task of our network trained with two loss functions are 61.8 and 74.6, respectively; with only the shape loss, the accuracy and IoU drop to 60.5 and 61.6; with only the point loss, the accuracy and IoU drop to 54.4 and 72.2.

*Augmentation scheme.* To study the impact of different augmentation schemes on the model performance, we train the network with three different augmentation schemes, each of which drops one kind of transformation (e.g., rotation, scaling, and translation) from the original augmentation scheme, respectively. Compared with the MID-Net trained with full augmentation, the performances of three networks trained with new augmentation schemes (denoted as NoRot-Augment, NoScale-Augment, and NoTrans-Augment shown in Table 8) decrease.

*Patch number.* To test the impact of the patch number on the model performance, we train the networks with different  $K$ . As shown in Table 8, as the number of clusters increases from 100 to 400, the performance of the resulting networks increases less than 0.2% in both shape segmentation and classification tests. We thus set  $K$  as 100 in our current implementation to achieve a good balance between training cost and model performance.

## 6 CONCLUSION

We propose an unsupervised pre-training method for learning a generic backbone network from unlabeled 3D shape collections for shape analysis. To this end, we design an octree-based HR-Net as backbone network architecture and a simple-yet-efficient MID loss for pre-training. The ablation study validates the advantages of joint shape and point feature encoding and training enabled by our design in the unsupervised pre-training and downstream shape analysis tasks. Our MID-Net offers state-of-the-art performance for various downstream shape analysis tasks, especially for tasks with small training sets.

Despite these advantages, our method still has a few limitations that can be resolved in future work.

*Scale to large dataset.* We reduce the computational cost and memory footprint of the point-instance discrimination loss by replacing it with the discrimination loss of over-segmented patches. However, this solution cannot totally resolve the issue as the scale of the data set significantly increases. An alternative solution to resolve this issue is the algorithm presented in the work [He et al. 2020]. It is interesting to implement this solution in our unsupervised 3D learning.

*Free-form 3D shapes.* The MID-Net presented in the paper is trained on the ShapeNet dataset that mainly consists of man-made and CAD-like objects. For shape analysis tasks on free-form 3D models, such as human body registration [Bogo et al. 2014], our initial experiments indicate that our pre-trained model does not provide any performance gain compared to a network trained from scratch. An interesting research direction is how to generalize our MID-Net to the shape analysis task of free-form shapes.

*Data imperfection.* The shape data used in our pre-training and the downstream analysis tasks are complete and noise-free. While the scanned 3D shapes are always incomplete and noisy. Another future research topic is to develop unsupervised 3D learning methods for scanned data.

*Shape correspondence.* For some 3D models with different overall shapes but similar local geometry, the pre-trained MID-Net cannot generate similar point-wise features at these local regions, which limits its application in 3D shape matching and correspondence. Figure 8 illustrates a failure case, in which the similar heads of two airplanes are encoded as different point-wise features.

*Pretext tasks and network structures.* A set of pretext tasks have been proposed for unsupervised 3D learning. It is interesting to explore the combination of our MID-Loss with other loss functions for unsupervised 3D learning. Also, it is interesting to develop efficient backbone network structures that can be used for different shape analysis tasks.

## REFERENCES

- Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. 2018. Learning representations and generative models for 3D point clouds. In *International Conference on Learning Representations (ICLR)*.
- Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. 2019. PointNetLK: Robust & Efficient Point Cloud Registration Using PointNet. In *Computer Vision and Pattern Recognition (CVPR)*.
- Matan Atzmon, Haggai Maron, and Yaron Lipman. 2018. Point Convolutional Neural Networks by Extension Operators. *ACM Trans. Graph. (SIGGRAPH)* 37, 4 (2018).
- Y. Bengio, A. Courville, and P. Vincent. 2013. Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (2013), 1798–1828.
- Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *Computer Vision and Pattern Recognition (CVPR)*.
- Davide Boscaini, Jonathan Masci, Simone Melzi, Michael M Bronstein, Umberto Castellani, and Pierre Vandergheynst. 2015. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Comput. Graph. Forum* 34, 5 (2015).
- Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael M. Bronstein. 2016. Learning shape correspondence with anisotropic convolutional neural networks. In *Neural Information Processing Systems (NeurIPS)*.
- Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision (ECCV)*.
- Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. *ShapeNet: An information-rich 3D model repository*. Technical Report arXiv:1512.03012 [cs.GR]. Stanford University – Princeton University – Toyota Technological Institute at Chicago.
- Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 2016. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *European Conference on Computer Vision (ECCV)*.
- Haowen Deng, Tolga Birdal, and Slobodan Ilic. 2018. PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors. In *European Conference on Computer Vision (ECCV)*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*.
- Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11, 19 (2010), 625–660.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016a. *Deep Learning*. MIT Press.
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2016b. Generative adversarial networks. In *Neural Information Processing Systems (NeurIPS)*.
- Ben Graham. 2015. Sparse 3D convolutional neural networks. In *British Machine Vision Conference (BMVC)*.
- Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 2018. 3D semantic segmentation with submanifold sparse convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)*.
- Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 2018. AtlasNet: A Papier-Mâché approach to learning 3D surface generation. In *Computer Vision and Pattern Recognition (CVPR)*.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Computer Vision and Pattern Recognition (CVPR)*.
- Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. MeshCNN: A network with an edge. *ACM Trans. Graph. (SIGGRAPH)* 38, 4, Article 90 (2019).
- Kaveh Hassani and Mike Haley. 2019. Unsupervised multi-task feature learning on point clouds. In *International Conference on Computer Vision (ICCV)*.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Computer Vision and Pattern Recognition (CVPR)*.
- K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*.
- Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 2017. 3D shape segmentation with projective convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)*.
- Hamid Laga, Yulan Guo, Hedi Tabia, Robert B. Fisher, and Mohammed Bennamoun. 2018. *3D Shape Analysis: Fundamentals, Theory, and Applications*. Wiley.
- Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. *International Conference on Learning Representations (ICLR)* (2017).
- Jiaxin Li, Ben M. Chen, and Gim Hee Lee. 2018b. SO-Net: Self-organizing network for point cloud analysis. In *Computer Vision and Pattern Recognition (CVPR)*.
- Xianzhi Li, Lequan Yu, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. 2020. Unsupervised detection of distinctive regions on 3D shapes. *ACM Trans. Graph.* (2020).
- Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhuan Di, and Baoquan Chen. 2018a. PointCNN: Convolution on X-transformed points. In *Neural Information Processing Systems (NeurIPS)*.

- Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. 2019. Relation-shape convolutional neural network for point cloud analysis. In *Computer Vision and Pattern Recognition (CVPR)*.
- Yu Liu, Guanglu Song, Jing Shao, Xiao Jin, and Xiaogang Wang. 2018. Transductive centroid projection for semi-supervised large-scale recognition. In *European Conference on Computer Vision (ECCV)*.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9 (2008).
- D. Maturana and S. Scherer. 2015. VoxNet: A 3D convolutional neural network for real-time object recognition. In *International Conference on Intelligent Robots and Systems (IROS)*.
- Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. 2019. PartNet: A Large-scale Benchmark for Fine-grained and Hierarchical Part-level 3D Object Understanding. In *Computer Vision and Pattern Recognition (CVPR)*.
- Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *Computer Vision and Pattern Recognition (CVPR)*.
- Tom Le Paine, Pooya Khorrami, Wei Han, and Thomas S Huang. 2015. An analysis of unsupervised pre-training in light of recent advances. In *ICLR (workshop)*.
- Deepak Pathak, Ross Girshick, Piotr Dollar, Trevor Darrell, and Bharath Hariharan. 2017. Learning features by watching objects move. In *Computer Vision and Pattern Recognition (CVPR)*.
- Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Computer Vision and Pattern Recognition (CVPR)*.
- Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Neural Information Processing Systems (NeurIPS)*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. *Improving language understanding by generative pre-training*. Technical Report, OpenAI.
- Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. 2017. OctNet: Learning deep 3D representations at high resolutions. In *Computer Vision and Pattern Recognition (CVPR)*.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*.
- Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Neural Information Processing Systems (NeurIPS)*.
- Jonathan Sauder and Bjarne Sievers. 2019. Self-supervised deep learning on point clouds by reconstructing space. In *Neural Information Processing Systems (NeurIPS)*.
- Zhenyu Shu, Chengwu Qi, Shiqing Xin, Chao Hu, Li Wang, Yu Zhang, and Ligang Liu. 2016. Unsupervised 3D shape segmentation and co-segmentation via deep learning. *Comput. Aided Geom. Des.* 43 (2016).
- Shuran Song. 2019. ModelNet benchmark leaderboard. <https://modelnet.cs.princeton.edu/>
- Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. 2018. SPLATNet: Sparse lattice networks for point cloud processing. In *Computer Vision and Pattern Recognition (CVPR)*.
- Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3D shape recognition. In *International Conference on Computer Vision (ICCV)*.
- Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Fleuret, and Leonidas J. Guibas. 2019. KPConv: Flexible and deformable convolution for point clouds. In *International Conference on Computer Vision (ICCV)*.
- Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018. Cosface: Large margin cosine loss for deep face recognition. In *Computer Vision and Pattern Recognition (CVPR)*.
- Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. 2019a. Deep high-resolution representation learning for visual recognition. arXiv preprint arXiv:1908.07919.
- Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. 2017. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Trans. Graph. (SIGGRAPH)* 36, 4 (2017).
- Yue Wang and Justin M. Solomon. 2019. Deep Closest Point: Learning Representations for Point Cloud Registration. In *International Conference on Computer Vision (ICCV)*.
- Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. 2019b. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* 38, 5 (2019).
- Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. 2016. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Neural Information Processing Systems (NeurIPS)*.
- Wenxuan Wu, Zhongang Qi, and Li Fuxin. 2019. PointConv: Deep Convolutional Networks on 3D Point Clouds. In *Computer Vision and Pattern Recognition (CVPR)*.
- Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D ShapeNets: A deep representation for volumetric shape modeling. In *Computer Vision and Pattern Recognition (CVPR)*.
- Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *Computer Vision and Pattern Recognition (CVPR)*.
- Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. 2018. SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters. In *European Conference on Computer Vision (ECCV)*.
- Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. 2015. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 11 (2015).
- Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. 2018. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *Computer Vision and Pattern Recognition (CVPR)*.
- Li Yi, Lin Shao, Manolis Savva, and et al. 2017a. Large-scale 3D shape reconstruction and segmentation from ShapeNet Core55. arXiv:1710.06104 [cs.CV].
- Li Yi, Hao Su, Xingwen Guo, and Leonidas Guibas. 2017b. SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. In *Computer Vision and Pattern Recognition (CVPR)*.
- Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. 2019. PartNet: A Recursive Part Decomposition Network for Fine-grained and Hierarchical Shape Segmentation. In *CVPR*.
- Ling Zhang and Zhigang Zhu. 2019. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. In *Proc. Int. Conf. on 3D Vision (3DV)*.
- Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*.
- Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 2019. 3D point capsule networks. In *Computer Vision and Pattern Recognition (CVPR)*.
- Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. 2016. Fast global registration. In *European Conference on Computer Vision (ECCV)*.