# 3D Intracranial Aneurysm Classification and Segmentation via Unsupervised Dual-Branch Learning

Di Shao, *Member, IEEE*, Xuequan Lu, *Member, IEEE*, and Xiao Liu, *Senior Member, IEEE*

***Abstract*—Intracranial aneurysms are common nowadays and how to detect them intelligently is of great significance in digital health. Whereas most existing deep learning research focused on medical images in a supervised way, we introduce an unsupervised method for the detection of intracranial aneurysms based on 3D point cloud data. In particular, our method consists of two stages: unsupervised pre-training and downstream tasks. As for the former, the main idea is to pair each point cloud with its jittering counterpart and maximise their correspondence. Then we design a dual-branch contrastive network with an encoder for each branch and a subsequent common projection head. As for the latter, we design simple networks for supervised classification and segmentation training. Experiments on the public dataset (IntrA) show that our unsupervised method achieves comparable or even better performance than some state-of-the-art supervised techniques, and it is most prominent in the detection of aneurysmal vessels. Experiments on the ModelNet-40 also show that our method achieves the accuracy of 90.79% which outperforms existing state-of-the-art unsupervised models.**

***Index Terms*—Deep learning, point cloud, three-dimensional (3D) intracranial aneurysm detection, unsupervised learning.**

## I. Introduction

INTRACRANIAL aneurysms can result in a high rate of mortality, and their classification and segmentation are of great significance. Existing research mainly focused on image data which involves regular pixels [1]–[6]. Whereas 3D geometric data such as point clouds can depict more useful information, the research on analysing intracranial aneurysms using point cloud data has been very sparsely exploited to date. Thanks to [7], a point cloud dataset including aneurysmal segments and healthy vessel segments has been published. They have conducted a benchmark using state-of-the-art point-based networks that can directly consume 3D points instead of 2D pixels. There has been research using the IntrA dataset [8], [9], and nearly all are supervised. Unfortunately, unsupervised learning remains underexplored on the IntrA dataset.

There are many networks available for consuming point cloud data, for example, PointNet [10], PointNet++ [11], SpiderCNN [12], PointCNN [13], SO-Net [14] and DGCNN [15]. PointNet is a seminal method for taking 3D points as input and used for 3D point cloud classification and segmentation. Later on, other point-based methods have been proposed to improve the performance. Since they are all supervised learning methods, annotated data are required for training. However, annotation often requires experts and significant amounts of time, especially for large datasets and medical data. The unsupervised learning methods can effectively reduce labelling costs, offering potential for decent performance in a limited label situation.

With the above analysis in mind, we design an unsupervised representation learning method that consumes point clouds of vessel segments. The contrastive learning concept inspires our method. In particular, we first generate a pair of augmented samples of the original point cloud which should have a distinct difference. Next, we design a dual-branch contrastive network with an encoder for each branch and a follow-up common projection head to facilitate the unsupervised training with a contrastive loss. As for the downstream tasks, we first use the unsupervised trained model to output the representations. Then, we design simple networks and train them by taking the representations as input to classify or segment intracranial aneurysms. Note that we design two unsupervised networks and two corresponding downstream networks to fulfil two different tasks (i.e. classification and segmentation). Supervised methods often need a large scale of labelled data for achieving satisfactory performance. Compared with them, our method does not require labels in unsupervised training, and it can utilise a small scale of labelled data for downstream training. In summary, our contributions in this paper include:

- We propose a simple yet effective method for unsupervised representation learning on 3D point clouds of vessel segments.
- We find an effective augmentation method for generating pairs of each vessel segment.
- We propose a dual-branch contrastive network with an encoder for each branch.
- We demonstrate the effectiveness of our method in a limited label situation. We conduct comprehensive

Manuscript received 8 February 2022; revised 5 May 2022; accepted 28 May 2022. Date of publication 13 June 2022; date of current version 5 April 2023. *(Corresponding authors: Xuequan Lu; Xiao Liu.)*

The authors are with the School of Information Technology, Deakin University, Geelong, VIC 3216, Australia (e-mail: shaod@deakin.edu.au; xuequan.lu@deakin.edu.au; xiao.liu@deakin.edu.au).

Digital Object Identifier 10.1109/JBHI.2022.3180326

experiments and compare with state-of-the-art point-based techniques to demonstrate the superior performance of our method.

## II. Related Work

### A. Deep Learning on Intracranial Aneurysms

Intracranial aneurysms are associated with a high mortality rate. Therefore, the detection of intracranial aneurysms is crucial for human health. Traditional methods rely greatly on prior knowledge, which is often inferior to deep learning in terms of capability and accuracy. Due to the excellent performance of deep learning in processing medical images, there are many deep learning methods to detect intracranial aneurysms [16]. Nakao *et al.* [1] proposed a convolutional neural network-based detection system. The system used a 6-layer (CNN) and maximum intensity projection (MIP) algorithm based on the MRA images. This method can achieve almost 100% accuracy for detecting aneurysms greater than 7 mm in diameter. However, it was less sensitive to small vascular aneurysms. To improve this, Joseph *et al.* [2] used the full U-net convolution architecture to predict aneurysm size based on the detection. Ueda *et al.* [3] applied the ResNet-18 network to the MRA images and performed a secondary evaluation on the already detected image data to enhance the detection sensitivity. To better segment the shape of intracranial aneurysms, Sichtermann *et al.* [4] utilized DeepMedic [17] with 2-pathway architecture and 11-layer convolution to segment intracranial aneurysms from the MRA images on the basis of detection. The above method for detecting intracranial aneurysms used data that are stacked with 2D images. To sum up, nearly all works focused on dealing with medical images rather than 3D geometry like point clouds.

### B. Point-Based Networks

Neural network models for the classification and segmentation of 3D point cloud data have achieved noticeable success. Qi *et al.* [10] proposed PointNet to directly process point sets. To obtain permutation invariance and transformation invariance of point clouds, PointNet used the symmetric function and T-net to design the network. It had good results for global features extraction of point clouds. However, it ignored the geometric relationship among points and limited the extraction of local features. To address this problem, Qi *et al.* [11] proposed PointNet++ using a hierarchical neural network. It used the point sampling and grouping strategy to extract local features of point clouds. However, PointNet++ did not reveal the spatial distribution of the input point cloud. Li *et al.* [14] constructed the Self-Organizing Map (SOM) [18] to model the spatial distribution of the input point cloud called SO-Net. It allowed SO-Net to adjust the receptive field overlap and performed hierarchical feature extraction. Unlike SO-Net with adjusting the perceptual field of the hierarchical network, Li *et al.* [13] proposed the $\chi$-transformation to process the point cloud data so that the point cloud data can be weighted or permuted. Thus, it improved the extraction of local features. In addition, Xu *et al.* [12] proposed SpiderConv, i.e. parameterized convolutional filters, to implement convolutional operations on disordered point clouds. Wang *et al.* [15] proposed a convolutional-like operation by constructing local neighbourhood graphs and applying convolutional operations on the edges. It connected adjacent point pairs to exploit the local geometric structure. Gong *et al.* [19] proposed boundary-aware geometric encoding for semantic segmentation. Meanwhile, Gong *et al.* [20] proposed omni-supervised gradual receptive field component reasoning for point cloud segmentation. In addition to common tasks like classification and segmentation, point based networks have also been developed to address other tasks [21], [22]. In summary, there has been great progress in analyzing point cloud data in a supervised manner.

### C. Unsupervised 3D Point Cloud Learning

All the above deep neural networks can classify and segment the point cloud data well. However, considering the complexity of labelling 3D data, it is difficult to get enough data with expert labelling for supervised training in many scenarios. Therefore, it is meaningful to exploit unsupervised or self-supervised learning for 3D point cloud data. Xie *et al.* [23] proposed PointContrast which is an unsupervised framework with U-net as the backbone network. And it demonstrated the transferability of representation learning to 3D point cloud data and the performance enhancement of pre-training to downstream tasks. Lu *et al.* [24], [25] attempted to address skeleton learning on point cloud sequence data. Jiang *et al.* [26] introduced a simple yet effective unsupervised learning method on point cloud that only considers rotation as the transformation. Sanghi *et al.* [27] proposed to extend the InfoMax [28] and contrastive learning principles on 3D shapes. It maximized the mutual information between 3D objects and their “chunks” to improve the representation in the aligned dataset. Yang *et al.* [29] proposed FoldingNet which is an autoencoder with graph pooling and MLP layers using the folding operation to deform 2D grids into object surfaces. However, in 3D medical point cloud data, unsupervised methods are still in great demand. We propose an unsupervised representation learning method, which shows excellent performance for the classification and segmentation of point cloud based intracranial aneurysms.

## III. Method

Our method consists of two stages which are unsupervised learning and downstream tasks. In stage 1, we first perform augmentation on each point cloud to get a pair of augmented samples which are different in pose (Section III-A). We then get two representations of a pair of data in a high-dimensional space by means of the dual-branch encoders, which enables each branch of the encoder to extract distinct features. Next, we map representations to a low-dimensional vector [30] with a projection head to improve network training speed (Section III-B). Last, we employ a contrastive loss to encourage the representations of the pair of point clouds output by the encoders to be similar in the high-dimensional space (Section III-C). In stage 2, the output of the trained model unsupervised representation is concatenated and used as the input for downstream tasks. (Section III-D). The

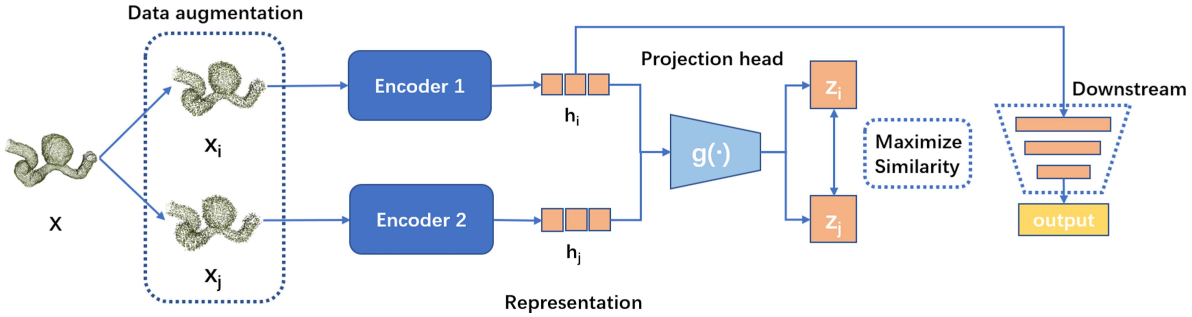

Fig. 1. The architecture of our method includes data augmentation, encoder, projection head, loss function and downstream tasks. We first jitter a point cloud $x$ to construct a pair $(x_i, x_j)$. The representation vectors $z_i$ and $z_j$ to the pair of point clouds are then extracted via the dual-branch encoders and mapping head, and network optimisation is performed by a contrastive loss. The representation $h$ obtained by dual-branch encoders will be used for downstream tasks.

downstream task will evaluate the effectiveness of unsupervised learning. Fig. 1 presents the architecture of our method.

## A. Data Augmentation

We use data augmentation to generate different samples for each point cloud. To generate a pair of data, we consider using data augmentation methods, including jittering, perturbation, crop and rotation transformations. After experiments, it was found that *jittering* as data augmentation in both branches gave the best results in the downstream tasks, indicating a more discriminative representation learned by the upstream network. Ablation experiments will be presented in Section IV-D.

We take a batch of point clouds with mini-batch size $N$ and input them into the data augmentation module. As shown in the top of Fig. 1, for a sample in the mini-batch, we use the jittering function to obtain a pair of samples: one is the jittering point cloud $x_i$ and the other is the jittering point cloud $x_j$. In this way, we have a batch size of $2N$ in this mini-batch. We randomly select $\{x_i, x_j\}$ as a positive pair, and the other $N-1$ pairs, which consist of one of positive sample and one of the other samples, are regarded as negative samples in this mini-batch.

## B. Dual-Branch Encoders and Projection Head

As shown in Fig. 1, each pair of samples needs to be passed through dual-branch encoders $f(\cdot)$ to obtain two representations $h_i$ and $h_j$. Features are respectively extracted from a pair of data using two different encoders. Experimentally, we have also compared two different encoders with a common encoder. Ablation experiments will be presented in Section IV-D. Two encoders are PointNet [10] and PointNet++ [11], respectively. The reason for choosing PointNet and PointNet++ is that PointNet can extract global features while PointNet++ can extract local features. This design highlights distinctions in features and allows for more distinctive representation.

**Classification.** The first encoder utilizes three consecutive 1D convolutional layers and a max-pooling layer to obtain the representation vector $h$ (1024-dimensional).

The second encoder consists of three abstraction levels. Each level abstracts and processes the point set to create a new point set with fewer elements. The abstraction level consists of three key layers: sampling layer, grouping layer and PointNet layer. The sampling layer selects a set of points from input points, which defines the centroids of local regions. The grouping layer then constructs local region sets by finding "neighbouring" points around the centroids. PointNet layer uses a mini-PointNet to encode local region patterns into feature vectors. The input to the abstraction level consists of an $n \times (d + c)$ matrix formed by $n$ points with $d$-dim coordinates and $c$-dim point features. $n$ is the number of points in a point cloud sample. A point set group of size $n_1 \times k \times (d + c)$ is the output by sampling $n_1$ centroids and grouping them, where each group corresponds to a local region. $k$ is the number of points sampled from the centroid point's neighbourhood. The subsequent PointNet layer outputs a local region feature vector $n_1 \times (d + c_1)$. We take all the sampled points as a group in the last abstraction level and output the representation vector $h$ (1024-dimensional). We design three linear layers as our projection head $g(\cdot)$ to map each 1024-dimensional representation vector to a 128-dimensional vector $z$.

**Segmentation.** The segmentation encoders are based on the encoders for classification. The first encoder output the 1024-dimensional representation vector. It is copied $n$ times to form an $n \times 1024$ tensor. Concatenating it with the $n \times 1024$ tensor obtained from the last convolutional layer gives a $n \times 2048$ tensor. As such, this tensor contains both global features and features for each point.

The second encoder extends the classification's second encoder by adding three propagation levels. We adopt distance-based interpolation and a skip-link across levels propagation strategy. As shown in the left bottom of Fig. 2, in a feature propagation level, we propagate point features from $n_1 \times (d + c_1)$ points to $n$ points. We achieve feature propagation by concatenating interpolating feature $c_1$ with point features $c$ from the set abstraction level. It outputs a $n \times (d + c_1 + c)$ vector, which is then passed through the unit PointNet to obtain a $n \times 1024$ tensor. The 1024-dimensional vector output from the abstraction layer is copied $n$ times and concatenated with the $n \times 1024$-dimensional tensor output from the propagation level to obtain the final $n \times 2048$ representation tensor.

The tensors obtained by both encoders are max-pooled separately to obtain a 2048-dimensional representation vector. These two vectors are used as the feature representation $h$ for the downstream segmentation network. We design two linear layers

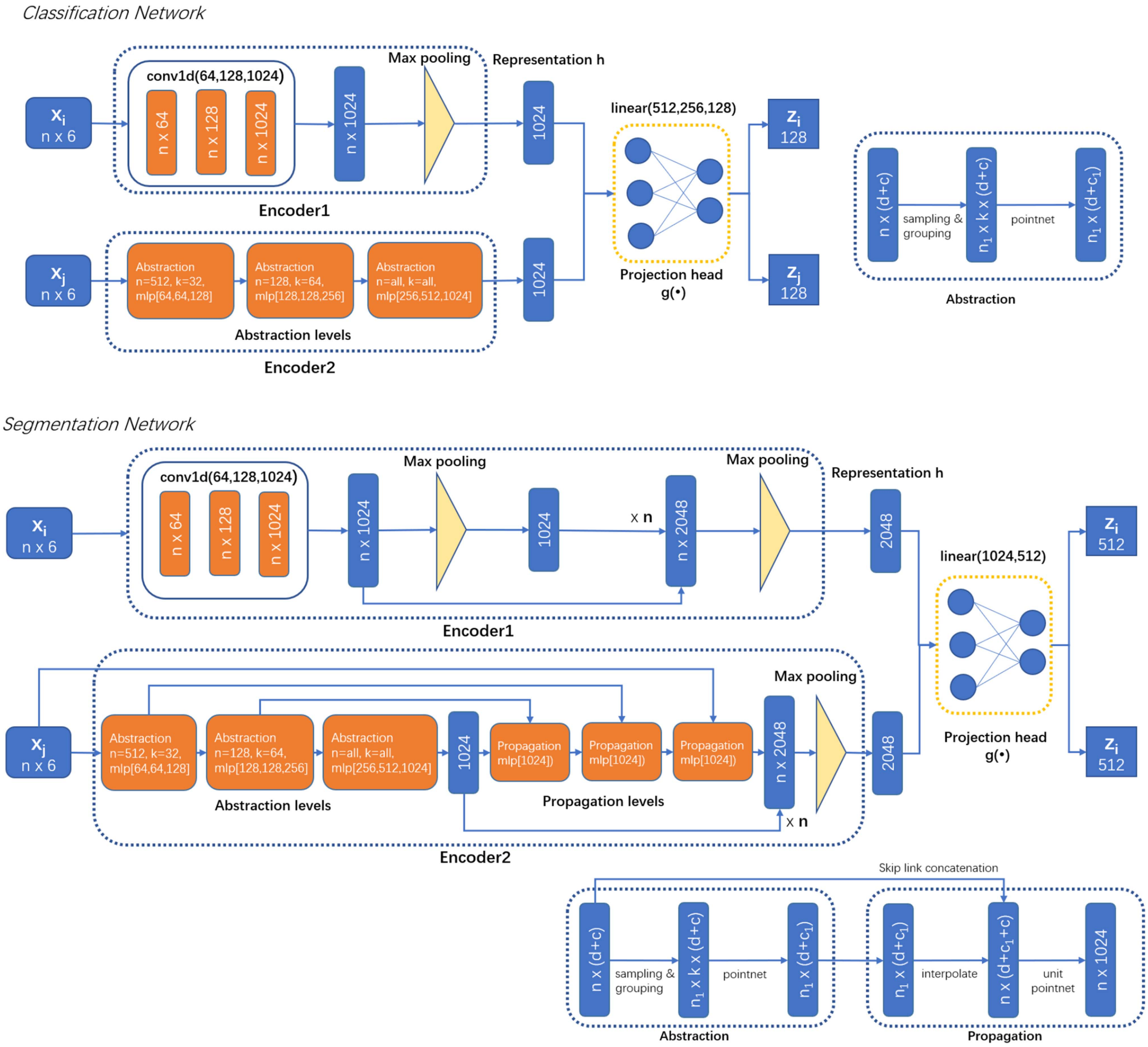

Fig. 2. Dual-branch Encoders and projection head. “conv1d”: 1D convolution, “linear”: fully connected layers, “mlp” stand for multi-layer perceptron. The numbers in brackets represent the layer sizes. All convolutions and fully connected layers include batchnorm and ELU.

as our projection head $g(\cdot)$ to map each 2048-dimensional representation vector to a 512-dimensional vector $z$.

### C. Contrastive Loss

We use a contrastive loss function similar to [31]. With this loss function, unsupervised learning can effectively learn separable features for point clouds. After the projection head $g(\cdot)$, for each sample in the mini-batch, we obtain the projection representation $z$. For the pair $x_i$ and $x_j$, we use their projection representations $z_i$ and $z_j$ to measure the cosine similarity between the two samples as follows:

$$s_{i,j} = \frac{{z_i}^{\top} z_j}{(\|z_i\| \, \|z_j\|)}. \tag{1}$$

Intuitively, the similarity for a positive sample pair should be high. A combination pair of a positive and a negative sample should be low. Then, we use it to get a similar probability of each positive sample pair in a mini-batch. $\mathbb{1}_{[k\neq i]} \in \{0,1\}$ is an indicator function evaluating to 1 iff $k \neq i$. The equation for calculating the probability of similarity is as follows:

$$S(i,j) = \frac{exp(s_{i,j})}{\sum_{k=1}^{2N} \mathbb{1}_{k\neq i} exp(s_{i,k})} \tag{2}$$

We use the negative logarithm to calculate the loss of the sample pair. This loss has been used in previous works [32]–[34]. $\tau$ denotes a temperature parameter which scales the input and expands the range of cosine similarity. This loss is known as the normalized temperature-scaled cross-entropy loss [35], [36] as follows:

$$l(i,j) = -log \frac{exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k\neq i} exp(s_{i,k}/\tau)} \tag{3}$$

We calculate the average loss of both $(i,j)$ and $(j,i)$ in the mini-batch. Based on this loss, the representation of the encoder and projection head improves over time, and the trained network places similar samples closer in the representation space. Specifically, the loss function is given by:

$$L = \frac{1}{2\,N}\sum_{k=1}^{N}[l(2k-1,2\,k)+l(2\,k,2k-1)]. \tag{4}$$

### D. Downstream Tasks

We design two simple downstream networks to evaluate the unsupervised learned representations for classification and segmentation, respectively. Each point cloud of a vessel segment is fed into the unsupervised dual-branch encoders to obtain two representations. We then concatenate the two representations into one and use this representation as input to train the downstream network. As for the classification task, we use four linear layers (512, 256, 128, 2) as the downstream network. Regarding the segmentation task, we employ four 1D convolutional layers (1024, 512, $m$), where $m$ is the number of segmentation labels.

## IV. Evaluation

### A. Datasets

**IntrA** [7] consists of complete models of aneurysms, generated vessel segments and annotated aneurysm segments. IntrA collected 103 3D models of the entire cerebral vasculature by reconstructing 2D MRA images scanned for patients. IntrA generated 1,909 vessel segments from the complete model, including 1,694 healthy vessel segments and 215 aneurysmal segments. Additionally, 116 aneurysm segments were manually annotated for each point. In IntrA, each sample was represented as a 3D point cloud. Each point $p$ is a 6D vector composed of its coordinates and normal vector. Following IntrA, we combined the generated vessel segments and manually annotated aneurysms to achieve a total of 2,025 samples. These 2,025 vessel segments will be used as the dataset for our unsupervised training. All 2,025 vessel segments will be used for the downstream classification task. 116 annotated aneurysm segments will be used for the downstream segmentation task.

**ModelNet-40** [36] is a collection of 40 categories and 12,311 models culled from the mesh surfaces of CAD models. Following previous practice, 9,843 models are employed for training and 2,468 for testing. Each point cloud has 2,048 points, and all of the points' coordinates are normalised to the unit sphere. Each point is a 6D vector made up of its coordinate and a normal vector. We take 1,024 points from each point cloud and augment the data by jittering. This dataset is used for comparisons of our method with other unsupervised methods.

**ModelNet-10** [36] is taken from the ModelNet-40 dataset. For ModelNet-10, 3,991 models are employed for training and 908 models are employed for testing. It contains 10 categories. For each model, we use 1,024 points as input.

**ShapeNetCore** [37] contains 51,127 pre-aligned shapes from 55 categories, which has 35,708 models for training, 5,158 models for validation and 10,261 models for testing. We use 1,024 points as input which is the same as PointContrast.

### B. Experimental Setting

For unsupervised training, we use the Adam optimizer with a weight decay of $10^{-6}$. The mini-batch size is set to 32. The number of epochs is 200. The initial learning rate is $10^{-3}$. The learning rate is scheduled to be multiplied by 0.5 every 10 epochs. We use jittering as the data augmentation method, which directly adds Guassian noise to every coordinate and normal information of input point clouds. In encoder 2, the number of points sampled from the centroid point's neighbourhood $k$ is set to [32, 64, "None"]. "None" means that all points are sampled. The projection head outputs a feature $z$, the dimension of which is set to 128 for the classification task and 512 for the segmentation task. In the loss function, the temperature parameter $\tau$ is set to 0.5.

For the downstream network, the optimizer, the number of epochs, representation dimensions, initial learning rate, learning rate decay schedule and mini-batch size are the same as those in unsupervised training. We sample 512,1024 and 2048 points separately in each point cloud for both experiments. For the classification task, weight decay is set to $10^{-6}$ and the size of linear is set to [512, 256, 128, $a$]. $a$ is the number of categories in the point cloud. For the segmentation task, weight decay is set to 1.0 and the size of MLPs is set to [1024, 512, $b$]. $b$ is the number of categories of points in the point cloud.

Experiments were implemented using PyTorch on a GeForce GTX 1080 GPU. For IntrA dataset, the time for both unsupervised training on classification and segmentation is approximately 1 hour. The downstream classification and segmentation tasks are approximately 40 minutes and 50 minutes, respectively.

### C. Experimental Results

We evaluate the classification task and the segmentation task separately on **IntrA** [7]. To demonstrate the generalisation of our method, we also perform the classification task on **ModelNet-40**, **ModelNet-10** [36] and **ShapeNetCore** [37], then compare our method with start-of-the-art unsupervised methods to verify the effectiveness of our method.

**Classification task.** On IntrA, we evaluate the performance using three metrics: (1) V. Accuracy, measuring the percentage of correctly predicted healthy vessels' samples overall healthy vessels' samples, (2) A. Accuracy, indicating the percentage of correctly predicted aneurysm vessels' samples overall aneurysm vessels' samples, (3) F1 score, representing the harmonic average of precision and recall and evaluating the quality of the model. On ModelNet-40, ModelNet-10 and ShapeNetCore, we evaluate the performance using overall accuracy.

As shown in Table I, as for our dual-branch encoders method with PointNet and PointNet++ backbones (i.e. PN, PN++), 1,024 sample points have the best results in terms of the F1 score and V. Accuracy compared with other numbers of sample points. The results for 512 sample points are still impressive, though the number of points in each point cloud is much smaller. Although the 2,048 sample point result is not the best in terms of F1 score and V. Accuracy. *Notice that our method in 2,048 sample points has the best A. Accuracy results compared with all mentioned*

TABLE I

CLASSIFICATION RESULTS OF EACH METHOD. THE ADDITIONAL INPUT $K$ IS REQUIRED FOR DGCNN. PN: POINTNET, PN++: POINTNET++

| | Network | #.Points | V.(%) | A.(%) | F1-Score |
|---|---|---|---|---|---|
| Supervised | SpiderCNN [12] | 512 | 98.05 | 84.58 | 0.8692 |
| | | 1024 | 97.28 | 87.9 | 0.8722 |
| | | 2048 | 97.82 | 84.89 | 0.8662 |
| | SO-Net [14] | 512 | 98.76 | 84.24 | 0.8840 |
| | | 1024 | 98.88 | 81.21 | 0.8684 |
| | | 2048 | 98.88 | 83.94 | 0.8850 |
| | PointCNN [13] | 512 | 98.38 | 78.25 | 0.8494 |
| | | 1024 | 98.79 | 81.28 | 0.8748 |
| | | 2048 | **98.95** | 85.81 | **0.9044** |
| | DGCNN [15] | 512/10 | 95.22 | 60.73 | 0.6578 |
| | | 1024/20 | 95.34 | 72.21 | 0.7376 |
| | | 2048/40 | 97.93 | 83.40 | 0.8594 |
| | PointNet++ [11] | 512 | 98.52 | 86.69 | 0.8928 |
| | | 1024 | 98.52 | 88.51 | 0.9029 |
| | | 2048 | 98.76 | **87.31** | 0.9016 |
| | PointNet [10] | 512 | 94.45 | 67.66 | 0.6909 |
| | | 1024 | 94.98 | 64.96 | 0.6835 |
| | | 2048 | 93.74 | 69.50 | 0.6916 |
| Unsupervised | FoldingNet [29] | 512 | 91.37 | 77.41 | 0.6159 |
| | | 1024 | 91.83 | 78.28 | 0.6241 |
| | | 2048 | 91.64 | 79.54 | 0.6316 |
| | Our(single_PN) | 512 | 94.33 | 75.55 | 0.7233 |
| | | 1024 | 94.21 | 78.33 | 0.7424 |
| | | 2048 | 94.84 | 77.05 | 0.7408 |
| | Our(single_PN++) | 512 | 95.33 | 80.55 | 0.7679 |
| | | 1024 | 95.63 | 83.60 | 0.7968 |
| | | 2048 | 95.74 | 83.41 | 0.7988 |
| | Our(dual) | 512 | 96.74 | 82.35 | 0.8296 |
| | | 1024 | **97.45** | 84.28 | **0.8613** |
| | | 2048 | 95.41 | **89.47** | 0.8226 |

Bold numbers indicate the best results of supervised methods and unsupervised methods.

*methods. The ability to identify aneurysms is essential in this case.* Furthermore, we find that the A. Accuracy increase with more sample points. Compared with other supervised methods, our results are better than the supervised PointNet in all metrics. Besides, it also outperforms more advanced supervised networks such as DGCNN in general. The result of 1,024 sample points is very close to SO-Net and SpiderCNN in terms of F1-Score. We also compare our method with FoldingNet, one of the most representative unsupervised methods. Obviously, our method performs better on all metrics. The effectiveness of unsupervised learning is inherently limited due to the unsupervised nature, and the tubular structure of intracranial aneurysms is much less prominent compared with other data. It causes our method (dual) to perform less well than supervised PN++.

As shown in Table II, we use the test accuracy as the evaluation of our method and compare the performance of our model with other unsupervised methods on ModelNet-40 and ModelNet-10 [36]. We can see that our method outperforms most unsupervised methods, such as two recent methods

TABLE II

CLASSIFICATION ACCURACY OF UNSUPERVISED METHODS ON MODELNET-40 AND MODELNET-10

| Method | ModelNet-40(%) | ModelNet-10(%) |
|---|---|---|
| SPH [38] | 68.2% | - |
| LFD [39] | 75.5% | - |
| T-L Network [40] | 74.4% | - |
| VConv-DAE [41] | 75.5% | - |
| 3D-GAN [42] | 83.3% | - |
| FoldingNet [29] | 84.36% | 91.85% |
| FoldingNet* [29] | 88.40% | 94.40% |
| Latent-GAN [43] | 87.27% | 92.18% |
| Latent-GAN* [43] | 85.70% | 95.30% |
| PointCapsNet [44] | 87.46% | - |
| PointCapsNet* [44] | 88.9% | - |
| MultiTask [45] | 89.1% | - |
| PointHop [46] | 89.1% | - |
| MAP-VAE [47] | 90.15% | 94.82% |
| Our(dual) | **90.79%** | **95.01%** |

Bold numbers indicate the best results of unsupervised methods on ModelNet40 and ModelNet10; * means the model is trained on ShapeNet55.

TABLE III
COMPARISON RESULTS BETWEEN POINTCONTRAST AND OUR METHOD ON THE SHAPENETCORE DATASET WITH PRE-TRAINING EVALUATION

| Method | Accuracy(%) |
|---|---|
| PointContrast* [23] | 85.7% |
| Our(dual)** | **86.65%** |

* means the model is trained on ScanNet, ** Means the model is trained on ModelNet-40.
Bold number indicates the best result.

PointHop (1.69% gain) and MAP-VAE (0.64% gain) on ModelNet-40 and three recent methods MAP-VAE (0.19% gain), Latent-GAN (2.83% gain) and FoldingNet (3.16% gain) on ModelNet-10. This confirms the effectiveness of our unsupervised representation learning.

PointContrast [23] is a relevant contrastive learning based method to our work. PointContrast is based on the point-wise level while ours is based on point cloud level. PointContrast is pretrained on a rather large scene dataset (ScanNet). However, it is not suitable for pre-training our method on ScanNet, since this downstream task is for classification (requiring point cloud level features for classification) while ScanNet has point-wise labels. To provide a plausible comparison with it, we pretrain our model on ModelNet-40 which is much smaller than ScanNet and use the ShapeNetCore dataset for the classification task with finetuning. The comparison results are shown in Table III, and we can see that our method (dual) outperforms it by 0.95%. The good performance of our method is mainly due to the effective point cloud level contrastive learning that can better obtain global features.

**Segmentation task.** Following [7], we evaluate the segmentation performance using two metrics: (1) V. IoU, indicating the IoU of healthy vessels, and (2) A. IoU, indicating the IoU of aneurysm vessels.

As shown in Table IV, as for our method (dual), 1,024 sample points have the best results in terms of V. IoU. But 2,048 sample points have the best results in terms of A. IoU. In comparison, our method outperforms the supervised PointNet on both V. IoU and A. IoU. *Notice that our method is better than more advanced supervised networks like PointGrid.* Compared to the supervised PointNet, which is trained with only 116 labelled samples, our method is able to learn unsupervised features from a much wider range of data, thus facilitating downstream network training. Our method generally generates better results with increasing the point number, and produces better results than the supervised PointNet in both metrics. Our method (PN++) is still inferior to the supervised PointNet++, which is considered to be limited by the unsupervised nature. Our segmentation task achieves less decent performance than the classification task. It is mainly due to the fact that the segmentation task is more challenging than classification and requires a considerable amount of data to achieve good results. Therefore, with the same amount of data for training, it is a bit difficult for unsupervised methods to outperform supervised methods. Fortunately, our framework can still exceed several supervised methods.

TABLE IV
SEGMENTATION RESULTS OF EACH METHOD

| Network | #.Points | IoU_V.(%) | IoU_A.(%) |
|---|---|---|---|
| SO-Net [14] | 512 | 94.22 | 80.14 |
| | 1024 | 94.42 | 80.99 |
| | 2048 | **94.46** | **81.40** |
| PN++ [11] | 512 | 93.42 | 76.22 |
| | 1024 | 93.35 | 76.38 |
| | 2048 | 93.24 | 76.21 |
| PointCNN [13] | 512 | 92.49 | 70.65 |
| | 1024 | 93.47 | 74.11 |
| | 2048 | 93.59 | 73.58 |
| SpiderCNN [12] | 512 | 90.16 | 67.25 |
| | 1024 | 87.95 | 61.60 |
| | 2048 | 87.02 | 58.32 |
| PointGrid [48] | 16/2 | 78.32 | 35.82 |
| | 16/4 | 79.49 | 38.23 |
| | 32/2 | 80.11 | 42.42 |
| PointNet [10] | 512 | 73.99 | 37.30 |
| | 1024 | 75.23 | 37.07 |
| | 2048 | 74.22 | 37.75 |
| Our(PN) | 512 | 80.05 | 44.66 |
| | 1024 | 82.54 | 46.55 |
| | 2048 | 81.65 | 48.45 |
| Our(PN++) | 512 | 80.05 | 40.66 |
| | 1024 | 82.05 | 41.42 |
| | 2048 | 82.65 | 42.45 |
| Our(dual) | 512 | 82.25 | 48.66 |
| | 1024 | 84.35 | 50.92 |
| | 2048 | 82.65 | 51.45 |

Bold numbers indicate the best segmentation results.

**Reduction of labels cost.** In real-world situations, we frequently lack sufficient labelled data. One of the important motivations for studying unsupervised methods is to reduce the labelling cost of training. Therefore, it is necessary to evaluate the performance of our method with limited labels experimentally. Due to the uniqueness of the IntrA dataset, it is difficult to find another dataset similar to it. To achieve a similar goal, we divide the IntrA dataset into two parts, A and B. We perform unsupervised pre-training on dataset A, i.e., without using any labels from the A dataset. Then we finetune it on the target dataset B with a limited number of labels. The percentages of labelled data are set to 100%, 75%, 50%, 25% and 10%, respectively. Through experiments, we can simulate the results with limited labels and can evaluate how our method performs in a limited label situation. The setting such as batch size, learning rate etc., are the same as the classification task on IntrA. The comparison results are shown in Table V, and we can see the accuracy of the classification gradually decreases as the amount of labelled data decreases. With 100% labels, the performance of our method is generally better than PointNet++ but not much, and V. Accaurcy of PointNet++ is even better than our method. However, as the labels decrease, the accuracy decay of PointNet++ is much more obvious than our method. With 50% labels, our method's

TABLE V
PERFORMANCE OF OUR METHOD WITH LIMITED LABELS

| Label(%) | Network | V.(%) | A.(%) | F1-Score |
|---|---|---|---|---|
| 100% | PointNet | 94.37 | 74.36 | 0.7250 |
| | PointNet++ | **98.73** | 87.80 | 0.9020 |
| | Our(dual) | 98.67 | **92.85** | **0.9197** |
| 75% | PointNet | 93.06 | 72.22 | 0.6753 |
| | PointNet++ | **97.76** | 86.84 | 0.8654 |
| | Our(dual) | 97.10 | **92.10** | **0.8889** |
| 50% | PointNet | 89.43 | 68.02 | 0.5151 |
| | PointNet++ | 96.10 | 80.78 | 0.8148 |
| | Our(dual) | **96.82** | **87.17** | **0.8595** |
| 25% | PointNet | 87.23 | 64.70 | 0.3793 |
| | PointNet++ | 95.87 | 73.72 | 0.7419 |
| | Our(dual) | **96.09** | **83.41** | **0.8019** |
| 10% | PointNet | 86.01 | 51.85 | 0.3078 |
| | PointNet++ | 93.82 | 65.69 | 0.6617 |
| | Our(dual) | **95.34** | **78.19** | **0.7494** |

Bold numbers indicate the best results.

TABLE VI
ABLATION STUDY ON AUGMENTATION

| Augmentation | V.(%) | A.(%) | F1-Score |
|---|---|---|---|
| rotation | 95.23 | 75.52 | 0.7637 |
| perturbation | 95.35 | 81.87 | 0.8121 |
| crop | 96.53 | 83.58 | 0.8358 |
| jittering & perturbation | 95.63 | 81.76 | 0.8240 |
| jittering | **97.45** | **84.28** | **0.8613** |

Bold numbers indicate the best results.

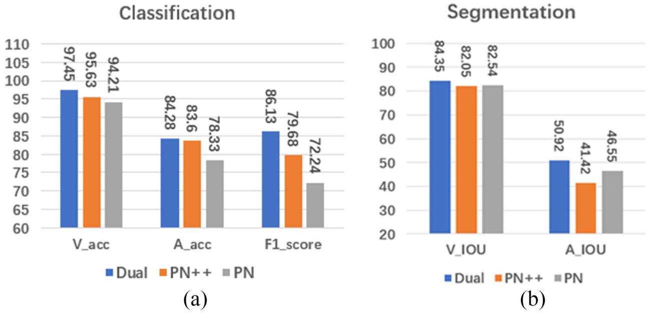

Fig. 3. Ablation study on dual-branch encoders in terms of (a) Classification. (b) Segmentation.

TABLE VII
ABLATION STUDY FOR ROBUSTNESS TO ROTATION

| Transform | Network | V.(%) | A.(%) | F1-Score |
|---|---|---|---|---|
| None | PointNet | 94.98 | 64.96 | 0.6835 |
| | PointNet++ | **98.52** | **88.51** | **0.9029** |
| | Our(dual) | 97.45 | 84.28 | 0.8613 |
| Rotation | PointNet | 92.37 | 61.79 | 0.6067 |
| | PointNet++ | **97.16** | 82.11 | **0.8418** |
| | Our(dual) | 96.52 | **82.65** | 0.8296 |

Bold numbers indicate the best results.

performance outperforms PointNet++ in all metrics. With 10% labels, the F1-score of our method is 8.77% higher than that of PointNet++. As for PointNet, its result with 100% labels is approximately equal to the result of our method with 10% labels. Results demonstrate the effectiveness of our method in reducing labelling costs. We consider our pre-training method can effectively extract the features of the point cloud data, which enables the model to achieve better performance with limited labels.

## *D. Ablation Studies*

We explore the factors that make our method effective through ablation experiments. We conduct two ablation experiments to further understand which data augmentation is more effective, and the effect of dual-branch encoders. We also analyze robustness to transformations like rotation in the testing phase. The ablation experiments sample 1,024 points in each point cloud.

**Augmentation.** We try to find the best data augmentation method for our unsupervised method, by considering three different augmentation methods including *rotation, perturbation, crop and jittering.*

Rotation means randomly rotating the point cloud along the $Y$-axis. Perturbation means randomly rotating the point cloud by a small angle along the $XYZ$-axis. Crop means to crop a part of the point cloud. Jittering is the addition of Gaussian noise to the $XYZ$ coordinates and normal information of the point cloud. As shown in Table VI, jittering for both branches is the best data augmentation, crop for both branches is the second best data augmentation, and the method with both jittering and perturbation is generally better than the perturbation for both branches. The data augmentation with both branches rotation is the least effective. Based on the results, we can find that the data augmentation using jittering allows the encoder to learn the distinctive features of the point cloud more effectively, thereby giving better results. We consider that our encoder PointNet++ extracts local features and thus affects the effect of rotation on the point cloud. In addition, other works [49] also indicate that jitter produces more uniformly distributed features than rotation, and these features are beneficial to downstream tasks. Crop is also effective, but it is more destructive to the partial structure of the point cloud, which makes it inferior to jittering.

**Dual-branch encoders.** In order to investigate the effectiveness of dual-branch encoders, the experiments are designed to compare it with the traditional single encoder method. As shown in Fig. 3, the dual-branch encoders method has the best performance for both classification and segmentation tasks, in particular for the classification task. Besides, the single encoder method based on the more advanced PN++ is inferior to the PN-based method in the segmentation task. This is probably due to that single encoder contrastive learning can not reveal the distinction between global and local information.

**Robustness to rotation.** To analyze the robustness of our method in the testing phase for transformations like rotation, we compare the effect of using randomly rotated data augmentation in PointNet, PointNet++ and our method during the testing phase. As shown in Table VII, we can see that the rotation has the least effect on our method, decaying only 3.17% of the F1-score. PointNet++ and PointNet decrease the F1-score by 6.11% and 7.68%, respectively. *Notice that the A. Accuracy of our method is the best result among three results with rotation. Its influence is also the smallest among the three methods.* We consider that contrastive learning can improve the learning of the distinctive features between different categories of objects. Also our method has two encoders (PN and PN++), which are able to learn the relationship between the whole and the local information. So our method has better robustness to rotation.

## V. Discussion

### A. Unsupervised Dual-Branch Learning

Deep learning is a rapidly developing field. It offers great promise for solving healthcare problems. However, there is often lack of labelled datasets in the medical field, such as intracranial aneurysms. Lack of labelled data imposes a considerable challenge to training deep learning models. However, a more powerful network model can be trained by feeding unlabeled data using an unsupervised method. It has significant implications for boosting the accuracy of deep learning techniques. In addition, most data used to train deep learning models are 2D slices of images, rather than 3D geometric data currently [6]. However, 3D geometric data contain a large amount of spatial information, which facilitates the acquisition of more accurate results. We proposed a novel unsupervised method for detecting intracranial aneurysms based on 3D point cloud data. Our method can extract the features of 3D vessel segment point clouds to detect the presence of intracranial aneurysms effectively. As shown in Table I and Table II, our method achieves good results in both IntrA dataset [7] and ModelNet-40 dataset [36]. The experiments demonstrate the effectiveness of our method. It outperforms the current mainstream unsupervised and partially supervised methods in terms of accuracy.

The popular contrastive learning framework [31] obtains two samples by data augmentation and then optimizes the encoder and the projection head by contrastive loss so that the projection representations of two samples are close in latent space. In this work, we design a novel method involving dual-branch encoders to boost performance. In particular, the two encoders are built upon PointNet [10] and PointNet++ [11]. We concatenate the feature representations output from the dual-branch encoders for the downstream task, so that more discriminative features can be extracted. In addition, contrastive learning essentially encourages similar samples to cluster together by reducing the discrepancy between the feature representations of the two samples. Dual-branch encoders extract features from two different “perspectives”. Therefore, our dual-branch encoders are more effective than a single encoder. In our design, PointNet branch focuses on global features and PointNet++ branch focuses on local features. PointNet and PointNet++ as encoders can better emphasize the distinction between local and global features of point cloud samples, enabling more effective optimization of the network. As shown in Table I, our method even outperforms most of the mainstream supervised methods in the accuracy of identifying vessel segments with aneurysms. We consider that our method is effective in identifying data with distinctive features such as aneurysmal vessel segments. As shown in Fig. 3, we conduct experiments on single encoder and dual-branch encoders. The experimental results illustrate the important contribution of our dual-branch encoders on performance.

We have explored various strategies of data augmentation. According to the experiments in Table VI, we notice that jittering is more effective than rotation. We consider that jittering changes the spatial position relationship of the points in the point cloud, whereas rotation changes the angle of the whole point cloud. Our method is able to advance local and global information of point clouds. Therefore it will be more sensitive to changes in the location relationships of points. In addition, we simulate the training scenarios with the lack of labelled data. According to Table V, the experiments demonstrate that our method can effectively extract features in unsupervised manner, thus enhancing the performance and robustness of the model.

### B. Limitations and Future Work

Despite the significant advantages of our method described above, there are still several limitations that could be addressed in future work.

First, dual-branch encoders of our method are not chosen arbitrarily. It requires choosing encoders that extract features from different scales, such as PointNet [10] and PointNet++ [11] that extract features from global and local scales, respectively. Second, our dual-branch encoders structure can effectively extract richer features than a single shared encoder. Compared to general contrastive learning frameworks, our multi-branch can extract distinct features from multiple encoders and can extract much richer features. We speculate that better features can be learned when more branches are used. Also we think this property of multi-branch can be used for multimodal data fusion, for example, fusing features from different modalities such as position coordinates, depth information and colour information. In addition, multi-branch will result in richer positive samples in contrastive learning, which is useful to bring the representation of same-origin objects closer in the latent space. We would like to investigate this in future.

## VI. Conclusion

In this work, we have presented an unsupervised representation learning method for the classification and segmentation of 3D intracranial aneurysms. It first augments a point cloud into two samples and pairs them up for going through the dual-branch encoders and a subsequent common projection head. Distinctive features are learned by maximising the correspondence for a pair. The representations learned by the unsupervised trained encoders are used as input for the downstream tasks. Experiments demonstrated that our method is effective in learning unsupervised representations and can achieve better or comparable performance than state-of-the-art supervised and unsupervised learning methods.

## References

[1] T. Nakao *et al.*, "Deep neural network-based computer-assisted detection of cerebral aneurysms in MR angiography," *J. Magn. Reson. Imag.*, vol. 47, no. 4, pp. 948–953, 2018.

[2] J. N. Stember *et al.*, "Convolutional neural networks for the detection and measurement of cerebral aneurysms on magnetic resonance angiography," *J. Digit. Imag.*, vol. 32, no. 5, pp. 808–815, 2019.

[3] D. Ueda *et al.*, "Deep learning for MR angiography: Automated detection of cerebral aneurysms," *Radiology*, vol. 290, no. 1, pp. 187–194, 2019.

[4] T. Sichtermann, A. Faron, R. Sijben, N. Teichert, J. Freiherr, and M. Wiesmann, "Deep learning-based detection of intracranial aneurysms in 3D TOF-MRA," *AJNR Am J. Neuroradiol.*, vol. 40, no. 1, pp. 25–32, 2019.

[5] Z. Shi *et al.*, "A clinically applicable deep-learning model for detecting intracranial aneurysm in computed tomography angiography images," *Nat. Commun.*, vol. 11, no. 1, pp. 1–11, 2020.

[6] B. Joo *et al.*, "A deep learning algorithm may automate intracranial aneurysm detection on MR angiography with high diagnostic performance," *Eur. J. Radiol.*, vol. 30, pp. 5785–5793, 2020.

[7] X. Yang, D. Xia, T. Kin, and T. Igarashi, "Intra: 3D intracranial aneurysm dataset for deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2656–2666.

[8] J. Yu *et al.*, "3D medical point transformer: Introducing convolution to attention networks for medical point cloud analysis," 2021, *arXiv:2112.04863*.

[9] L. Schneider, A. Niemann, O. Beuing, B. Preim, and S. Saalfeld, "MedmeshCNN-Enabling MeshCNN for medical surface models," *Comput. Methods Programs Biomed.*, vol. 210, 2021, Art. no. 106372.

[10] C. R. Qi, H. Su, K. Mo, and L. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[11] C. R. Qi, L. Yi, H. Su, and L. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[12] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 87–102.

[13] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on $\chi$-transformed points," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 828–838.

[14] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9397–9406.

[15] T. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[16] Z. Shi *et al.*, "Artificial intelligence in the management of intracranial aneurysms: Current status and future perspectives," *Amer. J. Neuroradiol.*, vol. 41, no. 3, pp. 373–379, Jan. 2020.

[17] K. Kamnitsas *et al.*, "DeepMedic for brain tumor segmentation," in *Proc. Int. Workshop Brainlesion: Glioma, Mult. Sclerosis, Stroke Traumatic Brain Injuries*, 2016, pp. 138–149.

[18] T. Kohonen, "The self-organizing map," *Proc. IEEE.*, vol. 78, no. 9, pp. 1464–1480, Sep. 1990.

[19] J. Gong *et al.*, "Boundary-aware geometric encoding for semantic segmentation of point clouds," in *Proc. AAAI Conf. Artif. Intell*, 2021, pp. 1424–1432.

[20] J. Gong *et al.*, "Omni-supervised point cloud segmentation via gradual receptive field component reasoning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11673–11682.

[21] D. Zhang, X. Lu, H. Qin, and Y. He, "Pointfilter: Point cloud filtering via encoder-decoder modeling," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 3, pp. 2015–2027, Mar. 2021.

[22] W. Wang, X. Lu, D. S. Edirimuni, X. Liu, and A. Robles-Kelly, "Deep point cloud normal estimation via triplet learning," 2021, *arXiv:2110.10494*.

[23] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "PointContrast: Unsupervised pre-training for 3D point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 574–591.

[24] X. Lu, H. Chen, S. K. Yeung, Z. Deng, and W. Chen, "Unsupervised articulated skeleton extraction from point set sequences captured by a single depth camera," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7226–7234.

[25] X. Lu, Z. Deng, J. Luo, W. Chen, S. K. Yeung, and Y, "He 3D articulated skeleton extraction using a single consumer-grade depth camera," *Comput. Vis. Image Underst.*, vol. 188, 2019, Art. no. 102792.

[26] J. Jiang, X. Lu, W. Ouyang, and M. Wang, "Unsupervised representation learning for 3D point cloud data," 2021, *arXiv:2110.06632*.

[27] A. Sanghi, "Info3D: Representation learning on 3D objects using mutual information maximization and contrastive learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 626–642.

[28] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," 2018, *arXiv:1809.10341*.

[29] Y. Yang, F. Chen, Y. Shen, and T. Dong, "FoldingNet: Point cloud autoencoder via deep grid deformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 206–215.

[30] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Soc. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 1735–1742.

[31] T. Chen, K. Simon, N. Mohammad, and H. Geoffrey, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[32] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proc. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.

[33] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.

[34] A. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[35] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 15535–15545.

[36] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.

[37] A. X. Chang *et al.*, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

[38] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," *Symp. Geometry Process.*, vol. 6, pp. 156–164, 2003.

[39] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Comput. Graph Forum.*, vol. 22, no. 3, pp. 223–232, 2003.

[40] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 484–499.

[41] A. Sharma, O. Grau, and M. Fritz, "VConv-DAE: Deep volumetric shape learning without object labels," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 236–250.

[42] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Conf. Neural Inf. Process. Syst.*, 2016, pp. 82–90.

[43] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Representation learning and adversarial generation of 3D point clouds," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 40–49.

[44] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, "3D point capsule networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1009–1018.

[45] K. Hassani and M. Haley, "Unsupervised multi-task feature learning on point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8160–8171.

[46] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "PointHop: An explainable machine learning method for point cloud classification," in *IEEE Trans. Multimedia.*, vol. 22, no. 7, pp. 1744–1755, Jul. 2020.

[47] Z. Han, X. Wang, Y.-S. Liu, and M. 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10441–10450.

[48] T. Le and Y. Duan, "PointGrid: A deep network for 3D shape understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9204–9214.

[49] Y. Liu, L. Yi, S. Zhang, Q. Fan, T. Funkhouser, and H. Dong, " P4Contrast: Contrastive learning with pairs of point-pixel pairs for RGB-D scene understanding," 2020, *arXiv:2012.13089*.