2019 International Conference on 3D Vision (3DV)

Pairwise Attention Encoding for Point Cloud Feature Learning

Yunxiao Shi Haoyu Fang Jing Zhu Yi Fang*

NYU Multimedia and Visual Computing Lab, USA
New York University, USA
New York University Abu Dhabi, UAE

{yunxiao.shi, fanghyvl, jingzhu, yfang}@nyu.edu

Abstract

Compared to hand-crafted ones, learning a 3D point signature has attracted increasing attention in the research community to better address challenging issues such as deformation and structural variation in 3D objects. PointNet is a pioneering work in introducing learning 3D point signature directly by consuming raw point cloud as input and applying convolution on each one of these points. Groundbreaking as it is, PointNet has limited capability in capturing local structure when learning visual features from each individual point. Recent variants of PointNet improved the quality of 3D point signature learning by taking neighbourhood information into account, but typically do so through hard-coded mechanisms (e.g. manually setting 'k' for k-Nearest Neighbour search, radius 'r' for Ball Query, etc). In this paper, we developed a novel point signature learning approach by considering pairwise interaction between every two individual points that moves beyond hard-coded neighbourhood exploitation, which further improves the quality of 3D point signature learning by encouraging the model to be aware of both neighbourhood information and global context. Specifically, we first introduce a novel pairwise reference tensor (PRT) in the original input point space to represent the influence of every two individual points that have on each other. Then, by passing the pairwise reference tensor through a multi-layer perceptron (MLP), we obtain a high-dimensional attention tensor that encodes pairwise relationships in high dimensional space that acts as an attention mechanism. Next we further fuse learned point features with the attention weights to obtain global visual features. Our proposed method has demonstrated superior performance on various 3D visual recognition tasks (e.g. object classification, part segmentation and scene semantic segmentation).

1. Introduction

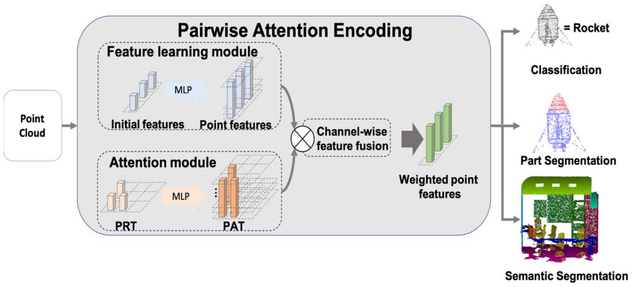

Figure 1: Pairwise attention encoding architecture learns features on raw point cloud data. We introduce a novel point reference tensor (PRT) which encodes the pairwise points influence in the Euclidean space and then learn a point attention tensor (PAT) by passing the point reference tensor through an MLP. Lastly we fuse the learned point features and the attention weights through a channel-wise fusion operator to obtain final global features.

Recent developments in 3D range sensors (e.g. LiDAR, RGB-D cameras like Microsoft Kinect, Xception PRO, etc) have made them more and more ubiquitous in diverse fields like engineering, medical imaging and AR/VR industry. Along with it the amount of available 3D visual data has experienced an explosive growth leading to a variety of important applications. Therefore it is imperative to develop automated methods that are able to analyze large amount of point cloud data both effectively and efficiently, which is the basis for various 3D visual recognition tasks (e.g. object classification, part segmentation, scene semantic segmentation, etc) that takes root in widespread applications like autonomous vehicles [15], SLAM technologies [4] and robotics [18].

To be able to process 3D point clouds on a large scale, given the established success of deep learning methods in a variety of vision tasks (especially in 2D vision) [9, 22, 14],

*indicates corresponding author.

2475-7888/19/$31.00 ©2019 IEEE
DOI 10.1109/3DV.2019.00024

much efforts have been devoted to learn robust 3D point signatures and 3D shape descriptors [33, 3, 11, 16, 29, 34]. The standard Convolutional Neural Networks (CNN), which features discrete convolution on a regular spatially ordered grid structure, has limited capability when being directly applied to spatially irregular data (e.g. 3D point clouds or 3D meshes). Previous researches often first transform irregular 3D geometric data to equally spaced 3D voxels in order to take advantage of the representational power of deep neural networks. The resulting performance is encouraging in various 3D vision tasks which indicates the effectiveness of convolutional neural networks in learning informative 3D point signature and shape descriptors. However, there are also researches [20] that point out the inevitable information loss due to the conversion of data from 3D point cloud to 3D voxels, therefore methods of directly dealing with point clouds that avoids this kind of information loss is urgently needed.

As a first step in this direction, researchers proposed PointNet [19] which is a deep neural network that directly consumes raw point cloud as input, a pioneering work in applying standard convolution to data of spatially irregular structure. By only using a simple feature mapping network, point features can be directly extracted using point-wise multi-layer-peceptron (MLP), followed by a max-pooling operation to obtain the global feature representation. Although PointNet proved to be the first great success of using deep learning methods directly in learning 3D point signatures, the local context of a point were not fully utilized because the neighbourhood information of a particular point are not considered during the whole feature learning process. PointNet++ [21] introduced a multi-scale feature aggregation in sub-groups of the entire point clouds but did not take the spatial distribution of the input point clouds into account. SO-Net [12], PointGrid [10] and Kd-Net [8] mitigated this issue by considering the spatial distribution of the input point cloud into account and demonstrated further improved performance in various 3D visual tasks, validating that neighbourhood information of a particular point is useful when extracting visual features from point clouds.

Exploitation of neighbourhood information is useful in point signature learning, but the above methods typically do so using a hard-coded mechanism (e.g. k-Nearest Neighbour search). In this paper, we develop a novel point signature learning paradigm, named pairwise attention encoding, which equips the model with the capability of being aware of both local information and global context. First, we introduce a novel data-driven pairwise reference tensor that moves beyond using neighbourhood information in a hard-coded fashion. Figure 1 gives us a more detailed description of our pairwise attention encoding mechanism which consists of four components. The first component is a "feature learning module". In this component, we learn point-wise features by passing input point cloud through a transformation network first (same as T-Net in PointNet [19]) and then an MLP. This gives us informative visual features but does not take local information into account. The second component is an "attention module", in the sense we consider those pairwise interaction encoding weights as a kind of attention, where we generate the pairwise reference tensor by computing the Euclidean distance between every two individual points and then mapping them into high-dimensional space by passing them through an MLP to obtain the pairwise attention tensor. The third component is "channel-wise feature fusion". In this component, we fuse the learned point features in the first component and the high-dimensional attention weights obtained in the second component by performing an inner product of the two, taking both local neighbourhood and global context into account. For the last step we map the visual features obtained in the third component into high-dimensional space by passing them through another MLP followed by a max-pooling operation to obtain the global visual features. Therefore the main contributions of our work can be summarized as follows:

- We introduced a novel data-driven pairwise reference tensor that moves beyond hard-coded neighbourhood exploitation when considering local information in 3D point signature learning.
- We developed a new mechanism of learning a pairwise attention tensor that encodes the influence every other point has on a particular point.
- We proposed a feature fusion process that fuses learned point features and attention weights into local and global context aware visual features.
- Under the same experimental settings, our method is able to achieve superior performance over various 3D point cloud visual recognition tasks compared to other approaches.

2. Related work

Deep representation learning powered methods has transformed modern (2D) computer vision by setting records on various task benchmarks [9, 22, 23]. Following this tide researchers have been trying to replicate this success in the field of 3D computer vision especially with point cloud data. Works like [19, 21] are the pioneering explorations towards this goal which achieved remarkable performance. In recent years, researches in 3D point cloud learning mainly follow these four trajectories: regularize point cloud into a volumetric representation, converting point cloud data into multi-view images, view point cloud as a graph structure and directly consume point cloud

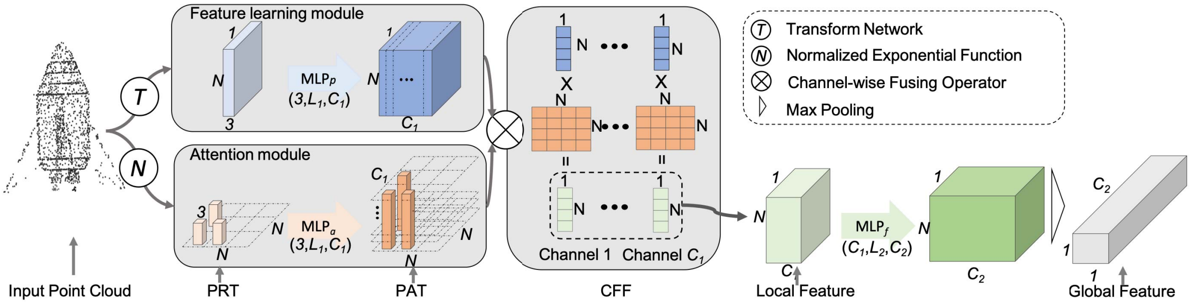

Figure 2: Pipeline of our proposed network. The input data is a normalized point cloud. MLP_p, MLP_a, MLP_f denotes pointwise multilayer perceptrons (MLP) for learning point features, attention weights and final global features respectively. L_i and C_i $(i = 1, 2)$ stand for layer input and output sizes. Batchnorm is used for all layers with ReLU activation except for the last output layer where we do not impose any non-linearity. Here PRT denotes the point reference tensor, PAT denotes the point attention tensor and CFF denotes a channel-wise feature fusing module. We use the transform network bearing the same structure as in [19] to preprocess raw point cloud data and a Normalized Exponential Function to regularize the relative positions between a point pair.

as input. Volumetric-based approaches partition 3D point sets into grid voxels and then use 3D CNN to extract visual features from regular voxels [36, 2]. However, the volumetric representation can incur heavy memory and computation bottleneck. Approaches like oct-tree [5] or kd-tree [8] tries to address this issue by partitioning the input space using a tree structure, but is still time-consuming to convert point cloud data into voxel representation. Another major problem with volumetric-based methods is that they suffer from the inevitable loss of information [19] using such conversion.

Another line of work is based on multi-view images which renders 3D point cloud into multiple 2D images of different views and then apply traditional 2D CNN to extract visual features. [26] utilized a 2D CNN to learn several independent shape representations and designed a novel view-pooling layer to fuse information from multiple views. [31] took a further step to propose a view clustering and pooling module recurrently to aggregate information from similar views. With carefully-designed multi-view shape rendering, these methods have demonstrated state-of-the-art performance on 3D vision tasks like shape classification and retrieval tasks. However, they still tend to suffer from information loss and can not directly applied to other 3D vision tasks like part segmentation and semantic segmentation. Graph-based methods typically treat a point cloud as a graph and try to design convolution-like operators on graphs to learn useful representations. For example recent works [13, 28] designed a particular convolution-like operator based on spectral graph theory to carry out convolution in the spectral domain by computing the eigenvector of the graph Laplacian matrix and utilizing Chebyshev polynomials and its approximation scheme.

Since the inception of PointNet [19] directly consuming raw point cloud as input has become an important stream of research in 3D point cloud feature learning. Typically they use a point-wise MLP followed by a symmetric max-pooling operation to obtain the global features. PointNet provided a simple and efficient structure to learn point signature but despite the success it achieved, it lacks the capability to capture local structures. PointNet++ [21] used hierarchical point sampling and grouping techniques trying to overcome this issue. PointSIFT [7] encodes multi-scale neighbourhood information of different orientations using a SIFT-like operator. SO-Net [12] made use of self-organizing map (SOM) to model the spatial distribution of point sets and conducts hierarchical feature extraction on SOM nodes. Other than the above MLP based models, KC-Net [24] proposed kernel correlation layers to capture the local geometric structures of a point cloud. Kd-Net [8] builds a Kd-tree for the input point cloud and performs hierarchical feature learning in a bottom-up fashion.

3. Approach

In this section, we give a detailed description of the proposed deep network with pairwise attention encoding. The proposed network (shown in Fig. 2) consists of three parts including a feature learning module, an attention module (learning point reference tensor (PRT) and point attention tensor (PAT)) and a channel-wise feature fusion process. Section 3.1 introduces the feature learning module that we used to learn point features. Section 3.2 describes our attention module that consists of how we generate our point reference tensor in a data-driven manner and how we define the point attention tensor that is end-to-end learnable. Section 3.3 states the feature fusion process that is used to obtain the final global visual features.

3.1. Feature Learning Module

Previous researches have shown that point features can be learned through an MLP-like structure [19, 21]. Here we also regard the point feature learning process as a non-linear transformation function $F_p(\cdot) : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{N\times C}$ (shown in Figure 2), which is

$$F_p(p_i) = \sum_{i,j=1}^{N} v_{pi} \sum_{i,j=1}^{N} (w_{pi} p_i + \alpha_p) + \beta_p, \quad (1)$$

where $H = \{h_i = F_p(p_i) \in \mathbb{R}^{1\times C}, i = 1, 2, ..., N\}$ is learned point features. In essence we also design $F(\cdot)$ as an MLP. Since all operators in non-linear transformation functions $F_p(\cdot)$ are symmetric binary functions (because $F_p(\cdot)$ consists of only "+" and "×"), therefore the proposed feature learning module is invariant to input permutation.

3.2. Attention Module

Point Reference Tensor We propose a novel *point reference tensor (PRT)* which captures the interaction between each individual two points in a distance-driven way. As shown in Figure 2, we generate the point reference tensor by computing the Euclidean distance between every two points. Therefore it is completely data-driven instead of hard-coded mechanisms such as k-Nearest Neighbour search. Formally, we define the raw point reference tensor as $\Delta = \{\delta_{ij}, i, j = 1, 2, ..., N\}$, where

$$\delta_{ij} = p_j - p_i = [\Delta x_{ij}, \Delta y_{ij}, \Delta z_{ij}]^T. \quad (2)$$

Each element $\delta_{ij} \in \Delta$ is a 1×3 vector which is the Euclidean distance between point p_j and p_i. We further normalize the point reference tensor Δ by Eq. 3, denoted as $PRT = \{r_{ij} \in \mathbb{R}^{1\times 1\times 3}, i, j = 1, 2, ..., N\}$, where

$$r_{ij} = \frac{exp(\delta_{ij})}{\sum_{j=1}^{N} exp(\delta_{ij})} \quad (3)$$

is a normalized relation between different p_j and p_i. The point reference tensor PRT encodes the influence that every two individual points have on each other under the Euclidean distance metric.

Point Attention Tensor Point reference tensor encodes the influences that every other individual point has on a particular point in the Euclidean space. But as the features go up into high-dimensional spaces, point reference tensor is no longer equipped to encode the pairwise relations of corresponding features, meanwhile having the means to describe the influences of the high dimensional representations of corresponding points that have on each other is important for learning visual features of good quality. Therefore we propose a novel point attention tensor (*PAT*) to achieve such a goal through a learning mechanism. As shown in the attention module in Figure 2, we define the point attention tensor $PAT = \{a_{ij} \in \mathbb{R}^{1\times 1\times C}, i, j = 1, 2, ..., N\} \in R^{N\times N\times C}$ to extract point interaction knowledge from the distance-based information in *PRT*. Concretely, we learn the point attention tensor as

$$a_{ij} = F_r(r_{ij}) = \sum_{i,j=1}^{N} \Big[v_{rij} \sum_{i,j=1}^{N} (w_{rij} r_{ij} + \alpha_r) + \beta_r \Big], \quad (4)$$

where $F_r(\cdot) : \mathbb{R}^{N\times N\times 3} \rightarrow \mathbb{R}^{N\times N\times C}$ is a non-linear transformation function processing with the point reference tensor as input, w_{rij} and v_{rij} are learnable weights, α_r and β_r are biases in the transformation function. We note that this non-linear transformation function can be implemented as a MLP.

3.3. Channel-Wise Feature Fusion

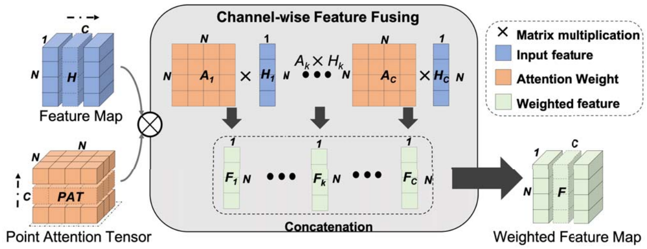

Figure 3: The channel-wise feature fusion. The inputs of the channel-wise fusing operator are a latent feature map $H \in \mathbb{R}^{N\times C}$ and the point attention tensor $PAT \in \mathbb{R}^{N\times N\times C}$, the output of the process is a weighted feature map $F \in \mathbb{R}^{N\times C}$, H_k, A_k, F_k denotes features in the k^{th} channel. C denotes the number of channels in input tensors.

Exploiting local structure has proven to be important for learning high-quality point features using convolutional architectures. Fusing information of point neighbors into individual point features according to the learned pairwise interaction will enable the network to be aware of both local neighbourhood and global context. We design a novel encoding mechanism with a channel-wise fusing operator, which fuses influence of surrounding points interaction into points features in a channel-wise fashion. As shown in Figure 3, the channel-wise feature fusion process is done by conducting an inner product of the learned point features and the attention weights that are from the point attention tensor. Formally the process can be defined as:

$$\begin{aligned} F &= A \otimes H \\ &= \Big[F_1, .., F_k, .., F_C \Big] \\ &= \Big[A_1 H_1, .., A_k H_k, .., A_C H_C \Big], \end{aligned} \quad (5)$$

where $A_k \in \mathbb{R}^{N\times N}$ denotes the k^{th} channel in A, $H_k \in \mathbb{R}^{N\times 1}$ denotes k^{th} channel in H and $\otimes$ denotes the channel-wise feature fusion operator. $F_k = A_k H_k \in \mathbb{R}^{N\times 1}$ represents the weighted feature of the k^{th} channel. $F \in R^{N\times C}$ represents the weighted feature map for all channels. The output feature map after the channel-wise feature fusion process encodes knowledge of both local information and global context of a each particular point, which gives us the final high-quality point features to carry out various 3D vision tasks.

Method	Input	Accuracy Avg. Class	Accuracy Overall
3DShapeNets [32]	volume	77.3	84.7
VoxNet [36]	volume	83.0	85.9
PointNet [19]	point	86.2	89.2
PointNet++ [21]	point	-	90.7
SO-Net [12]	point	87.3	90.9
Ours	point	**88.2**	**91.1**

Table 1: **Object classification results on ModelNet40**. Our network achieves state-of-the-art performance.

4. Experiments

In this section, we evaluation our network on three different tasks, namely 3D object classification, 3D object part segmentation and 3D scene semantic segmentation. We demonstrate that our model performs better than current state-of-the-art approaches under the same settings.

4.1. 3D Object Classification

Our network learns both locally and globally informative visual features to do object classification. We evaluate our model on the ModelNet40 [32] object shape classification benchmark. There are 12,311 CAD Models from 40 man-made object categories. Here we use the official split of 9,843 for training and 2,468 for testing.

We first uniformly sample 1,024 points from the mesh surfaces and then normalize them into a unit sphere (i.e. all the input coordinates are in the range of $[-1, 1]$). Following the same setting in [19], during the training phase we randomly rotate the object along the z-axis and apply a small perturbation of adding a Gaussian noise of mean $\mu = 0$ and standard deviation $\sigma = 0.02$ to all the points as data augmentation. The whole process is done in an online fashion. The final evaluation metric is mean accuracy across all forty object categories. For a given object category, the classification accuracy is computed as

$$acc_i = \frac{TP_i}{TP_i + FP_i}, \tag{6}$$

where TP stands for true positives and FP for false positives.

In Table 1, we compare our model with several strong models that also use raw point cloud as input. Our model out-performs PointNet [19] by a considerable margin and achieves state-of-the-art performance among methods dealing with raw point cloud data. Also while PointGrid [10] has an overall accuracy of 92% outperforming all of the above methods including ours, it essentially modified the structure of raw point cloud to incorporate a constant number of points in each cell that is no longer of the same experiment setting as ours (and others as well), which is the reason why we did not list its entry here since it is not a completely fair comparison.

4.2. 3D Object Part Segmentation

Part segmentation is considered a challenging 3D visual recognition task. Given a 3D point cloud or a mesh surface model, the task is to assign part category label (e.g. mug handle, skateboard wheel, etc) to each point or a mesh face.

We evaluate our model on the ShapeNet [35] dataset which contains 16,881 shapes from 16 object categories with 50 annotated object part classes in total. Most of the categories has two to four object parts. Ground truth annotations are labeled on sampled points of the shapes. Given the highly imbalanced number of samples between different categories, the task of part segmentation on this dataset pose a great challenge to all methods including our own.

We also formulate the problem of part segmentation as per-point classification. During training we uniformly sample 2,048 points per each object instance on the fly and perform a unit-sphere normalization. Because of the highly imbalanced number of parts that each different category has, if only training on each category separately the network might have difficulty attending to the categories that have fewer parts leading to poor performance. Therefore during training we follow the setting of [19], which is to add a one-hot vector indicating the class of the input and concatenate it with the max-pooling layer's output to help our model to better attend to the categories that has small number of object parts. The final evaluation metric is mean Intersection over Union (mIoU) on points. Given shape S_i of category C_j, the shape's IoU is calculated as the intersection over union between ground truth and prediction. Specifically we have

$$IoU_i = \frac{TP_i}{TP_i + FP_i + FN_i}, \tag{7}$$

where TP stands for the true positives, FP for the false positives and FN for the false negatives. If the union of ground truth and prediction is empty, the part IoU is computed as 1. Then we average all part IoUs in that category to get mIoU for that shape. To calculate mIoU for a certain category, we take average of all mIoUs for all shapes in that category. We compare our model with several latest works that are based on point cloud input. In Table 2, we report per-category and mean IoU (in percentile) scores, and we show visualization of example segmentation results of the 4 categories of earphone, guitar, lamp and rocket in Figure 4. (the results for the rest 12 ShapeNet categories can be found in Figure 5).

From Table 2 we can see our model achieves remarkable

Method	# Shapes	Yi [35]	PointNet [19]	Kd-Net [8]	PointNet++ [21]	SPLATNet [25]	SO-Net [12]	Ours
mIoU	-	81.4	83.7	77.2	**85.1**	84.6	84.6	85.0
aiplane	2690	81.0	**83.4**	79.9	82.4	81.9	81.9	82.7
bag	76	78.4	78.7	71.2	79.0	**83.9**	83.5	82.3
cap	55	77.7	82.5	80.9	**87.7**	88.6	84.8	86.9
car	898	75.7	74.9	68.8	77.3	**79.5**	78.1	77.2
chair	3758	87.6	89.6	88.0	**90.8**	90.1	**90.8**	89.1
earphone	69	61.9	73.0	72.4	71.8	73.5	72.2	**74.3**
guitar	787	**92.0**	91.5	88.9	91.0	91.3	90.1	90.9
knife	392	85.4	85.9	**86.4**	85.9	84.7	83.6	83.9
lamp	1547	82.5	80.8	79.8	83.7	**84.5**	82.3	81.2
laptop	451	95.7	95.3	94.9	95.3	**96.3**	95.2	95.2
motorbike	202	70.6	65.2	55.8	**71.6**	69.7	69.3	69.1
mug	184	91.9	93.0	86.5	94.1	**95.0**	94.2	94.1
pistol	283	**85.9**	81.2	79.3	81.3	81.7	80.0	80.4
rocket	66	53.1	57.9	50.4	58.7	**59.2**	51.6	**59.3**
skateboard	152	69.8	72.8	71.1	**76.4**	70.4	72.1	76.2
table	5271	75.3	80.6	80.2	**82.6**	81.3	**82.6**	81.9

Table 2: **Segmentation results on ShapeNet dataset**. We compare our model with several latest approaches. Our model achieves state-of-the-art performance.

performance. While PointNet++ [21] gains a slight advantage (0.1% in overall mIoU) over us, it utilized extra point normal information whereas our method relies solely on the raw point cloud data, making our model the state-of-the-art approach among methods that only use point coordinates so far.

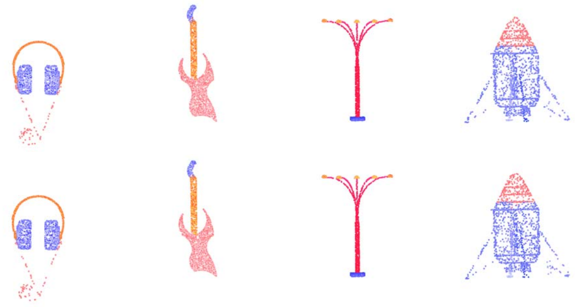

Figure 4: Example results of part segmentation on the ShapeNet dataset. The first row is our model's predicted results of the categories of earphone, guitar, lamp and rocket (from left to right). The second row is ground truth.

4.3. 3D Scene Semantic Segmentation

In the last experiment we present the performance of our model on the Stanford Large-Scale 3D Indoor Indoor Spaces Dataset (S3DIS) [1]. The S3DIS dataset provides point clouds for six fully reconstructed large-scale areas, originating from three different buildings. We follow the same train/test split as proposed in the original paper [1] (Area 1, Area 2, Area 3, Area 4, Area 6 for training, and Area 5 for testing). We compare our model against the Qi [19] (PointNet) which performs a six-cross validation across areas rather than buildings. As SegCloud points out [27], this experimental setting will cause areas from the same building end up both in training and test set leading to an increased performance. Therefore we adopt the more principled strategy of testing on the fifth fold (Area 5) which is the most comprehensive fold of all the six folds and train on the rest. During training we uniformly sample 4,096 points on the fly and rotate along the upright axis as data augmentation. The point inputs are normalized into the range $[0, 1]^3$ as relative to the scene. The evaluation criteria is mIoU across all 14 categories, the computation of IoU for each category is the same as defined in Eq. 6. We compare quantitatively with the results obtained from recent state-of-the-art approaches in Table 3. We also show qualitative results consisting of a variety of different scene areas (offices, lounges and conference rooms) in Figure 6. Note that we removed some clutters and walls that belong to the 'clutter' and 'wall' classes respectively for the purpose of clearer visualization.

From Table 3 we can see that our model outperforms PointNet [19] by a large margin and achieves state-of-the-art performance. Meanwhile we would like to note although PCNN [6] outperforms all the other methods including ours by quite a margin, it firstly uses point coordinates + RGB values as input and projects point features onto an ordered sequence of feature vectors, which introduces extra constraints in the point feature learning process.

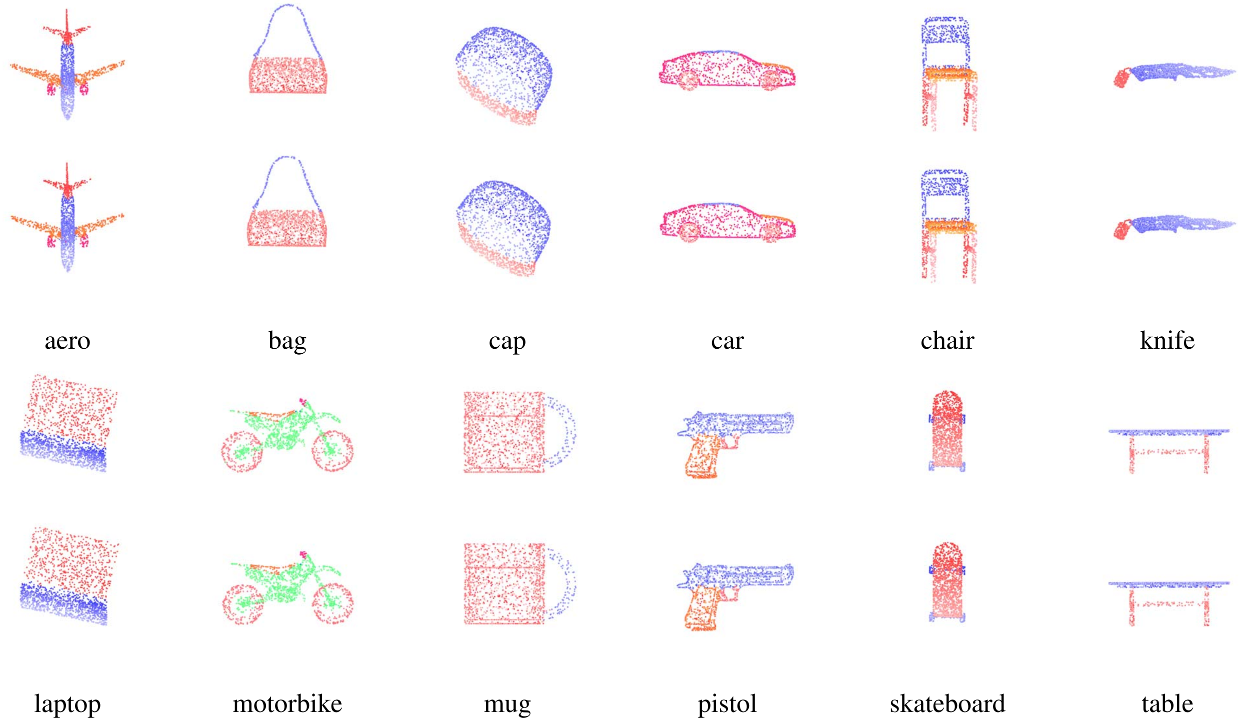

Figure 5: Visualization of results of part segmentation for the rest 12 categories on the ShapeNet part dataset. The first row in every two rows are the outputs of our segmentation model. The second row in every two rows are the corresponding ground truths. Best viewed in color.

Method	3D-CNN [27]	PointNet [19]	3D-FCN-TI [27]	SegCloud [27]	PCNN [30]	RCNN [6]	Ours
mIoU	43.67	41.09	47.46	48.92	**58.27**	51.93	53.16
ceiling	-	88.80	90.17	90.06	92.26	93.34	**93.50**
floor	-	97.33	96.48	96.05	96.20	**98.36**	96.24
wall	-	68.90	70.16	69.86	75.89	**79.18**	74.68
beam	-	0.05	0.00	0.00	**0.27**	0.00	0.10
column	-	3.92	11.40	**18.37**	5.98	15.75	16.78
window	-	46.26	33.36	38.35	**69.49**	45.37	50.14
door	-	10.76	21.12	23.12	**63.45**	50.10	45.57
chair	-	52.61	**76.12**	75.89	66.87	65.52	70.26
table	-	58.93	70.07	**70.40**	65.63	67.87	**71.12**
bookcase	-	40.28	57.89	**58.42**	47.28	22.45	45.27
sofa	-	5.85	37.46	40.88	**68.91**	52.45	50.94
board	-	26.38	11.16	12.96	**59.10**	41.02	38.76
clutter	-	33.22	41.61	41.60	46.22	43.64	**47.12**

Table 3: **Scene semantic segmentation results on the S3DIS dataset [1].** We compare our model with several latest approaches. Our model achieves state-of-the-art performance.

4.4. Implementation Details

We implemented our pairwise-attention model using the PyTorch [17] framework, which is an open-sourced deep learning platform that provides strong GPU support for computation efficiency. We implement the transformation network in our model as the same one in PointNet [19] which takes raw point clouds as input and regresses them into a 3 × 3 matrix. For classification network we apply a dropout ratio of 0.6 on the last fully connected layer before class prediction. The decay rate for batch normalization starts with 0.5 and is gradually increased to 0.99. We use SGD optimizer with a momentum of 0.9 and an initial

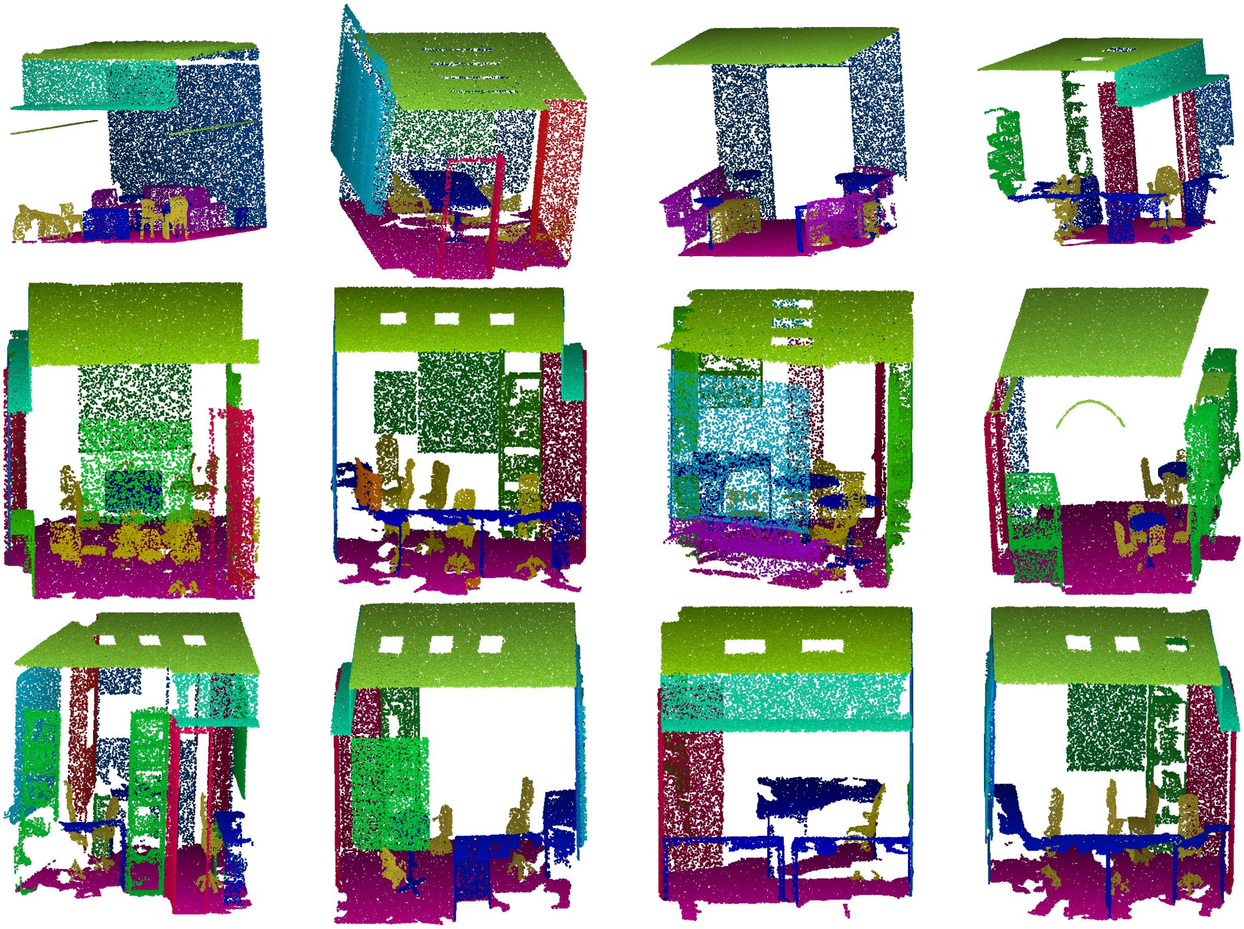

Figure 6: Visualization of scene semantic segmentation results on the S3DIS dataset. We choose different view angles for each scene for better visualization quality. Best viewed in color.

learning rate of 0.001. We use an early stopping mechanism to monitor training, if the performance on test set has not improve for 10 consecutive epochs then we decay the learning rate by half. Our model takes around eight hours to converge on ModelNet40 with PyTorch using a NVIDIA GTX 1080Ti GPU. The segmentation network shares the same base architecture as our classification network, except we concatenate the features after the first phase of 1-D convolution whose dimension of 64 with the final output global features to help our network to attend to local information in various object parts. We do not use dropout layer in any part of segmentation network. Except an initial learning of 0.0005 the other training parameters and training monitor mechanism are the same as our classification network. The segmentation takes around twelve hours to converge on ShapeNet and twenty hours to converge on Stanford S3DIS.

5. Discussion and conclusion

In this work, we proposed a novel paradigm named pairwise attention encoding for 3D point feature learning. We first introduce a novel pairwise reference tensor (PRT) in the original input point space to represent the influence of every two individual points that have on each other. By passing the pairwise reference tensor through an MLP, we obtain a high-dimensional point attention tensor (PAT) that encodes pair-wise relationships in high dimensional space that acts as an attention mechanism. Finally, we further fuse learned point features with the attention weights to obtain global visual features through a channel-wise feature fusion (CFF) module. The fused point features encourage the model to be aware of both neighbourhood information and global context. With the pairwise attention encoding architecture, the proposed network provides a integrated approach to a list of challenging 3D point cloud learning tasks and achieves state-of-the-art or even better performance on standard benchmarks. For the future work, we will continue to design proper loss function for the attention-based network to converge faster and make the pairwise attention encoding architecture a more powerful tool for point cloud feature learning.

References

[1] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. 6, 7

[2] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016. 3

[3] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong. 3d deep shape descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2319–2328, 2015. 2

[4] N. Fioraio and K. Konolige. Realtime visual and point cloud slam. In *Proc. of the RGB-D workshop on advanced reasoning with depth cameras at robotics: Science and Systems Conf.(RSS)*, volume 27. Citeseer, 2011. 1

[5] S. Gumhold, Z. Kami, M. Isenburg, and H.-P. Seidel. Predictive point-cloud compression. In *ACM SIGGRAPH 2005 Sketches*, page 137. ACM, 2005. 3

[6] Q. Huang, W. Wang, and U. Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2635, 2018. 6, 7

[7] M. Jiang, Y. Wu, and C. Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*, 2018. 3

[8] R. Klokov and V. S. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 2, 3, 6

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 2

[10] T. Le and Y. Duan. Pointgrid: A deep network for 3d shape understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9204–9214, 2018. 2, 5

[11] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, Q. Chen, N. K. Chowdhury, B. Fang, et al. A comparison of 3d shape retrieval methods based on a large-scale benchmark supporting multimodal queries. *Computer Vision and Image Understanding*, 131:1–27, 2015. 2

[12] J. Li, B. M. Chen, and G. Hee Lee. So-net: Self-organizing network for point cloud analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3, 5, 6

[13] R. Li, S. Wang, F. Zhu, and J. Huang. Adaptive graph convolutional neural networks. 2018. 3

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[15] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 1

[16] P. Papadakis, I. Pratikakis, T. Theoharis, and S. Perantonis. Panorama: A 3d shape descriptor based on panoramic views for unsupervised 3d object retrieval. *International Journal of Computer Vision*, 89(2-3):177–192, 2010. 2

[17] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 7

[18] F. Pomerleau, F. Colas, R. Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015. 1

[19] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 2, 3, 5, 6, 7

[20] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016. 2

[21] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 2, 3, 5, 6

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015. 1, 2

[23] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, Apr. 2017. 2

[24] Y. Shen, C. Feng, Y. Yang, and D. Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 3

[25] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018. 6

[26] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 3

[27] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *International Conference on 3D Vision (3DV)*, 2017. 6, 7

[28] G. Te, W. Hu, Z. Guo, and A. Zheng. Adaptive graph convolutional neural networks. 2018. 3

[29] J. Wang, X. Bai, X. You, W. Liu, and L. J. Latecki. Shape matching and classification using height functions. *Pattern Recognition Letters*, 33(2):134–143, 2012. 2

[30] S. Wang, S. Suo, W.-C. M. A. Pokrovsky, and R. Urtasun. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2589–2597, 2018. 7

[31] W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. In *International Conference on Machine Learning*, pages 1083–1092, 2015. 3

[32] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 5

[33] J. Xie, Y. Fang, F. Zhu, and E. Wong. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1275–1283, 2015. 2

[34] X. Yang, X. Bai, L. J. Latecki, and Z. Tu. Improving shape retrieval by learning graph transduction. In *European Conference on Computer Vision*, pages 788–801. Springer, 2008. 2

[35] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016. 5, 6

[36] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *arXiv preprint arXiv:1711.06396*, 2017. 3, 5