VoxSegNet: Volumetric CNNs for Semantic Part Segmentation of 3D Shapes

Zongji Wang and Feng Lu, *Member, IEEE*

Abstract—Volumetric representation has been widely used for 3D deep learning in shape analysis due to its generalization ability and regular data format. However, for fine-grained tasks like part segmentation, volumetric data has not been widely adopted compared to other representations. Aiming at delivering an effective volumetric method for 3D shape part segmentation, this paper proposes a novel volumetric convolutional neural network. Our method can extract discriminative features encoding detailed information from voxelized 3D data under limited resolution. To this purpose, a spatial dense extraction (SDE) module is designed to preserve spatial resolution during feature extraction procedure, alleviating the loss of details caused by sub-sampling operations such as max pooling. An attention feature aggregation (AFA) module is also introduced to adaptively select informative features from different abstraction levels, leading to segmentation with both semantic consistency and high accuracy of details. Experimental results demonstrate that promising results can be achieved by using volumetric data, with part segmentation accuracy comparable or superior to state-of-the-art non-volumetric methods.

Index Terms—Shape analysis, semantic segmentation, convolutional neural networks, volumetric models

1 Introduction

In the past few decades, with the advances in user-friendly 3D modeling tools (e.g., SketchUp) and low-cost 3D shape capturing devices, the amount of available 3D shapes on the Internet has increased significantly, which induced a surge of research interest in 3D shape analyzing and understanding, for which 3D shape semantic segmentation is considered as a fundamental yet challenging task. 3D shape segmentation can be defined as the partition of a 3D shape into meaningful parts, which can greatly benefit a large number of applications such as modeling [1], [2], [3], shape editing [4], [5] and object classification [6].

The rapidly growing body of 3D models on the Internet also provide an opportunity to introduce deep learning methods into the field of 3D shape analysis. Inspired by the success of applying convolutional neural networks (CNNs) on various computer vision tasks such as image classification [7], semantic segmentation [8], and image caption generation [9], an intuitive idea for 3D deep learning is to apply CNNs to extract high-quality 3D geometric features. However, directly applying CNNs to 3D shapes faces obstacles, one of which is the irregular data formats of 3D shapes.

In order to solve the problem due to irregular 3D formats, several different representations for 3D shapes are investigated. *Volumetric representation* depicts a shape's occupancy in a gridded cubic space, thus 3D convolution can be applied to voxelized shapes [10], [11]. *Multi-view representation* models a 3D shape by a set of 2D images rendered from different viewpoints, which can be fed into 2D CNNs [12], [13], [14]. For *triangular mesh representation*, there are methods extending CNNs to the graphs defined by meshes [15], [16], [17]. Such convolution is conducted in a non-euclidean space, extracting features robust to isometric deformation. Without 3D convolution, there are also methods directly processing *unorganized point sets* in 3D. In PointNet [18], all 3D points are processed individually sharing the same set of network weights, and then max pooling is applied to extract global features.

Why Volumetric. Volumetric data naturally encodes the spatial distribution of 3D shapes. Similar to pixels in a bitmap, voxels excel at representing regularly sampled spaces, explicitly describing the spatial relationships between elements forming a shape. This makes it convenient to apply 3D convolution on voxels. In addition, volumetric data can be transformed from other 3D data formats through an effortless sampling procedure, which demonstrates its generalization ability. These advantages make volumetric representation a good choice for 3D deep learning.

Besides volumetric representation, meshes and point clouds are popular 3D data formats. However, mesh convolution methods require smooth manifold meshes as input, which are not so common in today's large shape collections. Point clouds lack local structure, making it hard to extract

- *Z. Wang is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China. E-mail: wzjgintoki@buaa.edu.cn.*
- *F. Lu is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and with the Beijing Advanced Innovation Center for Big Data-based Precision Medicine, Beihang University, Beijing 100191, China, and also with the Peng Cheng Laboratory, Shenzhen 518000, China. E-mail: lufeng@buaa.edu.cn.*

Manuscript received 2 Sept. 2018; revised 6 Jan. 2019; accepted 19 Jan. 2019. Date of publication 30 Jan. 2019; date of current version 4 Aug. 2020. (Corresponding author: Feng Lu.)
Recommended for acceptance by R. Zhang.
Digital Object Identifier no. 10.1109/TVCG.2019.2896310

contextual information. Although multi-view representation agrees with human's habit of observing 3D shapes through visual cues, viewpoint selection and lighting parameter setting influence the system's robustness. Therefore, in this paper, we select volumetric representation to model 3D shapes.

Nowadays, volumetric CNNs have already gained success in shape classification and retrieval [10], [11], [19]. Although detailed structures would have been lost during convolution procedures, the features encoding global information are sufficient for tasks like classification. However, it is less preferred to use volumetric representation for dense prediction tasks like part segmentation, due to difficulties in preserving and processing detailed information with limited 3D resolution. Some attempts have been made [20], [21], [22] recently to solve this issue, but they required the use of either more complex data formats or specially designed convolutional operations. Moreover, these methods still face problems dealing with the information loss caused by sub-sampling operations.

In this paper, we propose a novel deep network architecture (VoxSegNet) for volumetric semantic segmentation. The key idea of our method is to extract discriminative features encoding detailed information under limited resolution. To this purpose, we propose two network modules. First, a Spatial Dense Extraction (SDE) module is designed to extract discriminative features from sparse volumetric data. Without reducing spatial resolution, this module effectively alleviates the loss of detailed structural information caused by sub-sampling operations in CNNs. Second, we introduce an Attention Feature Aggregation (AFA) module, which uses attention mechanism to aggregate features according to their different levels of abstraction. This module enables us to contextually select more informative components from an input signal. Integrating the above modules, our method can obtain dense prediction with both semantic consistency and high accuracy of details.

Through this paper, we try to show that a well designed voxel-based method is able to produce state-of-the-art segmentation results, or even outperform existing methods using other shape representations. The technical contributions include:

1) A novel approach for 3D object semantic segmentation using volumetric representation.
2) A voxel-based feature extraction method and an attention-based feature aggregation method, which can preserve detailed structural information.
3) Experiments on large-scale datasets demonstrate the effectiveness of our method for 3D part segmentation. Each of the proposed modules is validated via ablation study.

2 Related Work

3D shape part segmentation has long been studied. Many early researches for mesh segmentation have tried to use a single effective feature for mesh label identification [23], [24], [25], [26], [27], [28]. However, meshes in different 3D models may vary remarkably. A single feature is often insufficient to cover all kinds of scenarios. To address this problem, Kalogerakis et al. [29] presented a learning-based method to segment and label 3D meshes by combining various geometric features. Later, Xie et al. [30] applied extreme learning machine to a group of hand-engineered geometric features for shape face labeling. These methods rely on human-designed geometric features extracted from high quality smooth manifold meshes. In this section, we mainly cover works of shape part segmentation using deep learning methods.

Recently, with the availability of large-scale 3D shape collections, there has been a growing research interest in solving 3D shape analyzing tasks via deep learning methods. Volumetric representation of 3D shapes makes it convenient to automatically extract deep features from raw data. Wu et al. [11] trained a convolutional deep belief network on voxelized shapes for object classification, shape completion and next best view prediction. Maturana et al. [10] and Qi et al. [19] proposed volumetric CNN architectures to learn discriminative features to identify or classify shapes. These methods restrict their input voxel data to a limited grid size like 30^3 or 32^3, due to the high computational and memory cost. Li et al. [31] proposed a field probing scheme to simultaneously train the weights and locations of the filters, reducing computational complexity. Graham et al. [22], [32] used the 3D sparse CNNs that apply convolutions to the active voxels in order to increase computation efficiency. Riegler et al. [20] presented a non-uniform volumetric representation using the concept of octree structure, making it possible to compute 3D CNNs with high-resolution inputs. Recently, Wang et al. [21] presented a 3D CNN based on octree representation, which can largely improve the computation efficiency.

Although volumetric CNNs have been widely used in tasks like shape classification and retrieval, voxel grids are relatively less preferred compared to other representations for semantic segmentation task. Partially because of the difficulties in preserving and processing detailed information with limited 3D resolution. With the sparsity of volumetric data being studied, it becomes possible to compute high-resolution 3D CNNs more efficiently. Several works paid attention to 3D CNNs for semantic segmentation. Riegler et al. [20] performed 3D semantic segmentation on a colored 3D point cloud facades dataset. This method maps the point clouds to grid-octree data, and then feed the data into a U-shaped encoder-decoder convolutional neural network to predict the label for each voxel. Wang et al. [21] also used a U-shaped convolutional neural network for shape part segmentation, achieving state-of-the-art results with the 64^3 resolution of leaf octant. In [22], utilizing submanifold sparse convolution, the authors trained FCN [8] and U-Net [33] for part semantic segmentation of voxelized volumes. While the inputs with high-resolution benefit part segmentation task, there is still an inevitable loss of detailed information due to the sub-sampling operation during feature extraction.

Besides volumetric CNNs, there are methods segmenting point clouds without voxelizing the input. PointNet [18] operates on unordered points set directly. The independently processed points are aggregated into global feature by max-pooling. In the following work PointNet++ [34], the authors improved PointNet by incorporating local

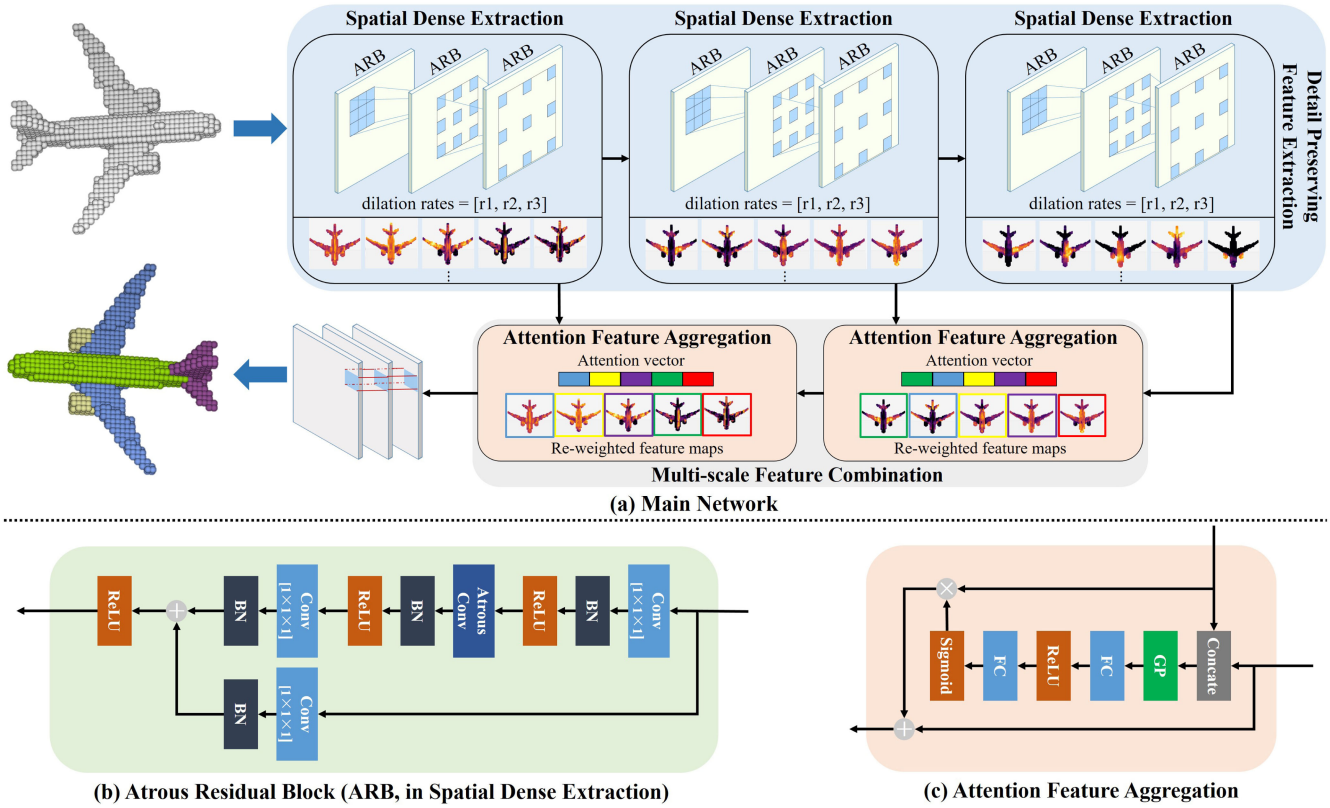


Fig. 1. An overview of the proposed Voxel Segmentation Network (VoxSegNet). (a) depicts the main network architecture. The input voxelized shape is passed through the spatial dense extraction units to extract multi-scale features preserving detailed information. Some of the visualized filter responses are shown within the units. After that, the features from different extraction stages are combined by the attention feature aggregation units. Finally, three $1 \times 1 \times 1$ convolutional layers are used to predict part labels for each voxel. In (b), the inner structure of the Atrous Residual Block (ARB) is presented. ARB is the basic component of the spatial dense extraction unit. (c) is the inner structure of the attention feature aggregation unit. This unit utilizes high stage filters to guide the feature selection of low stage.

dependencies and hierarchical feature learning in the network. Kd-Networks [35] builds a Kd-tree by recursively partitioning the space along the axis of the largest variation on the input point clouds. Li et al. [36] presented a framework generalizing typical CNNs to feature learning from point clouds. Huang et al. [37] proposed a Recurrent Slice Network (RSNet) to directly segment point clouds. The sliced points are input to a stack of bidirectional RNNs sequentially, generating features by interacting with adjacent points.

3 Method

3.1 Problem Statement

A voxelized shape can be easily obtained from a point cloud or a triangular mesh, capturing the spatial occupancy within a 3D space regularized to 3D lattices. Denote $V_{i,j,k}$ as the state value on the specific discrete voxel coordinate (i, j, k), reflecting the state of whether this grid is occupied. This is an intuitive and effective representation for a 3D object, free from the variance introduced by different meshing methods or the lack of topology resulted from applying point clouds representation.

Given a 3D object represented by volumetric matrix V, the goal of part segmentation is to assign a part category label to each occupied element $\{v \in V | V_{location(v)} = 1\}$. In this paper, we design a deep learning framework to model the function $f(v) = l_p$, where $f : V \mapsto L_p$, $l_p \in L_p$ and $L_p = \{1, 2, \ldots, K\}$ is the set of part labels.

3.2 Network Architecture

Our full network architecture is visualized in Fig. 1. Taking the voxelized shape as input, the Spatial Dense Extraction (SDE) units are used to extract discriminative features encoding detailed information from raw data. As shown in Fig. 1a, an SDE unit consists of stacked Atrous Residual Blocks (ARBs) with a user specified dilation rate for each ARB. The inner structure of an ARB is depicted in Fig. 1b. The $1 \times 1 \times 1$ convolutional layers are used to change the channel number of feature maps, reducing the computational complexity of the atrous convolutional layer. Batch normalization and ReLU activation are used after both the first $1 \times 1 \times 1$ convolutional layer and the atrous convolutional layer. The sequence of SDEs extract multi-scale features, which encode information from low-level geometry to high-level semantics and preserve the spatial resolution of input signals. Such features from different extraction stages are combined by the Attention Feature Aggregation (AFA) units. In an AFA unit, after concatenation between feature maps from high- and low-stages, global pooling layer is applied to abstract features channel-wise. Then, fully connected layers are used to compute the attention weights for low-stage features. Finally, stacked $1 \times 1 \times 1$ convolutional layers are used to predict the semantic label per voxel, which can be considered as Multilayer Perceptrons (MLPs). Using Softmax as the cost function, the optimization goal can be represented as $\min_\theta J(\theta)$:

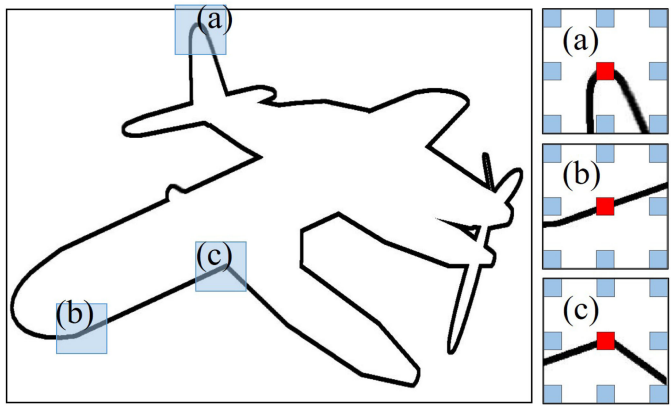


Fig. 2. Atrous convolution may face problem extracting discriminative features from sparse data. Without loss of generality, we take a 2D case as an example. An atrous kernel with size 3×3 and dilation rate 3 is used to extract features from an airplane boundary image. At three locations (a), (b), and (c), only the weight at the center of the kernel is activated, resulting in the same convolution value. This means the filter failed to capture different local patterns from the sparse boundary image.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{\#L_p} 1\{y^{(i)} = j\} log \frac{e^{\hat{f}_j(v^{(i)}|\theta)}}{\sum_{l_p=1}^{\#L_p} e^{\hat{f}_{l_p}(v^{(i)}|\theta)}}, \quad (1)$$

where $y^{(i)}$ is the ground truth label of voxel i, $\#L_p$ is the number of semantic labels, $\hat{f}_j(\cdot)$ is the last layer output in channel j, and m is the voxel number of the input 3D shape.

Our network has two key modules: the Spatial Dense Extraction (SDE) and the Attention Feature Aggregation (AFA). In the rest of this section, we first discuss the characteristics of 3D atrous convolution, and then detailedly introduce the modules proposed to extract features from 3D volumetric data.

3.3 Atrous Convolution in 3D

Atrous (dilated) convolution is originally introduced in *algorithme àtrous* for wavelet decomposition. Recently, it is widely applied in 2D segmentation tasks to extract semantic-rich feature maps, instead of using stacked spatial pooling and strided convolution [38], [39]. The key idea is to enlarge the receptive fields by inserting 'holes' in the convolutional kernels to remove down-sampling operations and to benefit from fewer training weights in sparse kernels. In this paper, we further extend the atrous convolution to 3D for discriminative feature extraction from volumetric data.

In 3D, atrous convolution can be defined as:

$$g[i,j,k] = \sum_{x=1}^{X}\sum_{y=1}^{Y}\sum_{z=1}^{Z} f[i+rx, j+ry, k+rz]h[x,y,z], \quad (2)$$

where $f[\cdot]$ is the input signal, $g[\cdot]$ is the output signal, $h[\cdot]$ denotes the filter, and $r \in \mathbb{Z}_+$ corresponds to the dilation rate for sampling $f[\cdot]$. By changing the dilation rate, we cloud control the field-of-view of the corresponding convolutional kernel easily, thus extract features with different levels of abstraction [40].

However, different from the traditional convolutional networks usually applied to dense smooth data like photos and videos, voxelized volumes of a 3D object often have sparse structure (2D manifolds in a 3D euclidean space). In addition, an atrous convolutional kernel also has sparse structure, making it only capture the information with non-zero weights (active weights). It is highly possible an atrous kernel can not discriminate any local pattern due to the sparsities of both the input signal and the kernel weights. As we can see in Fig. 2, the filter failed to capture discriminative features from the three local patterns (a), (b) and (c). Specifically, the atrous convolution kernel is degenerated to 1×1. A detailed discussion can be found in Section 4.3.

In summary, directly applying atrous convolution to 3D voxelized volumes will result in information loss. In the next subsection, we present a method to overcome this problem.

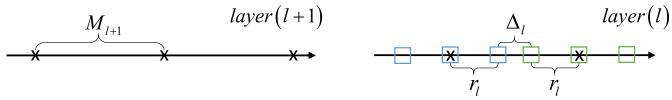


Fig. 3. Maximum distance between active weights of adjacent layers (shown in 1-D).

3.4 Spatial Dense Extraction

Similar to the Hybrid Dilated Convolution [41] in 2D scenario, we introduce the Spatial Dense Extraction (SDE) unit to overcome the information loss in feature extraction from 3D volumetric data. An SDE unit consists of multiple stacked atrous convolutional layers with dilation rates of $[r_1, r_2, \ldots, r_n]$ and kernel size $K \times K \times K$. The basic design principle is to ensure the receptive field of the top layer in an SDE unit fully cover a cubic region without any holes.

To model the sparsity of an atrous convolutional kernel, the "maximum distance between active weights" is defined as

$$\begin{aligned} M_l &= max[|M_{l+1} - 2r_l|, r_l], \\ M_n &= r_n, \end{aligned} \quad (3)$$

in which $|\cdot|$ returns 1-norm value, and $l \in \{1, 2, \ldots, n\}$ represents the layer index in an SDE unit. As shown in Fig. 3, in 1-D situation, the 'maximum distance between active weights (non-zero weights)' from a higher layer's atrous kernel defines a line segment M_{l+1}. The current layer's active weights split this line segment into at most two kinds of segments: the distance between the current layer's active weights $\Delta_l = |M_{l+1} - 2r_l|$, and that between the active weights from the current layer and the higher layer r_l.

At the same time, r and M should be subject to:

$$\begin{aligned} 1 \le r_1 \le r_2 &\le \cdots \le r_n, \\ M_2 &\le K. \end{aligned} \quad (4)$$

The second layer's M_2 is not greater than the kernel size (i.e., the largest hole is smaller than the kernel size). Even for an extreme condition, the holes will be filled by the lowest layer convolution with dilation rate $r_1 = 1$ in an SDE unit.

As we mentioned earlier in Section 3.3, 3D volumetric shapes are sparse, making it harder to extract discriminative features through dilated kernels. By stacking the atrous convolutions with dilation rates subjected to Equations (3) and (4), the receptive field of an SDE unit can fully cover a cubic region of the input feature maps, which helps to encode more discriminative information from sparse data.

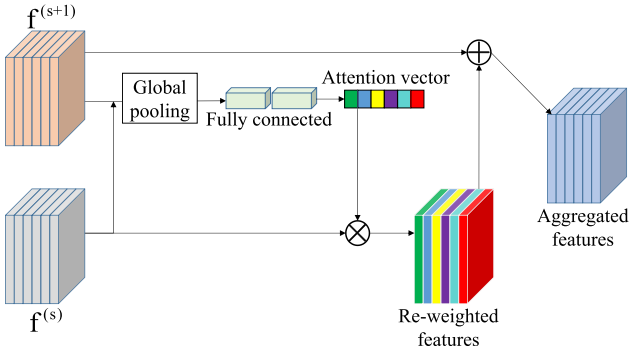


Fig. 4. Architecture of attention feature aggregation. $\mathbf{f}^{(s)}$ and $\mathbf{f}^{(s+1)}$ are feature maps from two adjacent SDE stages. The low-stage feature maps $\mathbf{f}^{(s)}$ are re-weighted by the attention vector, and then added to the high-stage ones, generating more informative aggregated features.

In our network architecture, the SDE module is a basic component of multi-stage feature extraction from volumetric data. In one stage, we use three residual blocks [42], each of them has a bottleneck structure with an atrous convolutional layer with dilation rate r_l.

3.5 Attention Feature Aggregation

In convolutional neural networks, features from deep layers learn more about high-level semantic information, while features of shallow layers keep rich spatial structural details. To accurately segment an object into semantic parts, it is necessary to combine multi-level features together. However, previous encoder-decoder networks for fine-grained and dense tasks usually directly integrate multi-level features indiscriminately (e.g., U-Net [33]). The equal weights for different channels of integrated features are defective due to the redundant details and distractions from different parts.

To address the problem, we propose the Attention Feature Aggregation (AFA) unit leveraging attention mechanism [43], [44], which can extract informative features and suppress indiscriminative ones. As shown in Fig. 4, the AFA unit takes both low- and high-stage feature maps as input to compute the channel attention weights. Then, the low-stage feature maps are re-weighted by the channel attention weights and added to the high-stage ones. Using high-level semantic information to guide the selection of low-level detailed information, multi-scale features with stronger discriminative ability are aggregated, leading to a segmentation which is both semantically consistent and accurate.

To formulate the AFA, we unfold convolutional features $\mathbf{f}$ as $\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_C]$, where $\mathbf{f}_i \in \mathbb{R}^{W \times H \times D}$ is the ith slice of $\mathbf{f}$ and C is the channel number. First, the input feature maps from two feature extraction stages are concatenated and then passed through a global average pooling layer to extract the channel-wise global context:

$$\mathbf{z}(i) = \frac{1}{W \times H \times D} [\![\mathbf{f}^{(s)}, \mathbf{f}^{(s+1)}]\!]_i, \tag{5}$$

in which $[\![\cdot]\!]$ means concatenation, $\mathbf{f}^{(s)}$ is the feature maps from the sth extraction stage, and $\mathbf{z} \in \mathbb{R}^{C_s + C_{s+1}}$ is the channel-wise global feature vector. Note that this equation computes the ith element of the vector. Then, two fully connected layers are used to learn the aggregation feature for each channel:

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{W}_1 \bullet \mathbf{z} + \mathbf{b}_1, \\ \mathbf{u}_2 &= \mathbf{W}_2 \bullet ReLU(\mathbf{u}_1) + \mathbf{b}_2, \end{aligned} \tag{6}$$

where $\bullet$ denotes matrix multiplication, $\mathbf{W}_1 \in \mathbb{R}^{C_s \times (C_s + C_{s+1})}$, $\mathbf{W}_2 \in \mathbb{R}^{C_s \times C_s}$ are fully connected weights, and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{C_s}$ are the bias parameters. To define the attention for the channels of feature maps, a Sigmoid operation is applied to $\mathbf{u}_2$ to generate attention of each channel i:

$$\mathbf{a}(i) = \frac{e^{\mathbf{u}_2(i)}}{e^{\mathbf{u}_2(i)} + 1}, \tag{7}$$

where $\mathbf{u}_2(i)$ is the feature value of channel i and $\mathbf{a} \in \mathbb{R}^{C_s}$ is the channel-wise attention vector.

In the decoding phase of our method, AFA units are applied progressively between adjacent SDE units. The attentive information from one unit serves as a guidance for the next to adaptively generate new attentions. Finally, multi-level features are aggregated to form more discriminative ones:

$$\tilde{\mathbf{f}}^{(s)} = \mathbf{f}^{(s)} \bullet \mathbf{a} \oplus \mathbf{f}^{(s+1)}, \tag{8}$$

in which $\oplus$ represents element-wise addition operation. In the prediction phase, unit stride convolutional layers followed by a Softmax operation are used to generate voxel-wise part label prediction.

4 Experiments and Discussion

Datasets. We conduct an experiment on a large-scale shape part annotation dataset (ShapeNetSem) introduced by Yi et al. [45], which augments a subset of the ShapeNet models with semantic part annotations. The dataset contains 16 categories of shapes, with 2 to 6 parts per category. In total, there are 16,881 models and 50 parts. Wang et al. [21] augmented the dataset by projecting the point label back to the triangle faces of the corresponding 3D mesh, and condensed the point clouds by uniformly re-sampling the triangle faces. We use the same augmented dense point clouds data to generate volumetric data, and the same training/testing split. We also evaluate the performances on the three major subsets of a 3D shape co-segmenation dataset (COSEG) [46].

Training Details. In a data preprocessing step, each point cloud is centered and rescaled to a unit sphere. After that, the normalized point cloud is voxelized into a 3D grid with a user-specified size. The input feature to our VoxSegNet is the voxelized 3D shape with size $48 \times 48 \times 48$, in which each value represents the occupancy of the corresponding voxel grid. As for the network structure, the dilation rates of the three SDE units in our VoxSegNet are set to be [1, 1, 1], [1, 3, 5], and [1, 3, 5] respectively. Thus the SDE units can extract features from low-level to high-level. We use kernel size 3 for all of the atrous convolutional kernels in this work. We train the network using Adaptive Moment Estimation (Adam) optimization [47] with batch size 4. The initial learning rate is 0.001, β_1 is 0.9, β_2 is 0.999, and ϵ is 10^{-8}. We augment the training data by rotating each shape $n\pi/6$ around the upright axis, where $n \in \{0, 1, 2, \ldots, 11\}$.

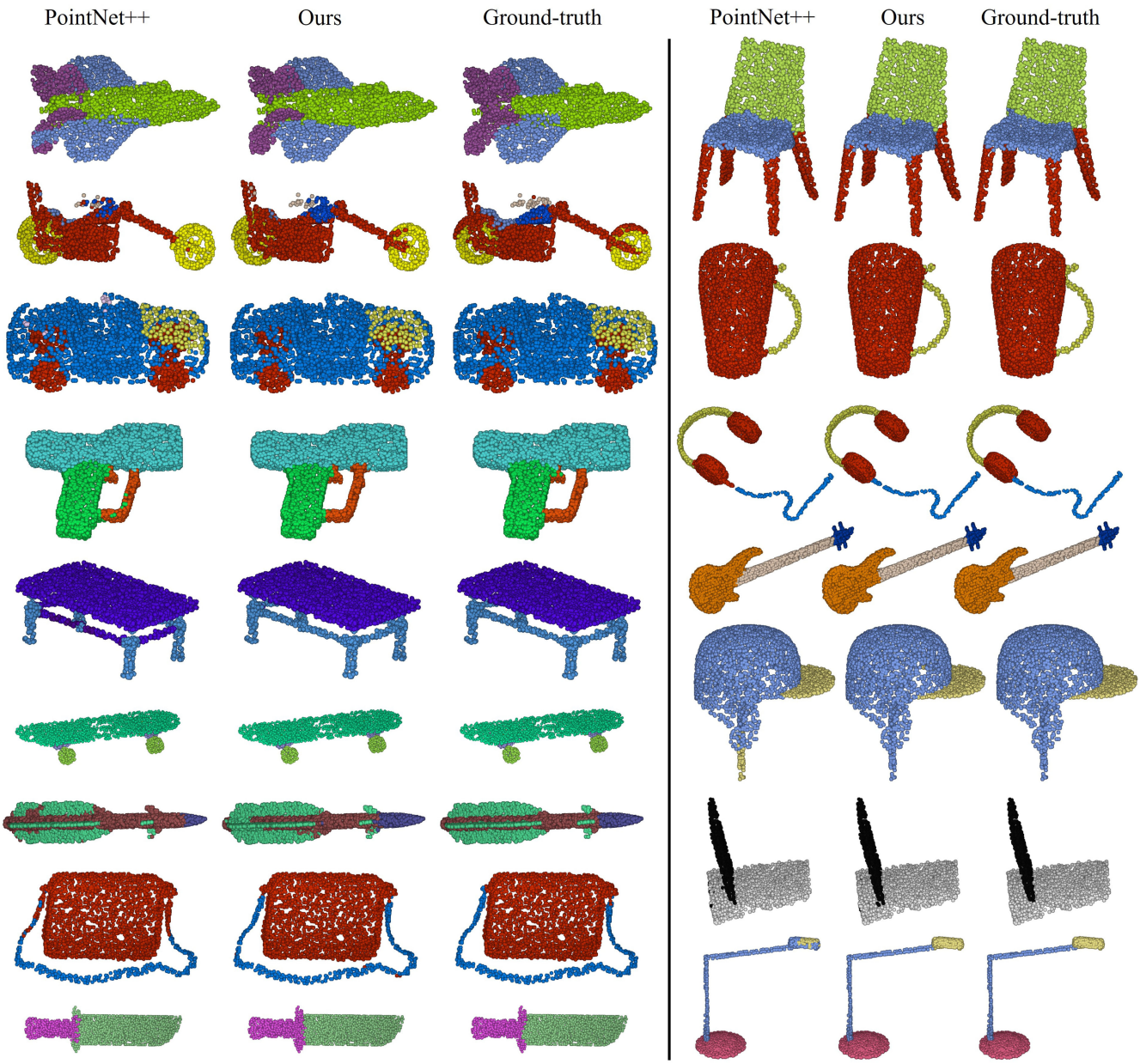


Fig. 5. Qualitative comparison of object part segmentation between PointNet++ and our method (VoxSegNet) on ShapeSegSem.

4.1 In Comparison to State-of-the-Art Methods

To evaluate the part segmentation quality, the predicted results of voxelized volumes are projected to the corresponding point clouds. We then compute the Intersection over Union (IoU) as the metric. Specifically, for each shape, all the part class IoUs are averaged to obtain the object IoU. Per category average IoU is computed by averaging across all shapes with the certain category label. Then an overall average IoU is computed through a weighted average of per category IoU. The weights are just the number of shapes in each category.

Fig. 5 presents some examples of segmentation results of our VoxSegNet compared with those of PointNet++ [34]. As illustrated, our results are visually better in most cases. Specifically, the boundary between different parts can be separated more accurately. Taking the lamp as an example, the lamp base, head, and the connection part are predicted properly by our method, while the PointNet++ result shows some artifacts around the lamp head (see the right-bottom of Fig. 5). As for the segmentation results of the rocket, our VoxSegNet can separate out the fin (near the rocket nose) from the frame of the rocket while PointNet++ cannot. Similar observations can be seen for other shapes such as the motorbike, pistol, and bag, etc.

The numerical comparison on ShapeNetSem is conducted with a learning-based technique [45] which uses per-point local geometric features and correspondences between shapes, and six latest deep learning based methods [18], [21], [22], [34], [36], [48]. As shown in Table 1. Our proposed method performs better or comparable to other methods in most of the categories. Specifically, we achieve the best overall average IoU. In individual categories, we rank the best in 9 out of 16 categories. We achieve promising object part predictions without post-processing such as Conditional Random Field (CRF) refinement as is done in O-CNN [21].

TABLE 1
Object Part Segmentation Results on ShapeSegSem, Measured by Averaged IoU (%)

	mean	plane	bag	cap	car	chair	e.ph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate	table
# shapes	-	2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Yi2016[45]	81.4	81.0	78.4	77.7	75.7	87.6	61.9	92.0	85.4	82.5	95.7	70.6	91.9	**85.9**	53.1	69.8	75.3
SpecCNN[48]	84.7	81.6	81.7	81.9	75.2	90.2	74.9	**93.0**	86.1	84.7	95.6	66.7	92.7	81.6	60.6	**82.9**	82.1
PointNet[18]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++[34]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
O-CNN[21]	85.9	85.5	87.1	84.7	77.0	91.1	**85.1**	91.9	87.4	83.3	95.4	56.9	96.2	81.6	53.5	74.1	84.4
SSCN[22]	86.0	84.1	83.0	84.0	**80.8**	91.4	78.2	91.6	**89.1**	85.0	95.8	73.7	95.2	84.0	58.5	76.0	82.7
PointCNN[36]	86.1	84.1	86.5	86.0	**80.8**	90.6	79.7	92.3	88.4	**85.3**	**96.1**	77.2	95.3	84.2	64.2	80.0	83.0
VoxSegNet	**87.5**	**86.2**	**88.7**	**91.9**	79.8	**92.0**	76.5	92.0	86.4	84.2	**96.1**	**78.4**	**96.3**	83.7	**65.4**	77.0	**86.2**

(***Bold number***: *the highest score; underlined number: the second highest score).*

We also extend our evaluation to the three major subsets of COSEG. Since this dataset is much smaller than ShapeNetSem, and other methods do not report results on it, we choose two latest methods using different data representations for performance comparison. In particular, ShapePFCN [12] is chosen as the representative of the projective images-based methods, and PointCNN [36] for the point clouds-based methods. To ensure comparison fairness, the labeling accuracy and part IoU values (%) are measured on point clouds uniformly sampled from original meshes. All results are provided in Table 2, where our method achieves comparable results with the state-of-the-art methods using different shape representations.

4.2 Visualization and Analysis

For image understanding using deep convolutional neural networks, it is known that the learned filters are activated when important image features appear, and the filters in different convolutional stages capture different levels of features. In other words, filters in the first convolutional layer are usually activated by the object edges in the image, while the higher level layer's filters capture more complex patterns [49]. A similar phenomenon can be observed on our VoxSegNet, which facilitates our understanding of the segmentation network.

In Fig. 6, we visualize some filters in different feature extraction stages of VoxSegNet by color-coding the responses of the input voxelized shapes. In detail, (a) is the input voxelized shape. For the first feature extraction stage, (b1) and (b2) are the responses of two filters which capture low-level geometric patterns. While the responses of filters from the second feature extraction stage (c1), (c2), and (c3) show the filters' ability to capture high-level shape features. In the airplane example (the first row of Fig. 6), filter (b1) tends to capture large planar regions such as wings of the airplane, while filter (b2) tends to capture sharp and pointed areas. In stage two, filters tend to capture regions possessing semantic information, discriminating wings (c2) from fuselage (c3) and engines (c1). Similar observations hold true for the other two samples. Note that for the chair in the third row, despite that the legs and the back are composed of bars with similar low-level geometric features, (c1) and (c3) discriminate them successfully.

TABLE 2
Evaluation on the three Major Subsets of COSEG

	Chairs (400)		Vases (300)		Aliens(200)		mean	
Methods	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU
ShapePFCN [12]	92.42	87.17	92.72	80.73	95.62	90.55	93.18	85.83
PointCNN [36]	97.81	94.61	93.60	78.54	95.52	90.26	96.01	88.63
VoxSegNet	97.84	95.11	93.14	79.46	94.05	87.98	95.57	88.69

4.3 Ablation Study

In this section, in order to evaluate the effects of proposed modules, we conduct several experiments gradually adding SDE and AFA modules to the baseline network. The 3D version of U-Net [33] is used as the baseline segmentation network (U-Net 3DCNN). Then dilated convolutional layers are used to remove downsampling operations (Atrous 3DCNN). After that, AFA and SDE modules are added to Atrous 3DCNN respectively (Atrous+AFA, SDE+concat), investigating the two modules' effects individually. Finally, we combine the two proposed modules to form a new architecture (SDE+AFA).

The performances of these network architectures are reported in Table 3. Besides the average IoU across all categories, we also compute the precision and recall on a subset of ShapeNetSem, in which the objects contain more than 3 part labels (i.e., Airplane, Car, Chair, Lamp, and Motorbike). The corresponding precision-recall values are also visualized in Fig. 7, for better illustration. To provide more detailed information of these architectures, we report the number of parameters, training/testing time, and memory usage in Table 4. All of these networks are trained and tested on an NVIDIA Titan X card. The training time is measured per batch with size 4, and the testing time is measured per instance. As for the number of parameters, the proposed architectures have the same order of magnitude with the baseline model.

In summary, by comparing the baseline (U-Net 3DCNN) with the proposed networks, it is clear that including SDE and AFA modules improves the segmentation performance by a large margin. The results also show that each of our proposed modules can improve the performance.

In detail, we conduct experiments on the following network architectures:

1) U-Net 3DCNN. We use the popular encoder-decoder architecture U-Net [33] as a baseline. In the encoding

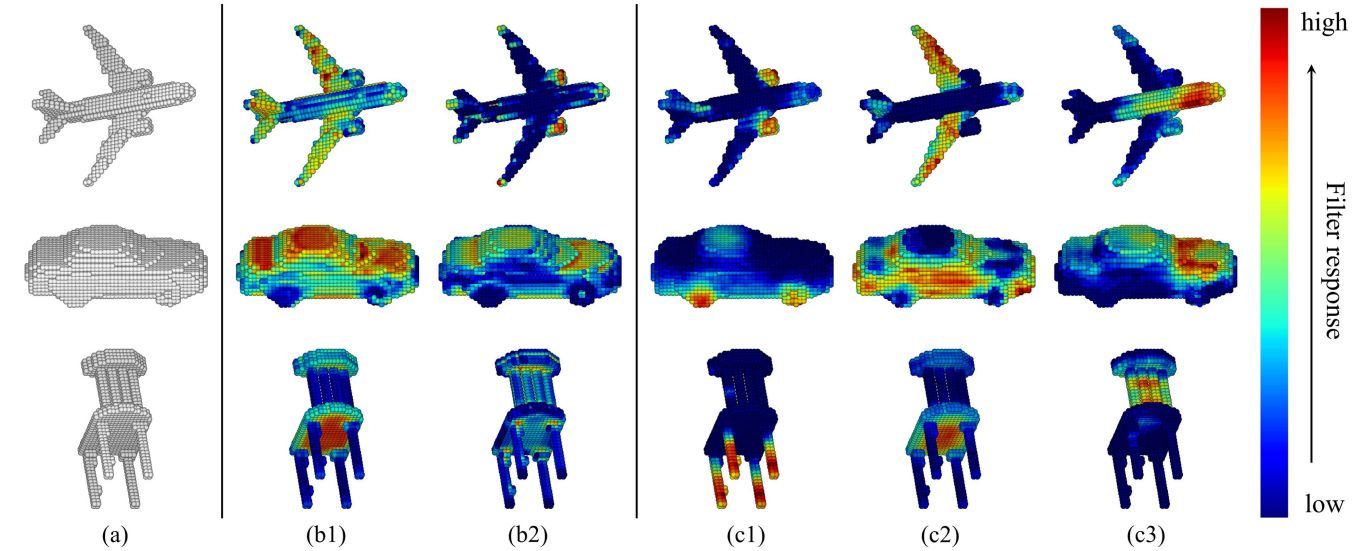


Fig. 6. Visualization of filter responses from different feature extraction stages. For each row, (a) is the input voxelized shape; (b1) and (b2) show filter responses from the first SDE module; (c1), (c2), and (c3) show filter responses from the second SDE module in feature extraction procedure of the proposed segmentation network. Note that low-stage filters capture geometric patterns while high-stage ones capture regions with semantic meanings, which demonstrates the effectiveness of our proposed feature extraction module.

phase, 3D convolutions with kernel size 3 and stride 1 are applied. The encoding phase contains three $(conv, bn, relu, maxpooling)$ blocks. The decoding phase consists of three $(deconv, bn, relu)$ blocks with skip layer connections. The deconvolution has kernel size 3 and stride 2. After that, three stacked 3D convolutional layers with unit kernel size and stride are applied to get voxel-wise prediction. This architecture achieves quite good object part segmentation results with an average IoU of 83.97 percent.

2) Atrous 3DCNN. In this experiment, we try to find out the effect of directly applying 3D dilated convolutions to extract features from volumetric data. In the Atrous 3DCNN architecture, three ARBs with dilation rates 2, 3, 4 are applied to extract features from three different scales respectively. After this encoding phase, three residual blocks with skip layer connections are used to aggregate different levels of features. Then, convolutional layers with unit kernel size and stride are used to predict voxel-wise labels. This architecture achieves an average IoU of 83.71 percent.

Different from intuition, the Atrous 3DCNN does not improve segmentation performance even though the spatial resolution is preserved by dilated convolutions. As shown in Fig. 8, almost the whole shape is activated by the first ARB's filter, which indicates that the filter failed to capture discriminative features. As we discussed earlier in Section 3.3, for sparse volumetric data, it is highly possible that a sparse kernel can not discriminate different local patterns. In the Atrous 3DCNN, directly concatenating the less discriminative information through skip

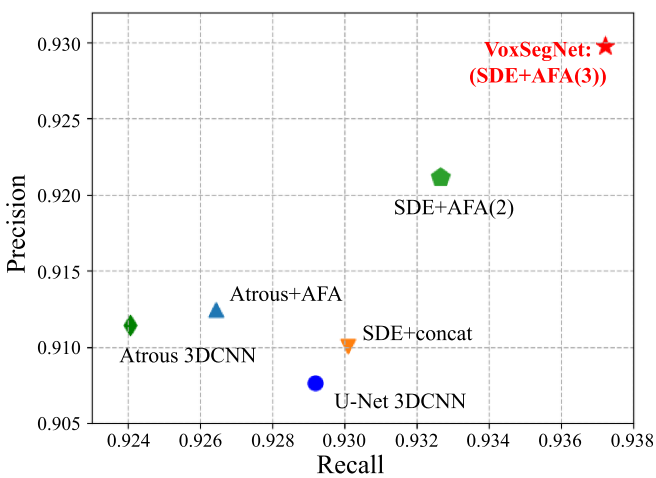


Fig. 7. Visualized precision-recall comparison between different network architectures (on categories with more than 3 part labels).

TABLE 3
Comparison between Different Network Architectures

Method	IoU	Precision ($>$ 3 labels)	Recall ($>$ 3 labels)
U-Net 3DCNN	83.97	90.76	92.92
Atrous 3DCNN	83.71	91.14	92.40
Atrous+AFA	84.22	91.25	92.65
SDE+concat	84.34	91.01	93.01
SDE+AFA(2)	86.24	92.12	93.27
SDE+AFA(3)	**87.46**	**92.98**	**93.72**

TABLE 4
Parameter Number, Running Time, and Memory Usage

Methods	#Parameters	Training time	Testing time	Memory cost
U-Net 3DCNN	291.4K	0.0684s	0.0115s	2451MB
Atrous 3DCNN	199.0K	0.3175s	0.0293s	8611MB
Atrous+AFA	141.3K	0.2518s	0.0242s	8595MB
SDE+concat	203.3K	0.3354s	0.0299s	8611MB
SDE+AFA(2)	211.6K	0.3353s	0.0305s	8611MB
SDE+AFA(3)	325.3K	0.5483s	0.0444s	8595MB

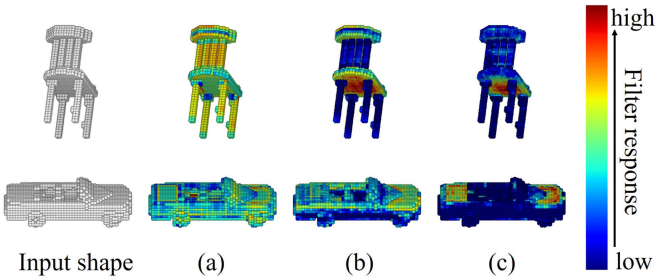


Fig. 8. Visualization of filter responses from different ARB modules in Atrous 3DCNN. For each row, (a) is the response for the first ARB module; (b) for the second ARB module ; (c) for the third ARB module. The progressively concentrating responses show the filters in higher layer ARB are able to capture more discriminative local patterns.

layer connections will decrease the quality of feature maps. This motivates us to propose the SDE and AFA modules.

3) Atrous+AFA. This experiment aims to investigate the effect of applying AFA modules. Different from the Atrous 3DCNN architecture, the Atrous+AFA uses two AFA modules to aggregate the feature maps from different ARBs rather than direct concatenation. Specifically, while being fed into AFA modules, the lower level ARB features are enhenced with a weight of 10.0, in order to preserve more low-level information. This architecture achieves an average IoU of 84.22 percent ($+0.25$ percent compared to the baseline).
4) SDE+concat. We study the effect of applying SDE modules. To capture robust features from sparse volumetric data, the SDE+concat uses three atrous residual blocks to form an SDE unit. The encoding phase consists of two SDE units with dilation rates [1, 1, 1] and [1, 3, 5] respectively. The first SDE unit extracts local features, while the second encodes long-range information. Then, the different levels of features are concatenated directly along the feature channel axis. Finally, voxel-wise results are predicted by three stacked convolutional layers. This architecture achieves an average IoU of 84.34 percent ($+0.37$ percent compared to the baseline).
5) SDE+AFA. In this experiment, we investigate the effect of using both SDE and AFA modules. Similar to the SDE+concat network, this architecture uses SDE units to extract different levels of features. In feature aggregation step, AFA modules are applied to combine the features from different SDE units. Attention mechanism can select discriminative and effective features according to inputs. Specifically, since high-stage features usually capture regions with semantic meanings, through the AFA module, low-stage feature maps activated by basic geometric patterns can be selected to better discriminate semantic parts. This architecture achieves an average IoU of 86.24 percent ($+1.90$ percent compared to the SDE+concat).

In Fig. 9, several part segmentation results with and without the use of AFA are reported. As we can see, with the help of AFA, falsely predicted regions of the chair back are correctly differentiated from its leg, even though they share similar low-level geometric patterns. Such observations can

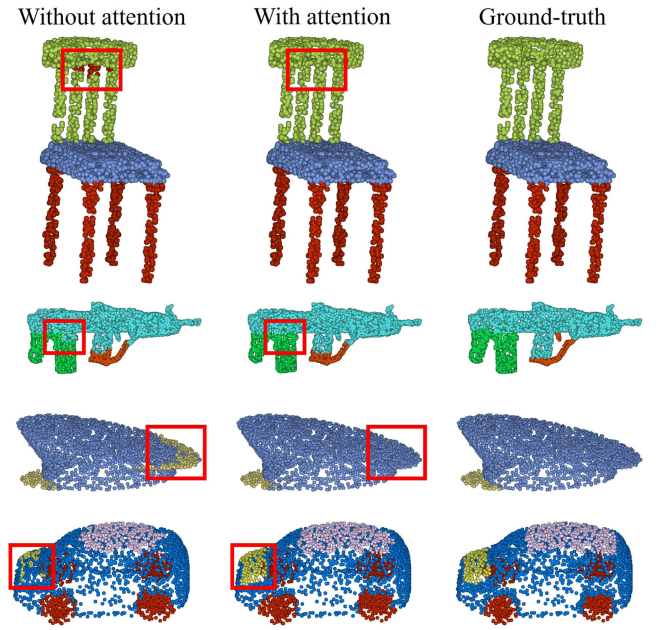


Fig. 9. Our proposed attention feature aggregation benefits part segmentation. There are several comparisons between segmentation results with and without the use of AFA. For each object category, the first column is the direct concatenation result (SDE+concat); the second column is the attention feature aggregation result (SDE+AFA); and the third column is the ground truth segmentation. The comparison regions are highlighted by red boxes.

be seen for other shapes such as the hat and the car. These demonstrates the effectiveness of using AFA modules for extracting semantic discriminative features.

We also note that the network structure with three SDE units outperforms that with two SDE units. The three-SDE version has a larger receptive field of 43, which informs the network with a much larger region of input data than the two-SDE version with a receptive field of 25. Moreover, by using three SDE units, the network can integrate information from more different scales. We apply this architecture in our VoxSegNet.

4.4 Application: Fine-Grained Shape Clustering

With the emergence of large shape collections, the shapes within each category exhibit significant variations. For example, chair models from the Trimble 3D Warehouse can be further classified into sub-classes such as chairs-with-arms, swivel chairs, rocking chairs, etc. Fine-grained category information is important for shape understanding and will benefit the exploration of the variability of a shape collection.

In this subsection, we show that with the help of semantic part information, fine-grained categories among a specific parent class can be investigated. In particular, given a collection of shapes from the same class (eg., chair), semantic part segmentation is conducted using our VoxSegNet. Then, we take the feature maps before the $1 \times 1 \times 1$ convolutional layers as the shape descriptor and extract features per part. In detail, for each semantic part, the feature maps are multiplied element-wise by a mask (according to the part existence) and then averaged along spatial axes, generating a 64 (number of channels) dimensional feature. The features from different parts are concatenated to form the

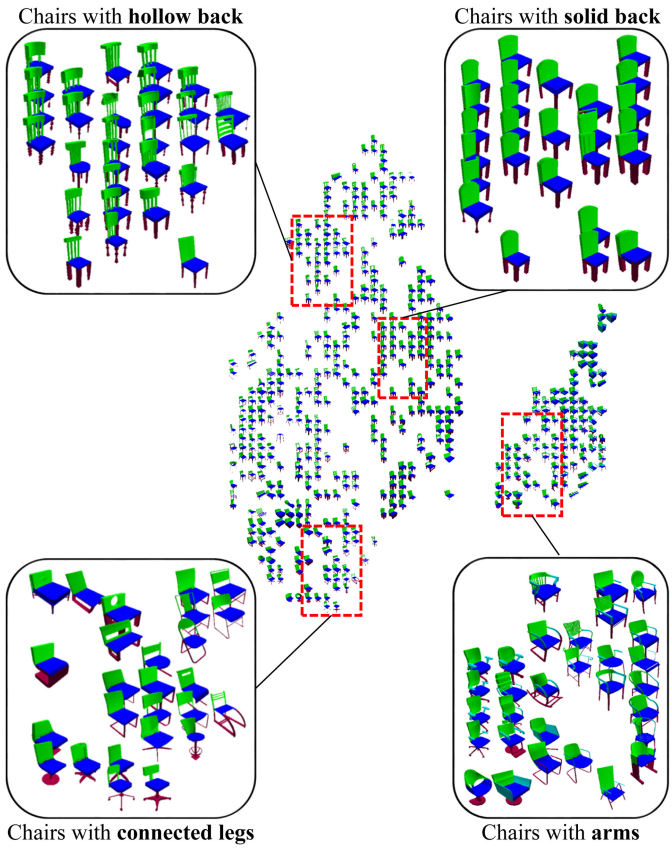


Fig. 10. Fine-grained clustering results using the proposed part-based features. This is a t-SNE visualization. We can observe that the chairs in the same dashed box share similar geometric structures, which demonstrates the good description ability of part-based features.

part-based feature, which describes a 3D model. Such features contain structural information, which can benefit fine-grained discrimination tasks. For example, in the chair category, the feature consists of four parts (back, seat, leg, and arm).

In Fig. 10, we show a t-SNE visualization of the part-based features extracted from the chair category. As we can see, chairs with arms are differentiated from those without arms, and shapes with similar geometric structures are near each other in the feature space. Note that the features are not specifically designed for object classification. In fact, semantic part segmentation features might be ambiguous for different detailed geometric patterns in the same part category. Acknowledging room for future improvement, the simple part-based features perform well in clustering the chairs according to shape structures.

5 Conclusion

In this paper, we have presented a novel deep neural network for shape part segmentation using volumetric data. Our method is motivated by extracting features better encoding detailed information under limited resolution. Specifically, we introduce SDE to alleviate the loss of detailed information caused by sub sampling and data sparsity. Moreover, we propose AFA to fuse the multi-scale information produced by SDEs through attention mechanism. The experimental results and the ablation study

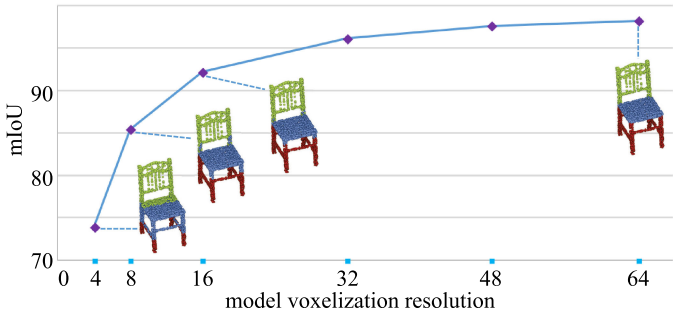


Fig. 11. Finding the segmentation upper bound. We change the voxel resolution and evaluate the segmentation performance by projecting the labels of the voxel grids to the corresponding point clouds.

demonstrate the effectiveness of our proposed method. We can draw the conclusion that among various 3D representations, volumetric data is able to achieve state-of-the-art results for part segmentation task.

Future Work. In order to further improve the segmentation accuracy, one can follow the previous work [20] to use larger resolution volumetric data, or to extract finer features better describing the detailed structures as we did in this paper. We argue that the latter way possesses much potential. As shown in Fig. 11, we performed an experiment to investigate the segmentation upper bound under different resolutions. Although a larger resolution results in a higher mIoU, it is harder to improve the upper bound as the resolution increases. In addition, there is still an unignorable gap between the performance upper bound and the state of the art, indicating room for improvement. Therefore, we believe it is necessary to conduct further research on methods focusing on finer feature extraction.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) under Grant 61602020, and Grant 61732016.

References

[1] V. Kreavoy, D. Julius, and A. Sheffer, "Model composition from interchangeable components," in *Proc. 15th Pacific Conf. Comput. Graph. Appl.*, Oct. 2007, pp. 129–138.

[2] C. Zhu, K. Xu, S. Chaudhuri, R. Yi, and H. Zhang, "SCORES: Shape composition with recursive substructure priors," *ACM Trans. Graph. (SIGGRAPH Asia 2018)*, vol. 37, no. 6, 2018, Art. no. 211.

[3] K. Xu, H. Zhang, D. Cohen-Or, and B. Chen, "Fit and diverse: Set evolution for inspiring 3D shape galleries," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 57:1–57:10, Jul. 2012. [Online]. Available: http://doi.acm.org/10.1145/2185520.2185553

[4] R. Hu, W. Li, O. van Kaick, A. Shamir, H. Zhang, and H. Huang, "Learning to predict part mobility from a single static snapshot," *ACM Trans. Graph.*, vol. 36, no. 6, 2017. Art. no. 227.

[5] Y. Yang, W. Xu, X. Guo, K. Zhou, and B. Guo, "Boundary-aware multidomain subspace deformation," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 10, pp. 1633–1645, Oct. 2013. [Online]. Available: doi.ieeecomputersociety.org/10.1109/TVCG.2013.12

[6] D. Huber, A. Kapuria, R. Donamukkala, and M. Hebert, "Parts-based 3D object classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 82–89. [Online]. Available: http://dl.acm.org/citation.cfm?id=1896300.1896313

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[9] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *CoRR*, vol. abs/1502.03044, 2015. [Online]. Available: http://arxiv.org/abs/1502.03044

[10] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 922–928.

[11] A. K. F. Y. L. Z. X. T. J. X. Z. Wu, S. Song, "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.

[12] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2017, pp. 6630–6639.

[13] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki, "Gift: A real-time and scalable 3d shape search engine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5023–5032.

[14] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.114

[15] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," in *Proc. Eurographics Symp. Geometry Process.*, 2015, pp. 13–23. [Online]. Available: http://dx.doi.org/10.1111/cgf.12693

[16] D. Boscaini, J. Masci, E. Rodoià, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3197–3205. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157382.3157455

[17] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2015, pp. 832–840. [Online]. Available: http://dx.doi.org/10.1109/ICCVW.2015.112

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 77–85, Jul. 2017.

[19] 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," *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 5648–5656, Jun. 2016.

[20] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6620–6629.

[21] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 72.

[22] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9224–9232.

[23] S. Katz and A. Tal, "Hierarchical mesh decomposition using fuzzy clustering and cuts," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 954–961, Jul. 2003. [Online]. Available: http://doi.acm.org/10.1145/882262.882369

[24] M. Ben-Chen and C. Gotsman, "Characterizing shape using conformal factors," in *Proc. 1st Eurographics Conf. 3D Object Retrieval*, 2008, pp. 1–8. [Online]. Available: http://dx.doi.org/10.2312/3DOR/3DOR08/001–008

[25] Q. Huang, M. Wicke, B. Adams, and L. Guibas, "Shape decomposition using modal analysis," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 407–416, Apr. 2009.

[26] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, and H. Zhang, "Contextual part analogies in 3D objects," *Int. J. Comput. Vis.*, vol. 89, no. 2–3, pp. 309–326, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0279-0

[27] J. Zhang, J. Zheng, C. Wu, and J. Cai, "Variational mesh decomposition," *ACM Trans. Graph.*, vol. 31, no. 3, pp. 21:1–21:14, Jun. 2012. [Online]. Available: http://doi.acm.org/10.1145/2167076.2167079

[28] O. Kin-Chung Au, Y. Zheng, M. Chen, P. Xu, and C.-L. Tai, "Mesh segmentation with concavity-aware fields," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 7, pp. 1125–1134, Jul. 2012. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.131

[29] E. Kalogerakis, A. Hertzmann, and K. Singh, "Learning 3D mesh segmentation and labeling," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 102:1–102:12, Jul. 2010. [Online]. Available: http://doi.acm.org/10.1145/1778765.1778839

[30] Z. Xie, K. Xu, L. Liu, and Y. Xiong, "3D shape segmentation and labeling via extreme learning machine," *Symp. Geometry Process.*, vol. 33, no. 5, pp. 85–95, 2014.

[31] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "FPNN: Field probing neural networks for 3D data," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 307–315. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157096.3157131

[32] B. Graham, "Sparse 3D convolutional neural networks," *CoRR*, vol. abs/1505.02890, 2015. [Online]. Available: http://arxiv.org/abs/1505.02890

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[34] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 5099–5108, 2017.

[35] R. Klokov and V. S. Lempitsky, "Escape from cells: Deep KD-networks for the recognition of 3D point cloud models," *CoRR*, vol. abs/1704.01222, 2017. [Online]. Available: http://arxiv.org/abs/1704.01222

[36] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 826–836. [Online]. Available: http://papers.nips.cc/paper/7362-pointcnn-convolution-on-x-transformed-points.pdf

[37] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation on point clouds," *CoRR*, vol. abs/1802.04402, 2018. [Online]. Available: http://arxiv.org/abs/1802.04402

[38] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. [Online]. Available: http://arxiv.org/abs/1706.05587

[39] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Representations*, 2016.

[40] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015, Art. no. 436.

[41] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," *Workshop Appl. Comput. Vis.*, pp. 1451–1460, 2018.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[43] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Rev. Neuroscience*, vol. 2, no. 3, 2001, Art. no. 194.

[44] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *Comput. Vis. Pattern Recognit.*, pp. 7132–7141, 2018.

[45] 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," *ACM Trans. Graph.*, vol. 35, 2016, Art. no. 210.

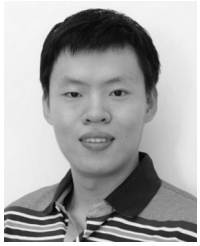
[46] Y. Wang, S. Asafi, O. van Kaick, H. Zhang, D. Cohen-Or, and B. Chen, "Active co-analysis of a set of shapes," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 165:1–165:10, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366145.2366184

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[48] L. Yi, H. Su, X. Guo, and L. J. Guibas, "Syncspeccnn: Synchronized spectral CNN for 3D shape segmentation," *Comput. Vis. Pattern Recognit.*, pp. 6584–6592, 2017.

[49] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

Zongji Wang received the BS degree in school of mathematics and systems science from Beihang University, in 2014. He is currently working toward the PhD degree with the State Key Laboratory of Virtual Reality Technology and System, School of Computer Science and Engineering, Beihang University. His research interests include computer vision and computer graphics.



Feng Lu received the BS and MS degrees in automation from Tsinghua University, in 2007 and 2010, respectively, and the PhD degree in information science and technology from the University of Tokyo, in 2013. He is currently a professor with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. His research interests include computer vision, human-computer interaction and augmented intelligence. He is a member of the IEEE.