PartNet: A Recursive Part Decomposition Network for Fine-grained and Hierarchical Shape Segmentation

Fenggen Yu[1*] Kun Liu[1*] Yan Zhang[1] Chenyang Zhu[2] Kai Xu[2†]
[1]Nanjing University [2]National University of Defense Technology

Abstract

Deep learning approaches to 3D shape segmentation are typically formulated as a multi-class labeling problem. Existing models are trained for a fixed set of labels, which greatly limits their flexibility and adaptivity. We opt for top-down recursive decomposition and develop the first deep learning model for hierarchical segmentation of 3D shapes, based on recursive neural networks. Starting from a full shape represented as a point cloud, our model performs recursive binary decomposition, where the decomposition network at all nodes in the hierarchy share weights. At each node, a node classifier is trained to determine the type (adjacency or symmetry) and stopping criteria of its decomposition. The features extracted in higher level nodes are recursively propagated to lower level ones. Thus, the meaningful decompositions in higher levels provide strong contextual cues constraining the segmentations in lower levels. Meanwhile, to increase the segmentation accuracy at each node, we enhance the recursive contextual feature with the shape feature extracted for the corresponding part. Our method segments a 3D shape in point cloud into an unfixed number of parts, depending on the shape complexity, showing strong generality and flexibility. It achieves the state-of-the-art performance, both for fine-grained and semantic segmentation, on the public benchmark and a new benchmark of fine-grained segmentation proposed in this work. We also demonstrate its application for fine-grained part refinements in image-to-shape reconstruction.

1. Introduction

Segmentation is a long-standing problem in 3D shape analysis, on which data-driven approach has shown clear advantage over traditional geometric methods [38]. With the proliferation of deep learning techniques, researchers have been seeking for exploiting the powerful feature learning ability of deep neural networks to replace the hand-crafted features used in previous data-driven approaches. In these works, deep networks are trained for multi-class labeling task, which outputs a semantic label for each geometric primitive (such as voxels [23] or points [20]).

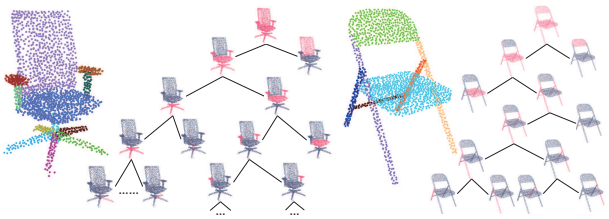

Figure 1. PartNet segments 3D point clouds in a top-down recursive fashion, leading to a hierarchy of fine-grained parts. The same model trained for Chair class can be used to segment different chair models into different number of parts, depending on the structure complexity of the input shapes.

There are two issues with these existing models. First, the models are trained targeting a *fixed* set of labels, which greatly limits its flexibility and adaptivity. For example, a model trained to segment a chair into three semantic parts cannot be used to correctly segment a chair with four parts, even they both belong to the same shape family. Training different models for different targeted label sets is neither general nor efficient. Second, labeling all primitives simultaneously cannot exploit the hierarchical nature of shape decomposition. Hierarchical shape segmentation reduces the difficulty of shape segmentation through dividing the multi-class labeling problem into a cascade of binary labeling problems [1, 39]. On the other hand, hierarchical segmentation can utilize structural constraints across different levels: The segmentations in higher levels provide strong cues constraining those in the lower levels. This enables accurate segmentation into very fine-grained levels (Figure 1).

In this work, we opt for the top-down decomposition and propose the first deep learning model for hierarchical segmentation of 3D shapes into fine-grained parts, based on recursive neural networks (RvNN). Starting from a full shape represented as a point cloud, our model performs recursive binary decomposition, where the decomposition network at

*Joint first authors
†Corresponding author: kevin.kai.xu@gmail.com

all nodes in the hierarchy share weights. At each node, a node classifier is trained to determine the type of its decomposition (adjacency or symmetry node) and whether the decomposition should stop (leaf node). The features extracted in higher level nodes are recursively propagated to lower level ones through the tree structure, which we refer to as *recursive context features*. Therefore, the meaningful decompositions in higher levels constrain the segmentations in lower levels. Meanwhile, to increase the segmentation accuracy at each node, we enhance the recursive context feature with the *part shape feature* extracted for the corresponding point cloud.

The network is trained with point sampled 3D models from ShapeNet [3] which are typically composed of semantically labeled parts. For each shape, a hierarchy is constructed with an existing rule-based method [36]. Such principled training hierarchies help the training converges faster. The loss is computed at each node, including node classification loss and binary point labeling loss.

Our method produces accurate segmentation, even for highly fine-grained decomposition into arbitrary number of parts, due to the flexibility of dynamic, RvNN-based architecture. Moreover, it recovers the part relations (adjacency or symmetry) which further improves the labeling accuracy, e.g., symmetric parts can be identified and thus correctly labeled (Figure 1). Our method achieves state-of-the-art performance, both on the public benchmark and a new benchmark of fine-grained segmentation proposed in this work. We also demonstrate its utility in image-to-shape reconstruction with fine-grained structure recovery.

Our contributions include:

- A deep learning model for top-down hierarchical, fine-grained segmentation of 3D shapes based on dynamic RvNN-based architecture.
- A part feature learning scheme which integrates both contextual information and per-part shape geometry.
- A benchmark for fine-grained, part instance segmentation of 3D shapes.
- An application of our fine-grained structure recovery for high-quality image-to-shape reconstruction.

2. Related work

Learning 3D shape segmentation. Semantic segmentation of 3D shapes has gained significant research progress in recent year, benefiting from the advances in machine learning techniques [10, 24, 35, 37]. A comprehensive survey on learning-based 3D shape segmentation can be found in [38]. The basic idea of these approaches is to learn a shape primitive (e.g., a triangle, a point or a voxel) classifier, based on the geometric features of the shape primitives.

Recently, several deep learning models have been developed for supervised segmentation of 3D shapes in various representations including volumetric grid [23, 32], point cloud [20, 11, 7], multi-view rendering [9] or surface mesh [41, 31]. The main idea is to replace the hand-crafted geometric features employed in the traditional methods with data-driven learned ones. All these models, however, are trained targeting a *fixed* set of semantic labels. Given a different set of targeting labels, the model has to be re-trained, using a training dataset annotated with the new labels.

Hierarchical segmentation of 3D shapes. 3D shapes are usually modeled with parts in a hierarchical construction manner. This is evidenced in part by the wide availability of scene graphs in human-created 3D models of objects or scenes, and by the well-known hierarchical modeling paradigm of Constructive Solid Geometry (CSG) [21]. This naturally leads to the idea of hierarchical decomposition of 3D shapes. Hierarchical shape segmentation can be achieved either with a bottom-up grouping approach [1, 34], or in a top-down fashion based on a global topological analysis [22, 8, 42]. Given a pre-segmented 3D shape, Wang et al. [36] infer a hierarchical organization of the parts based on proximity and symmetry relations. Later, this heuristic method is improved with an unsupervised learning approach [30]. Yi et al. [39] propose a supervised learning approach to hierarchical segmentation of 3D shapes. Their model is, again, trained for a fixed set of semantic tags. The tag sets are determined in a pre-processing of part label analysis and organized with a pre-defined canonical hierarchy. Our method, on the other hand, does not require a prescribed canonical hierarchy and learns the decomposition hierarchies in a data-driven manner, thank to the recursively trained node classification. Our method is, to our knowledge, the first end-to-end trainable model for hierarchical shape segmentation.

Recursive neural networks. Recursive neural nets (RvNN) are developed by Socher et al. [25], for text and image understanding [27], and for 3D shape classification [26]. Recently, Li et al. [12] introduce a generative recursive auto-encoder for generating 3D shape structures. Given a collection of pre-segmented 3D shapes, a variational auto-encoder (VAE) model is learned with RvNN-based encoding and decoding of part structures. Following that, RvNN-based VAE is also trained for 3D scene generation [13], structure-aware single-view 3D reconstruction [18] and substructure prior learning for part group composition [43]. We are not aware of a previous work on using RvNN for hierarchical 3D shape segmentation.

3. Method

We first introduce the overall architecture of PartNet, which is a recursive part decomposition network. Several key designs in the network will then follow.

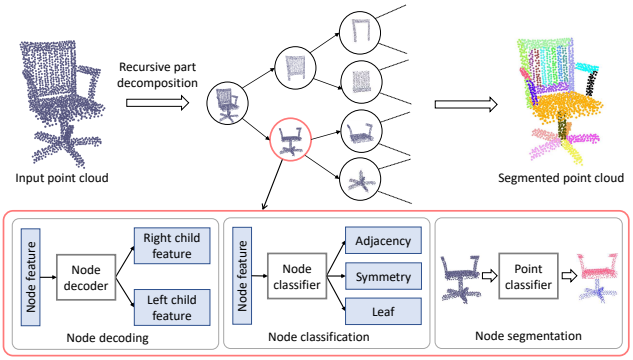

Figure 2. The architecture of PartNet. At each node, there are three modules devised for context propagation, hierarchy construction and point cloud segmentation, respectively. Being a recursive network, these modules are shared by all node in the hierarchy.

3.1. Recursive part decomposition network

Figure 2 shows the architecture of PartNet. Taking a point cloud of 3D shape as input, PartNet performs a top-down decomposition and and outputs a segmented point cloud at the level of part instances. At each node, three modules are devised:

- *Node decoding module* used to pass the global contextual information from the current node to its children. Such information constraints the segmentation of a node with higher level context.
- *Node classification module* devised to construct the topological structure of the decomposition hierarchy. This is achieved by learning to predict the node type which determines how to decompose a node and when to stop the decomposition.
- *Node segmentation module* used for performing actual segmentation of the point cloud of the current node. This is achieved by learning a point classification network shared across all nodes.

Below we elaborate the discussion on these modules.

Node decoding module. To bootstrap, we first extract a 128D PointNet [19] feature for the full shape point cloud, which is then duplicated and concatenated, forming a 256D the *root node feature*. This 256D feature is then decoded into two 128D features, one for each of its two child nodes, which we refer to as *recursive context feature*. At each non-root node, we also extract a 128D PointNet feature for the partial point cloud corresponding to that node, which is called *part shape feature*. This 128D part shape feature is then concatenated with the 128D recursive context feature passed down from the parent node, forming the *current node feature*. Please see Figure 3 for a visual explanation of these features. The decoding module is implemented with a

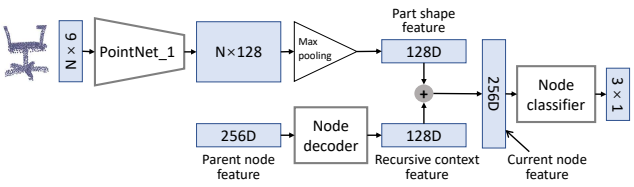

Figure 3. Network design of the node decoding module and the node classification module. The *recursive contextual feature* and *part shape feature* are concatenated and fed into the node classifier.

two-layer fully connected network with $tanh$ nonlinearity. This PointNet used in this module is referred to as PointNet_1, to distinguish with the one to be used in the node segmentation module (see below).

Node classification module. At a given node, taking its current node feature as input, the node classification module predicts its *node type* as one of the following three ones: *adjacency*, *symmetry* or *leaf*. Through determining the how and whether a node is split, this module constructs the topological structure of the hierarchy. This node classification module is implemented with two fully-connected layers with $tanh$ nonlinearity. It can be trained with the ground-truth hierarchical segmentation of a point cloud.

Note that when a node is classified as a symmetry node, we interpret its left child as a symmetry generator (representing a part) and its right child as symmetry parameters, similar to [12]. Applying the symmetry parameters on the symmetry generator part obtains the complete point cloud of that symmetry node. For example, the node corresponding to the spokes in the leg part of a swivel chair (Figure 1, left) is a rotational symmetry node. Its left child represents the point cloud of one of the spokes and the right child encodes the symmetry axis and symmetry fold.

Node segmentation module. This module performs point labeling based on both the current node feature and per-point PointNet features. Specifically, we use another Point-

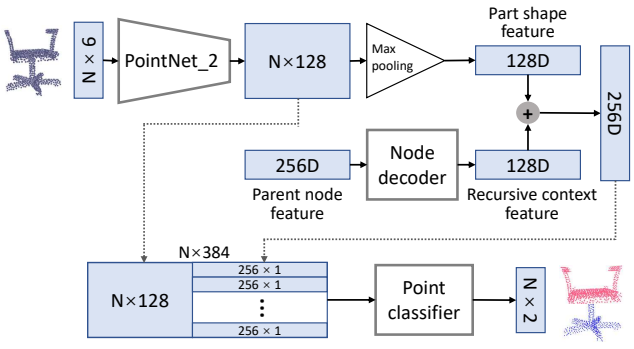

Figure 4. Network design of the node segmentation module. The concatenation of *recursive contextual feature* and *part shape feature* is enhanced with point-wise PointNet features for the purpose of point label prediction (point cloud segmentation).

Net (denoted as PointNet_2) to extract per-point feature, leading to a $N \times 128$ feature matrix, with N being the number of points of the current node. Then for each row (point feature), we enhance it by concatenating the 256D current node feature. This results in a $N \times 384$ feature matrix, which is fed into a point classification network to produce point-wise binary labels. This point classifier is implemented with the last five layers of a PointNet. Note that these layers do not share weight with PointNet_1 or PointNet_2.

For a symmetry node, only its left child, i.e., the symmetry generator, needs to be segmented. After the segmentation, the point labels are transferred to all other symmetric counterparts, based on the predicted symmetry parameters.

3.2. Loss function

For each training point cloud, the overall loss function for PartNet, L_{partnet}, consists of the average node classification loss and average node segmentation loss over all relevant nodes:

$$L_{\text{partnet}} = \frac{1}{|\mathcal{H}|} \sum_{n \in \mathcal{H}} L_{\text{class}}(n) + \frac{1}{|\mathcal{T}|} \sum_{n \in \mathcal{T}} L_{\text{seg}}(n) \quad (1)$$

where $L_{\text{class}}(n)$ and $L_{\text{seg}}(n)$ are the classification loss and segmentation loss of node n, respectively. Both losses are defined as the cross-entropy loss. $\mathcal{H}$ is the set of all nodes in the hierarchy, and $\mathcal{T}$ the set of all non-leaf nodes.

3.3. Training details

The PointNet_1 for node classification (Figure 3) uses six point convolution layers with 64, 128, 128, 256, 256 and 128 filters, respectively. The PointNet_2 for node segmentation (Figure 4) uses four point convolution layers with 64, 64, 128 and 128 filters, respectively. The point cloud segmentation network in Figure 4 consists of four point convolution layers, with 512, 256, 128 and 128 filters, respectively, plus the output layer with a 2 filters for binary label prediction. 20% random feature dropout are used between every two of the last three layers in all these networks. Batch normalization are used between every two layers. We use the Adam optimizer for training, with a batch size of 10 and the initial learning rate of 0.001.

The size of input point cloud is 2048. The training point clouds are obtained by point sampling 3D models. Gaussian noise is added for data enhancement. All PointNets use point normals to improve fine-grained part segmentation performance. Therefore, the dimension of input tensors to PartNet is 2048×6.

4. Results and evaluations

4.1. Benchmark

The Fine-grained Segmentation Benchmark (FineSeg). With the advances in deep learning based 3D shape segmentation, a benchmark for instance segmentation of fine-grained parts is called for. A nice benchmark for evaluating fine-grained shape segmentation is recently proposed in a concurrent work in [17]. In this work, we propose FineSeg. The dataset contains about 3000 3D shapes over six shape categories: chair (1000), table (500), airplanes (600), sofa (600), helicopter (100) and bike (140). The models are collected from a subset of ShapeNet [3] used in the work of Sung et al. [29]. These models are consistently aligned and uniformly scaled. For those model whose segmentation is not fine-grained enough (e.g., no instance part segmentation), we manually segment the models. We then build a part hierarchy for each shape, using the method proposed in [36]. We point sample each 3D model, thus generating a ground-truth fine-grained segmentation of the corresponding 3D point cloud. The hierarchies can be used for training our recursive part decomposition network. This benchmark can be used to quantitatively evaluate fine-grained segmentation of 3D point clouds, based on *Average Precision (AP)* for part detection (with the IoU against ground-truth greater than a threshold). The benchmark is publicly available at: www.kevinkaixu.net/projects/partnet.html.

4.2. Segmentation results and evaluation

Our PartNet model is trained with 80% models of FineSeg, leaving the rest 20% for testing. The discussion of complexity and timing (for both training and testing) can be found in the supplemental material.

Visual results on FineSeg. We first show in Figure 5 some visual examples of fine-grained point cloud segmentation obtained by PartNet. For side-by-side comparison, we also show the ground-truth segmentation for each example. Our method produces precise fine-grained segmentation on the noisy point clouds with complicated part structures. Furthermore, once trained, the same model can be used to segment the test (unseen) point clouds into varying number of parts, demonstrating its flexibility and generality. Figure 6 demonstrates how the same model of PartNet can segment different shapes in a category into for an arbitrary number of targeting parts, depending on structure complexity. More results can be found in the supplemental material. In the supplemental material, we also show a visual comparison of hierarchical segmentation with two traditional (non-learned) baseline methods.

Quantitative evaluation with ablation study. In quantitative evaluation on FineSeg, we compare to two baselines which are ablated versions of our method. Specially, we are interested in the effect of the two important node features used in PartNet: recursive context feature (RCF) and part shape feature (PSF).

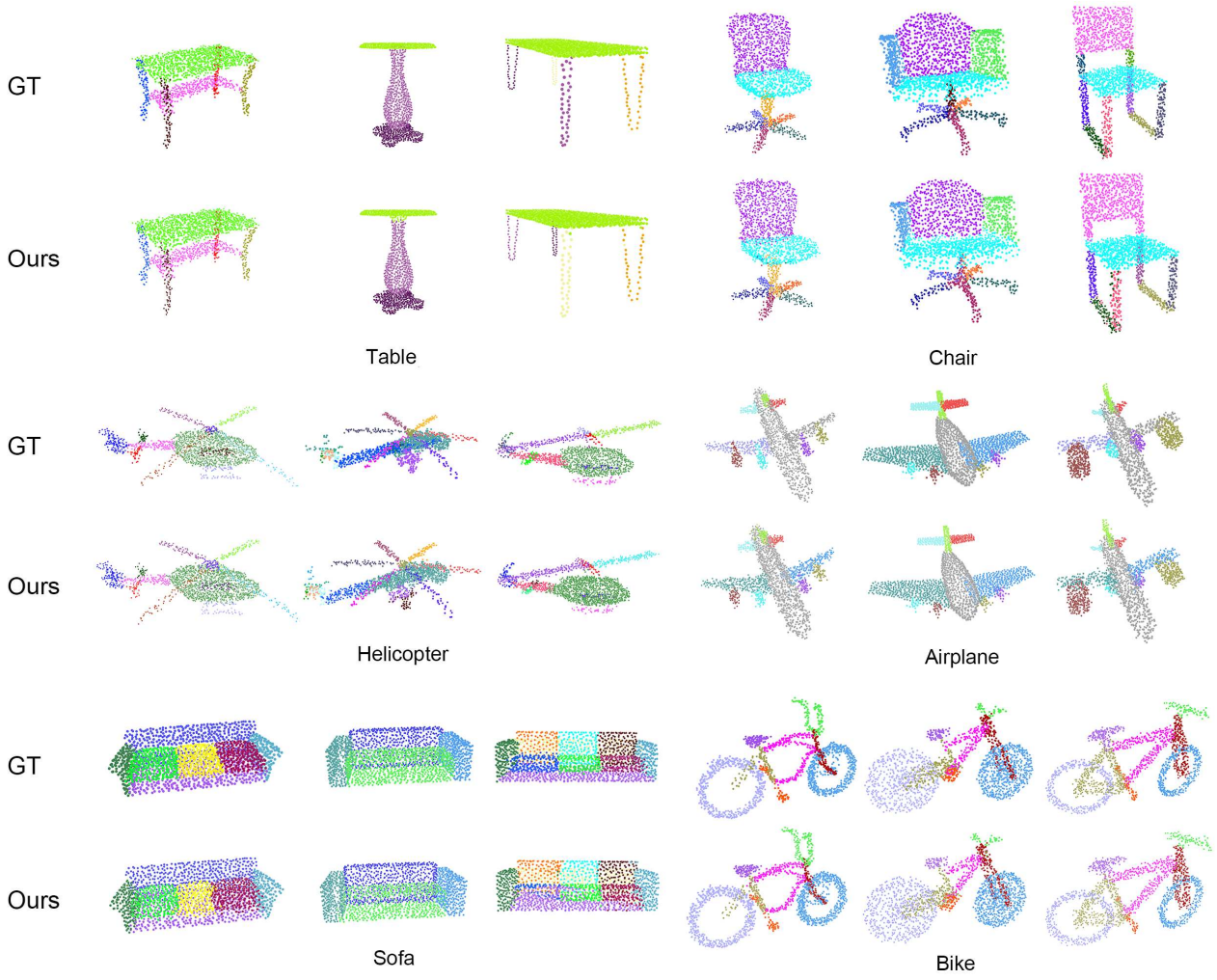

Figure 5. Fine-grained point cloud segmentation by PartNet. For comparison, we show for each shape the fine-grained segmentation result (bottom) and the corresponding ground-truth (top).

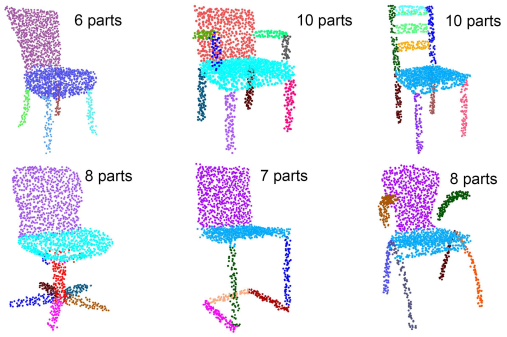

Figure 6. The same PartNet model trained on the Chair set, can be used to segment different chair models into different number of parts, depending on the structure complexity of the input shapes.

In the first baseline (w/o RCF), recursive context feature is removed from both node classification (see Figure 3) and node segmentation (see Figure 4). To compensate the missing of recursive context feature, the 128D part shape feature is duplicated into a 256D feature. The ablated network is re-trained using the training set of FineSeg. In the second baseline (w/o PSF), PSF is removed only from the node classification module.

		mean	aero	bike	chair	heli.	sofa	table
IoU > 0.25	Full	84.8	95.2	97.0	91.1	83.0	65.4	77.2
	w/o RCF	79.2	92.8	92.0	87.1	71.1	61.6	70.8
	w/o PSF	77.6	90.8	95.1	83.6	77.8	54.1	64.0
IoU > 0.5	Full	72.8	88.0	89.4	80.5	69.4	46.7	62.6
	w/o RCF	66.0	85.3	83.4	71.8	56.7	42.5	56.4
	w/o PSF	64.9	85.2	88.4	65.6	57.5	36.9	55.6

Table 1. Comparing our full model with two baselines (w/o RCF and w/o PSF) on FineSeg. AP(%) is measured with IoU threshold being 0.25 and 0.5, respectively.

Table 1 reports AP on all six categories of the testing set, with the IoU thresholds being 0.25 and 0.5, respectively.

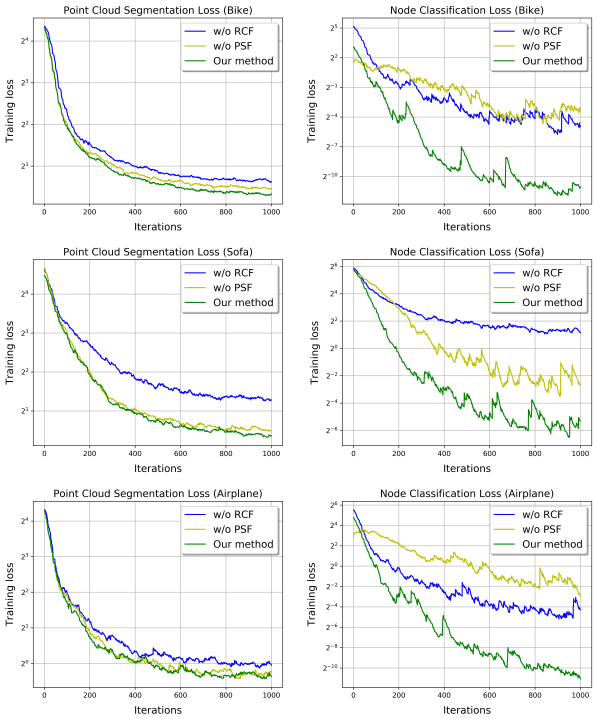

Figure 7. Training loss over iterations in the ablation study of the two key node features (RCF and PSF), on three shape categories (Bike, Sofa and Airplane). For each category (row), we plot both node segmentation loss (left) and node classification loss (right).

The consistent superiority of our full model demonstrates the importance of the two features. Figure 7 plots the training loss over iterations for the three methods, on three shape categories (Bike, Sofa and Airplane). The results show that both features are critical for fast training of the node classification module and the node segmentation module. This evidences the importance of both global context information and local shape geometry on learning hierarchical segmentation. More results are in the supplemental material.

ShapeNet challenge for fine-grained segmentation. In addition, we conduct a moderate-scale stress test using ShapeNet [3] challenge for fine-grained segmentation. We randomly select a collection of shapes from ShapNet and use PartNet to segment them. Since we don't have ground-truth fine-grained segmentation for ShapeNet models, we resort to a subjective study to evaluate our segmentation results. We ask the participants to rate the quality of fine-grained segmentation in the range from 1 to 5. The user study shows that our method attains > 4.0 average ratings for all the categories tested, much higher than the results of the "w/o RCF" baseline. The details and results of this study are provided in the supplemental material. In Figure 8, we show a few visual examples, from which one can see that our method produces fine-grained segmentation these unseen shapes with complicated structures. Moreover, our method obtains the adjacency and symmetry relations of the decomposed parts, which can be used for many downstream structure-aware applications [16].

Figure 8. A few results from the ShapeNet fine-grained segmentation challenge. Besides segmentation, PartNet can also recover the relations (adjacency and symmetry) between the segmented parts. We visualize the recovered symmetry relations with colored arrows (Reflective: Red; Translational: Blue; Rotational: Green).

4.3. Comparison of semantic segmentation

Although PartNet is designed for fine-grained segmentation, the recursive decomposition should work even better for semantic segmentation since the latter is usually a much coarser-level segmentation. We evaluate PartNet for semantic segmentation of 3D point clouds on the ShapeNet part dataset [40], through comparing with seven state-of-the-art methods on this task. Similar to PointNet [19], we re-sample the point cloud for each shape into 2048 points. We use the same training/testing split setting as those state-of-the-arts, and compute part-wise average IoU as metric.

Note that PartNet does not produce semantic labels for points, so it cannot perform labeled segmentation. To enable the comparison, we add an extra module to PartNet to predict a semantic label for each part it decomposes. The part label prediction module takes the node feature of the leaf nodes as input and outputs a semantic label for all points included in that leaf node. This module is implemented with three fully-connected layers and is trained with cross-entropy loss.

Method	mean	aero	bag	cap	car	chair	eph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate.	table
PointNet [19]	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++ [20]	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 [32]	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 [6]	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
PCNN [2]	85.1	82.4	80.1	85.5	79.5	90.8	73.2	91.3	86.0	85.0	95.7	73.2	94.8	83.3	51.0	75.0	81.8
SPLATNet [28]	85.4	83.2	84.3	89.1	80.3	90.7	75.5	92.1	87.1	83.9	96.3	75.6	95.8	83.8	64.0	75.5	81.8
PointCNN [14]	86.1	84.1	86.4	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
Ours	**87.4**	**87.8**	86.7	**89.7**	80.5	**91.9**	75.7	91.8	85.9	83.6	**97.0**	74.6	**97.3**	83.6	**64.6**	78.4	**85.8**

Table 2. Comparison of semantic segmentation on the ShapeNet part dataset [40]. Metric is part-wise IoU (%).

The results are reported in Table 2. PartNet, augmented with a part label prediction module, achieves better performance in most of the categories, and the highest mean accuracy over all categories. Furthermore, our method works especially well for those categories with complex structures such as chair, table, and aeroplane, etc. We believe that the divide-and-conquer nature of recursive decomposition does help reduce the difficulty of segmentation learning. Another key benefit of recursive decomposition is that the segmentation of higher levels provides contextual cues constraining that of the lower levels. Similar results can also be observed in testing our trained model on the Princeton Segmentation Benchmark [4] (see supplemental material).

For semantic segmentation, PartNet can be trained with a consistent hierarchy for all shapes in a category. The training is can be done with *any* hierarchy that is consistent across all training shapes. Therefore, we do *not* need an extra process (such as the one [36] used in fine-grained segmentation) for hierarchy construction. Taking *any random hierarchy* of one training shape as a "template", we unify the hierarchies of all the other shapes based on the semantic part labels. Therefore, PartNet does *not* require an extra supervision of part hierarchy for training for semantic segmentation. Consequently, the comparison reported in Table 2 of the main paper is a fair one.

4.4. Comparison of instance segmentation

SGPN [33] is the first deep learning model that learns instance segmentation on 3D point clouds. It can segment object instances and predict a class label for each instance, which is very similar to our method (augmented the label prediction module), except that SGPN cannot obtain part relations as our method does. We make a comparison to SGPN on our FineSeg dataset, using again AP with IoU thresholds of 0.25 and 0.5.

Figure 9 shows a few visual comparisons, where incorrectly segmented regions are marked out. Table 3 shows the quantitative comparison on our datasets. We attribute the consistent improvement over SGPN to two factors. First, the instance group learning of SGPN is based on point clustering. Such a one-shot point grouping over the entire shape is hard to learn. Our method, on the other hand, performs top-down recursive decomposition which breaks the full shape segmentation into a cascade of partial shape segmentations. Second, the point features used by SGPN are solely point convolutional features [19] while our features accounts for both local part shape and global context.

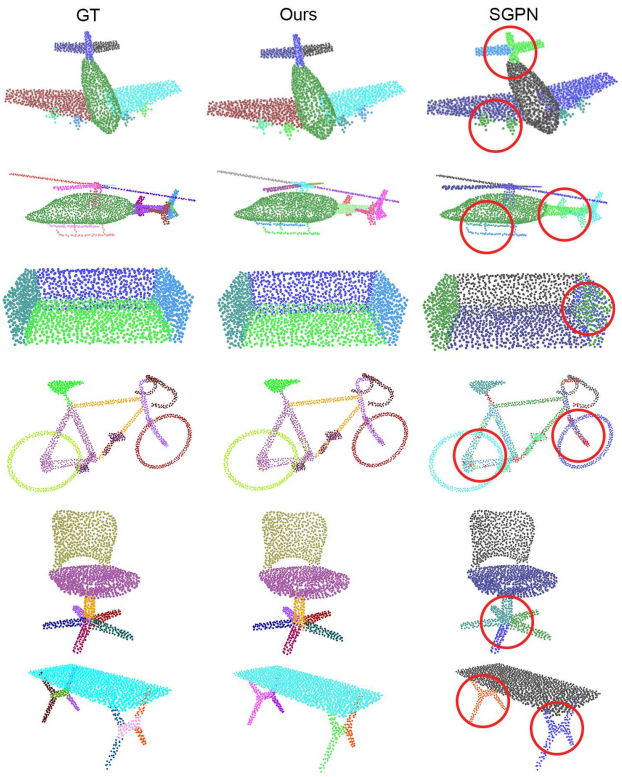

Figure 9. Visual comparison of fine-grained part instance segmentation with SGPN [33]. Left: Ground-truths. Middle: Segmentation results by PartNet. Right: Results by SGPN. Incorrect segmentations (w.r.t. ground-truth) are marked with red circles.

		mean	aero	bike	chair	heli.	sofa	table
IoU	SGPN [12]	62.2	67.8	75.8	66.2	59.4	50.4	53.6
> 0.25	Ours	84.8	95.2	97.0	91.1	83.0	65.4	77.2
IoU	SGPN [12]	47.0	56.7	63.7	54.6	38.9	29.5	38.4
> 0.5	Ours	72.8	88.0	89.4	80.5	69.4	46.7	62.6

Table 3. Comparison with SGPN [33] on fine-grained instance segmentation over the FineSeg dataset. The metric is AP (%) with IoU threshold being 0.25 and 0.5, respectively.

5. Applications

We demonstrate an application of the fine-grained segmentation of PartNet in refining 3D point clouds reconstructed from single view images. The basic idea is a *segment-and-refine* process. Given a 3D point cloud reconstructed in a holistic fashion (using, e.g., the method of Fan et al. [5]), we first perform a recursive decomposition of the point cloud, resulting in a hierarchical organization of part point clouds. We then train a network to refine the part point cloud at each leaf node, yielding a high-quality point cloud for that part. These refined part point clouds together constitute a refined point cloud of the full shape.

The part refiner network used at each leaf node is composed of two channels of PointNet, to encode the point clouds of the part and the full shape, respectively. The resulting two features are concatenated and fed into a four layer fully-connected networks to generate a refined part point cloud. To train this refiner network, we use reconstruction loss computed as the Chamfer distance and the earth mover's distance between point clouds [5]. To gain more training signals, we opt to train the refiner with a hierarchical reconstruction loss, through a bottom-up composition of the refined part point clouds, following the hierarchy obtained by PartNet segmentation. This way, we can compute a reconstruction loss at each node of the hierarchy, with the corresponding point cloud composed from the part point clouds within its subtree. Please refer to the supplemental material for more details on the network architecture.

Figure 10 shows a few examples of point cloud refinement, guided by the fine-grained, hierarchical segmentation of PartNet. Although the refinement may sometimes lose part fidelity w.r.t. the input images, it does produce highly detailed point clouds and with plausible part structure, thanks to the fine-grained part decomposition of PartNet. See more examples in the supplemental material.

Figure 10. A few examples on refining point clouds reconstructed from single view images, guided by the fine-grained segmentation of PartNet. In each row, we show from left to right the input image, result of holistic reconstruction, fine-grained segmentation of the reconstruction, and the final refinement result by our method.

6. Conclusion

We have presented a top-down recursive decomposition network for fine-grained segmentation of 3D point clouds. With the hierarchical decomposition scheme, our model obtains fine-grained and accurate segmentation even for highly complex shapes. Different from most existing deep-learning based segmentation models, our method segments a shape into an arbitrary number of parts, depending on its structural complexity, instead of producing a labeled segmentation with a fixed label set. Even for semantic segmentation, our model also achieves superior performance, benefiting from our divide-and-conquer segmentation learning.

Limitations and future work. Our current method has a few limitations. *First*, although PartNet segments a shape in a hierarchical fashion, the resulting segment hierarchy is not necessarily as meaningful as the those learned purposively for shapes [30, 39] or scenes [15]. *Second*, although our model can be used to segment different shapes into different number of parts, instead of targeting a fixed part label set. It still needs to be trained for each shape category separately. Learning a more general model for recursive decomposition of shapes from multiple classes would be a very interesting future direction to look into. *Third*, PartNet is trained with reasonable ground-truth hierarchies built with the method in [36]. Training with totally random hierarchy would lead to performance degrading. We show this effect in the supplemental material. Therefore, another future direction is to learn hierarchical segmentation in a unsupervised manner, without the need of building ground-truth hierarchies.

Acknowledgement

We thank the anonymous reviewers for their valuable comments. We are also grateful to Hao (Richard) Zhang for the fruitful discussions, and to Yuan Gan and Pengyu Wang for the tremendous help on data preparation. This work was supported in part by NSFC (61572507, 61532003, 61622212) and Natural Science Foundation of Hunan Province for Distinguished Young Scientists (2017JJ1002).

References

[1] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006. 1, 2

[2] M. Atzmon, H. Maron, and Y. Lipman. Point convolutional neural networks by extension operators. *ACM SIGGRAPH*, 2018. 7

[3] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 4, 6

[4] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), Aug. 2009. 7

[5] H. Fan, H. Su, and L. Guibas. A point set generation network for 3d object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2463–2471, 2017. 8

[6] B. Graham, M. Engelcke, and L. van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proc. CVPR*, pages 9224–9232, 2018. 7

[7] 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. 2

[8] Q.-X. Huang, M. Wicke, B. Adams, and L. Guibas. Shape decomposition using modal analysis. *Computer Graphics Forum*, 28(2):407–416, 2009. 2

[9] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri. 3d shape segmentation with projective convolutional networks. In *Proc. CVPR*, volume 1, page 8, 2017. 2

[10] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3d mesh segmentation and labeling. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(4):102:1–102:12, 2010. 2

[11] R. Klokov and V. Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 863–872. IEEE, 2017. 2

[12] J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas. GRASS: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):52, 2017. 2, 3, 7

[13] M. Li, A. G. Patil, K. Xu, S. Chaudhuri, O. Khan, A. Shamir, C. Tu, B. Chen, D. Cohen-Or, and H. Zhang. Grains: Generative recursive autoencoders for indoor scenes. *arXiv preprint arXiv:1807.09193*, 2018. 2

[14] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: Convolution on x-transformed points. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 820–830. Curran Associates, Inc., 2018. 7

[15] T. Liu, S. Chaudhuri, V. G. Kim, Q. Huang, N. J. Mitra, and T. Funkhouser. Creating consistent scene graphs using a probabilistic grammar. *ACM Transactions on Graphics (TOG)*, 33(6):211, 2014. 8

[16] N. Mitra, M. Wand, H. R. Zhang, D. Cohen-Or, V. Kim, and Q.-X. Huang. Structure-aware shape processing. In *SIGGRAPH Asia 2013 Courses*, page 1. ACM, 2013. 6

[17] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. *arXiv preprint arXiv:1812.02713*, 2018. 4

[18] C. Niu, J. Li, and K. Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *Proc. CVPR*, pages 4521–4529, 2018. 2

[19] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 3, 6, 7

[20] 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. 1, 2, 7

[21] A. A. Requicha and H. B. Voelcker. Constructive solid geometry. 1977. 2

[22] M. Reuter. Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions. *International Journal of Computer Vision*, 89(2-3):287–308, 2010. 2

[23] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 3, 2017. 1, 2

[24] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. on Graph. (SIGGRAPH Asia)*, 30(6), 2011. 2

[25] R. Socher. *Recursive deep learning for natural language processing and computer vision*. PhD thesis, Citeseer, 2014. 2

[26] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng. Convolutional-recursive deep learning for 3d object classification. In *Advances in neural information processing systems*, pages 656–664, 2012. 2

[27] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 129–136, 2011. 2

[28] 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. 7

[29] M. Sung, H. Su, V. G. Kim, S. Chaudhuri, and L. Guibas. ComplementMe: Weakly-supervised component suggestions for 3D modeling. *ACM Trans. on Graph. (SIGGRAPH Asia)*, 2017. 4

[30] O. Van Kaick, K. Xu, H. Zhang, Y. Wang, S. Sun, A. Shamir, and D. Cohen-Or. Co-hierarchical analysis of shape structures. *ACM Trans. on Graph. (SIGGRAPH)*, 32(4):69, 2013. 2, 8

[31] P. Wang, Y. Gan, P. Shui, F. Yu, Y. Zhang, S. Chen, and Z. Sun. 3d shape segmentation via shape fully convolutional networks. *Computers & Graphics*, 70:128–139, 2018. 2

[32] 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 Transactions on Graphics (SIGGRAPH)*, 36(4), 2017. 2, 7

[33] W. Wang, R. Yu, Q. Huang, and U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. 2017. 7

[34] X. Wang, B. Zhou, H. Fang, X. Chen, Q. Zhao, and K. Xu. Learning to group and label fine-grained shape components. In *SIGGRAPH Asia 2018 Technical Papers*, page 210. ACM, 2018. 2

[35] Y. Wang, M. Gong, T. Wang, D. Cohen-Or, H. Zhang, and B. Chen. Projective analysis for 3d shape segmentation. *ACM Transactions on Graphics (TOG)*, 32(6):192, 2013. 2

[36] Y. Wang, K. Xu, J. Li, H. Zhang, A. Shamir, L. Liu, Z. Cheng, and Y. Xiong. Symmetry hierarchy of man-made objects. *Computer Graphics Forum*, 30(2):287–296, 2011. 2, 4, 7, 8

[37] Z. Xie, K. Xu, and L. L. abd Yueshan Xiong. 3d shape segmentation and labeling via extreme learning machine. *Computer Graphics Forum (Proc. SGP)*, 33(5):85–95, 2014. 2

[38] K. Xu, V. G. Kim, Q. Huang, N. Mitra, and E. Kalogerakis. Data-driven shape analysis and processing. In *SIGGRAPH ASIA 2016 Courses*, page 4. ACM, 2016. 1, 2

[39] L. Yi, L. Guibas, A. Hertzmann, V. G. Kim, H. Su, and E. Yumer. Learning hierarchical shape segmentation and labeling from online repositories. *ACM Trans. on Graph. (SIGGRAPH)*, 2017. 1, 2, 8

[40] 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. 6, 7

[41] L. Yi, H. Su, X. Guo, and L. J. Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *CVPR*, pages 6584–6592, 2017. 2

[42] Y. Zhou, K. Yin, H. Huang, H. Zhang, M. Gong, and D. Cohen-Or. Generalized cylinder decomposition. *ACM Trans. on Graph. (SIGGRAPH Asia)*, 34(6):171–1, 2015. 2

[43] C. Zhu, K. Xu, S. Chaudhuri, R. Yi, and H. Zhang. Scores: Shape composition with recursive substructure priors. *ACM Transactions on Graphics (SIGGRAPH Asia 2018)*, 37(6):to appear, 2018. 2