PartNet: A Large-scale Benchmark for Fine-grained and Hierarchical Part-level 3D Object Understanding

Kaichun Mo[1] Shilin Zhu[2] Angel X. Chang[3] Li Yi[1] Subarna Tripathi[4] Leonidas J. Guibas[1,5] Hao Su[2]
[1]Stanford University [2]UC San Diego [3]Simon Fraser University [4]Intel AI Lab [5]Facebook AI Research
https://cs.stanford.edu/~kaichun/partnet/

Abstract

We present PartNet: a consistent, large-scale dataset of 3D objects annotated with fine-grained, instance-level, and hierarchical 3D part information. Our dataset consists of 573,585 part instances over 26,671 3D models covering 24 object categories. This dataset enables and serves as a catalyst for many tasks such as shape analysis, dynamic 3D scene modeling and simulation, affordance analysis, and others. Using our dataset, we establish three benchmarking tasks for evaluating 3D part recognition: fine-grained semantic segmentation, hierarchical semantic segmentation, and instance segmentation. We benchmark four state-of-the-art 3D deep learning algorithms for fine-grained semantic segmentation and three baseline methods for hierarchical semantic segmentation. We also propose a baseline method for part instance segmentation and demonstrate its superior performance over existing methods.

1. Introduction

Being able to parse objects into parts is critical for humans to understand and interact with the world. People recognize, categorize, and organize objects based on the knowledge of their parts [10]. Many actions that people take in the real world require detection of parts and reasoning over parts. For instance, we open doors using doorknobs and pull out drawers by grasping their handles. Teaching machines to analyze parts is thus essential for many vision, graphics, and robotics applications, such as predicting object functionality [13, 14], human-object interactions [18], shape editing [28, 16], and shape generation [25, 41].

To enable part-level object understanding by learning approaches, 3D data with part annotations are in high demand. Many cutting-edge learning algorithms, especially for 3D understanding [45, 44, 30, 8], intuitive physics [27], and reinforcement learning [48, 29], require such data to train the networks and benchmark the performances. Researchers are also increasingly interested in synthesizing dynamic data through physical simulation engines [20, 43, 29]. Creation of large-scale animatable scenes will require a large amount of 3D data with affordances and mobility information. Object parts serve as a critical stepping stone to access this information. Thus it is necessary to have a large 3D object dataset with part annotation.

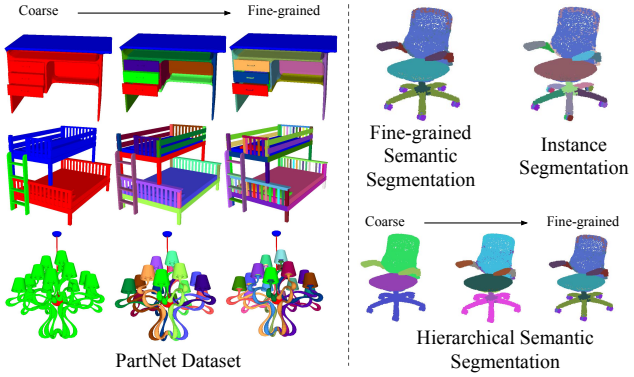


Figure 1. PartNet dataset and three benchmarking tasks.

With the availability of the existing 3D shape datasets with part annotations [5, 3, 45], we witness increasing research interests and advances in 3D part-level object understanding. Recently, a variety of learning methods have been proposed to push the state-of-the-art for 3D shape segmentation [30, 31, 46, 19, 35, 24, 9, 39, 40, 42, 33, 7, 26, 23]. However, existing datasets only provide part annotations on relatively small numbers of object instances [5], or on coarse yet non-hierarchical part annotations [45], restricting the applications that involves understanding fine-grained and hierarchical shape segmentation.

In this paper, we introduce PartNet: a *consistent, large-scale* dataset on top of ShapeNet [3] with *fine-grained, hierarchical, instance-level* 3D part information. Collecting such fine-grained and hierarchical segmentation is challenging. The boundary between fine-grained part concepts are more obscure than defining coarse parts. Thus, we define a common set of part concepts by carefully examining the 3D objects to annotate, balancing over several criteria: well-

Dataset	#Shape	#Part	#Category	Granularity	Semantics	Hierarchical	Instance-level	Consistent
Chen et al. [5]	380	4,300	19	**Fine-grained**	No	No	**Yes**	No
MCL [37]	1,016	7,537	10	**Fine-grained**	**Yes**	No	No	**Yes**
Chang et al. [4]	2,278	27,477	**90**	**Fine-grained**	**Yes**	No	**Yes**	No
Yi et al. [45]	**31,963**	80,323	16	Coarse	**Yes**	No	No	**Yes**
PartNet (ours)	26,671	**573,585**	24	**Fine-grained**	**Yes**	**Yes**	**Yes**	**Yes**

Table 1. Comparison to the other shape part datasets.

defined, consistent, compact, hierarchical, atomic and complete. Shape segmentation involves multiple levels of granularity. Coarse parts describe more global semantics and fine-grained parts convey richer geometric and semantic details. We organize expert-defined part concepts in hierarchical segmentation templates to guide annotation.

PartNet provides a large-scale benchmark for many part-level object understanding tasks. In this paper, we focus on three shape segmentation tasks: fine-grained semantic segmentation, hierarchical semantic segmentation, and instance segmentation. We benchmark four state-of-the-art algorithms on fine-grained semantic segmentation and propose three baseline methods for hierarchical semantic segmentation. We propose the task of part instance segmentation using PartNet. By taking advantages of rich shape structures, we propose a method that outperforms the existing baseline algorithm by a clear margin.

PartNet contains highly structured, fine-grained and heterogeneous parts. Our experiments reveal that existing algorithms developed for coarse and homogeneous part understanding do not work well on PartNet. First, small and fine-grained parts, *e.g.* door handles and keyboard buttons, are abundant and present new challenges for part recognition. Second, many geometrically similar but semantically different parts require more global shape context to distinguish. Third, understanding the heterogeneous variation of shapes and parts necessitate hierarchical understanding. We expect that PartNet could serve as a better platform for part-level object understanding in the next few years.

In summary, we make the following contributions:

- We introduce PartNet, consisting of 573,585 fine-grained part annotations for 26,671 shapes across 24 object categories. To the best of our knowledge, it is the first *large-scale* dataset with *fine-grained, hierarchical, instance-level* part annotations;
- We propose three part-level object understanding tasks to demonstrate the usefulness of this data: fine-grained semantic segmentation, hierarchical semantic segmentation, and instance segmentation;
- We benchmark four state-of-the-art algorithms for semantic segmentation and three baseline methods for hierarchical segmentation using PartNet;
- We propose the task of part instance segmentation on PartNet and describe a baseline method that outperforms the existing baseline method by a large margin.

2. Related Work

Understanding shape parts is a long-standing problem in computer vision and graphics. Lacking large-scale annotated datasets, early research efforts evaluated algorithm results qualitatively and conducted quantitative comparison on small sets of 3D models. Attene *et al*. [1] compared 5 mesh segmentation algorithms using 11 3D surface meshes and presented side-by-side qualitative comparison. Chen *et al*. [5] collected 380 surface meshes with instance-level part decomposition and proposed quantitative metrics for evaluation. Concurrently, Benhabiles *et al*. [2] proposed similar evaluation criteria and methodology. Kalogerakis *et al*. [17] further assigned semantic labels to the segmented components. Shape co-segmentation benchmarks [38, 11] were proposed to study co-segmentation among similar shapes.

Recent advances in deep learning have demonstrated the power and efficiency of data-driven methods on 3D shape understanding tasks. ShapeNet [3] collected a large-scale 3D CAD models from online open-sourced 3D repositories, including more than 3,000,000 models and 3,135 object categories. Yi *et al*. [45] took an active learning approach to annotate the ShapeNet models with semantic segmentation for 31,963 shapes covering 16 object categories. In their dataset, each object is usually decomposed into 2∼5 coarse semantic parts. PartNet provides more fine-grained part annotations that contains 18 parts per shape on average.

Many recent works studied fine-grained and hierarchical shape segmentation. Yi *et al*. [44] leveraged the noisy part decomposition inputs in the CAD model designs to learn consistent shape hierarchies. Chang *et al*. [4] collected 27,477 part instances from 2,278 models covering 90 object categories and studied the part properties related to language. Wang *et al*. [37] collected 1,016 3D models from 10 object categories and trained neural networks for grouping and labeling fine-grained part components. A concurrent work [47] proposed a recursive binary decomposition network for shape hierarchical segmentation. PartNet provides a large-scale testbed with 573,585 fine-grained and hierarchical shape parts to support this direction of research.

There are also many previous works that attempted to understand parts by their functionality and articulation. Hu *et al*. [13] constructed a dataset of 608 objects from 15 object categories annotated with the object functionality and introduced a co-analysis method to learns category-wise object functionality. Hu *et al*. [12] proposed a dataset of 368 mo-

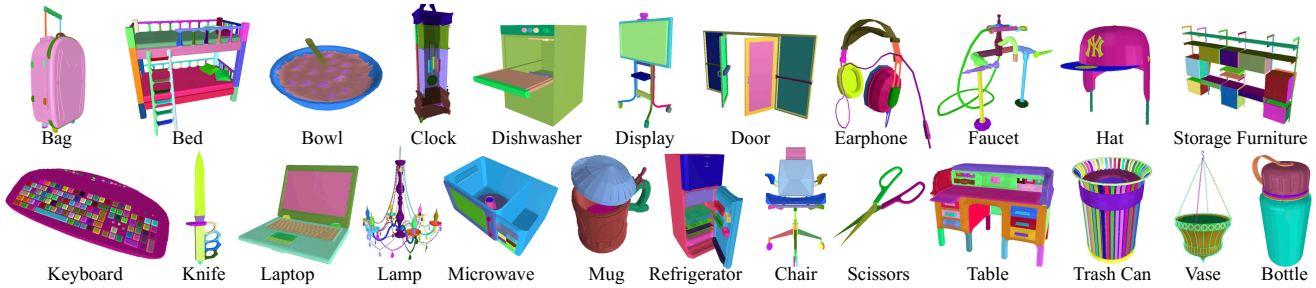


Figure 2. **PartNet dataset.** We visualize example shapes with fine-grained part annotations for the 24 object categories in PartNet.

	All	Bag	Bed	Bott	Bowl	Chair	Clock	Dish	Disp	Door	Ear	Fauc	Hat	Key	Knife	Lamp	Lap	Micro	Mug	Frid	Scis	Stora	Table	Trash	Vase
#A	32537	186	248	519	247	8176	624	241	1005	285	285	840	287	210	514	3408	485	268	252	247	127	2639	9906	378	1160
#S	26671	146	212	464	208	6400	579	201	954	245	247	708	250	174	384	2271	453	212	212	207	88	2303	8309	340	1104
#M	771	20	18	28	20	77	25	20	26	20	19	60	19	18	57	64	20	28	20	20	20	34	91	19	28
#PS	480	4	24	12	4	57	23	12	8	8	15	18	8	3	16	83	8	12	4	13	5	36	82	15	10
#PI	573K	664	9K	2K	615	176K	4K	2K	7K	2K	3K	8K	1K	20K	3K	50K	3K	2K	839	2K	981	77K	177K	8K	5K
$\mathbf{P}_{\mathbf{med}}$	14	4	33	5	2	19	5	9	8	7	12	9	4	106	7	12	8	7	3	9	8	24	15	9	4
$\mathbf{P}_{\mathbf{max}}$	230	7	169	7	4	153	32	16	12	20	14	34	5	127	10	230	8	17	6	33	9	220	214	143	200
$\mathbf{D}_{\mathbf{med}}$	3	1	5	2	1	3	3	3	3	3	3	3	2	1	3	5	2	3	1	3	2	4	4	2	2
$\mathbf{D}_{\mathbf{max}}$	7	1	5	2	1	5	4	3	3	3	3	3	2	1	3	7	2	3	1	3	2	5	6	2	3

Table 2. **PartNet statistics.** Row **#A**, **#S**, **#M** respectively show the number of shape annotations, the number of distinct shape instances and the number of shapes that we collect multiple annotations. Row **#PS** and **#PI** show the number of different part semantics and part instances that we finally collect. Row $\mathbf{P}_{\mathbf{med}}$ and $\mathbf{P}_{\mathbf{max}}$ respectively indicate the median and maximum number of part instances per shape. Row $\mathbf{D}_{\mathbf{med}}$ and $\mathbf{D}_{\mathbf{max}}$ respectively indicate the median and maximum hierarchy depth per shape, with root node as depth 0.

bility units with diverse types of articulation and learned to predict part mobility information from a single static segmented 3D mesh. In PartNet, we assign consistent semantic labels that entail such functionality and articulation information for part components within each object category, which makes PartNet useful for such research.

3. Data Annotation

The data annotation is performed in a hierarchical manner. Expert-defined hierarchical part templates are provided to guarantee labeling consistency among multiple annotators. We design a single-thread question-answering 3D GUI to guide the annotation. We hire 66 professional annotators and train them for the annotation. The average annotation time per shape is 8 minutes, and at least one pass of verification is performed for each annotation to ensure accuracy.

3.1. Expert-Defined Part Hierarchy

Shape segmentation naturally involves hierarchical understanding. People understand shapes at different granularities. Coarse parts convey global semantics while fine-grained parts provide more detailed understanding. Moreover, fine-grained part concepts are more obscure to define than coarse parts. Different annotators have different knowledge and background so that they may name parts differently when using free-form annotation [4]. To address these issues, we introduce And-Or-Graph-style hierarchical templates and collect part annotations according to the predefined templates.

Since there are no standard rules of thumb for defining good templates, it is non-trivial to design good hierarchical part templates for a category. Furthermore, the requirement for the designed template to cover all variations of shapes and parts, makes the problem even more challenging. Below we summarize the criteria that we used to guide our template design:

- **Well-defined**: Part concepts are well-delineated such that parts are identifiable by multiple annotators;
- **Consistent:** Part concepts are shared and reused across different parts, shapes and object categories;
- **Compact:** There is no unnecessary part concept and part concepts are reused when it is possible;
- **Hierarchical:** Part concepts are organized in a taxonomy to cover both coarse and fine-grained parts;
- **Atomic:** Leaf nodes in the part taxonomy consist of primitive, non-decomposable shapes;
- **Complete:** The part taxonomy covers a heterogeneous variety of shapes as completely as possible.

Guided by these general principles, we build an And-Or-Graph-style part template for each object category. The templates are defined by experts after examining a broad variety of objects in the category. Each template is designed in a hierarchical manner from the coarse semantic parts to the fine-grained primitive-level components. Figure 3 (middle) shows the lamp template. And-nodes segment a part into small subcomponents. Or-nodes indicate subcategorization for the current part. The combination of And-nodes and Or-nodes allows us to cover structurally different shapes

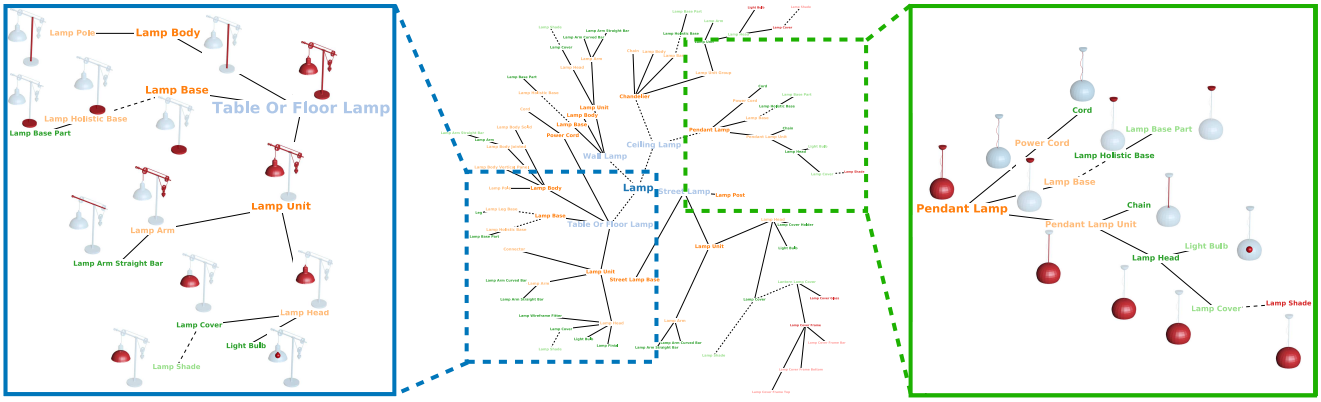


Figure 3. We show the expert-defined hierarchical template for lamp (middle) and the instantiations for a table lamp (left) and a ceiling lamp (right). The And-nodes are drawn in solid lines and Or-nodes in dash lines. The template is deep and comprehensive to cover structurally different types of lamps. In the meantime, the same part concepts, such as light bulb and lamp shade, are shared across the different types.

using the same template while sharing as much common part labels as possible. As shown in Figure 3 (left) and (right), both table lamps and ceiling lamps are explained by the same template through the first-level Or-node for lamp types.

Despite the depth and comprehensiveness of these templates, it is still impossible to cover all cases. Thus, we allow our annotators to improve upon the structure of the template and to annotate parts that are out of the scope of our definition. We also conduct template refinements to resolve part ambiguity after we obtain the data annotation according to the original templates. To systematically identify ambiguities, we reserve a subset of shapes from each class and collect multiple human annotations for each shape. We compute the confusion matrix among different annotators and address data inconsistencies. For example, we merge two concepts with high confusion scores or remove a part if it is frequently segmented in the wrong way. We provide more details about this in the supplementary material.

3.2. Annotation Interface

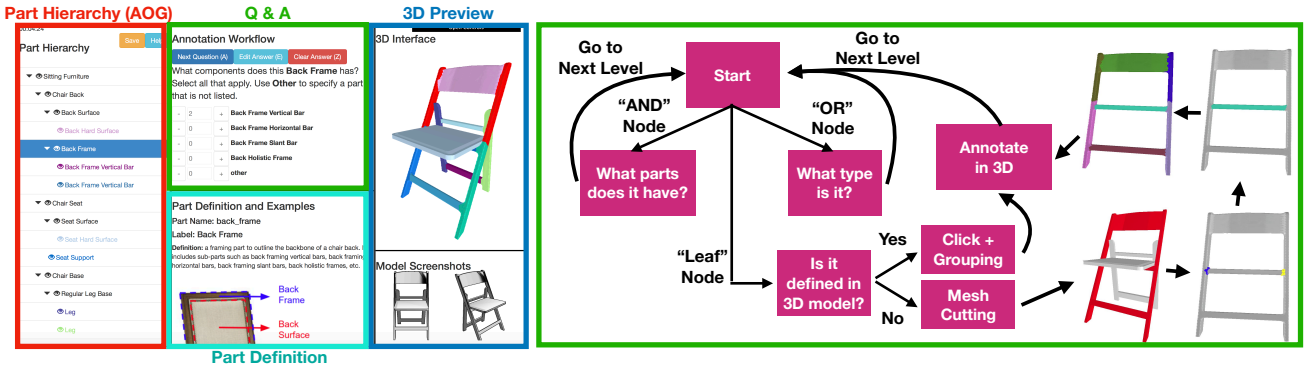
Figure 4 (a) shows our web-based annotation interface. Based on the template hierarchy, the annotation process is designed to be a single-thread question-answering workflow, traversing the template graph in a depth-first manner, as shown in Figure 4 (b). Starting from the root node, the annotator is asked a sequence of questions. The answers automatically construct the final hierarchical segmentation for the current shape instance. For each question, the annotator is asked to mark the number of subparts (And-node) or pick one among all subtypes (Or-node) for a given part. For each leaf node part, the annotator annotates the part geometry in the 3D interface. To help them understand the part definition and specification, we provide rich textual definitions and visual examples for each part. In addition, our interface supports cross-section and visibility control to annotate the interior structure of a 3D model.

The collected 3D CAD models often include original mesh subgroups and part information. Some of the grouping information is detailed enough to determine the final segmentation we need. Considering this, we provide the annotators with the original groupings at the beginning of the annotation, to speed up annotation. The annotators can simply select multiple predefined pieces to form a part of the final segmentation. We also provide mesh cutting tools to split large pieces into smaller ones following [5], when the original groupings are coarser than the desired segmentation, as shown in Figure 4 (c). The annotators draw boundary lines on the remeshed watertight surface [15] and the mesh cutting algorithm automatically splits the mesh into multiple smaller subcomponents.

In contrast to prior work, our UI is designed for operating directly on 3D models and collecting fine-grained and hierarchical part instances. Compared to Yi *et al*. [45] where the annotation is performed in 2D, our approach allows the annotators to directly annotate on the 3D shapes and thus be able to pick up more subtle part details that are hidden from 2D renderings. Chang *et al*. [4] proposes a 3D UI that paints regions on mesh surfaces for part labeling. However, their interface is limited to existing over-segmentations on part components and does not support hierarchical annotations.

4. PartNet Dataset

The final PartNet dataset provides fine-grained and hierarchical instance-level part segmentation annotation for $26,671$ shapes with $573,585$ part instances from 24 object categories. We select categories from ShapeNetCore [3] that 1) are mostly seen in indoor scenes; 2) contain interesting intra-class variation; and 3) provide a huge number of parts. We add 3 new object categories that are commonly present in indoor scenes (*i.e.* scissors, refrigerators, and doors) and augment 7 of the existing categories with more



(a) PartNet Main Web Interface (b) Human Annotation Workflow (c) Mesh Cutting Interface

Figure 4. We show our annotation interface with its components, the proposed question-answering workflow and the mesh cutting interface.

3D models from 3D Warehouse[1].

Figure 2 and Table 2 show the PartNet data and statistics. More visualization and statistics are included in the supplemental materials. Our templates define hierarchical segmentation a median depth of 3 and maximum depth of 7. In total, we annotate $573,585$ fine-grained part instances, with a median of 14 parts per shape and a maximum of 230. To study annotation consistency, we also collected multiple annotations per shape for a subset of 771 shapes.

5. Tasks and Benchmarks

We benchmark three part-level object understanding tasks: fine-grained semantic segmentation, hierarchical semantic segmentation and instance segmentation.

Data Preparation. We only consider parts that can be fully determined by their shape geometry[2]. In evaluation, we ignore parts that require additional information to identify, such as glass parts on cabinet doors which requires opacity to identify, and buttons on microwaves which requires texture or color information to distinguish it. We also remove infrequent parts from the evaluation due to the lack of data samples.

We sample $10,000$ points from each CAD model with furthest point sampling and use the 3D coordinates as the neural network inputs for all the experiments in the paper. The proposed dataset is split into train, validation and test sets with the ratio 70%: 10%: 20%. The shapes with multiple human annotations are not used in the experiments.

5.1. Fine-grained Semantic Segmentation

Recent advances of 3D semantic segmentation [30, 31, 46, 19, 35, 24, 9, 39, 40, 42, 33, 7, 26, 23] have accomplished promising performance in coarse-level segmentation on the ShapeNet Part dataset [3, 45]. However, few work focus on the fine-grained 3D semantic segmentation, due to the lack of large-scale fine-grained dataset. With the help of the proposed PartNet dataset, researchers can now work on this more challenging task with little overhead.

Fine-grained 3D semantic segmentation requires recognizing and distinguishing small and similar semantic parts. For example, door handles are usually small, 77 out of $10,000$ points on average in PartNet, but semantically important on doors. Beds have several geometrically similar parts such as side vertical bars, post bars and base legs. To recognize the subtle part details, segmentation systems need to understand them locally, through discriminative features, and globally, in the context of the whole shape.

Benchmark Algorithms. We benchmark four state-of-the-art semantic segmentation algorithms: PointNet [30], PointNet++ [31], SpiderCNN [42] and PointCNN [26][3]. PointNet [30] takes unordered point sets as inputs and extracts features for shape classification and segmentation. To better learn local geometric features, PointNet++ [31] proposes a hierarchical feature extraction scheme. SpiderCNN [42] extends traditional convolution operations on 2D images to 3D point clouds by parameterizing a family of convolutional filters. To organize the unordered points into latent canonical order, PointCNN [26] proposes to learn $\mathcal{X}$-transformation, and applies $\mathcal{X}$-convolution operations on the canonical points.

We train the four methods on the dataset, using the default network architectures and hyperparameters described in their papers. Instead of training a single network for all object categories as done in most of these papers, we train a network for each category at each segmentation level. We input only the 3D coordinates for fair comparison[4] and train the networks until convergence. More training details are described in the supplementary material.

[1] https://3dwarehouse.sketchup.com

[2] Although 3D models in ShapeNet [3] come with face normal, textures, material and other information, there is no guarantee for the quality of such information. Thus, we leave this as a future work.

[3] There are many other algorithm candidates: [46, 19, 35, 24, 9, 39, 40, 33, 7, 23]. We will host an online leadboard to report the performances.

[4] PointNet++ [31] and SpiderCNN [42] use point normals as additional inputs. For fair comparison, we only input the 3D coordinates.

	Avg	Bag	Bed	Bott	Bowl	Chair	Clock	Dish	Disp	Door	Ear	Fauc	Hat	Key	Knife	Lamp	Lap	Micro	Mug	Frid	Scis	Stora	Table	Trash	Vase
P1	57.9	42.5	32.0	33.8	58.0	64.6	33.2	76.0	86.8	**64.4**	53.2	58.6	55.9	65.6	62.2	29.7	96.5	49.4	80.0	49.6	86.4	51.9	50.5	55.2	54.7
P2	37.3	–	20.1	–	–	38.2	–	55.6	–	38.3	–	–	–	–	–	27.0	–	41.7	–	35.5	–	44.6	34.3	–	–
P3	35.6	–	13.4	29.5	–	27.8	28.4	48.9	76.5	30.4	33.4	47.6	–	–	32.9	18.9	–	37.2	–	33.5	–	38.0	29.0	34.8	44.4
Avg	51.2	42.5	21.8	31.7	58.0	43.5	30.8	60.2	81.7	44.4	43.3	53.1	55.9	65.6	47.6	25.2	96.5	42.8	80.0	39.5	86.4	44.8	37.9	45.0	49.6
P$^+$1	**65.5**	59.7	51.8	53.2	67.3	68.0	**48.0**	80.6	89.7	59.3	**68.5**	64.7	62.4	62.2	**64.9**	**39.0**	96.6	55.7	83.9	51.8	87.4	58.0	**69.5**	64.3	**64.4**
P$^+$2	44.5	–	38.8	–	–	43.6	–	55.3	–	49.3	–	–	–	–	–	**32.6**	–	48.2	–	41.9	–	49.6	**41.1**	–	–
P$^+$3	42.5	–	30.3	41.4	–	39.2	**41.6**	50.1	80.7	32.6	38.4	52.4	–	–	**34.1**	**25.3**	–	48.5	–	36.4	–	40.5	**33.9**	46.7	49.8
Avg	58.1	59.7	40.3	**47.3**	67.3	50.3	**44.8**	62.0	85.2	47.1	53.5	58.6	62.4	62.2	**49.5**	**32.3**	96.6	50.8	83.9	43.4	87.4	49.4	**48.2**	55.5	57.1
S1	60.4	57.2	55.5	**54.5**	**70.6**	67.4	33.3	70.4	90.6	52.6	46.2	59.8	63.9	64.9	37.6	30.2	**97.0**	49.2	83.6	50.4	75.6	**61.9**	50.0	62.9	63.8
S2	41.7	–	40.8	–	–	39.6	–	59.0	–	48.1	–	–	–	–	–	24.9	–	47.6	–	34.8	–	46.0	34.5	–	–
S3	37.0	–	36.2	32.2	–	30.0	24.8	50.0	80.1	30.5	37.2	44.1	–	–	22.2	19.6	–	43.9	–	39.1	–	44.6	20.1	42.4	32.4
Avg	53.6	57.2	44.2	43.4	**70.6**	45.7	29.1	59.8	85.4	43.7	41.7	52.0	63.9	64.9	29.9	24.9	**97.0**	46.9	83.6	41.4	75.6	50.8	34.9	52.7	48.1
C1	64.3	**66.5**	**55.8**	49.7	61.7	**69.6**	42.7	**82.4**	**92.2**	63.3	64.1	**68.7**	**72.3**	**70.6**	62.6	21.3	**97.0**	**58.7**	**86.5**	**55.2**	**92.4**	61.4	17.3	**66.8**	63.4
C2	**46.5**	–	**42.6**	–	–	**47.4**	–	**65.1**	–	**49.4**	–	–	–	–	–	22.9	–	**62.2**	–	**42.6**	–	**57.2**	29.1	–	–
C3	**46.4**	–	**41.9**	**41.8**	–	**43.9**	36.3	**58.7**	**82.5**	**37.8**	**48.9**	**60.5**	–	–	**34.1**	20.1	–	**58.2**	–	**42.9**	–	**49.4**	21.3	**53.1**	**58.9**
Avg	**59.8**	**66.5**	**46.8**	45.8	61.7	**53.6**	39.5	**68.7**	**87.4**	**50.2**	**56.5**	**64.6**	**72.3**	**70.6**	48.4	21.4	**97.0**	**59.7**	**86.5**	**46.9**	**92.4**	**56.0**	22.6	**60.0**	**61.2**

Table 3. **Fine-grained semantic segmentation results (part-category mIoU %).** Algorithm **P**, **P$^+$**, **S** and **C** refer to PointNet [30], PointNet++ [31], SpiderCNN [42] and PointCNN [26], respectively. The number **1**, **2** and **3** refer to the three levels of segmentation: coarse-, middle- and fine-grained. We put short lines for the levels that are not defined.

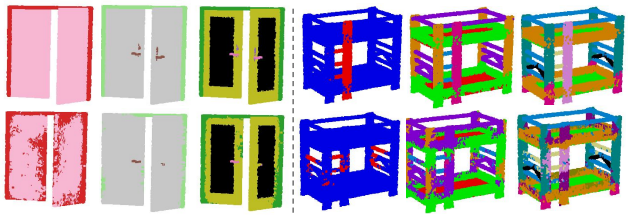


Figure 5. **Qualitative results for semantic segmentation.** The top row shows the ground-truth and the bottom row shows the PointCNN prediction. The black points indicate unlabeled points.

Evaluation and Results. We evaluate the algorithms at three segmentation levels for each object category: coarse-, middle- and fine-grained. The coarse level approximately corresponds to the granularity in Yi *et al*. [45]. The fine-grained level refers to the segmentation down to leaf levels in the segmentation hierarchies. For structurally deep hierarchies, we define a middle level in between. Among 24 object categories, all of them have the coarse level, while 9 have the middle level and 17 have the fine level. Overall, we define 50 segmentation levels for 24 object categories.

In Table 3, we report semantic segmentation results at multiple levels of granularity on PartNet. We use the mean Intersection-over-Union (mIoU) scores as the evaluation metric. After removing unlabeled ground-truth points, for each object category, we first calculate the IoU between the predicted point set and the ground-truth point set for each semantic part category across all test shapes. Then, we average the per-part-category IoUs to compute the mIoU for the object category. We further calculate the average mIoU across different levels for each object category and finally report the average cross all object categories.

Unsurprisingly, performance for all four algorithms drop by a large margin from the coarse level to the fine-grained level. Figure 5 shows qualitative results from PointCNN. The method does not perform well on small parts, such as the door handle on the door example, and visually similar parts, such as stair steps and the horizontal bars on the bed frame. How to learn discriminative features that better capture both local geometry and global context for these issues would be an interest topic for future works.

5.2. Hierarchical Semantic Segmentation

Shape segmentation is hierarchical by its nature. We study hierarchical semantic segmentation that predicts semantic part labels in the entire shape hierarchies that cover both coarse- and fine-grained part concepts. A key problem towards hierarchical segmentation is how to leverage the rich part relationships on the given shape templates in the learning procedure. Recognizing a chair base as a swivel base significantly reduces the solution space for detecting more fine-grained parts such as central supporting bars, star-base legs and wheels. On the other hand, the lack of a chair back increases the possibility that the object is a stool. In contrast to Sec. 5.1 where we consider the problem at each segmentation level separately, hierarchical segmentation requires a holistic understanding on the entire part hierarchy.

Benchmark Algorithms. We propose three baseline methods to tackle hierarchical segmentation: bottom-up, top-down and ensemble. The bottom-up method considers only the leaf-node parts during training and groups the prediction of the children nodes to parent nodes as defined in the hierarchies in bottom-up inference. The top-down method learns a multi-labeling task over all part semantic labels on the tree and conducts a top-down inference by classifying coarser-level nodes first and then finer-level ones. For the ensemble method, we train flat segmentation at multiple levels as defined in Sec. 5.1 and conduct joint inference by calculating the average log-likelihood scores over all the root-to-leaf paths on the tree. We use PointNet++ [31] as the

	Avg	Bed	Bott	Chair	Clock	Dish	Disp	Door	Ear	Fauc	Knife	Lamp	Micro	Frid	Stora	Table	Trash	Vase
Bottom-Up	51.2	40.8	56.1	47.2	38.3	**61.5**	84.1	**52.6**	54.3	**63.4**	52.3	36.8	48.2	41.0	46.8	38.3	**53.6**	54.4
Top-Down	50.8	41.1	**56.2**	46.5	34.3	54.5	**84.7**	50.6	**59.5**	61.4	**55.6**	37.1	**48.8**	**41.6**	45.2	37.0	53.5	**55.6**
Ensemble	**51.7**	**42.0**	54.7	**48.1**	**44.5**	58.8	**84.7**	51.4	57.2	61.9	51.9	**37.6**	47.5	41.4	**47.3**	**44.0**	52.8	53.1

Table 4. **Hierarchical segmentation results (part-category mIoU %).** We present the hierarchical segmentation performances for three baseline methods: bottom-up, top-down and ensemble. We conduct experiments on 17 out of 24 categories with tree depth bigger than 1.

backbone network[5] in this work. Note that methods listed in Sec. 5.1 can also be used. More architecture and training details are described in the supplementary material.

Evaluation and Results. Table 4 demonstrates the performances of the three baseline methods. We calculate mIoU for each part category and compute the average over all the tree nodes as the evaluation metric. The experimental results show that the three methods perform similarly with small performance gaps. The ensemble method performs slightly better over the other two, especially for the categories with rich structural and sub-categorization variation, such as chair, table and clock.

The bottom-up method only considers leaf-node parts in the training. Although the template structure is not directly used, the parent-node semantics of leaf nodes are implicitly encoded in the leaf-node part definitions. For example, the vertical bars for chair backs and chair arms are two different leaf nodes. The top-down method explicitly leverages the tree structures in both the training and the testing phases. However, prediction errors are accumulated through top-down inference. The ensemble method decouples the hierarchical segmentation task into individual tasks at multiple levels and performs joint inference, taking the predictions at all levels into consideration. Though demonstrating better performances, it has more hyper-parameters and requires longer training time for the multiple networks.

5.3. Instance Segmentation

The goal of instance segmentation is to detect every individual part instance and segment it out from the context of the shape. Many applications in computer graphics, vision and robotics, including manufacturing, assembly, interaction and manipulation, require the instance-level part recognition. Compared to detecting objects from scenes, parts on objects usually have stronger and more intertwined structural relationships. The existence of many visually-similar but semantically-different parts makes the part detection problem challenging. To the best of our knowledge, this work is the first to provide a large-scale shape part instance-level segmentation benchmark.

[5]In our experiments, PointNet++ and PointCNN give the top ranked performance under two different evaluation metrics: part-category mIoU (Table 3) and shape mIoU (Table 2 in supplementary material). We choose PointNet++ because previous works on ShapeNet mostly use shape mIoU as the metric. We reported part-category mIoU in the main paper to make it consistent with mIoU and mAP evaluation metrics used in ScanNet [6].

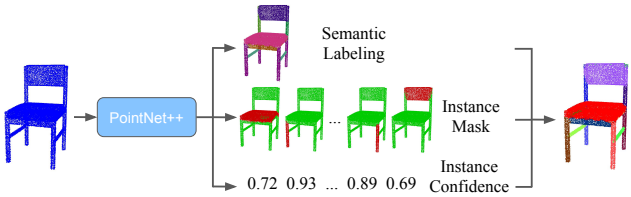


Figure 6. The proposed method for instance segmentation.

Given a shape point cloud as input, the task of part instance segmentation is to provide several disjoint masks over the entire point cloud, each of which corresponds to an individual part instance on the object. We adopt the part semantics from the defined segmentation levels in Sec. 5.1. The detected masks should have no overlaps, but they together do not necessarily cover the entire point cloud, as some points may not belong to any part of interests.

Benchmark Algorithms. We propose a part instance segmentation network (Figure 6) inspired by [32] to address instance segmentation. We use PointNet++ [31] as the backbone network for extracting features and predicting both semantic segmentation for each point and K instance segmentation masks $\{\hat{y}_i \in [0,1]^N | i = 1, 2, \cdots, K\}$ over the input point cloud of size N. Moreover, we train a separate mask $\hat{y}_{other}$ for the points without semantic labels in the ground-truth. A softmax activation layer is applied to encourage the mutual exclusiveness among different masks so that $\hat{y}_1 + \hat{y}_2 + \cdots + \hat{y}_K + \hat{y}_{other} = \mathbf{1}$. To train the network, we apply the Hungarian algorithm [22] to find a bipartite matching $\mathcal{M} : \{i \to \mathcal{M}(i) | i = 1, 2, \cdots, T\}$ between the prediction masks $\{\hat{y}_i | i = 1, 2, \cdots, K\}$ and the ground-truth masks $\{y_i | i = 1, 2, \cdots, T\}$, and regress each prediction $\hat{y}_{\mathcal{M}(t)}$ to the matched ground-truth mask y_t. We employ a relaxed version of IoU [21] defined as $\text{IoU}(p, q) = \langle p, q \rangle / (\|p\|_1 + \|q\|_1 - \langle p, q \rangle)$, as the metric for the Hungarian algorithm. A separate branch is trained to predict confidence scores for the predicted masks $\{C_i | i = 1, 2, \cdots, K\}$.

The loss function is defined as $L = L_{sem} + \lambda_{ins} L_{ins} + \lambda_{other} L_{other} + \lambda_{conf} L_{conf} + \lambda_{l21} L_{l21}$, combining five terms: 1) a cross-entropy semantic segmentation loss $L_{sem} = -\sum_{i=1}^{N} \sum_{k=1}^{C} t_{ik} \log(s_{ik})$ where C is the number of part semantics, t_i is a one-hot vector for the ground-truth part semantics of point i and s_i is the post-softmax scores of point i predicted by the semantic branch; 2) an IoU loss for mask regression $L_{ins} = \sum_{i=1}^{T} \text{IoU}(\hat{y}_{\mathcal{M}(i)}, y_i)$; 3) an IoU loss for the unlabeled points $L_{other} = \text{IoU}(\hat{y}_{other}, y_{other})$; 4) a prediction-confidence loss $L_{conf} = \sum_{i=1}^{T} (C_{\mathcal{M}(i)} -$

	Avg	Bag	Bed	Bott	Bowl	Chair	Clock	Dish	Disp	Door	Ear	Fauc	Hat	Key	Knife	Lamp	Lap	Micro	Mug	Frid	Scis	Stora	Table	Trash	Vase
S1	55.7	38.8	29.8	61.9	56.9	72.4	20.3	72.2	89.3	49.0	**57.8**	63.2	68.7	20.0	**63.2**	32.7	**100**	**50.6**	82.2	**50.6**	71.7	32.9	49.2	56.8	46.6
S2	29.7	–	15.4	–	–	25.4	–	58.1	–	25.4	–	–	–	–	–	21.7	–	**49.4**	–	22.1	–	30.5	18.9	–	–
S3	29.5	–	11.8	45.1	–	19.4	18.2	38.3	78.8	15.4	35.9	37.8	–	–	38.3	14.4	–	32.7	–	18.2	–	21.5	14.6	24.9	36.5
Avg	46.8	38.8	19.0	53.5	56.9	39.1	19.3	56.2	84.0	29.9	46.9	50.5	68.7	20.0	50.7	22.9	**100**	**44.2**	82.2	30.3	71.7	28.3	27.5	40.9	41.6
O1	**62.6**	**64.7**	**48.4**	**63.6**	**59.7**	**74.4**	**42.8**	**76.3**	**93.3**	**52.9**	57.7	**69.6**	**70.9**	**43.9**	58.4	**37.2**	**100**	50.0	**86.0**	50.0	**80.9**	**45.2**	**54.2**	**71.7**	**49.8**
O2	**37.4**	–	**23.0**	–	–	**35.5**	–	**62.8**	–	**39.7**	–	–	–	–	–	**26.9**	–	47.8	–	**35.2**	–	**35.0**	**31.0**	–	–
O3	**36.6**	–	**15.0**	**48.6**	–	**29.0**	**32.3**	**53.3**	**80.1**	**17.2**	**39.4**	**44.7**	–	–	**45.8**	**18.7**	–	**34.8**	–	**26.5**	–	**27.5**	**23.9**	**33.7**	**52.0**
Avg	**54.4**	**64.7**	**28.8**	**56.1**	**59.7**	**46.3**	**37.5**	**64.1**	**86.7**	**36.6**	**48.5**	**57.1**	**70.9**	**43.9**	**52.1**	**27.6**	**100**	**44.2**	**86.0**	**37.2**	**80.9**	**35.9**	**36.4**	**52.7**	**50.9**

Table 5. **Instance segmentation results (part-category mAP %, IoU threshold 0.5).** Algorithm **S** and **O** refer to SGPN [36] and our proposed method respectively. The number **1**, **2** and **3** refer to the three levels of segmentation: coarse-, middle- and fine-grained.

Figure 7. **Qualitative results for instance segmentation.** Our method produces more robust and cleaner results than SGPN.

Figure 8. **Learned instance correspondences.** The corresponding parts are marked with the same color.

$\text{IoU}(\hat{y}_{\mathcal{M}(i)}, y_i))^2$; and 5) a $l_{2,1}$-norm regularization term $L_{l21} = \sum_{i=1}^{K} \|\hat{y}_i\|_2 + \|\hat{y}_{other}\|_2$ to encourage unused prediction masks to vanish [34]. We use $N = 10,000$, $K = 200$, $\lambda_{ins} = 1.0$, $\lambda_{other} = 1.0$, $\lambda_{conf} = 1.0$, $\lambda_{l21} = 0.1$.

We compare the proposed method with SGPN [36], which learns similarity scores among all pairs of points and detect part instances by grouping points that share similar features. We follow most of the default settings and hyper-parameters described in their paper. We first pre-train PointNet++ semantic segmentation branch and then fine-tune it for improving the per-point feature similarity matrix and confidence maps. We use margin values of 1 and 2 for the double-hinge loss as suggested by the authors of [36], instead of 10 and 80 in the original paper. We feed 10,000 points to the network at a time, and use a batch-size of 32 in the pre-training and 1 in the fine-tuning.

Evaluation and Results. Table 5 reports the per-category mean Average Precision (mAP) scores for SPGN and our proposed method. For each object category, the mAP score calculates the AP for each semantic part category across all test shapes and averages the AP across all part categories. Finally, we take the average of the mAP scores across different levels of segmentation within each object category and then report the average over all object categories. We compute the IoU between each prediction mask and the closest ground-truth mask and treat a prediction mask as a true positive when the IoU is larger than 0.5.

Figure 7 shows qualitative comparisons for our proposed method and SGPN. Our method produces more robust and cleaner instance predictions. After learning for point features, SGPN has a post-processing stage that merges points with similar features as one component. This process involves many hyper-parameter tuning. Even though most parameters are automatically inferred from the validation data, SPGN still suffers from predicting partial or noisy instances in case of bad thresholding. Our proposed method learns structural priors within each object category that is more instance-aware and robust in predicting complete instances. We observe that training for a set of disjoint masks across multiple shapes gives us consistent part instances. We show the learned part correspondence in Figure 8.

6. Conclusion

We introduce PartNet: a *large-scale* benchmark for *fine-grained, hierarchical, and instance-level* 3D shape segmentation. It contains 573,585 part annotations for 26,671 ShapeNet [3] models from 24 object categories. Based on the dataset, we propose three shape segmentation benchmarks: fine-grained semantic segmentation, hierarchical semantic segmentation and instance segmentation. We benchmark four state-of-the-art algorithms for semantic segmentation and propose a baseline method for instance segmentation that outperforms the existing baseline method.

Acknowledgements

This research was supported by NSF grants CRI-1729205, IIS-1763268 and CHS-1528025, a Vannevar Bush Faculty Fellowship, a Google fellowship, and gifts from Autodesk, Google and Intel AI Lab. We especially thank Zhe Hu from Hikvision for the help on data annotation and Linfeng Zhao for the help on preparing hierarchical templates. We appreciate the 66 annotators from Hikvision, Ytuuu and Data++ on data annotation.

References

[1] Marco Attene, Sagi Katz, Michela Mortara, Giuseppe Patané, Michela Spagnuolo, and Ayellet Tal. Mesh segmentation-a comparative study. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 7–7. IEEE, 2006. 2

[2] Halim Benhabiles, Jean-Philippe Vandeborre, Guillaume Lavoué, and Mohamed Daoudi. A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D-models. In *IEEE International Conference on Shape Modeling and Applications (SMI)*, pages Session–5, 2009. 2

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. (arXiv:1512.03012 [cs.GR]), 2015. 1, 2, 4, 5, 8

[4] Angel X Chang, Rishi Mago, Pranav Krishna, Manolis Savva, and Christiane Fellbaum. Linking WordNet to 3D shapes. In *Global WordNet Conference*, 2018. 2, 3, 4

[5] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2009. 1, 2, 4

[6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 7

[7] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 18–22, 2018. 1, 5

[8] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D Wegner, Konrad Schindler, and Marc Pollefeys. Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*, 2017. 1

[9] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte carlo convolution for learning on non-uniformly sampled point clouds. *arXiv preprint arXiv:1806.01759*, 2018. 1, 5

[10] Donald D Hoffman and Whitman A Richards. Parts of recognition. *Cognition*, 18(1-3):65–96, 1984. 1

[11] Ruizhen Hu, Lubin Fan, and Ligang Liu. Co-segmentation of 3D shapes via subspace clustering. In *Computer graphics forum*, volume 31, pages 1703–1713. Wiley Online Library, 2012. 2

[12] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Transactions on Graphics (TOG)*, 36(6):227, 2017. 2

[13] Ruizhen Hu, Oliver van Kaick, Bojian Wu, Hui Huang, Ariel Shamir, and Hao Zhang. Learning how objects function via co-analysis of interactions. *ACM Transactions on Graphics (TOG)*, 35(4):47, 2016. 1, 2

[14] Ruizhen Hu, Zihao Yan, Jingwen Zhang, Oliver van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Predictive and generative neural networks for object functionality. In *Computer Graphics Forum (Eurographics State-of-the-art report)*, volume 37, pages 603–624, 2018. 1

[15] Jingwei Huang, Hao Su, and Leonidas Guibas. Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*, 2018. 4

[16] Arjun Jain, Thorsten Thormählen, Tobias Ritschel, and Hans-Peter Seidel. Exploring shape variations by 3d-model decomposition and part-based recombination. In *Computer Graphics Forum*, volume 31, pages 631–640. Wiley Online Library, 2012. 1

[17] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3D mesh segmentation and labeling. *ACM Transactions on Graphics (TOG)*, 29(4):102, 2010. 2

[18] Vladimir G Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. Shape2pose: Human-centric shape analysis. *ACM Transactions on Graphics (TOG)*, 33(4):120, 2014. 1

[19] Roman Klokov and Victor 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. 1, 5

[20] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 1

[21] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning*, pages 513–521, 2013. 7

[22] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 7

[23] Truc Le and Ye Duan. PointGrid: A deep network for 3D shape understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9204–9214, 2018. 1, 5

[24] Jiaxin Li, Ben M Chen, and Gim Hee Lee. SO-Net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9397–9406, 2018. 1, 5

[25] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):52, 2017. 1

[26] Yangyan Li, Rui Bu, Mingchao Sun, and Baoquan Chen. PointCNN: Convolution on $\mathcal{X}$-transformed points. *Advances in neural information processing systems (NIPS)*, 2018. 1, 5, 6

[27] Zhijian Liu, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Physical primitive decomposition. *arXiv preprint arXiv:1809.05070*, 2018. 1

[28] Maks Ovsjanikov, Wilmot Li, Leonidas Guibas, and Niloy J Mitra. Exploration of continuous variability in collections of 3d shapes. In *ACM Transactions on Graphics (TOG)*, volume 30, page 33. ACM, 2011. 1

[29] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *CVPR*, 2018. 1

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, volume 1, page 4, 2017. 1, 5, 6

[31] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas 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, 5, 6, 7

[32] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *European conference on computer vision*, pages 312–329. Springer, 2016. 7

[33] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan 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. 1, 5

[34] Minhyuk Sung, Hao Su, Ronald Yu, and Leonidas Guibas. Deep functional dictionaries: Learning consistent semantic structures on 3D models from functions. *Advances in neural information processing systems (NIPS)*, 2018. 8

[35] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Transactions on Graphics (TOG)*, 36(4):72, 2017. 1, 5

[36] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2569–2578, 2018. 8

[37] Xiaogang Wang, Bin Zhou, Haiyue Fang, Xiaowu Chen, Qinping Zhao, and Kai Xu. Learning to group and label fine-grained shape components. *ACM Transactions on Graphics (SIGGRAPH Asia 2018)*, 37(6), 2018. 2

[38] Yunhai Wang, Shmulik Asafi, Oliver Van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Active co-analysis of a set of shapes. *ACM Transactions on Graphics (TOG)*, 31(6):165, 2012. 2

[39] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018. 1, 5

[40] Zongji Wang and Feng Lu. VoxSegNet: Volumetric CNNs for semantic part segmentation of 3D shapes. *arXiv preprint arXiv:1809.00226*, 2018. 1, 5

[41] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Structure-aware generative network for 3d-shape modeling. *arXiv preprint arXiv:1808.03981*, 2018. 1

[42] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. SpiderCNN: Deep learning on point sets with parameterized convolutional filters. *European Conference on Computer Vision (ECCV)*, 2018. 1, 5, 6

[43] Claudia Yan, Dipendra Misra, Andrew Bennnett, Aaron Walsman, Yonatan Bisk, and Yoav Artzi. Chalet: Cornell house agent learning environment. *arXiv preprint arXiv:1801.07357*, 2018. 1

[44] Li Yi, Leonidas Guibas, Aaron Hertzmann, Vladimir G Kim, Hao Su, and Ersin Yumer. Learning hierarchical shape segmentation and labeling from online repositories. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 2017. 1, 2

[45] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, Leonidas Guibas, et al. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics (TOG)*, 35(6):210, 2016. 1, 2, 4, 5, 6

[46] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6584–6592, 2017. 1, 5

[47] Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. *arXiv preprint arXiv:1903.00709*, 2019. 2

[48] Yuke Zhu, Daniel Gordon, Eric Kolve, Dieter Fox, Li Fei-Fei, Abhinav Gupta, Roozbeh Mottaghi, and Ali Farhadi. Visual semantic planning using deep successor representations. *arXiv preprint ArXiv:1705.08080*, pages 1–13, 2017. 1