CoSegNet: Deep Co-Segmentation of 3D Shapes with Group Consistency Loss

Chenyang Zhu¹,²  Kai Xu²*  Siddhartha Chaudhuri³  Li Yi⁴  Leonidas Guibas⁴  Hao Zhang¹
¹Simon Fraser University  ²National University of Defense Technology  ³Adobe Research and IIT Bombay  ⁴Stanford University

Abstract

We introduce CoSegNet, a deep neural network architecture for co-segmentation of a set of 3D shapes represented as point clouds. CoSegNet takes as input a set of unsegmented shapes, proposes per-shape parts, and then jointly optimizes the part labelings across the set subjected to a novel group consistency loss expressed via matrix rank estimates. The proposals are refined in each iteration by an auxiliary network that acts as a weak regularizing prior, pre-trained to denoise noisy, unlabeled parts from a large collection of segmented 3D shapes, where the part compositions within the same object category can be highly inconsistent. The output is a consistent part labeling for the input set, with each shape segmented into up to K (a user-specified hyperparameter) parts. The overall pipeline is thus weakly supervised, producing consistent segmentations tailored to the test set, without consistent ground-truth segmentations. We show qualitative and quantitative results from CoSegNet and evaluate it via ablation studies and comparisons to state-of-the-art co-segmentation methods.

1. Introduction

With the proliferation of data-driven and deep learning techniques in computer vision and computer graphics, remarkable progress has been made on supervised image [17, 18, 1, 3, 16, 21] and shape segmentation [22, 23, 33, 10, 37]. Co-segmentation is an instance of the segmentation problem where the input consists of a collection, rather than one piece, of data and the collection shares certain common characteristics. Typically, for shape co-segmentation, the commonality is that the shapes all belong to the same category, e.g., airplanes. The goal for co-segmentation is to compute a consistent segmentation for all shapes in the collection. The consistency implies a correspondence between all the segmented parts, which is a critical requirement for knowledge and attribute transfer, collecting statistics over a dataset, and structure-aware shape editing [19].

Most shape co-segmentation methods to date have been unsupervised or weakly supervised [5, 26, 6, 34, 29], and this is due to two main reasons. First, accurate and consistent ground-truth segmentations are difficult to obtain, especially for large collections. An examination of existing large repositories, e.g., [27, 36], reveals that the shape segmentations therein can be highly inconsistent (e.g., see Figure 6). Second, as dictated by the consistency criterion, the same shape may be segmented differently depending on which collection it belongs to (Figure 1). In a way, the input shape collection serves as both training set and test set. Ideally, the learning scheme for co-segmentation should avoid expensive retraining and quickly adapt to new input sets.

In this paper, we introduce a deep neural network for shape co-segmentation which addresses both issues. The core component of the network is a co-segmentation module that takes as input a set of unsegmented shapes represented as point clouds, proposes per-shape parts, and then jointly optimizes the parts subject to a novel group consistency loss, expressed in terms of matrix rank estimates. The output is a K-way consistent part labeling for each shape, where K is a user-specified hyperparameter for the network. The network weights are initialized randomly and iteratively optimized via backpropagation based on the group loss.

While the co-segmentation component itself is unsupervised, and guided only by the group consistency loss, we found that the results are improved by adding a weak regularizing prior to refine the proposed part shapes. Thus,
Part refinement

Co-segmentation Network

Figure 2. Two-stage architecture of CoSegNet: a part refinement network (top) and a co-segmentation network (bottom). The part feature encoder and part refinement module in the first network learn a weak regularizing prior to denoise proposed part shapes. The co-segmentation network is trained with a novel group consistency loss, defined on a set of shapes, based on the ranks of part similarity matrices.

we pre-train a part refinement network which takes as input a possibly noisy proposed part, represented by an indicator function over the complete point cloud, and denoises or “snaps” it to a more plausible and clean shape. The part refinement network is similar to the pairwise potential of a conditional random field (CRF) in a traditional segmentation pipeline [11, 10]: while it is not a general prior (it is trained to remove only a small amount of noise), it suffices for boundary optimization. It is trained on a large collection of segmented 3D shapes, e.g., ShapeNet [2], where the part counts and part compositions within the same object category can be highly inconsistent. No segment labels are necessary: the model is label-agnostic.

Overall, our method, coined CoSegNet, is weakly supervised, since it produces consistent segmentations without consistent ground-truth segmentations. CoSegNet consists of an offline, supervised part refinement network, which is trained once on inconsistently segmented, unlabeled shapes, and a “runtime” co-segmentation network which is unsupervised and executed for each input set of shapes. It is important to note that consistency of the segmentations is not tied to the prescribed part count $K$, but to the geometric and structural features of the shape parts in the set, with $K$ serving as an upper bound for the part counts; see Figure 1. On the other hand, adjusting $K$ allows our method to produce consistent co-segmentations at varying levels of granularity, as shown in Figure 8.

Our part refinement network is trained using the ComplementMe dataset [27], and we adopt two datasets [34, 36] containing ground truth co-segmentations only for evaluation purposes. No ground truth data is needed to realize the co-segmentation network. While offline training required up to 20 hours to complete, it takes about 10 minutes to co-segment 20 shapes at a resolution of 2,048 points per shape. We show qualitative and quantitative results from CoSegNet and evaluate it through ablation studies and comparisons with state-of-the-art co-segmentation methods.

2. Related work

Deep learning for 3D shape segmentation. Many deep learning models have been developed for supervised segmentation of 3D shapes in various representations, such as voxel grids [23, 33], point cloud [22, 14, 8], multi-view projection [10], or surface mesh [37, 32]. The main idea is to replace the hand-crafted features employed in the traditional methods with data-driven learned ones. These models, however, are mostly trained targeting a fixed set of semantic labels. They cannot determine the label set dynamically based on the shapes being segmented, which is a key feature of co-segmentation. Moreover, it is costly to obtain a large training dataset with different target label sets. Relatively few works study deep learning for unsupervised segmentation of 3D shapes [25], where unsupervised learning is used only for feature learning but not for segmentation itself.

Image co-segmentation. The co-segmentation of a pair or a group of 2D images has been studied for many years in the field of computer vision, where the main goal is to segment out a common object from multiple images [30]. Most works formulate this problem as a multi-image Markov Random Field (MRF), with a foreground consistency constraint. Foreground consistency is measured based on color or object shape similarity, or dense image correspondence [31]. Recently, Li et al. [15] proposed a deep Siamese network to achieve object co-extraction from a pair of images. The general problem setting for all of these image co-segmentation works is significantly different from ours.

Co-segmentation of 3D shapes. Since the seminal work of consistent segmentation of 3D shapes [5], extensive research has been devoted to co-analysis of sets of shapes [35, 26, 7, 6, 34]. These methods often start with an over-segmentation and perform feature embedding and clustering of the over-segmented patches to obtain a consistent segmentation. While most of these methods are unsupervised, their analysis pipelines all adopt hand-craft features and heuristic-based clustering, often leading to unnatural results amid complex part or structure variations. In contrast, our part refinement network learns the part features from a large dataset and our co-segmentation network learns the network weights through a joint optimization.

Shu et al. [25] use deep auto-encoders for per-part feature learning. However, their co-segmentation component does not use a deep neural network (DNN) and it strictly constrains the final segmentations to parts learned in the first stage. In contrast, CoSegNet does not strictly adhere
to parts proposed by the refinement network, as the consistency loss can impact and adjust part labeling. In [20], a weakly-supervised method for tag-driven co-segmentation of 3D shapes is proposed. Their model is trained targeting a pre-defined label set. Sung et al. [28] attempt to relate a set of shapes with deep functional dictionaries, resulting in a co-segmentation. However, these dictionaries are learned offline, for individual shapes. Therefore, their model cannot dynamically determine the segmentation for a set of shapes. CoSetNet is split into an offline part which is transferrable across different shape sets, and a runtime co-segmentation network which is learned for a specific input set.

3. Overview

Our method works with point-set 3D shapes and formulates shape segmentation as a point labeling problem. The network has a two-stage architecture; see Figure 2.

Part refinement network. Given a point cloud, the network takes as input a noisy binary labeling, with the foreground representing a semantic part, and outputs a refined labeling that exactly segments out that part. To train this network, we employ the ComplementMe dataset [27] which is a subset of ShapeNet [2] and provides semantic part segmentation. The 3D models are point sampled and each part of a 3D model is used to generate a binary labeling. For each binary labeling, some random noise is added; the part refinement network is trained to denoise these binary labelings. Essentially, the part refinement network learns what a valid part looks like through training on a labeling denoising task. Meanwhile, it also learns a multi-scale and part-aware shape feature at each point, which can be used later in the co-segmentation network.

Co-segmentation network. Given an input set of 3D shapes represented by point clouds, our co-segmentation network learns the optimal network weights through back-propagation based on a group consistency loss defined over the input set. The network outputs a $K$-way labeling for each shape, with semantic consistency, where $K$ is a user prescribed network parameter specifying an upper bound of part counts; the final part counts are determined based on the input shape set and network optimization.

The co-segmentation network is unsupervised, without any ground-truth consistent segmentations. For each part generated by the $K$-way classification, a binary segmentation is formed and fed into the pre-trained part refinement network: (1) to compute a refined $K$-part segmentation, and (2) to extract a part-aware feature for each point. These together form a part feature for each segment. The corresponding part features with the same label for all shapes in the set constitute a part feature matrix. Then, weights of the co-segmentation network are optimized with the objective to maximize the part feature similarity within one label and minimize the similarity across different labels. This amounts to minimizing the rank of the part feature matrix for each semantic label while maximizing the rank of the joint part feature matrix for two semantic labels.

4. Method

In this section, we describe CoSegNet in details. The first, offline stage learns a weak regularizing prior on plausible shape parts. This stage takes a large, diverse repository of shapes, with generally inconsistent, unlabeled segmentations, as input. A part refinement network is trained on this dataset, to refine any proposed part to better resemble observed ones. The second, runtime stage jointly analyzes a collection of test shapes using a co-segmentation network that iteratively proposes (at most) $K$-way segmentations of each shape in order to optimize a consistency score.

4.1. Part Refinement Network

Dataset. In offline pre-training, we want to learn a general model to denoise all plausible part shapes at all granularities, using off-the-shelf data available in large quantities. This weak prior will be used to regularize any consistent segmentation of test shapes. Repositories with standard labeled segmentations [34, 36] are both limited in size and fixed at single pre-decided granularities. Instead, we use the 3D part dataset developed for ComplementMe [27].

This dataset, a subset of ShapeNet [2], exploits the fact that shapes in existing 3D repositories already have basic component structure, since artists designed them modularly. However, the segmentations are inconsistent: while a chair back may be an isolated part in one shape, the back and seat may be combined into a single part in another. ComplementMe does some basic heuristic-based merging of adjacent parts to eliminate very small parts from the collection, but otherwise leaves noisy part structures untouched. Further, the parts lack labels – while some tags may be present in the input shapes, we ignore them since the text is generally inconsistent and often semantically meaningless. Hence, this dataset is an excellent example of the weakly-supervised training data we can expect in a real-life situation. Our method trains a denoising prior on this noisy dataset, which will be used to refine consistent segmentations proposed in our co-segmentation stage.

Network architecture. The part refinement network learns to denoise an imperfectly segmented part, e.g. by refining its boundaries. Our architecture is based on components of PointNet++ [22]. The input to the network is a 3D shape represented as a point cloud $S$. The points belonging to the proposed part constitute the foreground $F \subset S$, the remaining points are the background $B = S \setminus F$. The output of the network is a probability for each point $q \in S$, such that the high probability points collectively define the ideal,
KNN Average \times Noisy

The architecture of our network is shown in Figure 3. The point cloud is processed by the multi-scale grouping (MSG) and multi-resolution grouping (MRG) modules of PointNet++, to produce two context-sensitive 128D feature vectors \( f_{MSG}(q) \) and \( f_{MRG}(q) \) for each point \( q \in S \). The MSG module captures the context of a point at multiple scales, by concatenating features over larger and larger neighborhoods. The MRG module computes a similar multi-scale feature, but (half of) the features of a large neighborhood are computed recursively, from the features of the next smaller neighborhood; see [22] for details.

We average the MSG features of foreground points to obtain a robust descriptor \( f_{fg} \), which is concatenated with the MRG feature of each point to produce \( [f_{MRG}(q), f_{fg}] \) pairs. The pairs are fed to a binary classifier with ReLU activation, where the output of the classifier indicates the “clean” foreground and background.

Training. The part refinement network is trained with single parts from the inconsistently segmented dataset. We add noise to each part (foreground) by randomly including some background points and excluding some foreground points (\( \sim 20-30\% \)), to simulate imperfect segmentation. The network takes the noisy part as input and tries to output the clean part’s indicator function, using a negative log-likelihood loss and Adam [13] to optimize its weights.

4.2. Co-segmentation Network

The runtime stage of our pipeline jointly segments a set of unsegmented test shapes \( T = \{ S_1, S_2, \ldots, S_N \} \) to maximize consistency between the segmented parts. To this end, we design a deep neural network that takes a shape’s point cloud as input and outputs a \( K \)-way segmentation; \( K \) is a user-specified hyperparameter specifying the part count. These outputs are compared across the test set to ensure geometric consistency of corresponding segments: our quantitative metric for this is a group consistency energy, which is used as a loss function to iteratively refine the output of the network using back-propagation.

Note that although we use a deep network to output per-shape segmentation maps, the trained network is not expected to generalize to new shape sets. Hence, the network performs essentially an unsupervised \( K \)-way clustering of the input points across all test shapes. Apart from the consistency loss, the network is guided by the offline prior that has learned to denoise plausible parts of various sizes, but has no notion of consistency or desired granularity.

Network architecture. Our co-segmentation architecture is shown in Figure 4. The network takes a minibatch of test shapes as input. The first part of the network is a classifier that independently assigns one of \( K \) abstract labels \( \{ L_1, L_2, \ldots, L_K \} \) to each point in each shape, with shared weights: the set of points in a shape with label \( L_i \) defines a single part with that label. Since the classifier output may be noisy, we pass the binary foreground/background map corresponding to each such part through the pre-trained (and frozen) offline denoising network (Section 4.1) and then re-compose these maps into a \( K \)-way map using a \( K \)-way softmax at each point to resolve overlaps. The recomposed output is the final (eventually consistent) segmentation.

The subsequent stages of the network are deterministic and have no trainable parameters: they are used to compute the group consistency energy. First, the MSG features [22] of the foreground points for each part are max-pooled to yield a part descriptor (we found max pooling to work better than average pooling). If the segmentation is consistent across shapes, all parts with a given label \( L_i \) should have similar descriptors. Therefore, we stack the descriptors for all parts with this label from all shapes in a matrix \( M_i \), one per row, and try to minimize its second singular value, a proxy for its rank (low rank = more consistent). Also, parts with different labels should be distinct, so the union of the rows of matrices \( M_i \) and \( M_j \neq i \) should have high rank. This time, we want to maximize the second singular value of \( \text{concat}(M_i, M_j) \), where the \( \text{concat} \) function constructs a new matrix with the union of the rows of its
Figure 5. Given a set of input point clouds, we construct a part similarity matrix for each abstract part label, based on the part features extracted for all shapes.

inputs. The overall energy function is:

$$E_{\text{coseg}} = 1 + \max_{i \in [1,K]} \text{rank}(M_i) - \min_{i \neq j \in [1,K]} \text{rank}(\text{concat}(M_i, M_j)),$$

where the \text{rank} function is the second singular value, computed by a (rather expensive) SVD decomposition [38]. As this energy is optimized by gradient descent, the initial layers of the network learn to propose more and more consistent segmentations across the test dataset. Additionally, we found that gaps between segments of a shape appeared frequently and noticeably before re-composition, and were resolved arbitrarily with the subsequent softmax. Hence, we add a second energy term that penalizes such gaps; see more details in the supplementary material.

Because the co-segmentation network has no access to ground truth and relies only on a weak geometry denoising prior, the consistency energy is the principal high-level influence on the final segmentation. We experimented with different ways to define this energy, and settled on SVD-based rank approximation as the best one. Note that the SVD operation makes this a technically non-decomposable loss, which usually needs special care to optimize [12]. However, consistency is in general a transitive property (even though its converse, inconsistency, is not). Hence, enforcing consistency over each of several overlapping batches is sufficient to ensure consistency over their union, and we can refine the segmentation maps iteratively using standard stochastic gradient descent.

Table 1. Dataset for training the part refinement network. For each category, we list the total number of shapes (#S) and parts (#P).

|   | Airplane | Bicycle | Car | Chair | Lamp | Table |
|---|----------|---------|-----|-------|------|-------|
| #S| 2,410    | 49      | 976 | 2,096 | 862  | 1,976 |
| #P| 9,134    | 299     | 5,119 | 9,433 | 3,296 | 6,608 |

Figure 6. High degrees of inconsistencies exist in the shape segmentations available from the ComplementMe dataset [27]. Top figure displays distribution of part counts in each object category, showing their diversity. Bottom shows several shapes, within the same category and having the same part counts (3 parts for the airplanes and 4 parts for the chairs), that exhibit much structural and geometric variation in their segmentations.

5. Results and Evaluations

We validate the two stages of CoSegNet through qualitative and quantitative evaluations, and compare our work with some state-of-the-art shape co-segmentation methods. We train our part refinement network on the shape part dataset from ComplementMe [27], which is a subset of ShapeNet [2], and test our method with the ShapeNet [36] and COSEG [34] semantic part datasets. We also manually labeled some small groups (6-12 shapes per group) of shapes from ShapeNet [36] to form a co-segmentation benchmark for quantitative evaluation.

5.1. Discriminative power of matrix rank features

We made a low rank assumption for the features of corresponding parts in the design of our co-segmentation network. That is: the MSG feature vectors of similar parts form a low rank matrix, while those dissimilar parts form a high rank matrix, where rank is estimated in a continuous way as the magnitude of the second singular value. To show
that this is a discriminative metric, we use the ShapeNet semantic part dataset [36], which has a consistent label for each part, as test data. The chair category for this dataset has four labels: back, seat, arm and leg. For each of the 14 (4 \choose 1) + (4 \choose 2) + (4 \choose 3) proper subsets of labels, we randomly sample a collection of 200 parts with labels from this subset. Our hypothesis is that matrix rank should make it easy to distinguish between collections with few distinct labels, and collections with many distinct labels. Figure 7 (right) plots the number of distinct labels in the part collection, vs increasing estimated rank (second singular value). As we can see, all part collections with a single label have a lower score than those with two labels, which in turn are all lower than those with 3 labels. In contrast, a naive variance metric such as mean squared error, as shown in Figure 7 (left), cannot correctly discriminate between part collections with 2 and 3 labels. We conclude that our rank-based metric accurately reflects consistency of a part collection.

5.2. Control, input dependency, and generalization

Our co-segmentation network is not strongly supervised with consistently segmented and labeled training data, unlike most prior deep networks for shape segmentation. Instead, the weakly-supervised part prior allows a fair amount of input-dependent flexibility in what the actual co-segmentation looks like. First, we can generate test set segmentations with different granularities, controlled by the cardinality bound $K$. Figure 8 shows co-segmentation of the same shapes for different values of $K$. In these examples, our method fortuitously produces coarse-to-fine part hierarchies. However, this nesting structure is not guaranteed by the method, and we leave this as future work.

Further, even for a fixed $K$, different test shape collections can induce different co-segmentations. Figure 1 shows co-segmentations of two different chair collections, both with $K = 4$. The collection on the left has several chairs with arms: hence, the optimization detects arms as one of the prominent parts and groups all chair legs into a single segment. The other collection has no arms, hence the four part types are assigned to back, seat, front, and back legs.

We present two further experiments to show that our part refinement network, trained on an unannotated offline database, is only a weak regularizer for co-segmentation. First, we show that reasonable co-segmentation results are obtained for a single category even when the offline training set includes shapes from multiple categories. Figure 9 (left) shows chairs co-segmented with the part refinement prior trained on both chairs and tables. Second, we show that weak denoising priors trained on one category can guide co-segmentation of another category. Figure 9 (right) shows tables co-segmented with the part refinement network trained only on chairs.

5.3. Quantitative evaluation

Since our method will produce co-segmentation results with different granularities, it is difficult to compare our results with a fixed segmentation ground truth, e.g. COSEG [34]. We adopt the following approach: first, we set $K$ to be the total number of ground truth labels for a shape category. Second, after segmentation, we manually map our abstract labels $\{L_1, L_2, \ldots, L_K\}$ to the semantic labels (arm, back, wing etc) present in the ground truth, us-
ing visual inspection of a few example shapes (this step could be easily automated, but that would not affect the overall argument). Now we can apply the standard Rand Index metric [4] for segmentation accuracy:

\[
RI = 1 - \left( \frac{2}{N} \right)^{-1} \sum_{i<j} (C_{ij}P_{ij} + (1 - C_{ij})(1 - P_{ij}))
\]

where \( i, j \) are different points of the input point cloud. \( C_{ij} = 1 \) if \( i \) and \( j \) have the same predicted label, and \( P_{ij} = 1 \) if they have the same ground truth label. A lower Rand Index implies a better match with the ground truth.

In Table 2, we compare the Rand Index scores of our method vs prior work [26, 6, 25]. Since our method trains category-specific weak priors by default, we evaluate on those categories of COSEG that are also present in the ComplementMe component dataset. Our method works natively with point clouds, whereas the three prior methods all have access to the original mesh data. Even so, we demonstrate the greatest overall accuracy (lowest RI).

To demonstrate that our co-segmentation does not rely on the initial segmentation of training data for our part refinement network, we present a quantitative consistency evaluation between the initial segmentation and our co-segmentation results on a subset of our training data, the ground truth of this evaluation is labeled by experts. In Table 3, we found that CoSegNet can improve the segmentation quality even on its own training data. More visual results can be found in supplemental material.

5.4. Ablation studies

We explore the effect of our design choices for our method through several ablation studies and show some results in Figure 11. These design choices include:

- No part refinement: Remove the part refinement module block in Figure 4 and connect the \( K \)-way classifier with point feature encoder directly.
Table 2. Rand Index scores for our method vs. prior works. With the exception of the vase category, CoSegNet performs the best. The hand-crafted features from Sidi et al. [26] prove to be best suited to this particular shape category.

| Category | CoSegNet | Shu [25] | Hu [6] | Sidi [26] |
|----------|----------|----------|--------|-----------|
| Chair    | 0.055    | 0.076    | 0.121  | 0.135     |
| Lamp     | 0.059    | 0.069    | 0.103  | 0.092     |
| Vase     | 0.189    | 0.198    | 0.230  | **0.102** |
| Guitar   | **0.032**| 0.041    | 0.037  | 0.081     |

Table 3. Rand Index score comparison between segmentations in training data (GT) and co-segmentation results by CoSegNet. Our CoSegNet improves consistency even in its own training data. Visual results can be found in the supplemental material.

|          | Chair | Table | Bicycle | Lamp | Car | Plane |
|----------|-------|-------|---------|------|-----|-------|
| GT       | 0.21  | 0.27  | 0.31    | 0.18 | 0.38| 0.24  |
| Ours     | **0.09** | **0.14** | **0.22** | **0.16** | **0.27** | **0.13** |

6. Conclusion, limitation, and future work

We present CoSegNet, a deep learning framework for shape co-segmentation. A novel feature of our method is that no ground truth consistent co-segmentations are needed to train our network. Consistent co-segmentations are learned by iteratively minimizing a group consistency loss via backpropagation over a deep neural network. The only supervision is applied to denoise part proposals on an individual shape basis, which can be trained using existing segmented shape datasets such as [27], with inconsistent segmentations. Experiments demonstrate the ability of CoSegNet to produce consistent co-segmentations amid large degrees of geometric and structural variations in the input sets, superior results over state-of-the-art methods, as well as co-segmentations at varying degrees of granularity.

Perhaps the most critical limitation of our current co-segmentation network is that it is not designed to generalize to new inputs. This is by design: the network weights are derived to minimize the loss function for the current input set and they are recomputed for each new set. That being said, we did look into whether learned weights for one set could serve as a good starting point for a next set to save optimization time. Results reveal that this is true when the new set is similar to the previous one. But when this is not the case, the network may be stuck in a local minima and the resulting co-segmentations are not meaningful; this is an expected outcome of gradient descent.

Another limitation is that our part refinement network is not trained over all object categories, which would have been ideal. Our current network appears capable of handling intra-category variations, but learning parts and their feature descriptions when all the object categories are mixed together is significantly more challenging.

In future work, we plan to extend our weakly supervised learning framework for cross-category part learning. We would also like to explore co-segmentation via online learning, which represents a family of machine learning algorithms that learn to update models incrementally from sequentially input data streams [24, 9]. In contrast, our current co-segmentation network does not really learn a generalizable model, and the learned network weights cannot be continuously updated as new shapes come in. An online learned model for unsupervised co-segmentation may need to create and maintain multiple segmentation templates.
References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. 1

[2] 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, 3, 5

[3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016. 1

[4] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. In *Trans. Graph.*, volume 28, page 73, 2009. 7

[5] A. Golovinskiy and T. Funkhouser. Consistent segmentation of 3D models. *Computers & Graphics*, 33(3):262–269, 2009. 1, 2

[6] R. Hu, L. Fan, and L. Liu. Co-segmentation of 3D shapes via subspace clustering. *Computer Graphics Forum*, 31(5):1703–1713, 2012. 1, 2, 7, 8

[7] Q. Huang, V. Koltun, and L. Guibas. Joint shape segmentation with linear programming. *ACM transactions on graphics (TOG)*, 30(6):125, 2011. 2

[8] 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

[9] R. Jin, S. C. Hoi, and T. Yang. Online multiple kernel learning: Algorithms and mistake bounds. In *International conference on algorithmic learning theory*, pages 390–404. Springer, 2010. 8

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

[11] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D mesh segmentation and labeling. *Trans. Graph. (SIGGRAPH)*, 29(3), 2010. 2

[12] P. Kar, H. Narasimhan, and P. Jain. Online and stochastic gradient methods for non-decomposable loss functions. In *NeurIPS*, 2014. 5

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[14] 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

[15] W. Li, O. H. Jafari, and C. Rother. Deep object co-segmentation. *arXiv preprint arXiv:1804.06423*, 2018. 2

[16] G. Lin, A. Milan, C. Shen, and I. D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. *CoRR*, abs/1611.06612, 2016. 1

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 1

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[19] 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, 2013. 1

[20] S. Muralikrishnan, V. G. Kim, and S. Chaudhuri. Tags2parts: Discovering semantic regions from shape tags. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2926–2935, 2018. 3

[21] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel matters - improve semantic segmentation by global convolutional network. *CoRR*, abs/1703.02719, 2017. 1

[22] 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, 3, 4

[23] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3D representations at high resolutions. In *CVPR*, volume 3, 2017. 1, 2

[24] S. Shalev-Shwartz and Y. Singer. Online learning: Theory, algorithms, and applications. 2007. 8

[25] Z. Shu, C. Qi, S. Xin, C. Hu, L. Wang, Y. Zhang, and L. Liu. Unsupervised 3D shape segmentation and co-segmentation via deep learning. *Computer Aided Geometric Design*, 43:39–52, 2016. 2, 7, 8

[26] 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. *Trans. Graph. (SIGGRAPH Asia)*, 30(6), 2011. 1, 2, 7, 8

[27] M. Sung, H. Su, V. G. Kim, S. Chaudhuri, and L. Guibas. ComplementMe: Weakly-supervised component suggestions for 3D modeling. *Trans. Graph. (SIGGRAPH Asia)*, 2017. 1, 2, 3, 5, 8

[28] M. Sung, H. Su, R. Yu, and L. Guibas. Deep functional dictionaries: Learning consistent semantic structures on 3D models from functions. *arXiv preprint arXiv:1805.09957*, 2018. 3

[29] O. van Kaick, K. Xu, H. Zhang, Y. Wang, S. Sun, A. Shamir, and D. Cohen-Or. Co-hierarchical analysis of shape structures. *Trans. Graph.*, 32(4):Article 69, 2013. 1

[30] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2217–2224. IEEE, 2011. 2

[31] F. Wang, Q. Huang, and L. Guibas. Image co-segmentation via consistent functional maps. In *CVPR*, 2013. 2

[32] 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

[33] 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*, 36(4):72, 2017. 1, 2

[34] Y. Wang, S. Asafi, O. Van Kaick, H. Zhang, D. Cohen-Or, and B. Chen. Active co-analysis of a set of shapes. *ACM Transactions on Graphics, 31(6):165*, 2012. 1, 2, 3, 5, 6

[35] K. Xu, H. Li, H. Zhang, D. Cohen-Or, Y. Xiong, and Z.-Q. Cheng. Style-content separation by anisotropic part scales. *ACM Transactions on Graphics (TOG)*, 29(6):184, 2010. 2
[36] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, et al. A scalable active framework for region annotation in 3D shape collections. ACM Transactions on Graphics, 35(6):210, 2016. 1, 2, 3, 5, 6

[37] L. Yi, H. Su, X. Guo, and L. J. Guibas. Syncspeccnn: Synchronized spectral cnn for 3D shape segmentation. In CVPR, pages 6584–6592, 2017. 1, 2

[38] R. Yi, C. Zhu, P. Tan, and S. Lin. Faces as lighting probes via unsupervised deep highlight extraction. In ECCV, 2018. 5