AdaCoSeg: Adaptive Shape Co-Segmentation with Group Consistency Loss

Chenyang Zhu[1,2]   Kai Xu[2]   Siddhartha Chaudhuri[3,4]   Li Yi[5]   Leonidas Guibas[6]   Hao Zhang[1]

[1]Simon Fraser University   [2]National University of Defense Technology
[3]Adobe Research   [4]IIT Bombay   [5]Google Research   [6]Stanford University

Abstract

We introduce AdaCoSeg, a deep neural network architecture for adaptive co-segmentation *of a set of 3D shapes represented as point clouds. Differently from the familiar single-instance segmentation problem, co-segmentation is intrinsically* contextual*: how a shape is segmented can vary depending on the set it is in. Hence, our network features an adaptive learning module to produce a consistent shape segmentation which adapts to a set. Specifically, given an input set of unsegmented shapes, we first employ an offline pre-trained* part prior network *to propose per-shape parts. Then, the co-segmentation network iteratively and jointly optimizes the part labelings across the set subjected to a novel* group consistency loss *defined by matrix ranks. While the part prior network can be trained with noisy and inconsistently segmented shapes, the final output of AdaCoSeg is a consistent part labeling for the input set, with each shape segmented into up to (a user-specified) K parts. Overall, our method is* weakly supervised*, producing segmentations tailored to the test set, without consistent ground-truth segmentations. We show qualitative and quantitative results from AdaCoSeg 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 [1,3] and shape segmentations [11,33]. *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., chairs or airplanes.

The goal of co-segmentation is to compute a *consistent* segmentation for all shapes in the input collection. The consistency of the segmentation implies a correspondence between all the segmented parts, which is a critical requirement for knowledge and attribute transfer, collecting
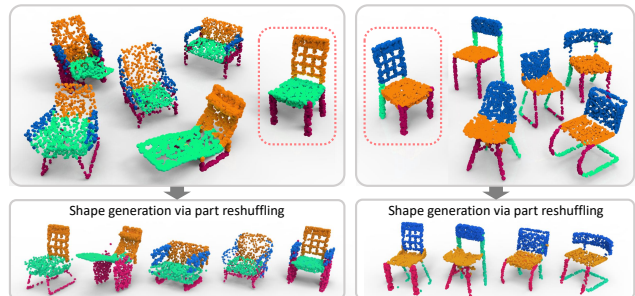

Figure 1. Our adaptive shape co-segmentation network, AdaCoSeg, produces structurally different segmentations (here up to 4 parts) for two sets of chairs — one with armrests, one without. For each set, the segmentations are semantically consistent, allowing shape generation via part reshuffling. However, the same shape can be segmented differently depending on its containing set (see the circled chair), showing the method's adaptivity.

statistics over a dataset, and structure-aware shape modeling [18]. Figure 1 shows such a modeling example based on part reshuffling induced by a co-segmentation.

In contrast to the familiar single-instance segmentation problem, a distinctive feature of co-segmentation is that it is inherently *contextual*. As dictated by the consistency criterion, the same shape may be segmented differently depending on which input set it belongs to; see Figure 1. From this perspective, the input shape collection serves both as the test set *and* the training set. Ideally, the co-segmentation network can *quickly adapt* to a new input set without expensive retraining. Such an adaptive network would change its behavior, i.e., the network weights, at the time it is run. This is different from the traditional label learning paradigm, where the trained model strives to generalize to new inputs without changing the network weights, either under the supervised [11,20] or weakly supervised settings [5,19,26].

In this paper, we introduce a deep neural network for shape co-segmentation, coined AdaCoSeg, which is designed to be adaptive. AdaCoSeg takes as input a set of unsegmented shapes represented as point clouds, proposes *per-shape* parts in the first stage, and then *jointly optimizes* the parts subject to a novel *group consistency loss* defined by *matrix rank estimates* for the specific input set. 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 is unsupervised, guided by the group consistency loss, we found that the results can be improved by adding a weak regularizing prior to boost the part proposal. Specifically, we pre-train a *part prior 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 part. The part prior network is similar to the pairwise potential of a conditional random field (CRF) in traditional segmentation [12]: while it is not a general prior, as 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 part counts and part compositions within the same object category can be highly inconsistent. No segment label is necessary: the model is label-agnostic.

Overall, our method is *weakly supervised*, since it produces consistent segmentations without consistent ground-truth segmentations. It consists of an offline, supervised part prior network, which is trained once on inconsistently segmented, unlabeled shapes, and a "runtime", adaptive 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 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 AdaCoSeg to produce consistent co-segmentations at varying levels of granularity; see Figure 7.

Our part prior network is trained using the dataset from ComplementMe [25]; the adaptive co-segmentation is unsupervised. For evaluation only, we also adopt two datasets [30, 32] containing ground truth co-segmentations. While offline training required up to 20 hours to complete, it takes about 7 minutes to co-segment 20 shapes at a resolution of 2,048 points per shape. We show qualitative and quantitative results from AdaCoSeg and evaluate it through ablation studies and comparisons with state-of-the-art co-segmentation methods. Our main contributions include:

- The first DNN for adaptive shape co-segmentation.
- A novel and effective group consistency loss based on low-rank approximations.
- A co-segmentation training framework that needs no ground-truth consistent segmentation labels.

2. Related work

Deep learning for shape segmentation. Deep models for supervised shape segmentation have been developed for
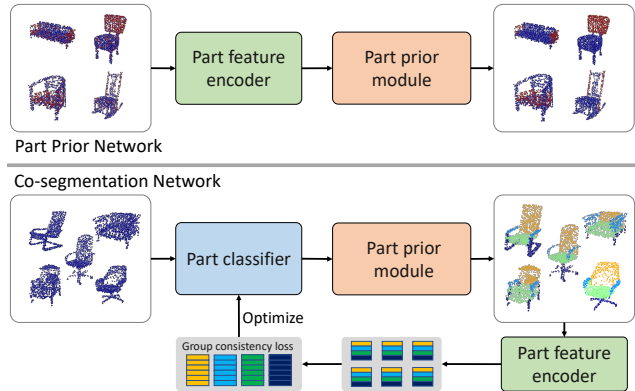

Figure 2. AdaCoSeg consists of a part prior network (top) and a co-segmentation network (bottom). The part feature encoder and part prior module in the first network learn a weak regularizing prior to denoise proposed part shapes. The co-segmentation network is trained with a novel group consistency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

various representations, such as voxel grids [21, 29], point clouds [9, 15, 20], multi-view projections [11], and surface meshes [28, 33]. The key is to replace hand-crafted features employed in traditional methods by features learned from data. However, these models are mostly trained to target a *fixed* set of semantic labels. The resulting segmentation for a given shape is also fixed and cannot be adaptive to the context of a shape set, a key feature of co-segmentation. Relatively few works study deep learning for unsupervised shape segmentation [5, 23].

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 [27]. Most works formulate this problem as a multi-image Markov Random Field (MRF), with a foreground consistency constraint. Recently, Li et al. [16] 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.

Shape co-segmentation. Extensive research has been devoted to co-analysis of sets of shapes [6, 7, 8, 24, 30, 31]. 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.

Recently, deep learning based approaches are emerging. Shu et al. [23] use deep auto-encoders for per-part feature learning. However, their co-segmentation module does not use a deep network and it strictly constrains the final

segmentations to parts learned in the first stage. In contrast, AdaCoSeg does not strictly adhere to proposals by the part prior network, as the consistency loss can impact and adjust part labeling. Muralikrishnan et al. [19] propose a weakly-supervised method for tag-driven 3D shape co-segmentation, but their model is trained to target a predefined label set. Sung et al. [26] 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, so their model cannot adaptively co-segment a set of shapes. In contrast, CoSetNet is split into an offline part which is transferrable across different shape sets, and an online, adaptive co-segmentation network which is learned for a specific input set.

In concurrent work, Chen et al. [5] present a branched autoencoder for weakly supervised shape co-segmentation. The key difference is that BAE-NET is essentially a more advanced part prior network, with each branch tasked to learn a simple representation for one universal part of an input shape collection; there is no explicit optimization for group consistency. As a result, BAE-NET tends to underperform comapred to AdaCoSeg on small input sets and in the presence of large part discrepancies; see Figure 11.

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 prior network. The network takes as input a point cloud with noisy binary labeling, where the foreground represents an imperfect part, and outputs a regularized labeling leading to a refined part. To train the network, we employ the ComplementMe dataset [25], a subset of ShapeNet [2], which provides semantic part segmentation. The 3D shapes are point sampled, with each shape part implying a binary labeling. For each binary labeling, some random noise is added; the part prior network is trained to denoise these binary labelings. Essentially, the part prior 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 backpropagation 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 prior 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

The offline stage of AdaCoSeg learns a weak regularizing prior for plausible shape parts, where a part prior network is trained on a large, diverse shape repository with generally inconsistent, unlabeled segmentations. The network serves to refine any proposed parts to better resemble observed ones. The runtime stage jointly analyzes a set of test shapes using a co-segmentation network that iteratively proposes (at most) K-way segmentations of each shape to optimize a group consistency score over the test set.

4.1. Part Prior 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 [30, 32] are both limited in size and fixed at single pre-decided granularities. Instead, we use the 3D part dataset developed for ComplementMe [25].

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.
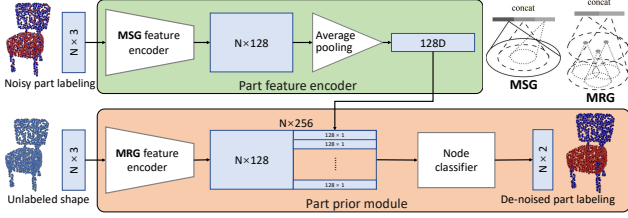
Figure 3. The architecture of the part prior network. The network encodes a shape with noisy part labeling and the whole shape, using the MSG and MRG feature encoders from PointNet++ [20], respectively. It is trained to denoise the input binary labeling and output a clean labeling, indicating a plausible part.

Network architecture. The part prior network learns to denoise an imperfectly segmented part, using an architecture based on components from PointNet++ [20]. The input to the network is a 3D point cloud shape S. Points belonging to the proposed part constitute the foreground $F \subset S$, while 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, "clean" part that best matches the proposed part, thereby denoising the noisy foreground.

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 128-D feature vectors $f_{\mathrm{MSG}}(q)$ and $f_{\mathrm{MRG}}(q)$ 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 [20] 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 $[f_{\mathrm{MRG}}(q), f_{\mathrm{fg}}]$ pairs. The pairs are fed to a binary classifier with ReLU activation, where the output of the classifier indicates the "cleaned" foreground and background.

Training. The part prior network is trained with single parts from the inconsistently segmented dataset. We add noise to each part (foreground) by randomly inserting some background points and excluding some foreground points ($\sim$20-30%). The network takes noisy parts as input and tries to output clean part indicator functions, using a negative log-likelihood loss and Adam [14] optimizer.

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 recompose 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 [20] 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 $concat(M_i, M_j)$, where the $concat$ function constructs a new matrix with the union of the rows of its inputs. The overall energy function is:

$$\mathcal{E}_{\mathrm{coseg}} = 1 + \max_{i \in \{1,2,\ldots,K\}} rank(M_i)$$
$$- \min_{i,j \in \{1,2,\ldots,K\}, i \neq j} rank\left(concat(M_i, M_j)\right),$$

where the $rank$ function is the second singular value, computed by a (rather expensive) SVD decomposition [34].
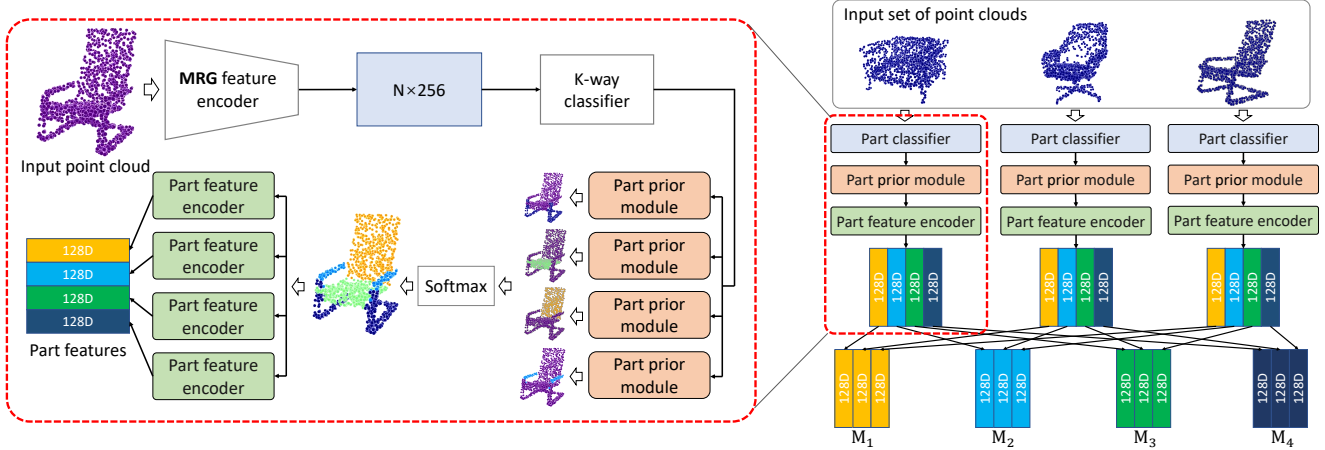
Figure 4. Left: Given an input point cloud, the K-way classifier segments it into K parts. These parts are then refined by the part prior module, resulting in a refined K-way segmentation of the input point cloud. After that, the part feature encoder is used to extract features for each refined part. Right: 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.

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 [13]. 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.

5. Results and Evaluations

We validate the two stages of AdaCoSeg through qualitative and quantitative evaluation, and compare to state-of-the-art methods. We train our part prior network on the shape part dataset from ComplementMe [25], which is a subset of ShapeNet [2], and test our method with the ShapeNet [32] and COSEG [30] semantic part datasets. We also manually labeled some small groups (6-12 shapes per group) of shapes from ShapeNet [32] to form a co-segmentation benchmark for quantitative evaluation.
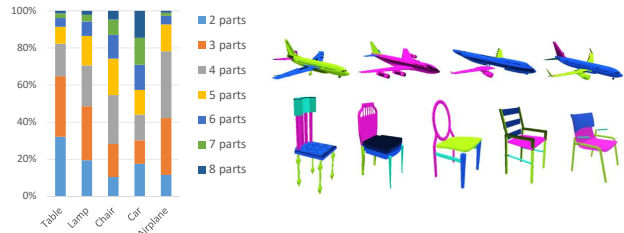

Figure 5. High degrees of inconsistencies exist in the shape segmentations available in the ComplementMe dataset [25]. The left figure charts the distribution of part counts in each object category, showing their diversity. The right figure shows several shapes, within the same category and having the same part counts (3 parts for airplanes, 4 parts for chairs), that exhibit much structural and geometric variation in their segmentations.

Table 1. Dataset for training the part prior network. For each category, we list the shape count (#S) and part count (#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

Discriminative power of matrix ranks. Our network design makes a low-rank assumption for the features of corresponding shape parts: the MSG feature vectors of similar parts form a low-rank matrix, while those dissimilar parts form a higher-rank matrix, where rank is estimated in a continuous way as the magnitude of the second singular value. To show that matrix ranks provide a discriminative metric, we use the ShapeNet semantic part dataset [32], 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*. From each of the 14 ($= \binom{4}{1} + \binom{4}{2} + \binom{4}{3}$) non-empty
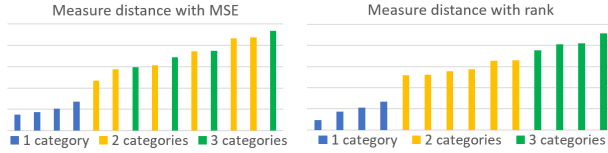
Figure 6. Number of distinct labels in a collection of parts (Y axis) vs increasing feature variation for that collection (X axis). The plot on the right uses the more discriminative matrix rank-based score, whereas the plot on the left uses MSE which cannot tell 2 and 3-label collections apart.

proper subsets of labels, we randomly sample a collection of 200 labeled parts. 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 6 (right) plots the number of distinct labels in the part collection, vs increasing rank estimates. 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 6 (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.

Control, adaptivity, and generalization. AdaCoSeg 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 7 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.

Quantitative evaluation. Since AdaCoSeg produces segmentations with varying granularity, it is difficult to compare its results to a fixed ground truth segmentation, e.g., [30]. We adopt the following strategy. 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
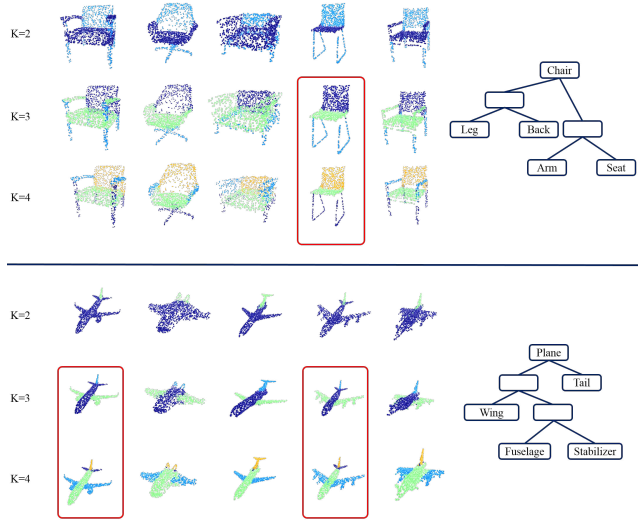

Figure 7. Coarse-to-fine co-segmentations of the same input shapes, generated by setting $K = 2, 3, 4$. The actual part count discovered per shape is adaptively selected and need not be exactly K, as shown in the examples bounded in red.

(*arm*, *back*, *wing* etc) present in the ground truth, using visual inspection of a few example shapes (this step could be automated, but it would not affect the overall argument). Now we can apply the standard Rand Index metric [4] for segmentation accuracy:

$$RI = 1 - \binom{2}{N}^{-1} \sum_{i<j} (C_{ij} P_{ij} + (1 - C_{ij})(1 - P_{ij}))$$

where i, j are different points of the input point cloud. $C_{ij} = 1$ iff i and j have the same predicted label, and $P_{ij} = 1$ iff they have the same ground truth label. A lower Rand Index implies a better match with the ground truth. Note that the main advantage of RI over IOU is that it computes segmentation overlap without needing segment correspondence. This makes it particularly suited for evaluating co-segmentation where the focus is on segmentation consistency without knowing part labeling or correspondence.

In Table 2, we compare the Rand Index scores of our method vs prior work [7, 23, 24]. 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 AdaCoSeg does not rely on the initial training segmentations for the part prior network, we present a quantitative consistency evaluation between the initial segmentations and our co-segmentation results on a subset of our training data; the ground truth of this evaluation is labeled by experts. Table 3 shows that AdaCoSeg
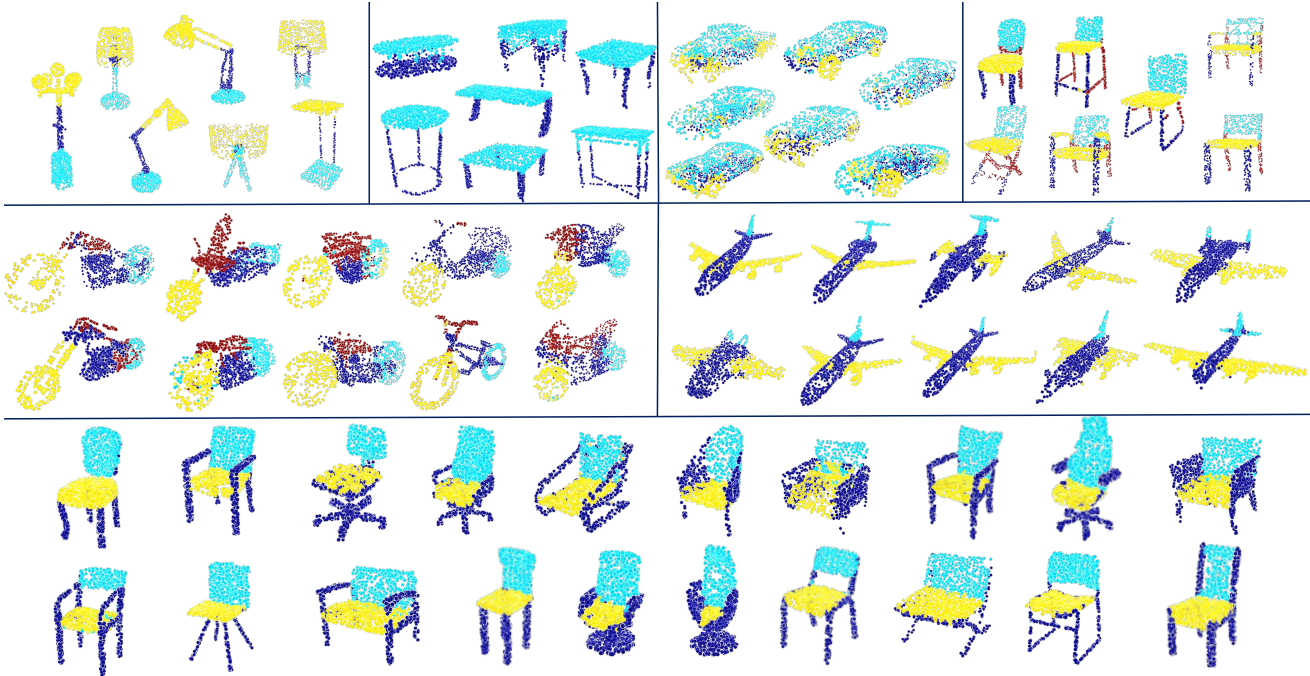
Figure 8. A gallery of co-segmentation results obtained by AdaCoSeg, for all the six object categories from the ComplementMe dataset. The input sets vary in size from 7 to 10. More results can be found in the supplementary material.
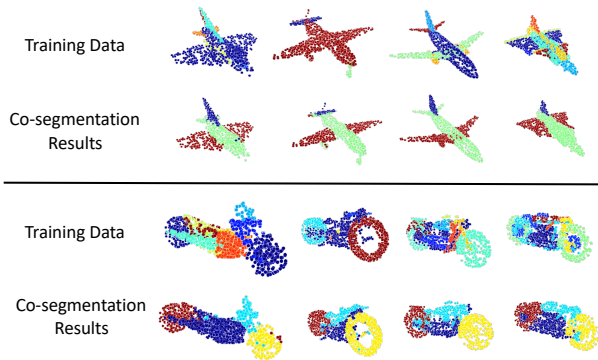

Figure 9. Co-segmentation results obtained by AdaCoSeg when using inconsistent training data. First and third rows show segmentations from the training data. Second and fourth rows show the co-segmentation results obtained by our network.

can even improve the segmentation quality of its own training data. Figure 9 demonstrates a significant improvement by our co-segmentation over the noisy training data. More results can be found in supplemental material.

Ablation study. We explore the effect of our design choices via several ablation studies and show some results in Figure 10. These design choices include:

- *No part prior*: Remove the part prior network and connect the K-way classifier to point feature encoder.

- *No de-noise*: No random noise is added when training of our part prior network.

- *No segmentation completeness loss*: Optimize AdaCoSeg by using only the group consistency loss.

- *No contrastive term in group consistency loss*: Only keep the second term in our loss function.

- *MSG vs. MRG for part feature encoder*: Using MRG instead of MSG for encoding each shape part.

We found that the loss cannot decrease significantly without the part prior module and the contrastive term during training. Refer to the supplemental material for visual segmentation results without the part prior. Further, the denoising is also important for training our co-segmentation network. Finally, we found that the MSG feature for the part encoder, which focuses more on local than global contexts, can achieve better performance over MRG in our task.

Comparison to BAE-NET. Figure 11 visually compares AdaCoSeg with one-shot learning of BAE-NET [5] using one perfect exemplar, on a small test set of 9 chairs; more comparison results can be found in the supplementary material. Both methods can be regarded as weakly supervised but with different supervision strategies. Our experiments show that with explicit optimization adapted to input sets, using the group consistency loss, AdaCoSeg generally outperforms BAE-NET over small test sets and in the presence of strong part discrepancies.

Category	AdaCoSeg	Shu	Hu	Sidi
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 2. Rand Index scores for AdaCoSeg vs. prior works. With the exception of the vases, AdaCoSeg performs the best. The hand-crafted features from Sidi et al. [24] prove to be best suited to the vase category.

	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**

Table 3. Rand Index score comparison between segmentations in training data (GT) and AdaCoSeg results. AdaCoSeg improves consistency even in its own training data. Visual results can be found in supplemental material.
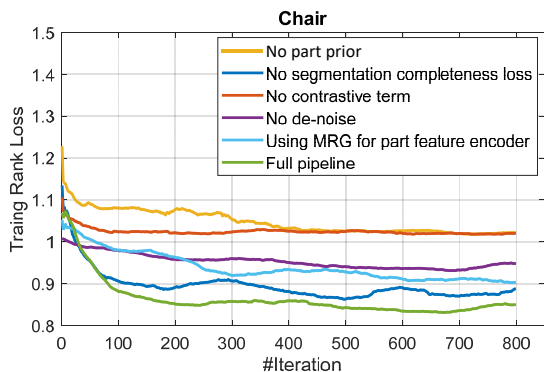

Figure 10. Training rank loss for ablation study on significant features. See supplemental material for more evaluation.
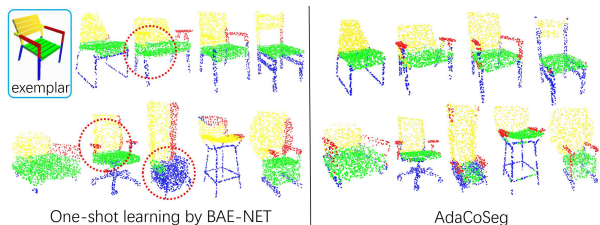

Figure 11. Comparing AdaCoSeg with BAE-NET on a small test set. AdaCoSeg, without needing any exemplars, leads to improved accuracy over BAE-NET with one exemplar.

6. Conclusion, limitation, and future work

We present AdaCoSeg, an adaptive deep learning framework for shape co-segmentation. A novel feature of our method is that beyond offline training by the part prior netowk, the online co-segmentation network is adaptive to the input set of shapes, producing a consistent co-segmentation by iteratively minimizing a group consistency loss via back-propagation over a deep network. Experiments demonstrate robustness of AdaCoSeg to large degrees of geometric and structural variations in the input sets, which is superior to state of the art.

No ground-truth consistent co-segmentations are needed to train AdaCoSeg. The offline and online stages are trained on different datasets, and for different tasks. The only supervision is at the first stage, to denoise part proposals on an individual shape basis, where the training can be carried out using existing datasets composed of inconsistent segmentations, e.g., [25]. The second optimizes a consistent segmentation on a specific test set, with the part prior as a regularizer. Our two-stage pipeline conserves computation by training the weak prior only once and reusing it across different co-segmentation tasks.

We reiterate that our online co-segmentation network does *not* generalize to new inputs, which 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. Also, AdaCoSeg is not trained end-to-end. While an end-to-end deep co-segmentation network is desirable, the challenges of developing such networks for an unsupervised problem are well known [17]. Another limitation is that our part prior network is not trained across different object categories. This would have been ideal, but per-category training is typical for most existing segmentation models [9, 15, 21, 29]. Our current network appears capable of handling some intra-category variations, but learning parts and their feature descriptions with all categories 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 [10, 22]. 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] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 39(12):2481–2495, 2017. 1

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 3, 5

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *CoRR*, abs/1606.00915, 2016. 1

[4] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. In *Trans. Graph.*, volume 28, 2009. 6

[5] Zhiqin Chen, Kangxue Yin, Matt Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-NET: Branched autoencoder for shape co-segmentation. In *ICCV*, 2019. 1, 2, 3, 7

[6] Aleksey Golovinskiy and Thomas Funkhouser. Consistent segmentation of 3D models. *Computers & Graphics*, 33(3):262–269, 2009. 2

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

[8] Qixing Huang, Vladlen Koltun, and Leonidas Guibas. Joint shape segmentation with linear programming. *Trans. Graph.*, 30(6), 2011. 2

[9] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3D segmentation of point clouds. In *CVPR*, 2018. 2, 8

[10] Rong Jin, Steven CH Hoi, and Tianbao Yang. Online multiple kernel learning: Algorithms and mistake bounds. In *Int'l Conf. on Algorithmic Learning Theory*, 2010. 8

[11] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3D shape segmentation with projective convolutional networks. In *CVPR*, 2017. 1, 2

[12] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3D mesh segmentation and labeling. *Trans. Graph. (SIGGRAPH)*, 29(3), 2010. 2

[13] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Online and stochastic gradient methods for non-decomposable loss functions. In *NeurIPS*, 2014. 5

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[15] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3D point cloud models. In *ICCV*, 2017. 2, 8

[16] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018. 2

[17] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019. 8

[18] Niloy Mitra, Michael Wand, Hao Richard Zhang, Daniel Cohen-Or, Vladimir Kim, and Qi-Xing Huang. Structure-aware shape processing. In *SIGGRAPH Asia 2013 Courses*, 2013. 1

[19] Sanjeev Muralikrishnan, Vladimir G Kim, and Siddhartha Chaudhuri. Tags2Parts: Discovering semantic regions from shape tags. In *CVPR*, 2018. 1, 3

[20] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 1, 2, 4

[21] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning deep 3D representations at high resolutions. In *CVPR*, 2017. 2, 8

[22] Shai Shalev-Shwartz and Yoram Singer. Online learning: Theory, algorithms, and applications. 2007. 8

[23] Zhenyu Shu, Chengwu Qi, Shiqing Xin, Chao Hu, Li Wang, Yu Zhang, and Ligang Liu. Unsupervised 3D shape segmentation and co-segmentation via deep learning. *Computer Aided Geometric Design*, 43:39–52, 2016. 2, 6

[24] Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *Trans. Graph. (SIGGRAPH Asia)*, 30(6), 2011. 2, 6, 8

[25] Minhyuk Sung, Hao Su, Vladimir G. Kim, Siddhartha Chaudhuri, and Leonidas Guibas. ComplementMe: Weakly-supervised component suggestions for 3D modeling. *Trans. Graph. (SIGGRAPH Asia)*, 2017. 2, 3, 5, 8

[26] Minhyuk Sung, Hao Su, Ronald Yu, and Leonidas Guibas. Deep functional dictionaries: Learning consistent semantic structures on 3D models from functions. In *NeurIPS*, 2018. 1, 3

[27] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, 2011. 2

[28] Pengyu Wang, Yuan Gan, Panpan Shui, Fenggen Yu, Yan Zhang, Songle Chen, and Zhengxing Sun. 3D shape segmentation via shape fully convolutional networks. *Computers & Graphics*, 70:128–139, 2018. 2

[29] 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*, 36(4), 2017. 2, 8

[30] Yunhai Wang, Shmulik Asafi, Oliver Van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Active co-analysis of a set of shapes. *Trans. Graph. (SIGGRAPH Asia)*, 31(6), 2012. 2, 3, 5, 6

[31] Kai Xu, Honghua Li, Hao Zhang, Daniel Cohen-Or, Yueshan Xiong, and Zhi-Quan Cheng. Style-content separation by anisotropic part scales. *Trans. Graph. (SIGGRAPH Asia)*, 29(6), 2010. 2

[32] 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. *Trans. Graph. (SIGGRAPH Asia)*, 35(6), 2016. 2, 3, 5

[33] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. In *CVPR*, 2017. 1, 2

[34] Renjiao Yi, Chenyang Zhu, Ping Tan, and Stephen Lin. Faces as lighting probes via unsupervised deep highlight extraction. In *ECCV*, 2018. 4