This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Tags2Parts: Discovering Semantic Regions from Shape Tags

Sanjeev Muralikrishnan[1] Vladimir G. Kim[2] Siddhartha Chaudhuri[1,2]

[1]IIT Bombay [2]Adobe Research

Abstract

We propose a novel method for discovering shape regions that strongly correlate with user-prescribed tags. For example, given a collection of chairs tagged as either "has armrest" or "lacks armrest", our system correctly highlights the armrest regions as the main distinctive parts between the two chair types. To obtain point-wise predictions from shape-wise tags we develop a novel neural network architecture that is trained with tag classification loss, but is designed to rely on segmentation to predict the tag. Our network is inspired by U-Net, but we replicate shallow U structures several times with new skip connections and pooling layers, and call the resulting architecture WU-Net*. We test our method on segmentation benchmarks and show that even with weak supervision of whole shape tags, our method can infer meaningful semantic regions, without ever observing shape segmentations. Further, once trained, the model can process shapes for which the tag is entirely unknown. As a bonus, our architecture is directly operational under full supervision and performs strongly on standard benchmarks. We validate our method through experiments with many variant architectures and prior baselines, and demonstrate several applications.*

1. Introduction

Online repositories contain millions of 3D shapes, providing rich data for data-driven 3D analysis and synthesis [26]. While these repositories often provide tags, textual descriptions, and soft categorization to facilitate text-based search, these labels are typically provided for the entire shape, and not at the region level. Many applications require finer shape understanding, e.g. parts and their labels are essential for assembly-based modeling interfaces. While one can obtain these labels by training a strongly supervised segmentation model [6], this level of supervision requires substantially more involved annotation interfaces and human effort, making it infeasible for massive and growing online repositories. Existing methods for discovering semantic regions without explicit supervision are typically guided by geometric cues (e.g. [5]), but they are prone to failure by being tailored to specific notions of parts, implicitly encoded by algorithm design.

Weakly- or semi-supervised methods have been proposed as a compromise between supervised and unsupervised techniques. For example, Yi et al. [27] leverage scene graph metadata in existing repositories, which provide some segments and labels for a small subset of shapes. This metadata is very sparse and specific to computer graphics models. In contrast, tags for entire shapes are abundant, often accompany scanned shapes, and are easy to crowdsource at scale.

In this work we propose a novel method for discovering regions from shape tags without explicit region-wise labeling or prior segmentation. For example, in a collection of shapes tagged as "has armrest" and "does not have armrest", we are able to identify the armrest components of the chairs in the former category (Figure 1). Further, once trained, our method can process shapes for which the tag is entirely unknown.

Our main challenge is that the weak supervisory signal (whole object tag) is different from the target output (point-wise labels). To address this, we use a neural network that jointly performs classification and segmentation, and train it for whole object tagging while relying on point-wise labels to infer the tags.

In particular, we propose a novel neural network architecture with skip connections, which we call *WU-Net* (Figure 2), inspired by the U-Net [16] architecture for *strongly*-supervised image segmentation. We make two key modifications. *First*, to regularize the network and improve localization of segments we replicate the 'U' structure thrice ('WU') and add skip connections both within *and* across them. *Second*, since the network is originally designed for strongly-supervised segmentation, we add two layers for tag classification from a hidden segmentation layer: average pooling followed by max pooling. Average pooling encourages coherence, forcing the network to train for segments that help tag classification. This network architecture is our main technical contribution.

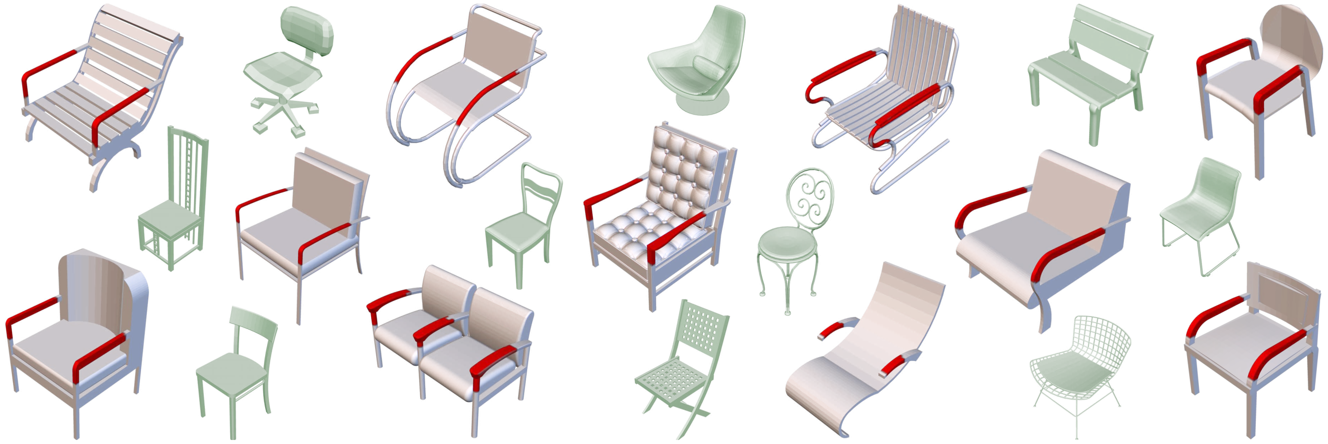


Figure 1: Chair armrests identified from only shape-level tags: “has armrest” (red/bisque) or “lacks armrest” (green). The weakly supervised problem is solved by a novel neural network which we call *WU-Net*. The highlighted red regions are the automatically generated outputs of the network, with no postprocessing except symmetrization.

To evaluate our approach, we use shapes from standard datasets [28], but withhold region labels and only tag shapes based on part presence and absence. Our method detects regions with remarkable accuracy without observing a single segmented shape. As a bonus, we observe that our approach is also suitable for strongly-supervised segmentation, and demonstrate that the architecture performs well under strong supervision. We validate our design through extensive experiments, including ablation studies, variant architectures, baseline comparisons, and prototype applications.

2. Related work

We overview related work on shape and image segmentation with various degrees of supervision.

Unsupervised shape segmentation One can leverage shape similarities and geometric cues to discover parts [3, 18, 5]. These methods encode generic part priors, but not all semantic regions conform to them. To bias unsupervised methods towards semantic regions, Yi et al. [27] leverage existing scene graphs, which are sparse and not always informative. No such method provides significant output control, which prevents discovery of user-prescribed regions.

Supervised shape segmentation A direct remedy is to use shapes with manually-labeled regions to train a model that can discover similar semantic regions in new shapes [7]. Recent methods exploit deep neural networks, based on 2D renderings [6], local descriptors after spectral alignment [30], unordered point sets [14], canonicalized meshes [9], and voxel octrees [15, 23].

The need to collect labeled data is the main bottleneck for supervised methods. Several approaches try to minimize this cost. E.g., Wang et al. [24] actively choose the next shape to label that most benefits a supervised method. Yi et al. [29] partially replace annotation with (quick) verification. These techniques still require tedious manual segmentation. Our goal is to avoid this altogether with a less taxing form of supervision, known as *weakly supervised analysis*.

Distinctive regions in shapes In prior work most relevant to ours, Shilane and Funkhouser [17] use hand-crafted local descriptors to highlight regions common to a category and different across categories. We learn such a representation via a neural network directly from a voxelized shape, and apply it to fine-grained shape segmentation within a single category. Further, unlike [17], our method directly applies to test shapes with unknown tags, since it implicitly performs classification. In evaluations, our method is significantly more accurate. In recent work, Hu et al. [4] identify small local elements correlated with object styles.

Weakly-supervised image segmentation Several computer vision methods can localize object data from whole-image tags (e.g. [20, 22, 2]). With the rise of deep neural networks, researchers observed that neurons in a classification network often activate on salient objects [19]. Oquab et al. [12] append global max-pooling to a convolutional segmentation network [8] to obtain a classification network suitable for object localization. In our work we focus on segmentation, and found that global max-pooling does not favor detecting coherent regions: we prefix it with average pooling for smoother segmentation. We also found that WU-Net’s skip connections improve results over sequentially stacked convolutions. Pathak et al. [13] study additional constraints, which we can potentially incorporate.

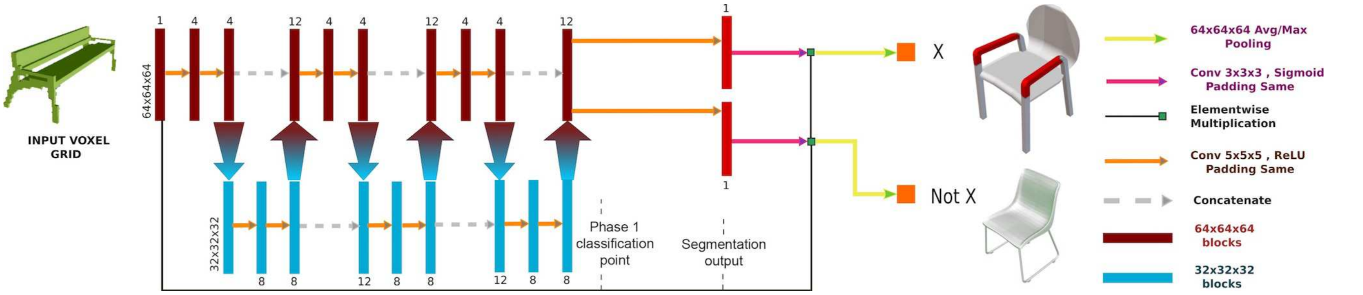


Figure 2: WU-Net architecture, showing three stacked down/upsampling 'U' structures linked by skip connections, ending in segmentation branches. Under weak supervision, the network is trained with only a classification loss.

3. Method

3.1. Data representation

We represent a 3D shape in voxelized form. Given a 64×64×64 cubical grid tightly fitting the shape, we set each voxel intersecting the surface to 1, and the rest to 0. We omitted interior voxels. Apart from being a natural domain for 3D convolution, this representation ensures we do not take advantage of inherent part structure in meshes. In fact, our input need not be a mesh at all, as long as we can densely sample it.

3.2. Network architecture

Our method for weakly-supervised 3D shape segmentation utilizes a novel feedforward neural network architecture, which we call WU-Net. It is inspired by the U-Net architecture of Ronneberger et al. [16], which was proposed as an effective way to segment biomedical images with limited training data in a strongly supervised setting. U-Net's prominent feature, from which it derives its name, is a sequence of fully convolutional downsampling layers (the "contracting" arm of an 'U'), followed by an inverse sequence of fully convolutional upsampling layers (the "expanding" arm of the 'U'), with the two sequences bridged by skip connections.

The WU-Net architecture leverages this building block by linking three fully convolutional U structures in sequence, i.e. a 'W' followed by an U (Figure 2). Data flowing through the network therefore goes through three successive cycles of down- and upsampling, from 64^3 to 32^3 and back to 64^3, encouraging spatial coherence and spread in the detected signal. Unlike U-Net, our U's are very shallow, each one involving a single downsampling/upsampling sequence. We explain this design choice below.

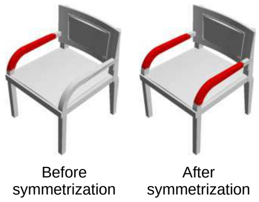
Our architecture also has skip connections like U-Net, which allow reasoning in later layers to be sensitive to structure in the original data which may have been lost during downsampling. *Unlike* the original U-Net, the WU-Net skip connections also provide bridging connections between *different* U structures (see the dashed arrows in the lower row of layers in Figure 2). This provides an elegant symmetry between the high and low resolution paths in the structure, with data winding back and forth between the resolutions while also having a secondary flow within the layers at each resolution. We also discuss this design choice below.

To map from a layer at one resolution to a layer at the same resolution (orange arrows), we employ several 5^3 convolutional kernels. To downsample, we use a max pooling operator over a 2^3 neighborhood. To upsample, we use bilinear interpolation of the feature map. All neurons in the 'WU' structure have ReLU activations.

Output segmentation map. The output of the final U is fed to two or more segmentation branches, one for each class. For weakly-supervised binary classification, e.g. "has back" vs "lacks back" for chairs, there are two branches. Under strong supervision, there is one branch for each part label: 'seat', 'back', 'arm' etc. Each branch has one 3^3 convolutional layer, with sigmoid activation. This layer acts as the *segmentation map* – it is in one-to-one correspondence with the input, and its output values are interpreted as the probabilities of voxels having particular class labels.

Loss function. Under strong supervision, a per-voxel cross-entropy loss is applied to the output segmentation map. Under weak supervision, we apply 2^3 average pooling to this output, and then take the maximum over the pooled response. Average pooling encourages a wider response region (Section 4 tests the effect of other pooling radii). The max-pooled prediction (across branches) is compared to the GT shape tag with a cross-entropy loss. To prevent activating empty voxels near shape boundaries, we multiply each segmentation map element by the corresponding element of the input voxel grid, letting the network focus only on errors over the shape.

Symmetrization. Our dataset shows prominent symmetries, chiefly reflectional. Since such shapes have redundant local information, a classifier can achieve high accuracy without seeing the complete shape. WU-Net is no exception, and our part detection often demonstrates consistent asymmetry, yielding high precision but lower recall, e.g. when only right arms of chairs are detected (inset). To correct this, we simply mirror inferred salient regions on both sides of the symmetry plane.



Discussion of design choices. WU-Net has three shallow U's instead of a single deep one, bridged by skip connections at both high and low resolutions. These design choices enable convolutional filters in later layers to have a high effective field of view (by composition with filters from preceding layers) even on high resolution data. Each shallow U mildly summarizes the signal and then immediately analyses it jointly with the unsummarized signal. The information flow is visualized in Figure 3. The "low resolution" skip connections provide each summarization step context from previous summaries. Section 4 shows that each successive stacked U improves performance, and weakly supervised shallow U's outperform deep U's.

Note the contrast to U-Net, where the latter half of the deeper architecture reconstructs successively higher resolution signals from a single drastic summary in the bottleneck layer. While skip connections do provide access to undecimated signals, the results of the joint high- and low-resolution analysis at each level are not further summarized, but simply upscaled to the next level. The filters in the final layer cannot have a high field of view on the original signal unmodified by downsampling. Thus, only excessively local information is incorporated from early layers by concatenation, which can drown out meaningful signals from the summarization layers when only weak supervision is available.

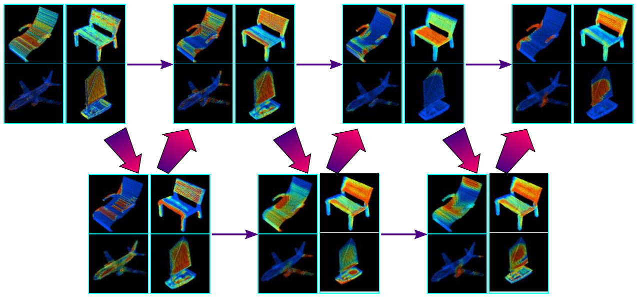


Figure 3: Information flow in weakly-supervised WU-Net, captured as layer activation maps (red: high) for detecting chair arms/backs, ship sails, and plane engines. Shallow U's prevent over-summarization, and the signal is not lost by repeated concatenations from distant layers, unlike a deep U (Figure 4).

In Figure 4, we show how activation maps in a single deep U suffer from excessive summarization, which the weak supervisory signal is not sufficient to repair despite skip connections: per-voxel strong supervision is required. Even though the bottleneck layer correctly localizes parts, multiple rounds of subsequent upsampling and ambiguous detail introduced from earlier layers spread the signal out incorrectly.

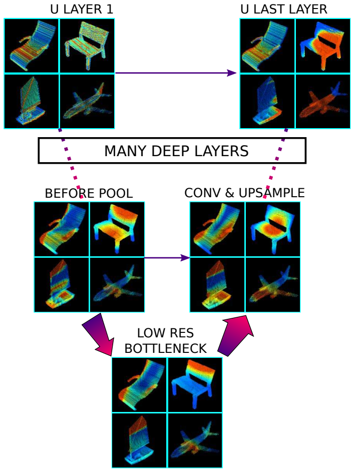


Figure 4: Arm, back, sail and engine signals are lost via over-summarization and ambiguous high-resolution concatenations in a weakly-supervised deep U-Net.

3.3. Training

In the weakly supervised setting, the WU-Net architecture is trained in two phases. We found the two-phase training to give better results than a single phase alone. The phases are described below.

Phase 1 (no output segmentation map). In this phase, the final segmentation branches are removed and a simple classification layer is temporarily appended to the 'WU'. This layer computes the maximum, over all voxels, of each of the 12 'WU' output channels, followed by a fully-connected map from the 12 maxima to two outputs (the complete shape label, e.g. "armrest" vs "no armrest"). This network is trained with cross-entropy loss until the classification accuracies on both training and validation sets exceed 95%. Once this happens, we adjudge the network to have high generalization accuracy and move to the next phase. Further phase 1 training tends to overfit.

Phase 2 (with output segmentation maps). We now remove the phase 1 classification layer, restore segmentation branches, and train the whole network end-to-end. We found benefit in slowly enlarging the average-pooling kernel, starting from 0 (no pooling) for 50 epochs, followed by 10 epochs for each expansion of the kernel. The best overall performance came from a 2^3 final kernel, and we report all comparative results with this setting. However, for specific datasets larger

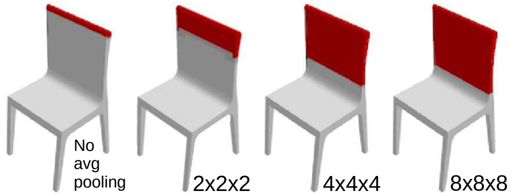


Figure 5: The effect of increasing the kernel size in average pooling. While here the largest kernel works best, for categories with finer parts this is not so.

kernels may help further, as we show in our evaluation. Generally, detection of larger salient parts is aided by larger average pooling kernels (Figure 5).

Under *strong* supervision, we dispense with two-phase training and final pooling, and directly train the network end-to-end with a per-voxel cross-entropy loss over the segmentation maps for each output label.

4. Results

We evaluate our method on standard datasets that contain various semantic region labels. For weakly-supervised segmentation, which is our principal focus, our extensive comparisons suggest that WU-Net is a "sweet spot" in the space of related architectures. We also show that the same network performs strongly under strong supervision on standard benchmarks.

4.1. Weakly Supervised Region Labeling

In these validation experiments, we test if our WU-Net architecture successfully detects salient parts that distinguish one category of shapes from another. We collated six different pairs of fine-grained shape classes, each pair distinguished by a prominent semantic component. These classes were: (a) *chairs* with and without **armrests**, (b) *chairs* w/wo **backs**, (c) *cars* w/wo **roofs**, (d) *airplanes* w/wo wing-mounted **engines**/propellers, (e) *ships* w/wo **sails**, and (f) *beds* w/wo **heads**. Fine-grained classes (a-d) are available in ShapeNet [1] with ground-truth labeled segmentations, although we had to collect chairs without backs (stools) from ModelNet [25]. We annotated ships (e) and beds (f) ourselves.

Each class was randomly split 50:50 into train/test sets. Under weak supervision, segmentation/labeling accuracy on the training set is as important as on the test set. Still, the test set allows us to directly compare with a strongly supervised baseline. Dataset statistics are in Table 1. The meshes were voxelized with Binvox [10]. Our data (and code) will be publicly available.

In evaluations presented in this section we report **area under the curve (AUC)** for precision/recall curves, where **higher** numbers indicate **better** performance. Our method is labeled "WU-Net + symmetrization". Full plots are in supplementary material.

Shape category	Part category	Has part	Lacks part
(a) Chair	Armrest	481	1359
(b) Chair	Back	150	75
(c) Car	Roof	806	106
(d) Airplane	Engine	1034	266
(e) Ship	Sail	95	674
(f) Bed	Head	19	4

Table 1: Weakly supervised segmentation dataset.

Segmentation performance. Table 2 reports the per-voxel labeling accuracy of WU-Net with identical hyperparameters (including 2^3 avg pooling and symmetrization) and automatic training protocol in the 6 weakly supervised segmentation tasks, on training shapes. (Training set segmentation accuracy is a relevant performance metric under weak supervision. When computing it, we do not use ground truth tags. Test set performance is similar, see supplementary.) For comparison we use these ablated alternatives:

- The *saliency map* of the trained WU-Net, computed as the gradient of output w.r.t. input.
- WU-Net *without skip connections*, representing a conventional fully convolutional architecture.
- WU-Net *without the final U*, dubbed W-Net.
- WU-Net *without 2 of the U's*, just a single shallow U, dubbed V-Net.

Further, we also test WU-Net without symmetrization.

WU-Net, with or without symmetrization, substantially improves upon these alternatives. Training of the ablated networks does not always converge. When it does converge, V-Net and W-Net perform reasonably well, though they don't match WU-Net. The version without skip connections is much worse.

In Table 3, we compare WU-Net with networks using deep U's. All outputs are symmetrized.

- A 3D analogue of the original U-Net [16], with *one deep U structure* that repeatedly halves the grid resolution to 4^3, then repeatedly doubles it back to 64^3, with skip connections at every resolution.
- *2 and 3 deep U's*, linked with high and low skip connections just like WU-Net.
- The above 3 networks, with *Inception-style blocks* [21] at every layer. Each 5^3 kernel has 2^3 and 3^3 kernels also applied in parallel.
- A 3D version of a *Stacked Hourglass Network* (SHN_{3D}) [11], modeled as 3 deep U's without low-resolution skip connections between different U's.

Deep 3D U-Net training converges, but it identifies incorrect parts (e.g. chair seats, not backs). Apart from

	Arm	Back	Roof	Propeller	Sail	Bed
WU-Net + symmetrization	**0.69**	**0.79**	0.32	**0.46**	**0.77**	**0.32**
WU-Net	0.61	0.76	**0.39**	0.39	0.76	**0.32**
W-Net	0.54	0.73	0.09	0.06	0.55	0.15
V-Net	0.60	0.76	0.03	0.34	0.52	0.12
No Skip Connections	0.07	0.62	0.05	0.09	0.30	0.17
Gradient saliency + symmetrization	0.03	0.27	0.12	0.18	0.00	0.29

Table 2: AUC of WU-Net vs various ablations for weakly-supervised segmentation (on training shapes).

	Arm	Back	Roof	Propeller	Sail	Bed
WU-Net + symmetrization	**0.69**	**0.79**	**0.32**	**0.46**	**0.77**	0.32
3 Deep U (Inception)	0.03	0.19	0.03	0.01	0.22	0.21
2 Deep U (Inception)	0.00	0.08	0.05	0.00	0.27	0.39
1 Deep U (Inception)	0.04	0.19	0.04	0.14	0.54	0.13
3 Deep U	0.08	0.10	0.03	0.00	0.00	**0.42**
2 Deep U	0.05	0.47	0.03	0.00	0.27	0.10
1 Deep U	0.31	0.01	0.03	0.11	0.16	0.06
SHN_{3D}	0.35	0.39	0.04	0.16	0.45	0.16

Table 3: AUC of WU-Net vs Deep U-Net variants (symmetrized) for weakly-supervised segmentation (on training shapes).

	Arm	Back	Roof	Propeller	Sail	Bed
WU-Net + symmetrization	0.71	0.73	0.35	0.42	**0.84**	0.37
Strong supervision without classifier	0.06	0.43	0.70	0.57	0.07	**0.48**
Strong supervision with classifier	**0.91**	**0.97**	**0.89**	**0.89**	0.68	**0.48**

Table 4: AUC of Weakly supervised WU-Net vs a strongly supervised baseline (on test shapes).

	Arm	Back	Roof	Propeller	Sail	Bed
WU-Net + symmetrization	**0.69**	**0.79**	0.32	**0.46**	0.77	0.32
SF 0.25	0.43	0.45	**0.51**	0.12	**0.87**	0.53
SF 0.5	0.52	0.59	0.27	0.13	0.67	**0.62**
SF 1.0	0.49	0.37	0.39	0.18	0.65	0.34
SF 2.0	0.39	0.42	0.42	0.29	0.61	0.34

Table 5: AUC of WU-Net vs Shilane and Funkhouser (SF) [16] at different scales (on training shapes). Note that SF requires knowledge of ground truth tags at test time, whereas our method does not use them.

	Arm	Back	Roof	Propeller	Sail	Bed
2x2x2 (default)	**0.69**	**0.79**	**0.32**	0.46	**0.77**	0.32
No avg pooling	0.63	0.67	**0.32**	**0.54**	0.70	0.31
4x4x4	0.49	0.42	0.05	0.26	**0.77**	0.32
8x8x8	0.08	0.23	0.00	0.01	0.58	**0.33**

Table 6: The statistical effect (AUC) of increasing the kernel size for average pooling at the end of WU-Net.

a single deep U for chair armrests and a double deep U for backs, the rest cannot identify meaningful parts. This validates our use of shallow U's for weakly supervised segmentation.

We also present visual examples of symmetrized WU-Net output, for a threshold of 0.9, in Figures 1 and 7. In addition we also show some visual results on swivel chairs, for which ground truth segmentations were not available: the roller wheels were identified as salient in these shapes. Visual results on all shapes in our datasets are provided in supplementary material.

Comparison to a strongly supervised baseline. For further insight, we train WU-Net with strong supervision, with a single segmentation branch which is thresholded for a precision-recall plot. (We cannot use the strongly supervised variant of Section 4.2, because it outputs the max over label branches per voxel, and has no tunable threshold.) This strongly supervised network is not a classifier, and hence can end up identifying a semantic part in a shape which lacks it. This leads to very poor accuracy (Table 4). If we aid it by using the trained weak network simply as a binary classification oracle (∼99% accurate), then it establishes a high baseline as expected. This indicates the very valuable role shape tags play in identifying semantic parts.

Comparison to Shilane and Funkhouser [17]. There is little prior work on weakly supervised 3D shape segmentation. The most relevant research is by Shilane and Funkhouser, who identify distinctive regions in different shape categories. While our scenario is slightly different (fine-grained intra-category differences), their method can be evaluated directly in our training setup. (Note that S-F relies on ground-truth shape tags to find distinctive regions. It does not work in our *test* setup, where the shape tag is unknown.) Table 5 shows results. The S-F results were

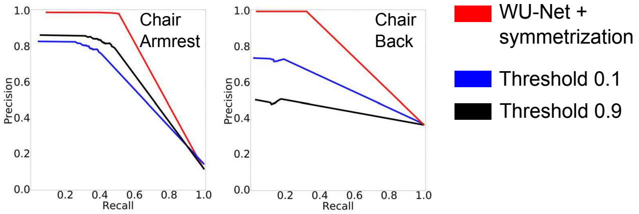


Figure 6: Chair armrest and back detected independently (red), vs detected in a multi-label setting for different thresholds on the label classifier outputs.

	ShapeBoost	ShapePFCN	1SU	2SU	**WU-Net** (3SU)	4SU	1DU	2DU	3DU	1DUI	2DUI	3DUI
Co-aligned	83.1	89.0	87.8	89.8	90.2	90.0	90.5	90.7	90.1	91.2	90.9	91.3
Randomly rotated	-	-	73.1	74.7	75.1	75.6	77.8	77.8	78.7	78.3	79.0	79.0

Table 7: Strongly-supervised segmentation and labeling accuracy (%), averaged over 16 categories, for test shapes in ShapeNetCore, versus ShapeBoost [7] and ShapePFCN [6]. Different WU-Net-style variants have abbreviated names: 3SU is a sequence of **3** **s**hallow **U**'s (i.e. WU-Net), 1DUI is **1** single **d**eep **U** (**I**nception-style); each variant trained for 100 epochs. Full per-category statistics are in supplementary material.

not symmetrized – symmetrization slightly hurt results because of false positives. In three out of six cases (chair arms/backs, airplanes), WU-Net with default settings significantly outperforms S-F at all manually specified scale settings. In the other cases (cars, ships, beds), WU-Net is a little worse, but the scales at which S-F outperform it turn out to be dramatically suboptimal in other cases. On average, WU-Net significantly outperforms Shilane-Funkhouser at any given scale.

The role of the average-pooling layer. The kernel size of the average-pooling layer after the segmentation map is a tunable hyperparameter that directly affects the identified regions in a visually interpretable way. For large semantic parts, a larger final kernel size often yields better results. The effect is one of degree, as seen in Table 6, and depends on the data. However, we found that a fixed 2^3 kernel achieves good performance in all cases, and this is the setting we present for our fully automatic method and for all evaluations.

Multi-label weak supervision. What does WU-Net predict when weakly supervised with *multiple* shape-level tags? While this is not this paper's focus, we find in preliminary investigations the framework can extend to some multi-label settings. For instance, we trained a single WU-Net on chairs tagged with "arm" and "back", where either, both or no labels could be present. (In fact, we could not find chairs with arms but no backs.) This WU-Net had two branches, one for each part label, and the segmentation map from each branch was output if the classification score exceeded a common threshold. In Figure 6, we show the multi-label output, vs training a different binary WU-Net for each part. While the multi-label scores are competitive, especially considering the training data lacks "arm but no back" combinations, they do not exceed the binary results. Extending weakly supervised 3D segmentation to a range of multi-label scenarios is a ripe avenue for future work. A major difficulty is that in real datasets, weak tags are often strongly correlated (e.g. chair legs and seats). A small amount of strong supervision may resolve this.

4.2. Strongly Supervised Region Labeling

The WU-Net architecture has the great advantage of being directly deployable in a strongly supervised setting, where per-point labels are available. We therefore test it on a standard benchmark: ShapeNetCore [1]. This dataset has manually annotated ground-truth segmentations for thousands of shapes in 16 categories. We compare our method to a recent state-of-the-art technique ([6]), using the same train/test splits. Our performance is summarized in Table 7 (more details in supplementary material). Since the benchmark shapes are co-aligned, WU-Net can take advantage of this to achieve state-of-the-art scores. However, WU-Net is not designed to be rotation-invariant by default, unlike

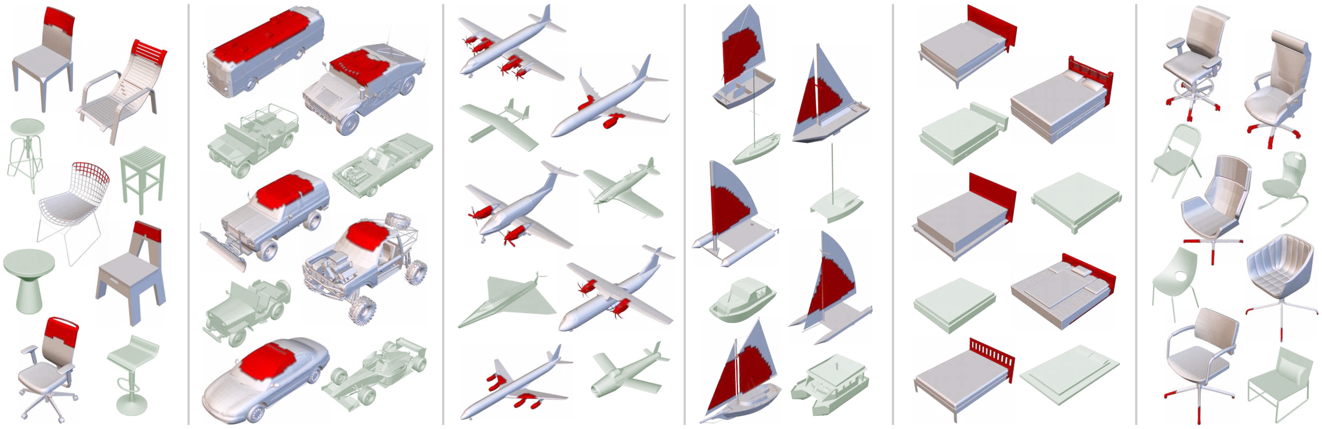


Figure 7: Examples of weakly supervised segmentation by WU-Net. Left to right: detecting chair backs, car roofs, wing-mounted airplane engines/propellers, ship sails, bed heads, and chair swivels.

ShapePFCN [6]. If we train and test with every shape independently and randomly rotated, scores drop by about 10-15%, since the network can no longer memorize rough absolute locations for parts. This is a familiar issue with voxel grid-based networks, and augmentation with many more rotations may help.

We also compare with a wide variety of variant architectures, described in the previous section, and present results in Table 7 (and full per-category results in supplementary material). From these, we can infer the following (especially from results with rotation): *under strong supervision*, (a) deep U's, (b) stacking multiple U's, and (c) Inception-style networks all improve performance a little. Combining all three factors yields the highest accuracy. Note that this improvement does not extend to weakly-supervised training.

4.3. Applications

We demonstrate the wide utility of our method with three mockup applications that focus on organizing a 3D shape database. First, we enable *part-sensitive shape search* (Figure 8) by computing a fine-grained shape similarity metric that focuses only on a user-selected tag (our simple implementation uses weighted average distance between the salient voxels, after aligning centroids of salient regions). Since we map tags to specific geometric regions, we can make queries like: "find chairs with similar armrests". Second, we show *fine-grained exploration of a shape dataset* (Figure 9), demonstrating that the entire dataset can be organized based on similarity metrics computed for a specific tag (our prototype uses the simple metric above), providing users with different tag-focused views of the database. Third, we demonstrate that our method facilitates better *thumbnail creation* (Figure 10) by focusing on salient regions that correspond to specific tags. Automatically-generated thumbnails are commonly used for rapid browsing, and demonstrating important surface regions can provide better shape understanding for the user.

5. Conclusion

We presented a method to obtain fine-grained semantic part annotations of 3D shapes from only weak shape-level tags. It achieves this through a deep neural network trained simply to classify the shape as possessing or lacking the part. The novel structure of this network, which forms our core technical contribution, encourages finding large consistent regions across shapes that characterize the differentiating part. We also presented compelling results on strongly supervised segmentation using the same network.

There are several avenues for future work enabled by unstructured user annotations in public online 3D repositories. It would be interesting to leverage natural language processing in addition to geometric analysis to automatically infer salient shape tags and corresponding parts from free-form shape descriptions provided by people. It would also be interesting to generalize our network architecture to handle a larger and more heterogeneous sets of tags, and to scale robustly to multi-label settings.

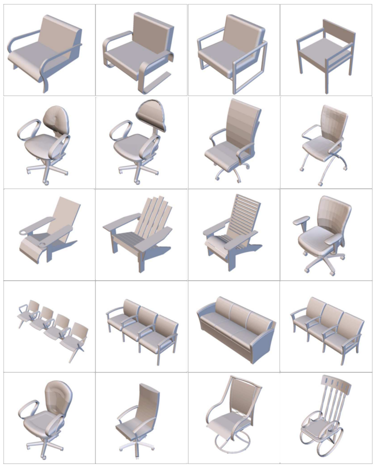


Figure 8: Each row shows the top 3 shapes with similar armrests, detected by WU-Net, retrieved for the query in the first column.

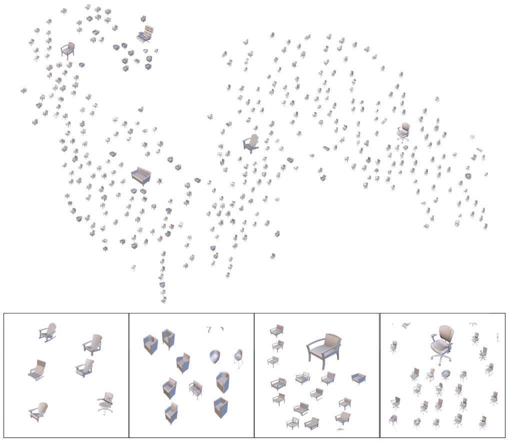


Figure 9: A t-SNE embedding of chairs organized by similarity of the "armrest" regions detected by WU-Net. Below, we show several zoomed-in regions of the image. The larger icons on top represent diverse representatives of the collection that can be obtained from this similarity metric.

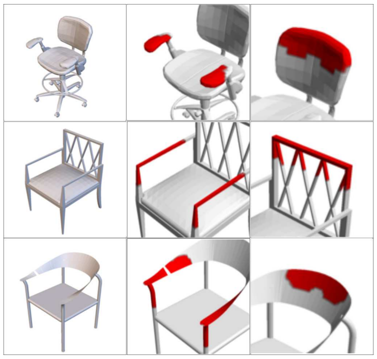


Figure 10: Different thumbnails of the same shapes (first column) created to highlight detected "armrest" (second column) and "back" (third column) regions.

References

[1] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q.-X. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An information-rich 3D model repository. arXiv:1512.03012, 2015. 5, 7

[2] R. G. Cinbis, J. J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *TPAMI*, 39(1), 2017. 2

[3] A. Golovinskiy and T. Funkhouser. Consistent segmentation of 3D models. *Computers and Graphics (SMI)*, 33(3), 2009. 2

[4] R. Hu, W. Li, O. van Kaick, H. Huang, M. Averkiou, D. Cohen-Or, and H. Zhang. Co-locating style-defining elements on 3D shapes. *Trans. Graph.*, 36(3), 2017. 2

[5] Q. Huang, V. Koltun, and L. Guibas. Joint shape segmentation with linear programming. *Trans. Graph.*, 30(6), 2011. 1, 2

[6] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri. 3D shape segmentation with projective convolutional networks. In *CVPR*, 2017. 1, 2, 7, 8

[7] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D mesh segmentation and labeling. *Trans. Graph.*, 29(4), 2010. 2, 7

[8] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional models for semantic segmentation. In *CVPR*, 2015. 2

[9] H. Maron, M. Galun, N. Aigerman, M. Trope, N. Dym, E. Yumer, V. G. Kim, and Y. Lipman. Convolutional neural networks on surfaces via seamless toric covers. *Trans. Graph.*, 36(4), 2017. 2

[10] P. Min. Binvox. `http://www.patrickmin.com/binvox/`, 2017. 5

[11] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 5

[12] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? – weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015. 2

[13] D. Pathak, P. Krähenbühl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 2015. 2

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *CVPR*, 2017. 2

[15] G. Riegler, A. O. Ulusoy, and A. Geiger. OctNet: Learning deep 3D representations at high resolution. In *CVPR*, 2017. 2

[16] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1, 3, 5

[17] P. Shilane and T. Funkhouser. Distinctive regions of 3D surfaces. *Trans. Graph.*, 26(2), 2007. 2, 6

[18] 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.*, 30(6), 2011. 2

[19] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2014. 2

[20] H. O. Song, R. B. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell. On learning to localize objects with minimal supervision. In *ICML*, 2014. 2

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 5

[22] C. Wang, W. Ren, K. Huang, and T. Tan. Weakly supervised object localization with latent category learning. In *ECCV*, 2014. 2

[23] 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. *Trans. Graph.*, 36(4), 2017. 2

[24] Y. Wang, S. Asafi, O. van Kaick, H. Zhang, D. Cohen-Or, and B. Chen. Active co-analysis of a set of shapes. *Trans. Graph.*, 31(6), 2012. 2

[25] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*, 2015. 5

[26] K. Xu, V. G. Kim, Q. Huang, and E. Kalogerakis. Data-driven shape analysis and processing. *Comp. Graph. For. (STAR)*, 2016. 1

[27] L. Yi, L. Guibas, A. Hertzmann, V. G. Kim, H. Su, and E. Yumer. Learning hierarchical shape segmentation and labeling from online repositories. *Trans. Graph.*, 36(4), 2017. 1, 2

[28] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3D shape collections. *Trans. Graph.*, 35(6), 2016. 2

[29] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3D shape collections. *Trans. Graph.*, 35(4), 2016. 2

[30] L. Yi, H. Su, X. Guo, and L. Guibas. SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. 2017. 2