HAL3D: Hierarchical Active Learning for Fine-Grained 3D Part Labeling

Fenggen Yu[1,2*] Yiming Qian[1] Francisca Gil-Ureta[1]
Brian Jackson[1] Eric Bennett[1] Hao Zhang[1,2]
[1]Amazon [2] Simon Fraser University

Abstract

We present the first active learning *tool for* fine-grained *3D part labeling, a problem which challenges even the most advanced deep learning (DL) methods due to the significant structural variations among the intricate parts. For the same reason, the necessary effort to annotate training data is tremendous, motivating approaches to minimize human involvement. Our labeling tool iteratively verifies or modifies part labels predicted by a deep neural network, with human feedback continually improving the network prediction. To effectively reduce human efforts, we develop two novel features in our tool, hierarchical and symmetry-aware active labeling. Our human-in-the-loop approach, coined HAL3D, achieves close to error-free fine-grained annotations on any test set with pre-defined hierarchical part labels, with 80% time-saving over manual effort. We will release the finely labeled models to serve the community.*

1. Introduction

Semantic shape segmentation and labeling [8, 27] is a classical problem with numerous applications [18], including shape/part recognition, retrieval, indexing, and attribute transfer. We consider the problem of *fine-grained* 3D part labeling, a task that has not received as much attention as the typical semantic segmentation into coarse/major parts (e.g., backs, seats, and legs for chairs). While these coarse semantic groups support high-level visual perception, they do not possess sufficient granularity to address shape properties related to motion, function, interaction, or construction. Indeed, fine-grained object parts, e.g., the individual wheels and the small mechanical parts in a swivel leg, are both function-revealing and build-aware (see Fig. 1), with strong ties to how the objects were physically assembled or virtually built by artists. Real-world applications such as product inspection, assembly, customization, and robot interaction all operate on fine-grained object parts.

Learning to label fine-grained parts is a highly challenging problem due to the great structural variations among the small and intricate parts, even for shapes that belong to the same category, as shown in Fig. 1. On a challenging dataset with average part counts per category ranging from 15 to more than 500, state-of-the-art learned models such as [33] can only achieve mean IoU scores below 50%, out of a maximum of 100% for ground truth (GT) results. Also challenging is the tremendous data annotation effort needed to train methods to segment and label fine-grained parts. On the other hand, for applications which demand de facto *full* accuracy, e.g., for diagnoses in medicine and quality assurance in e-commerce, one can hardly rely on a fully automatic method to reach the mark. It is arguable that the only means to attain full accuracy is to have humans validate all the results, as in the case of data annotation.

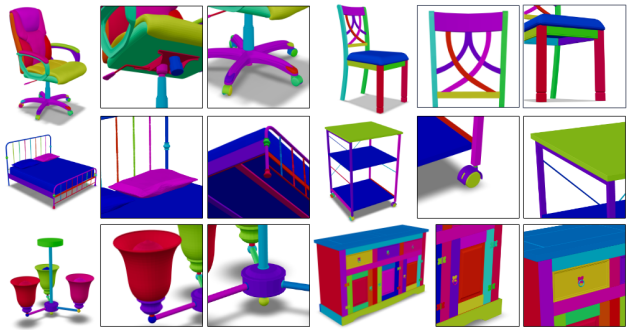

Figure 1. Fine-grained parts of the 3D models shown exhibit the kind of geometric and structural complexity and diversity that part labeling has to handle. No existing methods, whether learned or heuristic-based, could obtain close-to-fully-accurate labeling in these challenging cases, while our human-in-the-loop active learning tool can approach full accuracy, barring human errors.

In this paper, we present the first *active learning* tool for fine-grained 3D part labeling. Given a set of 3D shapes segmented into fine parts, our labeling tool assigns one of the predefined labels to each part. These input parts are deemed *atomic* (i.e., indivisible); they can be as low-level as mesh triangles and as high-level as results obtained by an unlabelled semantic segmentation, or mid-level components from an over-segmentation. In general, active learning iterates between automated labeling and human input for label rectification [44]. Compared to conventional learned models with full autonomy, such a *human-in-the-loop* approach provides a viable option to achieve close to error-free part labeling on test sets, barring human errors[1].

*Work carried out during internship at Amazon.

[1]During testing, accuracy is measured against GT labeling which has

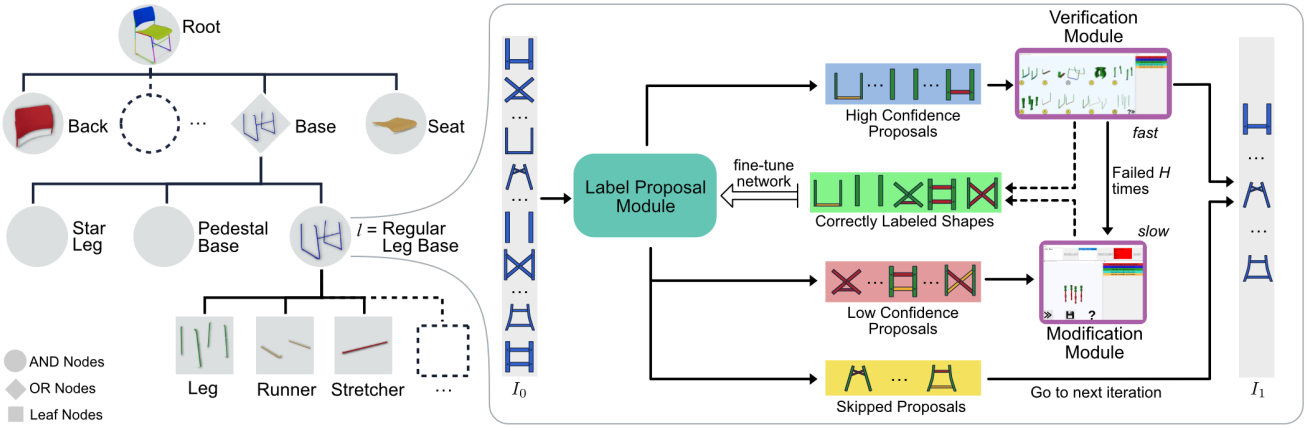

Figure 2. Our human-in-the-loop, hierarchical active learning (HAL3D) tool for fine-grained 3D part labeling. The input consists of a set of test shapes each pre-segmented into parts. The labeling proceeds hierarchically, following a tree structure (left) that organizes the hierarchical part labels, from coarse labels (top) to fine-grained labels (bottom). For a node in the label tree (with label l = "Regular Leg Base" in the illustration), the input I_0 is the subset of parts, from the entire set of shapes, that are labeled l by its parent. When labeling parts with l, the label proposal module first assigns refined labels for each part. Then, proposals are sorted by the mean label probability over the parts of each shape, with the high-confidence (HC) proposals passed to the verification step, which stops once the rate of failed shapes reaches a threshold. The low-confidence (LC) proposals are passed to the label modification module. Correctly labeled shapes fine-tune the network, while skipped and failed shapes, in set I_1, go to the next iteration for labeling. The iterations terminate when all shapes have passed human validation.

As shown in Fig. 2, our interactive labeling tool iteratively verifies or modifies part labels predicted by a deep neural network, with human feedback continually improving the network prediction. Specifically, our system consists of three modules for label proposal, verification, and modification. Label proposals are produced by dynamic graph CNN (DGCNN) [37], with the resulting label probabilities dictating whether to pass the predicted labels to the verification or modification module. Both human-verified and corrected labels are passed back to the label proposal network to fine-tune it. The iteration terminates once all part labels have passed human verification.

As the key criterion for success with active learning is the minimization of human effort, we design and incorporate two novel features into our labeling tool:

- *Hierarchical* labeling: We organize all the part labels in a hierarchical tree structure, which guides our prediction-verification-modification process so that labels further down in the hierarchy are dealt with only *after* their parent labels have been fully verified. This leads to labeling efficiency, since our labeling process is *coarse-to-fine*, with fewer labels to process in the coarser levels of the hierarchy. In addition, with both human users and the prediction network only needing to deal with labels at the same level of the hierarchy rather than across all levels, there is less load on the users to reduce possible errors and the prediction accuracy improves, which reduces the amount of label modifications, which are the most expensive task.

- Part *symmetry*: Since symmetric parts receive the same label, we employ detected symmetries to constrain and facilitate both label verification and modification.

Our hierarchical active learning tool is coined HAL3D. We adopt the hierarchical labels from PartNet [20] and train and test our labeling tool on both PartNet, which comes with semantic part segmentations, and Amazon-Berkeley Objects (ABO). ABO is a recently published dataset of high-quality, artist-created 3D models of real products sold online. These models are composed of build-aware connected components, However, since some of these components are too coarse, we first obtain an over-segmentation via convex decomposition [39] and then apply HAL3D.

We evaluate the core components of HAL3D, *i.e.*, the label proposal network being fine-tuned by human feedback, and both hierarchical and symmetry-aware active labeling, by comparing to baselines and performing an ablation. The results demonstrate clear efficiency of HAL3D over labeling tools without active learning and non-hierarchical designs. Overall, HAL3D achieves 80% speed-up over manual fine-grained labeling, on the PartNet test set. A supplementary video shows our interactive tool at work. We will

been produced by humans. To achieve full accuracy, two criteria must be met: (a) the *testing* human labeler's judgement of what is the correct labeling agrees with the GT; (b) the labeler does not make any error, e.g., due to carelessness or time pressure. We regard both types of violations as "human errors." One may also refer to the former as an inconsistency between human labeling, which is likely to occur near fuzzy part boundaries.

release the finely labeled ABO models to serve the community pending permission from the dataset owner.

2. Related Work

Most approaches to semantic 3D shape segmentation and labeling have been designed to reason about coarse- or high-level structures and to target fully autonomy. Considerably less work has been devoted to fine-grained segmentation and labeling or human-in-the-loop approaches.

Coarse 3D shape segmentation and labeling: Since the seminal work on learning 3D mesh segmentation and labeling [13] in 2010, many learning methods have been proposed [8]. Prominent supervised methods such as PointNet [24], PointNet++ [25], and DGCNN [37] perform feature learning over point clouds, while MeshCNN [7] develops convolution and pooling layers that operate on mesh edges. These methods, among many more, produce both parts and part labels, but they have only been trained and tested on datasets with coarse segmentations, such as the ShapeNet part [44] and the COSEG datasets [35]; most object categories therein were labeled with only 2-5 parts.

High-level parts, e.g., the legs, seats, and backs of chairs, typically possess sufficient structural consistency (e.g., in terms of relative positioning), hence predictability, to facilitate supervised learning. Most unsupervised or weakly supervised approaches also leverage such consistencies. In particular, earlier methods for co-segmentation, i.e., segmenting a *set* of related shapes altogether, all rely on alignment or clustering, either in the spatial domain [6] or feature spaces [29, 9]. More recently, resorting to DL, Ada-Coseg [50] optimizes a group consistency loss defined by matrix ranks. BAE-Net [3] and RIM-Net [21] obtain co-segmentations by learning branched neural fields.

Fine-grained segmentation: Fine-level parts are important in many real-world applications since, whether large or small, each of them has a function to serve and often a motion to carry out. However, the structure consistency possessed by major semantic groups is mostly lost among fine parts. Such parts could be obtained via shape decomposition based on geometric properties such as convexity [12, 39], cylindricity [49], or through primitive fitting using cuboids [42, 30] or quadrics [23, 45], deformed spheres [22, 15]. However, the resulting segmentations are neither semantic nor labeled.

Most learning-based solutions are unsupervised or self-supervised, mainly because annotating fine-grained parts is an extremely tedious task. Li et al. [43] learns part labeling from noisy online part graphs annotated by human artists. Yu et al. [46] trains a recursive part decomposition network for category-specific, fine-grained, hierarchical shape segmentation. Wang et al. [33] and SHRED [11] develop deep clustering to learn part priors from fine-grained segmentation without part labels, where no semantic labels are produced on test shapes. Sharma et al. [28] perform projective shape analysis [36], where a fine-grained 3D segmentation is obtained by aggregating segmented multi-view 2D images. Both of these methods operate fully automatically after training, but their performance falls far short of 100% accuracy: 32.6% mean IoU score achieved by [28] on PartNet [20] with fine-grained (Level-3) parts (part counts ranging from 4 to 51), and below 50% mean IoU by [33] on a more challenging dataset with part counts up to 500+.

Instead of operating on low-level shape elements such as points [33] and pixels [28], the neurally-guided shape parser [10] (NGSP) is trained to assign fine-grained semantic labels to *regions* of a 3D shape with improved labeling accuracy: mean IoU score up to 58% on PartNet. NGSP parses region label proposals generated by a guide neural network using likelihood modules to evaluate the global coherence of each proposal. It is still designed as a fully automatic method, but the authors did suggest the need for human-in-the-loop approaches to "scale beyond carefully-curated research datasets to 'in-the-wild' scenarios."

Hierarchical analysis: A fine-grained shape understanding is also attainable via hierarchical structural analysis. Some early works perform rule-based reasoning to construct symmetry hierarchies [38] and shape grammars [17], while others work with geometric priors, e.g., convexity [2], for hierarchical shape approximation. Similar to fine-grained segmentation, most learned hierarchical models are also unsupervised or weakly supervised, including co-hierarchical analysis [32], hierarchical abstraction [30], recursive implicit fields [21], and inverse constructed solid geometry (CSG) [45, 14]. All of these methods aim for full automation and work well only with shallow hierarchies.

Active learning for part labeling: The classical and core question for active learning [1, 47, 26] is: "How does one select instances from the underlying data to label, so as to achieve the most effective training for a given level of effort?" Past works have studied uncertainty and diversity for interactive segmentation and labeling of 3D shapes [35, 5], medical images [31], and large-scale point clouds [40], to name a few. We utilize active learning, aiming to approach full labeling accuracy, regardless whether the test shapes belong to a "research dataset" or are "in-the-wild."

Most closely related to our work is the active framework for region annotations by Yi et al. [44]. Both works share the common goal of producing accurate, human-verified labels. On the other hand, their work imposes a fixed human effort budget, while ours does not. Technically, their work relies on a conditional random field to automatically propagate human labels, while our work integrates a DL module [37] into an active part labeling tool. The only prior work that combines deep and active learning for 3D segmentation [5] focuses on boundary refinement, just as the earlier work [35] whose learning module is constrained k-

means clustering. However, none of these tools were designed to work with fine-grained labels, e.g., they still only operate on four part labels for chairs [5, 35, 44, 41].

Finally, while our work is the first active learning tool for fine-grained 3D part labeling, hierarchical active learning has been employed in other data domains such as documents. In hierarchical text classification[16, 4], text labels can also be organized hierarchically. In comparison, semantic labeling of 3D parts in a shape collection poses various challenges including richer structural variations, part relations, and more costly and delicate user interactions.

3. Method

Given a shape test set S, where each shape is pre-segmented into parts, our methods assigns a label $l \in L$ to each part of each shape. The set of labels, L, is predefined and organized in a tree structure such that coarse labels are at the top, and fine-grained labels at the bottom. The labeling progresses hierarchically top-to-bottom, assigning first coarse labels and refining them at each step, achieving high-accuracy fine-grained labeling results.

At each node, we use an active learning based tool to iteratively engage human interactions into the part labeling process. With this tool, we obtain a close to error-free part labeling on each node on any test set.

3.1. Hierarchical labeling strategy

We organize the part labels into a tree structure based on their semantics, as in PartNet [20]. Following the same terminology, two types of *internal* tree nodes are defined: an "AND" node indicates "what part(s) it has" (e.g., a regular base has legs and a runner), while an "OR" node indicates "what type it is" (e.g., a star- or pedestal-type base). The labeling on "AND" nodes is based on part features, while the labeling on "OR" nodes depends on group features; see Fig. 3. Each leaf node presents a final part label.

At each node of the tree, we combine a deep neural network with an active learning tool to perform part labeling. Consider a single shape $s_i \in S$ and its set of parts P_i. At the root node, the input are all the parts of shape s_i, that is $P_{root} = P_i$. After finishing the labeling task at the root node, the set of parts is split into one or more subsets $P_l \subseteq P_{root}$ corresponding to the children labels l. Then, each subset is input to the respective node for further labeling. By splitting the input set at each node, the size of the input decreases as the depth of the node increases. Finally, the labeling stops when we reach a leaf node.

The complexity of our labeling scheme depends on the structure of the tree. On one extreme, if all labels are direct children of the root node, training at the root would be more complex but there is only one node to train. The other extreme is a binary tree, where each node is trained to distinguish between only two labels, but we have to train more nodes, resulting in an increased training time. As we have found the original PartNet trees to have too many internal nodes, we modified this initial tree structure using a heuristic rule: we remove internal "AND" nodes which contain less than three children and keep all "OR" nodes. This way, our tree structure is better balanced for our task to lead to reduced training time. Examples of our modified tree structures are provided in the supplementary material.

3.2. Symmetry-assisted labeling

Another novel feature to accelerate HAL3D makes use of detectable part symmetries on the test set. Symmetry is ubiquitous in human-made objects and it is common practice for 3D artists to copy-and-paste symmetric or repeated parts when building 3D assets. Symmetry detection has been a well-studied subject in geometry processing [19] and there are a variety of applicable techniques. For simplicity, we simply use sizes of the part oriented bounding boxes as a crude and conservative feature to detect part symmetries.

As symmetric parts are expected to share the same label, we accelerate our active learning in two ways: if a proposal assigns different labels to symmetric parts, we automatically remove it from the "HC (High confidence)" set. Then, in the modification module, the user only needs to label one part of the symmetry group and our system automatically propagate it to the other parts. These features improve labeling efficiency by 20%; see Section 5.2.

3.3. Active labeling of fine-grained 3D parts

Fine-grained 3D shape segmentation and annotation is challenging because of the wide structural diversity found in small and intricate parts of each shape. This diversity makes it difficult for existing DL methods to achieve high prediction accuracy. On the other side, active learning techniques are commonly used by researchers to improve the performance of the learned classifier with the help of human efforts. We adopt these techniques and, at each node of our hierarchical labeling strategy, we use a deep active learning based method to achieve high part labeling accuracy. Our method comprises three main modules: the DL-based part label proposal network, the part label verification module, and the part label modification module.

Consider a node l in our tree structure, given the subset of parts labeled l of each shape, we iteratively update the part labels until they all pass human verification. At each iteration, we first use a DL-based neural network to make part label proposals based on part label probabilities. Then, the HC proposals are passed to the part label verification module for human verification. Next, we collect the LC, i.e., low-confidence, proposals, which are more likely to be inaccurate, and pass them to the label modification module. Finally, the verified proposals and modified proposals are combined and feed back to the proposal network to fine-

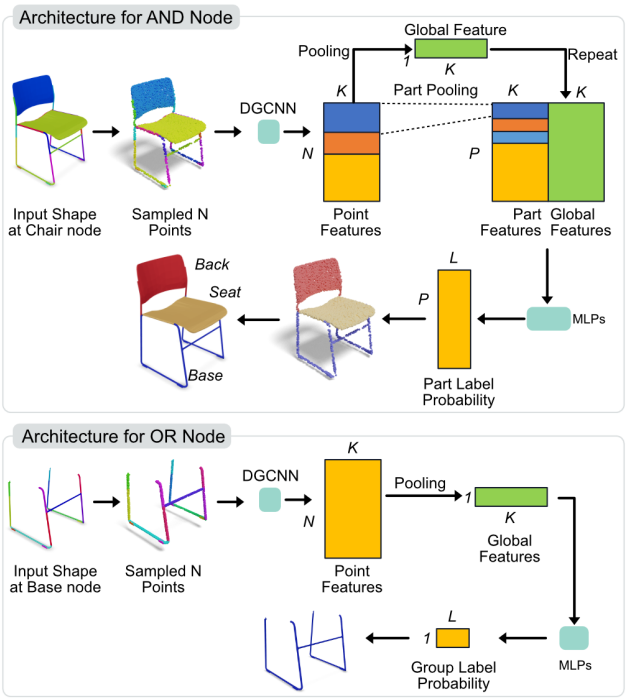

Figure 3. Label proposal architectures. Part label prediction is based on global and local, part-level features at the "AND" nodes and global features only for the "OR" nodes. The "AND" nodes would serve to predict the label for each part and the "OR" nodes would predict labels for the entire input shape. Notion-wise, N is the number of points, P is the number of parts, K is the feature dimension, and L is the number of labels.

tune the network weights. The active learning iteration will stop if the number of left shapes are less than 40. We explain next how we make HC vs. LC proposals.

Label proposal network. Given a set of parts from each shape $s_i \in S$, we sample $N = 8{,}192$ points over the part surfaces and input them into our proposal network. As shown in Fig. 3, our network uses the edge convolution (EdgeConv) module from DGCNN [37] to extract point features from the point clouds. This module consists of five EdgeConv layers whose feature dimensions are 64, 64, 128, 256, and 256, respectively. The point features from all layers are concatenated and passed into a convolution layer. The size of the resulting feature is 512.

Since our tree structure has different types of nodes, we apply two different network structures from DGCNN, where the architecture for "OR" nodes is the same as in the original DGCNN classification network. At an "AND" node, differently from DGCNN which is the SOTA method to predict point labels based on *point* features, we use extra max-pooling operations to aggregate the local *part* features and global shape features, as inspired by NGSP [10]. The concatenated features are fed into the part classifier, which contains 3 MLP layers to predict part labels. As for an "OR" node, labels of the input shape are predicted based on the global features only; see Fig. 3. Since different shapes have different part counts, we set the maximum part count $P_{max} = 150$ during training for batch processing and add zero-padding to the part feature matrix so that all the part feature matrices have the same dimension. More details of our network can be found in the supplementary material.

Label verification. Our goal is to get the labels of each shape verified by a human. To reduce verification and training times, we need to decide which shapes are verified first and how many to verify on each iteration. Shapes in the test set that have similar structure to shapes in the train set are more likely to be labeled well. We can quickly select these well-labeled shapes from the test set and use them to enhance our limited train set, improving the generalization ability of our proposal network. In our method, we first sort the test set based on average predicted part label probabilities from the proposal network of each shape and arrange the test set into batches, each containing $B = 10$ shapes. Users can mark a shape as well-labeled if they see all parts are labeled correctly, or bad-labeled when one part is wrong. We design a user-friendly interface to achieve this, where users can view and rotate a batch of 3D shapes at the same time. The interface also shows the corresponding label colors for reference. Please see supplementary material for more details about our part label verification interface.

Given the sorted shapes, we need to decide how many of them to show the user as HC proposals before fine-tuning the prediction, to achieve a balance between processing efficiency and utilization of user feedback. To this end, we adopt an adaptive strategy: given the sorted proposals, we stop the verification step when the number of human-verified shapes on a batch is below a threshold, which is set to 4 in our experiments. This simple strategy is effective since early batches tend to have higher labeling accuracy than later batches. This way, the number of high confidence proposals presented to the user would adapt to the classification purity of the current test set.

Label modification. Shapes in the test set whose part structures and geometries differ significantly from those in the train set are less likely to be labeled accurately by the network. These shapes can be regarded as outliers in the label regression task, but they can be especially valuable during fine-tuning. Human intervention is inevitable in this situation to correct the labels. Specifically, the shapes provided to the label modification module are collected from two sets: the $Q_1 = 20$ LC proposals with the lowest mean label probabilities, and those shapes that failed user verification more than $H = 2$ times. We ask users to modify the incorrect labels using our modification interface.

In our part annotation interface, labels are organized hierarchically so users can quickly look up the correct labels.

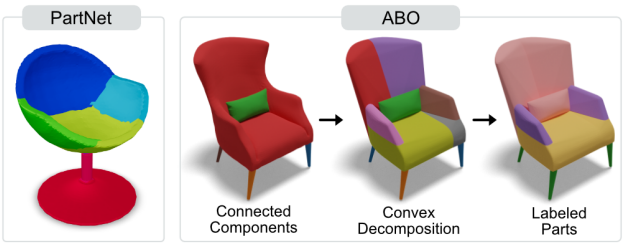

Figure 4. Comparison of pre-segmented parts in PartNet and ABO. ABO shapes were pre-segmented into connected components where each component may correspond to multiple labels, *e.g.*, the back and the arm of the chair are grouped into the same component, as indicated by the red color in the second column. Instead, our part labeling operates on the convex pieces obtained by a decomposition. In the first three figures, the colors are used to distinguish between different parts, and we can see that the parts in PartNet are manually segmented and exhibiting imperfect boundaries. In the last figure, the colors reflect semantic labels.

The label-to-color mapping table is also provided to help find the incorrect labels. More details about this interface are provided in the supplementary material.

Shapes that have been verified and modified by humans are added into the initial training set together to fine-tune our proposal network. On the other hand, we skip shapes excluded from the HC and LC proposals. These shapes are combined with shapes that failed to pass the verification less than H times to go to the next iteration for labeling.

4. Datasets and Metrics

Datasets. The Stanford PartNet [20] dataset is a common choice for fine-grained shape analysis [34, 10, 28]. We follow NGSP [10] and use the four major categories from PartNet for evaluation: chair, table, lamp, and storage furniture. Fine-grained labels in PartNet are organized into three granularity levels, and we adopt the second level in our experiments as in NGSP [10]. We first filter out noisy shapes with erroneous labels. Then from each category, we take 500 shapes and split them into 50/50/400 as train/val/test sets.

We also evaluate HAL3D on the recent Amazon Berkeley Objects (ABO) dataset, which is composed of about 8,000 high-quality 3D shapes created by artists from real product images, where each 3D model is a combination of build-aware connected components. Unlike the 3D shapes from PartNet in which each part corresponds to one single semantic label, a build-aware component from ABO may be appropriately associated with multiple labels, as shown in Fig. 4. To alleviate this issue, we apply an approximate convex decomposition [39] to further decompose each connected component into a fine-grained, i.e., over-segmented, set of convexes, where each convex piece is deemed to receive one and only one part label. Our active labeling is then applied to the set of convexes. When running the convex decomposition algorithm, we use the default parameters from the provided code, except for the concavity threshold setting, which is set as 0.2 in our experiments.

As GT labels are unavailable in ABO, three artists were hired to create them for the decomposed convex components. We built a test set containing 1,800 shapes from five categories: chair, table, lamp, storage furniture, and bed, with 200 for storage and 400 each for the other categories.

Metrics. Active learning systems are typically evaluated from two perspectives: labeling accuracy and the amount of human efforts or involvements. We employ two standard metrics to assess the labeling quality: (1) part label accuracy measuring how many parts are correctly labeled and (2) point cloud segmentation mIoU depicting how much of the shape surface area is correctly labeled.

The amount of human efforts is measured by the consumed time in the steps of label verification and modification. Inspired from [44], we firstly give artists the same training courses and tools, and conduct user studies on PartNet. We report the total recorded labeling time in Tab. 1. We also collect the operations they used in the user study and model the total human interaction time as a function of the number of processed parts and shapes:

$$t = T_v^p(P_v, L) + T_v^f(S_v, L) + T_v^p(P_m^c, L) + T_m^p(P_m^w, L), \tag{1}$$

where P_v is the number of correctly labeled parts in verification, S_v is the number of incorrectly shapes in verification, P_m^c is the number of correctly labeled parts in modification, P_m^w is the number of incorrectly labeled parts in modification, and finally L is the total number of labels.

Function (1) is devised based on several observations: a) users need to go through and verify all the part labels to ensure accuracy; b) they can easily pick up failure cases during verification if any part is incorrectly labeled, without checking other parts; and c) users need to check all the parts in each shape to know whether modification is needed.

The formulas and their coefficients in (2) are obtained via linear regression using the total operation and labeling times collected from all annotators in our user study:

$$\begin{aligned} T_v^p(P_v, L) &= (0.31 + 0.03 * L) * P_v, \\ T_v^f(S_v, L) &= (1.47 + 0.05 * L) * S_v, \\ T_v^p(P_m^c, L) &= (0.31 + 0.03 * L) * P_m^c, \\ T_m^p(P_m^w, L) &= (5.28 + 0.23 * L) * P_m^w, \end{aligned} \tag{2}$$

where T_v^p is the checking time for correctly labeled parts, T_v^f is the verification time for incorrectly labeled shapes, and T_m^p is the time for part label modification. We use these functions to evaluate the human interaction timing, in seconds, for hyper-parameters analysis on PartNet. We also report additional machine time cost from training and fine-tuning in the supplementary material.

5. Experiments

We have implemented our approach in PyTorch. The label proposal network at each node is pre-trained with the Adam optimizer on the training set of PartNet for 250 epochs, where the learning rate is 0.001 and it is decreased by 0.8 every 25 epochs. When fine-tuning the network in HAL3D, the learning rate is fixed to 0.0001, the maximum epoch number is 125, and the validation set is used for early stopping. Our experiments were ran on an NVIDIA Tesla V100 GPU, and the training time varies from 20 minutes to 3 hours according to the number of parts at each node.

5.1. Competing methods

We compare HAL3D, our hierarchical active labeling tool, with two automatic part labeling methods,

• **PartNet** [20], which employs the PointNet++ [25] backbone for point-wise feature extraction and labels parts by a voting scheme over points belonging to the same part.

• **NGSP** [10], which aggregates point-wise features via PointNet++ [25] to obtain part features and then leverages a likelihood-aware beam search for optimization; this represents the state of the art for fine-grained labeling.

And two variants of our labeling approach,

• **Our prop.** is one such variant without using the manual label verification or modification module; it predicts part labels using the label proposal network only.

• **DAL3D** is a deep active learning variant of our approach without using the hierarchical labeling strategy.

5.2. Evaluation on PartNet

All methods presented in this section are pre-trained and tested on the same training and test sets of PartNet.

Quantitative comparison. Tab. 1 compares the three automatic methods, where PartNet is inferior to both NGSP and our label proposal network due to its hand-crafted voting mechanism. Our simple network achieves comparable accuracy compared to NGSP (difference by 0.11 and 3.49 with respect to mIoU and part accuracy, respectively). However, NGSP suffers from significantly longer running time due to the intensive beam search; see the supplementary for a running time comparison. In contrast, our proposal network is effective and efficient, *i.e.*, capable of quickly producing accurate initial labels to reduce the manual workload in verification and modification for active learning.

Tab. 1 also shows that even with the sophisticated design in NGSP, the highest accuracy attained is only 70%. Hence the performance of automatic fine-grained labeling still falls far below that of active learning, where the largest and smallest accuracy gap is 40.77% and 22.49% between NGSP and HAL3D, respectively.

Table 1. Quantitative comparison against the competing labeling methods and variants. In the table, "AL" indicates whether a method uses active learning. Since PartNet, NGSP, and Our prop. are automatic methods, their entries are marked with a '-' instead. "Accu." denotes the part label accuracy, "mIoU" denotes the mean Intersection-over-Union, and "Lab. time" denotes the labeling time (hours in total) for the active learning methods.

Method	AL	Accu.↑	mIoU↑	Lab. time↓ (h)
Chair				
PartNet [20]	-	54.90	31.04	-
NGSP [10]	-	68.98	42.90	-
Our prop.	-	67.45	41.34	-
DAL3D	✓	89.84	72.30	5.99
HAL3D	✓	**94.13**	**84.92**	**4.34**
Table				
PartNet [20]	-	33.21	14.92	-
NGSP [10]	-	47.11	27.94	-
Our prop.	-	42.77	23.50	-
DAL3D	✓	78.53	69.82	9.14
HAL3D	✓	**84.18**	**74.65**	**4.41**
Storage furniture				
PartNet [20]	-	55.82	30.86	-
NGSP [10]	-	70.60	45.79	-
Our prop.	-	69.82	42.43	-
DAL3D	✓	84.57	71.99	4.81
HAL3D	✓	**93.09**	**88.72**	**3.87**
Lamp				
PartNet [20]	-	38.28	11.02	-
NGSP [10]	-	49.76	23.49	-
Our prop.	-	44.77	19.27	-
DAL3D	✓	83.14	69.35	4.52
HAL3D	✓	**90.53**	**79.51**	**2.16**

Table 2. Ablation on key modules of HAL3D for chair category.

Row ID	Prop.	Hier.	Sym.	AL	Lab-T↓	Accu↑
2nd	-	-	-	-	22.05	89.16
3rd	✓	-	-	-	8.65	88.53
4th	✓	✓	✓	-	6.37	93.87
5th	✓	-	✓	✓	5.99	89.84
6th	✓	✓	-	✓	5.21	93.45
7th	✓	✓	✓	✓	**4.34**	**94.13**

HAL3D consistently outperforms our DAL3D variant in all metrics, particularly 2.42 hours faster on average on labeling time, demonstrating the importance of our hierarchical labeling order for reducing manual workload. We also provide qualitative comparison in Fig. 5.

Ablation study. Tab. 2 verifies the contributions of the main components of HAL3D on minimizing human labeling efforts, a key criterion for success for a human-in-the-loop approach. The second to fifth columns indicate:

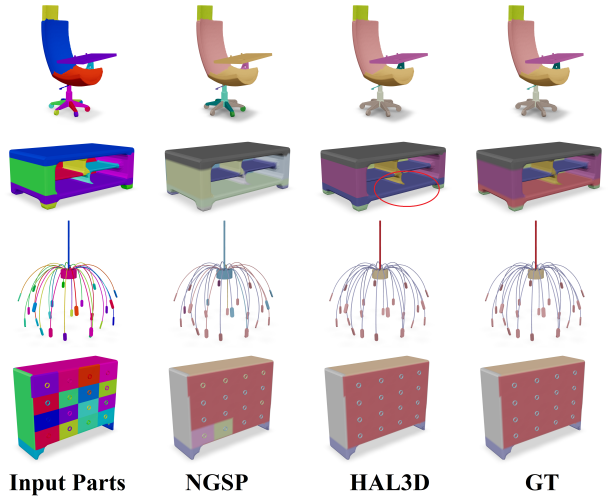

Figure 5. Visual comparison between labeling methods. HAL3D can achieve results approaching GT, barring human errors. The only mis-labeled part by HAL3D here is the bottom panel of the table (red oval), which could be easily labeled as shelf by users.

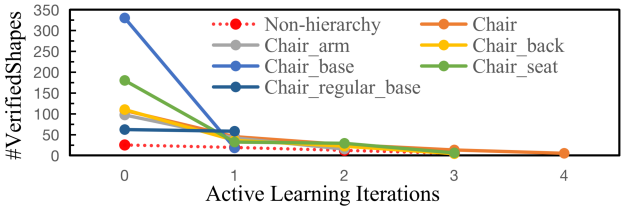

Figure 6. A plot of the number of human-verified shapes (max = 400) at different nodes (only those which existed in more than 100 shapes) of the chair category. The plot shows that with hierarchical active learning, many more shapes pass the verification, hence saving on the costly label modification, especially at higher hierarchy levels (i.e., earlier in the labeling iterations).

- **Prop.** if the label proposal network is used;
- **Hier.** if the hierarchical labeling strategy is used;
- **Sym.** if the symmetry checking is used.
- **AL.** if the active learning is used.

We compare our full approach against five baselines. The second row is a baseline that directly predicts labels by annotating with our label modification interface. The third row is a baseline that firstly uses the label proposal network and then manually modifies incorrect part labels. The fourth, fifth, and sixth rows represent variants of HAL3D without hierarchical labeling, symmetry checking, and active learning, respectively. The results in Tab. 2 clearly justify the design choices in our method, where our full approach (last row) is the most efficient.

To further evaluate the effectiveness of our hierarchical labeling, we plot in Fig. 6 the number of shapes which passed human verification over the active learning iterations. As we can see, HAL3D consistently verified more shapes as correct at all tree nodes when comparing to the baseline that had no hierarchical design, e.g., more than 300 shapes were marked as correct in the first iteration in the chair_base node. Essentially, our hierarchical verification splits the labeling task into smaller and easier tasks compared to the case of non-hierarchical design, which effectively reduces the burden arising from the subsequent and more costly modification step in terms of human effort.

In the supplementary material, we perform ablations on the four hyper parameters and thresholds employed in our method. In all the experiments, we fix these parameters and set them based only on a small dataset. A more careful parameter tuning should lead to better results.

Table 3. Inference time comparison between NGSP and our label proposal network. We report average seconds per shape.

Category	Chair	Table	Lamp	Storage
NGSP	5.410	18.01	12.69	12.87
Our prop.	0.012	0.014	0.012	0.011

Inference time comparison. Although NGSP achieved higher prediction accuracy than our proposal network, as shown in Tab. 1, our network is more efficient and better suited to be incorporated into an active learning framework. Tab. 3 compares the inference times between NGSP and our proposal network. NGSP is highly time-consuming due to the use of beam search, whereas our label proposal network takes only 12ms per shape on average.

5.3. Evaluation on ABO

We apply HAL3D to label our constructed test set for ABO, where the label proposal network is pre-trained on the training set of PartNet. As mentioned in Sec. 4, shapes in ABO are decomposed into convex shapes and HAL3D assigns a semantic label to each decomposed piece. Three artists were hired to use our HAL3D tool and interface for evaluation. Since GT labels are not provided in ABO, we inspect the labeling results from the artists via human cross validation with the final corrected labeling forming the GT. As different artists may report different accuracy and recorded labeling times, we report averages in the paper. The artists worked together with an expert to determine labels for uncertain shapes. The visual results from HAL3D, as shown in Fig. 7 and the supplemental material, were obtained on the test set by taking majority voting from labels provided by different artists.

As shown by quantitative results in Tab. 4, HAL3D can accurately label all categories with a 94% average accuracy using 3-7 hours of manual annotation, demonstrating the effectiveness and efficiency of our method. The reduction on human efforts against manual annotation becomes more significant when comparing against the last four rows in Tab. 4,

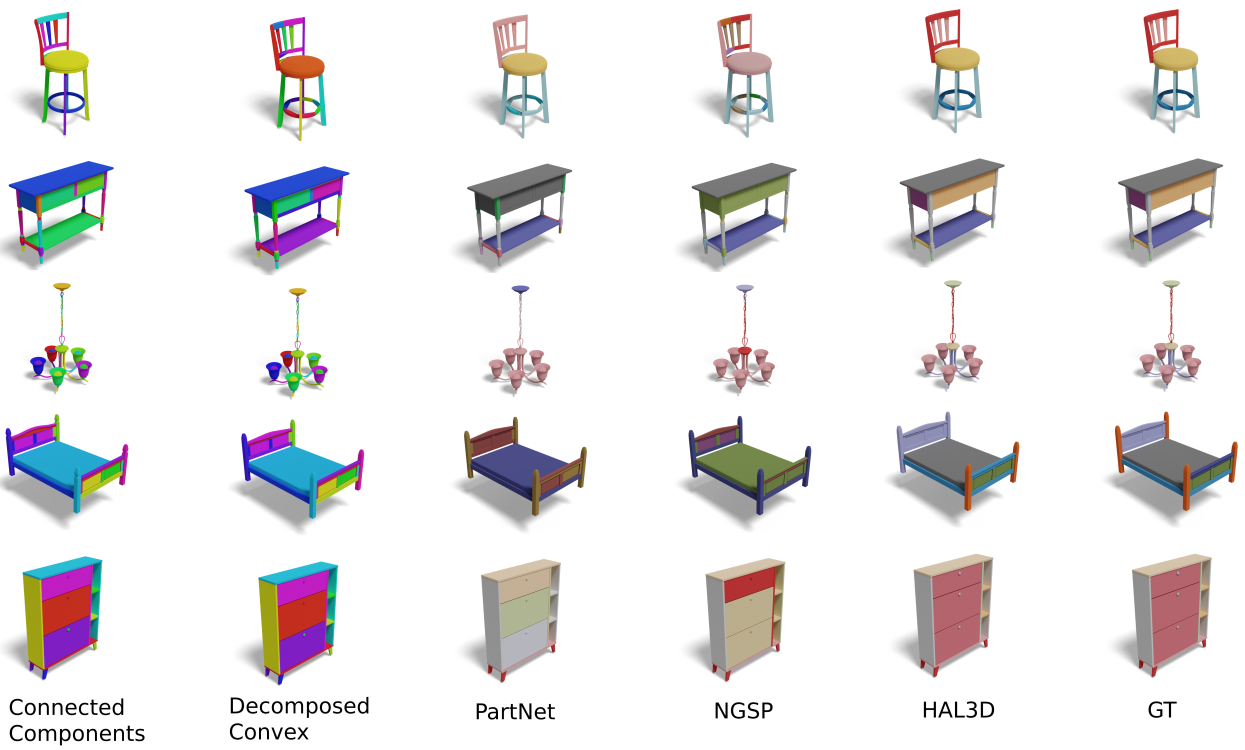

Figure 7. Visual comparison between several part labeling methods on the ABO dataset.

Table 4. Comparison for different methods to pass human verification on test set of ABO. The time is in hours.

	Chair	Table	Lamp	Cabinet	Bed
# shapes	400	400	400	200	400
HAL3D					
Time ↓	**6.46**	**7.41**	**4.38**	**3.34**	**4.51**
Accu. ↑	**93.07**	**92.55**	**96.29**	**94.48**	**93.71**
PartNet + modification					
Time ↓	28.25	33.94	31.53	13.35	25.02
Accu. ↑	88.96	86.51	90.91	90.24	89.37
NGSP + modification					
Time ↓	28.30	34.32	31.56	13.22	24.90
Accu. ↑	88.77	86.40	91.37	90.56	89.93

where we use the same manual modification interface to directly correct the PartNet and NGSP results. Please see supplementary material for results on ABO.

6. Conclusion

We present HAL3D, the first semantic labeling tool that is designed to operate on fine-grained 3D parts *and* achieve close to error-free labeling through full verification by humans, barring human misjudgement. Our human-in-the-loop approach is based on active learning, combining deep label prediction and human inputs to iteratively fine-tune and improve the network prediction. Extensive experiments demonstrate that human efforts are effectively reduced via hierarchical and symmetry-aware active labeling.

HAL3D is currently implemented with DGCNN [37] as the label prediction backbone, but it can be readily replaced by PointNet [24], which was adopted by PartNet [20] and NGSP [10], or a more performant alternative such as Point Transformer [48]. Aside from symmetry, inter-shape part correspondences can also be utilized to reduce labeling costs, assuming that they can be reliably obtained. In addition, our tool does not yet allow splitting of under-segmented parts. Since performing such operations by humans is time-consuming, incorporating auto-splitting schemes into HAL3D deserves future consideration.

We regard our work as only a preliminary step towards deep active learning for fine-grained structural analysis of 3D shapes. A natural focus for future work is to develop a more sophisticated ranking or selection mechanism for label verification and modification to achieve the most effective training for a given level of human effort.

Acknowledgement

We thank all the anonymous reviewers and area chairs for their constructive comments and the Amazon artists for performing the human annotations. Thanks also go to Jef-Aram Van Gorp for helpful discussions and other support.

References

[1] Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S. Yu. Active learning: A survey. In *Data Classification: Algorithms and Applications*, pages 571–597. 3

[2] Marco Attene, Michela Mortara, Michela Spagnuolo, and Bianca Falcidieno. Hierarchical convex approximation of 3D shapes for fast region selection. *Comput. Graph. Forum*, 27(5):1323–1332, 2008. 3

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

[4] Yu Cheng, Kunpeng Zhang, Yusheng Xie, Ankit Agrawal, and Alok Choudhary. On active learning in hierarchical classification. In *Proceedings of the 21st ACM International Conference on Information and knowledge Management*, pages 2467–2470, 2012. 4

[5] David George, Xianghua Xie, Yukun Lai, and Gary KL Tam. A deep learning driven active framework for segmentation of large 3d shape collections. *Computer-Aided Design*, 144:103179, 2022. 3, 4

[6] Aleksey Golovinskiy and Thomas Funkhouser. Consistent segmentation of 3D models. In *Proc. Shape Modeling International (SMI)*, 2009. 3

[7] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: A network with an edge. *ACM TOG*, 38(4), 2019. 3

[8] Yong He, Hongshan Yu, Xiaoyan Liu, Zhengeng Yang, Wei Sun, Yaonan Wang, Qiang Fu, Yanmei Zou, and Ajmal Mian. Deep learning based 3d segmentation: A survey. *CoRR*, abs/2103.05423, 2021. 1, 3

[9] Ruizhen Hu, Lubin Fan, and Ligang Liu. Co-segmentation of 3D shapes via subspace clustering. *Comput. Graph. Forum*, 31(5):1703—-1713, 2012. 3

[10] R. Kenny Jones, Aalia Habib, Rana Hanocka, and Daniel Ritchie. The neurally-guided shape parser: Grammar-based labeling of 3D shape regions with approximate inference. In *CVPR*, 2022. 3, 5, 6, 7, 9

[11] R Kenny Jones, Aalia Habib, and Daniel Ritchie. Shred: 3d shape region decomposition with learned local operations. *ACM Transactions on Graphics (TOG)*, 41(6):1–11, 2022. 3

[12] Oliver Van Kaick, Noa Fish, Yanir Kleiman, Shmuel Asafi, and Daniel Cohen-OR. Shape segmentation by approximate convexity analysis. *ACM TOG*, 34(1). 3

[13] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3D mesh segmentation and labeling. *ACM TOG*, 29(4), 2010. 3

[14] Kacper Kania, Maciej Zięba, and Tomasz Kajdanowicz. UCSG-NET - unsupervised discovering of constructive solid geometry tree. In *NeurIPS*, 2020. 3

[15] Yuki Kawana, Yusuke Mukuta, and Tatsuya Harada. Neural star domain as primitive representation. *Advances in Neural Information Processing Systems*, 33:7875–7886, 2020. 3

[16] Xiao Li, Da Kuang, and Charles X Ling. Active learning for hierarchical text classification. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 14–25. Springer, 2012. 4

[17] Tianqiang Liu, Siddhartha Chaudhuri, Vladimir G. Kim, Qixing Huang, Niloy J. Mitra, and Thomas Funkhouser. Creating consistent scene graphs using a probabilistic grammar. *ACM TOG*, 33, 2014. 3

[18] Niloy Mitra, Michael Wand, Hao Zhang, Daniel Cohen-Or, and Martin Bokeloh. Structure-aware shape processing. In *Eurographics State-of-the-Art Report (STAR)*, 2013. 1

[19] Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. Symmetry in 3D geometry: Extraction and applications. 32(6), 2013. 4

[20] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, June 2019. 2, 3, 4, 6, 7, 9

[21] Chengjie Niu, Manyi Li, Kai Xu, and Hao Zhang. RIM-Net: Recursive implicit fields for unsupervised learning of hierarchical shape structures. In *CVPR*, 2022. 3

[22] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3204–3215, 2021. 3

[23] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3D shape parsing beyond cuboids. In *CVPR*, pages 10344—10353, 2019. 3

[24] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 3, 9

[25] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 3, 7

[26] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning, 2020. 3

[27] Ariel Shamir. A survey on mesh segmentation techniques. *Comput. Graph. Forum*, 27(6):1539–1556. 1

[28] Gopal Sharma, Kangxue Yin, Subhransu Maji, Evangelos Kalogerakis, Or Litany, and Sanja Fidler. MvDeCor: Multi-view dense correspondence learning for fine-grained 3D segmentation. In *ECCV*, 2022. 3, 6

[29] 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. *ACM TOG*, 30(6):126:1–126:9, 2011. 3

[30] Chun-Yu Sun, Qian-Fang Zou, Xin Tong, and Yang Liu. Learning adaptive hierarchical cuboid abstractions of 3D shape collections. 38(6), 2019. 3

[31] Andrew Top, Ghassan Hamarneh, and Rafeef Abugharbieh. Active learning for interactive 3D image segmentation. In *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2011. 3

[32] Oliver van Kaick, Kai Xu, Hao Zhang, Yanzhen Wang, Shuyang Sun, Ariel Shamir, and Daniel Cohen-Or. Co-hierarchical analysis of shape structures. *ACM TOG*, 32(4), 2013. 3

[33] Xiaogang Wang, Xun Sun, Xinyu Cao, Kai Xu, and Bin Zhou. Learning fine-grained segmentation of 3D shapes without part labels. In *CVPR*, 2021. 1, 3

[34] Xiaogang Wang, Bin Zhou, Haiyue Fang, Xiaowu Chen, Qinping Zhao, and Kai Xu. Learning to group and label fine-grained shape components. 37(6), 2018. 6

[35] Yunhai Wang, Shmulik Asafi, Oliver van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Active co-analysis of a set of shapes. *ACM TOG*, 31(6), 2012. 3, 4

[36] Yunhai Wang, Minglun Gong, Tianhua Wang, Daniel Cohen-Or, Hao Zhang, and Baoquan Chen. Projective analysis for 3D shape segmentation. *ACM TOG*, 32(6), 2013. 3

[37] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM TOG*, 2019. 2, 3, 5, 9

[38] Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiquan Cheng, and Yueshan Xiong. Symmetry hierarchy of man-made objects. *Comput. Graph. Forum*, 30(2):287–296, 2011. 3

[39] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM TOG*, 41(4), 2022. 2, 3, 6

[40] Tsung-Han Wu, Yueh-Cheng Liu1, Yu-Kai Huang, Hsin-Ying Lee, Hung-Ting Su, Ping-Chia Huang, and Winston H. Hsu. ReDAL: Region-based and diversity-aware active learning for point cloud semantic segmentation. In *ICCV*, 2021. 3

[41] Zizhao Wu, Ruyang Shou, Yunhai Wang, and Xinguo Liu. Interactive shape co-segmentation via label propagation. In *Proc. of CAD/Graphics*, 2014. 4

[42] Kaizhi Yang and Xuejin Chen. Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. *ACM TOG*, 40(4), 2021. 3

[43] Li Yi, Leonidas Guibas, Aaron Hertzmann, Vladimir G. Kim, Hao Su, and Ersin Yumer. Learning hierarchical shape segmentation and labeling from online repositories. *SIGGRAPH*, 2017. 3

[44] Li Yi, Vladmir G. Kim, I.-Chao Shen Duygu Ceylan, Mengyuan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas J. Guibas. A scalable active framework for region annotation in 3D shape collections. *ACM TOG*, 35(6), 2016. 1, 3, 4, 6

[45] Fenggen Yu, Zhiqin Chen, Manyi Li, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. CAPRI-Net: Learning compact CAD shapes with adaptive primitive assembly. In *CVPR*, 2022. 3

[46] Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. PartNet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In *CVPR*, 2019. 3

[47] Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B. Chan. A comparative survey of deep active learning, 2022. 3

[48] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *CVPR*, 2021. 9

[49] Yang Zhou, Kangxue Yin, Hui Huang, Hao Zhang, Minglun Gong, and Daniel Cohen-Or. Generalized cylinder decomposition. *ACM TOG*, 34(6), 2015. 3

[50] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J. Guibas, and Hao Zhang. AdaCoSeg: Adaptive shape co-segmentation with group consistency loss. In *CVPR*, 2020. 3