SEG&STRUCT: The Interplay Between Part Segmentation and Structure Inference for 3D Shape Parsing

Jeonghyun Kim[1] Kaichun Mo[2] Minhyuk Sung[1†] Woontack Woo[1†]
[1]KAIST [2]Stanford University

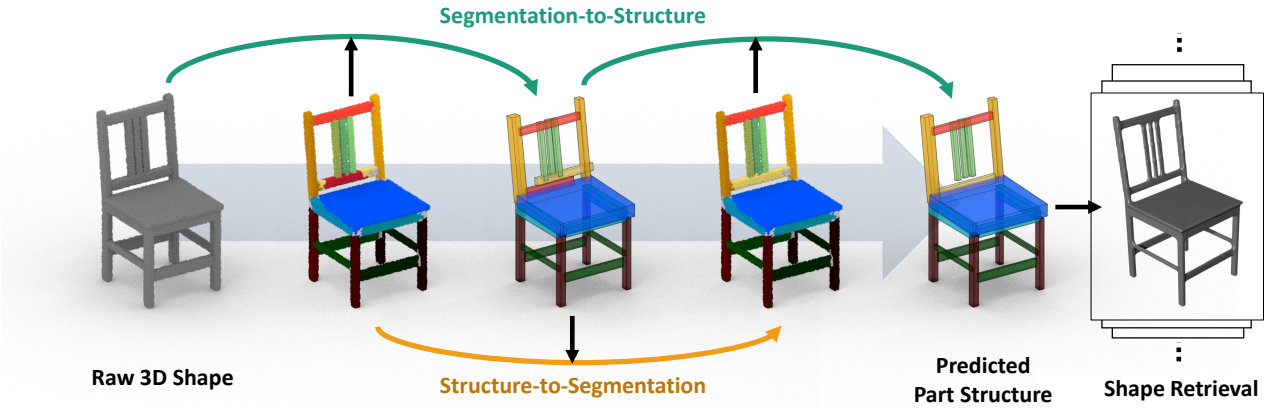

Figure 1. **Overivew.** We introduce SEG&STRUCT, a novel framework for the interplay between part segmentation and structure inference. In the forward path (green), our framework parses a raw 3D shape into part segments and predicts a part structure driven by an established *point-to-part associations*. In the backward path (yellow), the predicted structure is leveraged for segmentation refinement to relax the confusion of part boundaries. Through this interplay, we can achieve a more accurate part structure and improved segmentation. We also showcase that the predicted structure further can be used for shape retrieval by measuring a structural similarity between two shapes.

Abstract

We propose SEG&STRUCT, *a supervised learning framework leveraging the interplay between part segmentation and structure inference and demonstrating their synergy in an integrated framework. Both part segmentation and structure inference have been extensively studied in the recent deep learning literature, while the supervisions used for each task have not been fully exploited to assist the other task. Namely, structure inference has been typically conducted with an autoencoder that does not leverage the point-to-part associations. Also, segmentation has been mostly performed without structural priors that tell the plausibility of the output segments. We present how these two tasks can be best combined while fully utilizing supervision to improve performance. Our framework first decomposes a raw input shape into part segments using an off-the-shelf algorithm, whose outputs are then mapped to nodes in a part hierarchy, establishing point-to-part associations. Following this, ours predicts the structural information, e.g., part bounding boxes and part relationships. Lastly, the segmentation is rectified by examining the confusion of part boundaries using the structure-based part features. Our experimental results based on the StructureNet and PartNet demonstrate that the interplay between two tasks results in remarkable improvements in both tasks: 27.91% in structure inference and 0.5% in segmentation.*

1. Introduction

The importance of compositional understanding of 3D shapes has been reiterated for a long time in computer graphics and computer vision. Until recently, a large body of work has investigated how the part structure of 3D shapes can be utilized in various applications including object recognition [22, 34], shape completion [43, 44], shape editing [23, 28, 29], deformation [55, 3], functional attributes analysis [47, 30, 12, 42], shape retrieval [13, 2, 7, 46], and so on.

Due to the vast range of applications, there also have been lots of previous studies about learning representations of the part structure of 3d shapes, such as GRASS [23] and StructureNet [28]. StructureNet particularly offers highly

†Co-corresponding Authors.

curated information about the part structure including the part abstraction (e.g., part bounding boxes) and the relationships across the parts (e.g., symmetry across sibling parts). A straightforward idea leveraging such supervision for the part structure inference from a raw 3D shape is to build a simple encoder-decoder network, whose encoder takes the raw 3D shape and outputs a latent code, and the decoder takes the code and produces the part structure. The limitation of such an approach is, however, that the neural network does not exploit the supervision of association between the areas on the raw 3D shape and the semantic parts.

3D segmentation is another direction of learning the compositional structure from 3D shapes but focusing on decomposition. It also has been extensively studied in the deep learning literature, particularly to segment object instances from a scene [14, 48]. While recent segmentation techniques provide accurate results in many cases, the methods are still limited to learning the point-to-semantic part relationships without considering the global structure resulting from the segmentation. The global structural priors can assist in pruning such noises affecting the structure, while such as idea has not been explored in recent deep-learning-based approaches.

To overcome the limitations in both tasks, we propose SEG&STRUCT, a framework incorporating the interplay between them while fully exploiting the supervision of both part segmentation and point-to-part associations. Our pipeline is divided into two tasks: 1) a *Segmentation-to-structure inference* and 2) *Structure-to-segmentation refinement*. To infer a part structure from a raw 3d shape, we first extract part segments from the input geometry and then map them to nodes in a part hierarchy. This two-step approach establishing a correspondence between the part segments and nodes in the part hierarchy greatly helps the predicted structure resemble the input geometry. Subsequently, we take an additional step that uses the part structures to draw the candidates of incorrectly split parts and prune out the candidates assisted by the structure-aware features from the previous step. This reverse process and its rectification of the noisy segments close the loop of the pipeline and demonstrate the synergy between the two tasks.

Our experimental results based on the StructureNet dataset demonstrate our method outperforms the baselines using an encoder-decoder network, with significant margins of 27.91%. Our results also show that the segmentation can be better refined with the predicted structural information by 0.5% in the PartNet dataset.

To summarize, our contributions are following:

- We propose a supervised learning framework leveraging the interplay between part segmentation and structure inference and demonstrate its synergy in improving performance.
- We introduce a structure prediction module that takes advantage of a part segmentation network exploiting the supervision of point-to-part associations.
- We also introduce a part segmentation refinement module that learns the confusion of segmentation boundary from the predicted structure.
- Our experimental results demonstrate significant outperformance of our integrated framework compared with previous methods.

2. Related Work

Our work is primarily related to the two threads of research study on segmenting 3D shapes into parts and inferring the structure of 3D shapes. While both directions have been extensively explored in the literature, we propose to tackle the two tasks jointly and leverage their synergy to boost the performance for both tasks in this paper. We briefly review the two fields of study below.

2.1. Part Segmentation on 3D Shapes

Given a raw input 3D geometry, segmenting it into 3D parts is a long-standing important yet challenging research problem in 3D computer graphics and vision. Prior to the popularity of data-driven methods, early works have investigated various optimization based techniques for segmenting 3D mesh inputs into parts [10, 17, 25, 36, 52, 1, 40, 6]. Recently, driven by the advancement of modern machine learning techniques and the availability of large-scale data sets [4, 58, 32, 60], researchers have switched gears to work on data-driven solutions. While many works proposed learning based approaches to perform 3D semantic part segmentation – assigning semantic labels (*e.g.*, chair back, seat and base) to each point or face over the input geometry, such as [20, 53, 38, 18, 59], more related to us are the previous studies on 3D instance part segmentation where all different part instances (*e.g.*, the four chair legs) are separated, *e.g.*, [32, 26, 61]. There are also many works mostly focusing on 3D scene instance segmentation but also demonstrating good results on segmenting 3D shapes into parts, including [49, 50, 11, 14]. Researchers have been pushing state-of-the-art for 3D shape part segmentation using such learning-based methods driven by large-scale training data with part labels. While these works show impressive results, they are limited to decomposing the shape into parts, not connecting the parts to semantic and structural priors such as the part names and the relationships across the parts (e.g., parent-child, symmetry, etc).

In the vast literature on 3D shape part segmentation, only a few past works have explicitly exploited the structural information of 3D shapes. For example, researchers have attempted to leverage part templates [21, 8], part hierarchy [57, 51, 60], and shape grammar [16] to capture the

rich part relationships and constraints. While these works have demonstrated that the 3D part segmentation tasks benefit from estimating the shape structure, they have not explored if predicting 3D shape parts can inversely suggest better shape structural predictions. Our work studies and confirms the synergy between the two tasks.

2.2. 3D Shape Structure Inference

3D objects are often highly structured in their geometry, parts, and rich part relationships. For example, a physically stable chair is often governed by a set of rules specifying some strong relationships and constraints among the shape parts, *e.g.*, the four legs are distributed symmetrically and of the same length. Therefore, researchers have been investigating various approaches to infer the 3D shape structure from raw geometry input. Previous works have proposed different ways to represent the structure of 3D shapes, such as part-based templates [36, 21, 8], part-level or shape-level symmetry [27, 43], shape programs [33, 45, 15], shape grammars [5, 19], *etc.*. After estimating the 3D shape structure, these works can then leverage such information to perform diverse downstream tasks, such as shape generation [45, 15], editing [19, 8] and completion [43].

Recent works have explored designing learning models that represent and model the shape structure as part hierarchies or graphs [23, 57, 62, 54, 28, 9, 56]. Among these works, GRASS [23] is the pioneering work designing a novel learning framework to encode and decode binary part hierarchies, while StructureNet [28] further extends the system to handle *n*-ary hierarchical part trees and the rich part relationships. Follow-up works have tried to predict the hierarchical shape structure from a single input image of the 3D shape [35, 37] or a 3D input point cloud [60, 31]. Given the good performance in modeling 3D shape structure, in this paper, we adopt the hierarchical and relational structural shape representation introduced in StructureNet [28] and focus on investigating the interplay between the two tasks of structure prediction and part segmentation.

3. The Interplay-based Framework

3.1. Overview

Our goal is to build a synergy between the part segmentation and the structure inference via the bi-directional interplay between two tasks in an integrated framework. To this end, we propose two separate networks: a *Segmentation-to-Structure Inference* network (Sec 3.2) and a *Structure-to-Segmentation* network (Sec 3.3). The first network predicts a part structure from a given raw input geometry by exploiting an association between part regions and part nodes in the structure. Afterward, the second network relaxes the confusion of part segment boundaries by leveraging the structure-aware features derived from the earlier stage.

Notations. We denote the raw 3D shape as a point cloud $A = \{a_1, ...a_N\}$ for each point $a \in \mathbb{R}^3$, and a part segmentation output from A as $B = \{b_l\}_{l \in L}$, where b_l is l-th part segment which contains a set of points in the part region X_l and its semantic label y_l, and L is the number of part segments. For the part structure, we use a tree representation $S = (P, \mathbf{H}, \mathbf{R})$ defined in StructureNet [28]. Here, a set of part nodes is represented by P, and relationships for the hierarchical connection and part relations are denoted as $\mathbf{H}$ and $\mathbf{R}$, respectively. For M number of parts $P = \{m_1, ..., m_M\}$, a part node m_i contains a 128-dimensional feature vector and an one-hot encoded semantic label y_i as $m_i = (\mathbf{x}_i, y_i)$. At the end of the structure inference stage, the network predicts a part geometry represented in oriented bounding box parameters $\theta_i = (\mathbf{t}_i, \mathbf{s}_i, q_i)$ based on the part feature $\mathbf{x}_i$. The box parameters θ_i include a translation vector $\mathbf{t}_i \in \mathbb{R}^3$, a scaling vector $\mathbf{s}_i \in \mathbb{R}^3$, and an orientation in unit quaternion $q_i \in \mathbb{H}$. To describe a part relation between two nodes, we use a edge (m_i, m_j, τ) for a part relation type $\tau \in \mathcal{T}$, where $\mathcal{T}$ is a pre-defined set of part relations, e.g. symmetry and adjacency.

3.2. Segmentation-to-Structure Inference

We first propose a parsing-based structure encoder $\mathcal{F}$ that exploits the *segmentation prior*. To infer part structure from a raw 3D shape, one can naively utilize a simple autoencoder network, which encodes the input $A \in \mathbb{R}^{N \times 3}$ into a latent code and decodes it into a part structure S, adopting a decoder similar to StructureNet [28]. However, this approach hardly yields an accurate output. Since the information of 3D shape is just aggregated into a single latent code, the network does not see which point in the input shape belongs to which part in the structure.

To address this, we take a two-step approach that parses the input geometry A into a set of part segments B using a parser backbone ψ and then construct a structure hierarchy $\mathbf{H}$. For ψ, we utilize PointGroup [14], an off-the-shelf algorithm for 3d scene instance segmentation, by treating part instances in a single object as object instances in an indoor scene. To build the structure hierarchy, we treat these parsed segments B as leaf nodes and group them in a bottom-up manner recursively until we get a root node, according to the rule defined in StructureNet [28], which is based on the part labels. For example, a set of part nodes with a semantic label named *leg* has to be grouped together as sibling nodes under a *base* node. While the hierarchy is built starting from leaf nodes, the network also encodes each part geometry into a feature vector $\mathbf{x}_i \in \mathbb{R}^{128}$ and aggregates the vectors of sibling nodes to produce a feature of their parents in the same dimensionality. At the end of the encoding step, we get the root feature vector $\mathbf{x}^{root} \in \mathbb{R}^{128}$. Finally, we can establish a correspondence between part regions in input geometry and part nodes in structure hierar-

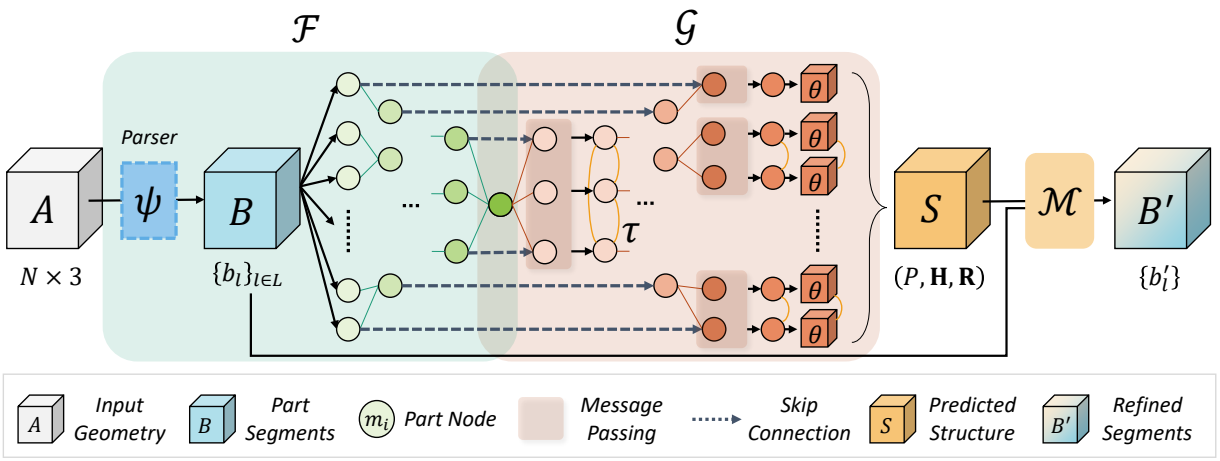

Figure 2. **Overall Framework Architecture.** First, the structure encoder $\mathcal{F}$ decomposes an input geometry A into part segments $B = \{b_l\}$ using the parser ψ. By treating these segments as the leaf nodes, $\mathcal{F}$ constructs a tree hierarchy in a bottom-up manner and aggregates the features of sibling nodes to create a feature vector of their parent node. The following structure decoder $\mathcal{G}$ propagates the root feature throughout the hierarchy in a top-down manner and aggregates the features of parent and sibling nodes indicated by the skip connection. Further, the refinement network $\mathcal{M}$ relaxes the confusion of part boundaries in the segmentation by leveraging the predicted structure.

chy, which largely helps the later network to associate the given shape and the part structure.

Next, we introduce a structure decoder $\mathcal{G}$, which recursively propagates the root feature $\mathbf{x}^{root}$ to leaf nodes, and predicts the part bounding boxes $\{\theta_i\}$ and the part relations $\{(m_i, m_j, \tau)\}$ between two nodes under the same parent node. For $\mathcal{G}$, we adapt a similar decoder network proposed in StructureNet [10], which uses the message passing network across the part features in the hierarchy. Through message passing, the network updates the part features to contain a much broader context across the hierarchy. We denote this updated part feature as $\mathbf{x}'_i \in \mathbb{R}^{128}$. However, it is not feasible to directly use the decoder from StructureNet, which produces the whole part structures by recursively decoding a single latent vector at the root node. When the input is given in the form of the raw geometry, it suffers to predict the existence of part nodes due to the domain gap between the geometry input and the structure output.

To tackle this, our decoder takes advantage of an *explicit* guidance given by the established correspondence between the part regions and part nodes in the previous stage. The previous structure construction step enables us to represent each part segment in the input shape as the hierarchical representation. Based on this, we can associate given part regions to the corresponding part nodes using *skip connection* (Figure 2). This association gives us strong supervision for the structure decoding step and enables the network to know the exact part region to which the part node is related. As a result, our network becomes less dependent on the implicit latent code and infers the part structure resembling the input geometry. We will discuss how this affects the performance of structure inference in the experiments (Sec 4.1).

3.3. Structure-to-Segmentation Refinement

In this section, we introduce a segmentation refinement network $\mathcal{M}$ on top of the structure inference. By predicting the merge operation at the confusion of part regions, our network relaxes noisy regions from the first segmentation output. We found that this task largely depends on the structural context since the local information does not suffice to decide which part has to be merged to other one. To this end, we exploit the features from earlier inference stage, which play a critical role for merge prediction.

To detect the confusion of part boundaries, we apply a simple heuristic using *Intersection over Union* (IoU). An IoU is computed using predicted part boxes and treated as *conflict score*. We filter out the candidates with a larger score than a threshold, i.e. 0.09. Note that if there are multiple confusions for one node, we take only one candidate with the largest score. This means we only consider one-directional merge cases where the merge operation is order-variant, e.g. a merge candidate for one node does not have to be vice-versa, as shown in Figure 3. The valid candidate pairs are assigned to $\mathbf{C}$, a $M \times M$ binary matrix having each row as an index of the part node to be merged and each column as an index of the target node. For example, $\mathbf{C}_{(i,j)} = 1$ describes i-th node have a chance to be merged to j-th node.

To predict merge operations across the nodes in $\mathbf{C}$, we first compute a candidate feature vector $\mathbf{c}_i \in \mathbb{R}^{256}$ for each candidate part by encoding the part segment $b_i = (X_i; y_i)$ using a candidate feature encoder f_c. For f_c, we use a vanilla PointNet [38] architecture where we treat a part segment as a single point.

$$\mathbf{c}_i = f_c([X_i; y_i]) \tag{1}$$

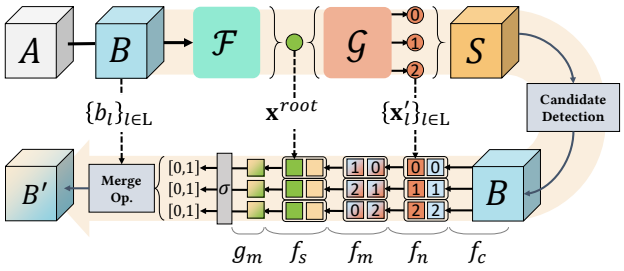

Figure 3. **Structure-induced Segmentation Refinement.** Our network first sample the part segments to be merged in predicted structure S through candidate detection. The number in the box from part segments B indicates the index of each part region. The order of indices inside the boxes grouped together means the direction of merge, e.g., 1-0 means the 1-th part segment can be merged to the 0-th part. Utilizing a series of features from S, refinement network further predicts merge scores for candidate pairs.

However, the feature $\mathbf{c}_i$ contains only fragmentary information of the local part region. Therefore, we opt to update the candidate feature with its corresponding part node feature $\mathbf{x}'_i$ from the inference stage. Using a single linear layer f_n, we aggregate this part-wise structural information to $\mathbf{c}_i$ producing an updated candidate feature as $\tilde{\mathbf{c}}_i = f_n([\mathbf{c}_i; \mathbf{x}_i])$.

After updating the candidate features, we compute a merge feature, an order-variant feature vector to predict merge operation from the candidate node to their defined target node. We denote the merge feature as a 256-dimensional vector $\mathbf{m}_{ij} \in \mathbb{R}^{256}$ and merge feature encoder as f_m, which uses a single linear layer.

$$\mathbf{m}_{ij} = f_m([\tilde{\mathbf{c}}_i; \tilde{\mathbf{c}}_j]) \tag{2}$$

Similar to the candidate feature case, we update $\mathbf{m}_{ij}$ to assist more comprehensive structural context to it from the predicted structure. To this end, we brought the root node feature $\mathbf{x}^{root}$ and treat it as a *structure code* and aggregate it to all merge features. We use an another single linear layer encoder f_s, which outputs the updated merge feature $\tilde{\mathbf{m}}_{ij} = f_s([\mathbf{m}_{ij}; \mathbf{x}^{root}])$.

After the series of the concatenations and feature encoding, we finally predict the probability score of each merge operation $\in [0, 1]$ similar to edge prediction:

$$p^{merge}_{(m_i,m_j)} = \sigma(g_m(\tilde{\mathbf{m}}_{ij})) \tag{3}$$

where g_m decodes the edge feature into logits, σ and p are the sigmoid and probability function, respectively. For the pairs with merge scores larger than the threshold (i.e., 0.7), we perform merge operation to update the first part segmentation by attaching the source part segments to their targets.

3.4. Training and Loss

Our goal is to train a category-specific framework that integrates two networks: structure inference and segmentation refinement. We train two networks separately and use the freezed part segmentation backbone ψ, which is pre-trained in advance.

First, the parameters for the encoder $\mathcal{F}$ and the decoder $\mathcal{G}$ are supervised both at the structure decoding step through backpropagation. Our loss design is mostly brought from StructureNet [28], which computes geometry loss for part bounding boxes, edge prediction loss for part relations, and structure consistency loss to make the relations at parent nodes transfer to their siblings. For more detail, we refer the readers to the original paper and our supplementary.

The training for our refinement network $\mathcal{M}$ is considered as the traditional binary classification problem. However, we face the imbalanced data distribution problem having the majority of merge operation label *True Negative*, which means most of the candidate pairs should not be merged due to the fairly good quality of the previous part segmentation output. To address this, we use a *Focal Loss* [24], a modified version of binary cross entropy loss to handle the data imbalance problem by setting a bigger weight for a label with a sparse number of samples. Please refer to the supplementary for more detail on training.

4. Experimental Results

In this section, we demonstrate our experimental results for two main tasks and one application: structure inference, segmentation refinement, and structure-aware shape retrieval. For more results including ablation study and discussion on failure cases, please refer to our supplementary.

Data Preparation. We prepared two kinds of datasets to test our method: PartNet [32] and StructureNet [28]. ParNet provides point cloud data sampled on surfaces of 3D mesh from ShapeNet [4] and its corresponding semantic-instance part annotation. We use PartNet to train our segmentation backbone ψ and evaluate the performance of segmentation refinement. StructureNet is built upon PartNet with an additional annotation on the structure hierarchy, part bounding boxes, and part relations. Same as StructureNet, we set a maximum number of parts in a subset of the tree as 10 and four types of part relations, i.e. translational, rotational, reflective symmetry, and adjacency. To test our method, we pick three shape categories from StructureNet, which have diverse and complex structures compared to other shapes: *chair*, *table*, and *storage furniture*. Since our framework learns from two datasets at the same time, we filter the invalid shape missing one of the annotations from them. In total, the remaining shapes for chair, table, and storage furniture are 3522, 1802, and 932, respectively. We split these samples into the train and test set following PartNet.

4.1. Evaluation on Structure Inference

Baselines. Since there are no directly comparable methods in previous studies, we build two encoder-decoder base-

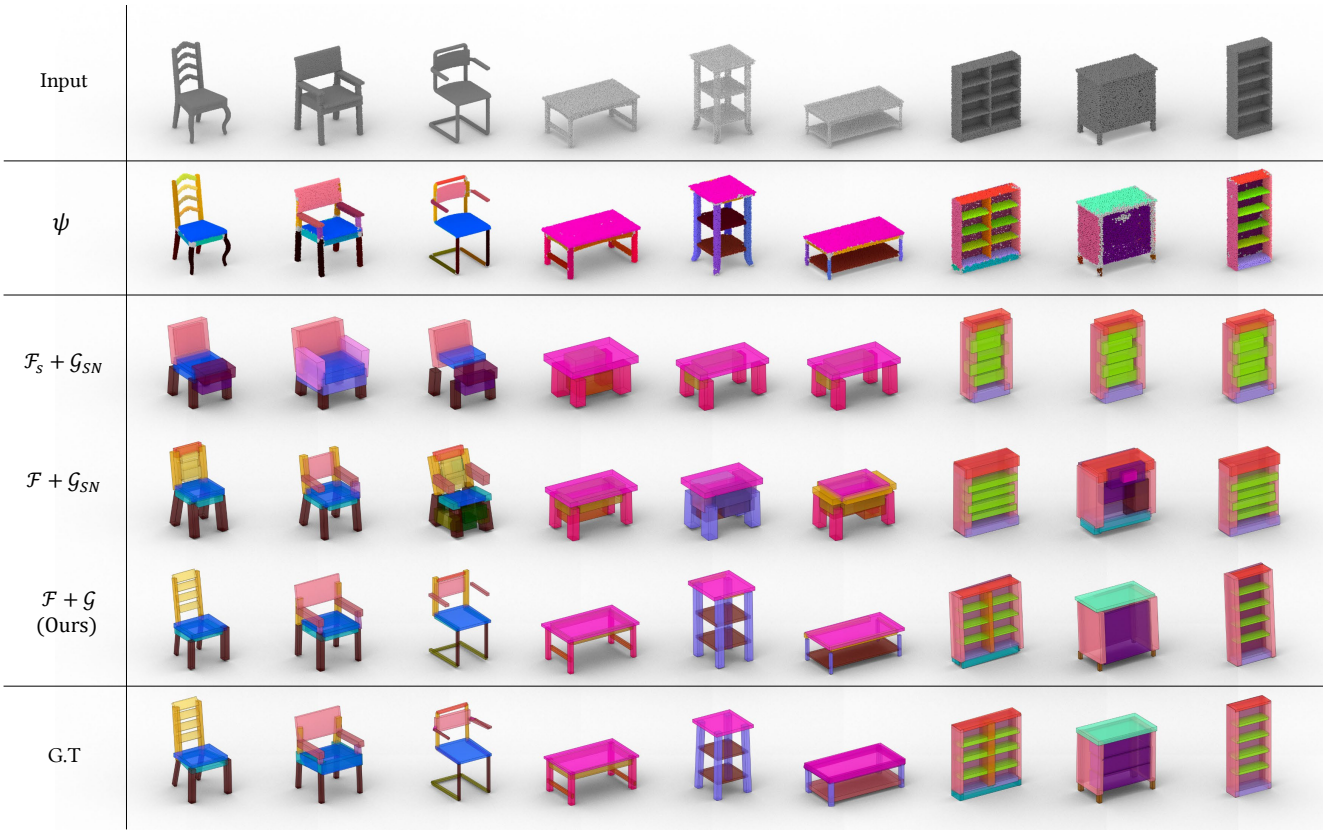

Figure 4. **Qualitative Comparison on Structure Inference.** The top row describes the input 3D shape and the bottom row describes ground-truth. In the second row, the part segmentation outputs from ψ are shown. Compared to other baselines, ours (bottom row) achieves the most accurate part structures also capturing a more diverse set of part structures.

lines. The first one ($\mathcal{F}_s + \mathcal{G}_{SN}$) encodes the input into a single feature vector using a shape encoder $\mathcal{F}_s$ without any segmentation prior and decodes it to predict the whole part structure using a structure decoder $\mathcal{G}_{SN}$. We use PointNet++ [39] for $\mathcal{F}_s$. and use the same decoder from StructureNet [28] for $\mathcal{G}_{SN}$. Meanwhile, the second baseline ($\mathcal{F} + \mathcal{G}_{SN}$) encodes the extracted part segments into a root feature in the hierarchy using our encoder $\mathcal{F}$ and decodes the vector sharing the same decoder $\mathcal{G}_{SN}$, not exploiting the point-to-part association. Both baselines expect the latent vector to contain all the information for the part structure without using one or any priors used in our method.

Metrics. We evaluate our method using two metrics: 1) part prediction accuracy and 2) edge prediction error. The part prediction accuracy measures how accurately are part bounding boxes predicted. We calculate this accuracy using *Average Precision (AP)*, which is widely used in the object detection problem [41]. Since the part semantics are given from segmentation backbone ψ, we use a class-agnostic AP with an IoU threshold 0.25. Note that the correspondence between the prediction and target structure is established only for the leaf nodes. Next, the edge prediction error (EE) measures the quality of classification on the part relations, borrowing the same metric from StructureNet [28]. The lower EE is likely to produce a more consistent structure, which means the predicted part boxes are co-related based on their relationships, e.g., symmetry and adjacency.

Results. We conducted the qualitative and quantitative evaluations and the results are summarized in Figure 4 and Table 1. We first illustrate the results of the qualitative evaluation in Figure 4. As illustrated, our method predicts the most plausible structures for all categories even for the cases of complex input geometry, resembling the target structures compared to other baselines. However, other baselines almost deliver inaccurate appearance even from given input shape and less realistic structure. Even in the case of an approximate structure predicted, they fail to recover the precise part geometry of small and thin parts. The second baseline, in particular, also fails to transfer the information of part regions extracted to the decoding step, although clear correspondence exists. Here, we can observe that naive encoder-decoder methods cannot deliver not only accurate structure prediction but also the diversity of output, where the predicted part structure has a similar appearance from apparently different input shapes. Therefore, we argue that fully utilizing part segmentation priors and the point-to-part

Table 1. **Quantitative Comparison on Structure Inference.** Please note that AP means part prediction accuracy (%) computed by average precision with IoU threshold 0.25, and EE means edge prediction error. The bold text is used for the best results for each column. The columns for key components describe which prior knowledge each method takes, i.e. segmentation prior or skip connection.

Id	Method	Key Comp.		Chair		Table		Storage		Average	
		Seg.	Skip.	AP (%)	EE (↓)	AP (%)	EE (↓)	AP (%)	EE (↓)	AP (%)	EE (↓)
1	$\mathcal{F}_s + \mathcal{G}_{SN}$			5.03	0.682	2.02	0.827	1.07	0.649	2.71	0.720
2	$\mathcal{F} + \mathcal{G}_{SN}$	✓		10.79	0.421	1.28	0.786	1.95	**0.519**	4.68	0.576
3	$\mathcal{F} + \mathcal{G}$ (Ours)	✓	✓	48.41	**0.273**	26.36	0.440	21.57	0.693	32.11	**0.469**
4	Ours $+\mathcal{M}$	✓	✓	**48.86**	**0.273**	**26.82**	**0.436**	**22.08**	0.697	**32.59**	**0.469**

association clearly helps the structure inference.

Next, we demonstrate our quantitative evaluation results in Table 1. Obviously, ours outperforms the naive encoder-decoder baselines in two metrics, leaving significant margins for both. The numbers in the first row describe the results of the first baseline without having any priors that our method uses, i.e., part segmentation and point-to-part association, which almost fails to predict the accurate structures. The other baseline also does not achieve good results either, only with a small improvement from the first baseline. Although this one takes the hierarchically aggregated latent code using both segmentation and hierarchy priors, the structure prediction accuracy slightly increased. By comparing it with ours, we find utilizing the priors in our method helpful showing a significant margin of 27.43% in part prediction accuracy and 0.107 in edge prediction error. Moreover, ours leaves more margin by 27.91% for the refined structure after the segmentation refinement (bottom row). We will discuss this later (Sec 4.2).

4.2. Evaluation on Segmentation Refinement

Baselines and Metric. We compared our method to the other state-of-the-art part segmentation methods, covering SGPN [49], PartNet [32], Probabilistic Embedding (PE) [61], and PointGroup [14]. To show the necessity of our proposed method using structural context, we compare ours to another simple baseline that predicts merge operation directly from the output from part segmentation without using any structural priors. The candidates are detected using the part bounding boxes from a principal component analysis (PCA)-based oriented bounding box estimator. As same as the introduced methods, the quantitative evaluation is performed based on a class-wise *mean Average Precision (mAP)* with IoU threshold 0.5.

Results. We discuss how the proposed method refines the given part segmentation output with quantitative and qualitative evaluations. Please note that we will focus on the *improvement* on the segmentation quality from our backbone ψ since we do not train any additional part segmentation network in our framework.

In Figure 5, we illustrate the visualization of this refinement process for clearance to demonstrate how the merge operation gives us the refined part segmentation. Given an

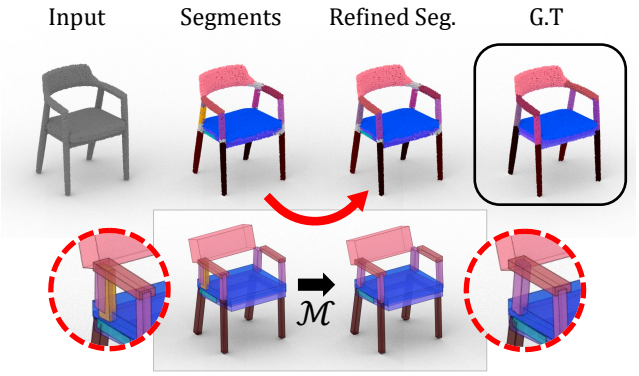

Figure 5. **Structure-to-Segmentation Refinement.** Utilizing structural information, ours rectify the first segmentation through refinement network $\mathcal{M}$. The circle describes a closer look at the region of conflict in the predicted part structure. After merge prediction, we can get refined segmentation updating noisy regions.

initial segmentation output and the prediction structure from it, we first predict merge operation by $\mathcal{M}$ for the candidate nodes detected by the conflict of the part boxes (indicated by the red arrow). By incorporating the set of structure-aware features, ours refines the first part segmentation result by merging the candidate part segment to its target segment. From the visuals, we observe the accurate merge prediction gives us more realistic and clear part segmentation. We provide more examples in Figure 6.

We demonstrate the result of quantitative evaluation in Table 2. Before our merge prediction, we observe that PointGroup itself achieves state-of-the-art performance on part instance segmentation overall. We observe that this quality can be much enhanced after structure-driven merge prediction (bottom row), showing the improved segmentation accuracy by 0.5% in average. However, for our compared baseline, we observe that the accuracy rather decreases on average, where the merge prediction is not aware of the structural information. For the chair category, the number seems not improved that much since most of the chair part segments are relatively small to make a bigger improvement even though with the correct merge prediction. On the other hand, the other categories have the bigger improvement where most of the merge cases occur in the bigger part regions, as shown in Figure 6. Based on the evaluations, we claim that the synergy between the part

Table 2. **Quantitative Results of Segmentation Refinement**. The numbers are calculated by mean average precision (mAP) for each shape category. Compared to the PCA-based baseline which rather decreases the performance, ours with structure-aware features has shown the clear advantage of the proposed method.

	Avg	Chair	Stora.	Table
SGPN [49]	18.5	19.4	21.5	14.6
PartNet [32]	26.8	29.0	27.5	23.9
PE [61]	31.5	34.7	**34.2**	25.5
PointGroup (ψ) [14]	32.0	40.7	26.8	28.5
ψ + PCA-box	31.6	40.7	26.9	27.2
ψ + Ours	**32.5**	**40.8**	27.5	**29.3**

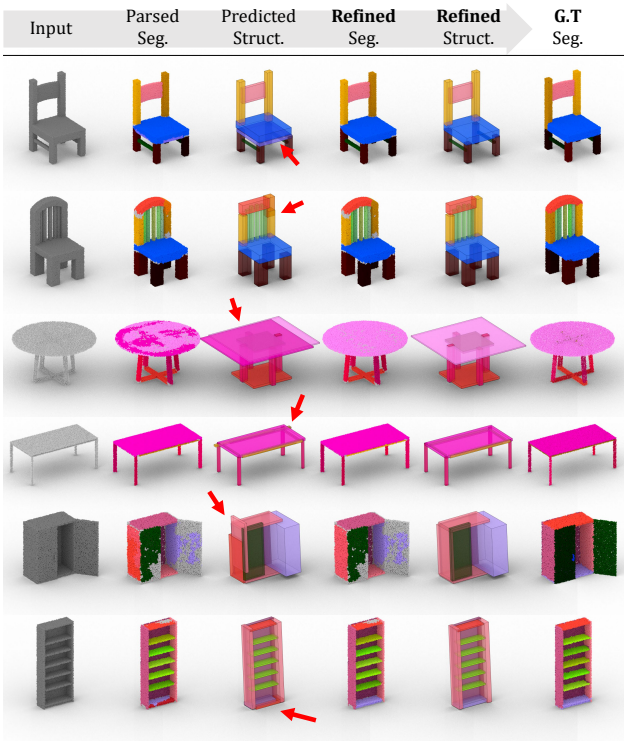

Figure 6. **The Interplay between Part Segmentation and Structure Inference.** The ground-truth part segmentation is at the rightmost column. We point the region of conflict using a red arrow.

segmentation and structure inference enables us to improve both tasks by exploiting the supervision for each task, i.e., segmentation and structural priors.

Moreover, we observe this interplay between two tasks further improves the quality of part structure again. In Table 1, we observe that rectified segmentation further can be used to enhance the quality of part structure once again, improved by 0.48% (bottom row). In Figure 6 (fifth column), we also draw the effectiveness of this refinement enabling our method to achieve more clean and plausible outputs.

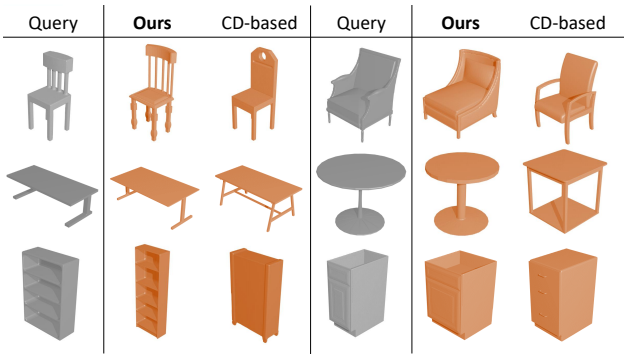

Figure 7. **The Top-1 Retrieval results of Structure-aware vs. CD-based approaches.**

4.3. Structure-Aware Shape Retrieval

As an application based on our proposed framework, we introduce a *structure-aware* shape retrieval. Shape retrieval, which is to search for the most resemble shape in the database given a query shape, has been one of the most practical applications to measure the shape difference. Currently, there has been a typical and dominant approach to comparing two shapes by measuring a *fitting distance*, which is usually computed by chamfer-distance (CD). This yields perceptional failure cases where we seek to find a similar shape in perspective of the semantics and structure.

To tackle this, we propose a structure-driven approach, measuring a *structure difference* that reflects the similarity of semantics between the query shape and shapes in the shape collection. We showcase the results of top-1 shape retrieval, comparing our structure-aware retrieval with the CD-based method in Figure 7. Here, we find that our method does not yield the smallest fitting distance, while the retrieved shapes have more similar *semantic* parts. For more results, we refer readers to the supplementary.

5. Conclusion

We proposed SEG&STRUCT, a framework leveraging the interplay between part segmentation and structure inference in a 3D shape to fully exploit two different supervisions, such as point-to-part associations and hierarchical part structure, and thus improve performance in both tasks making a loop between them. Our experimental results demonstrate that this interplay between segmentation and structure inference enables overcoming the performance barrier of existing methods solving only one of the tasks.

Acknowledgements This work was supported in part by the NRF grant (No. 2021R1A2C2011459) and NST grant (No. CRC 21011) funded by the Korea government(MSIT). Minhyuk Sung acknowledges the support of grants from Adobe, KT, Samsung Electronics, and ETRI.

References

[1] Marco Attene, Sagi Katz, Michela Mortara, Giuseppe Patané, Michela Spagnuolo, and Ayellet Tal. Mesh segmentation-a comparative study. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 7–7. IEEE, 2006.

[2] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2CAD: Learning CAD model alignment in RGB-D scans. In *CVPR*, 2019.

[3] Martin Bokeloh, Michael Wand, Hans-Peter Seidel, and Vladlen Koltun. An algebraic model for parameterized shape editing. *ACM TOG*, 2012.

[4] 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. *CoRR*, abs/1512.03012, 2015.

[5] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas J. Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3D modeling. *ACM Transactions on Graphics (TOG)*, 30:35, 2011.

[6] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3d mesh segmentation. *Acm transactions on graphics (tog)*, 28(3):1–12, 2009.

[7] Manuel Dahnert, Angela Dai, Leonidas J Guibas, and Matthias Nießner. Joint embedding of 3D scan and CAD objects. In *ICCV*, 2019.

[8] Vignesh Ganapathi-Subramanian, Olga Diamanti, Soeren Pirk, Chengcheng Tang, Matthias Niessner, and Leonidas Guibas. Parsing geometry using structure-aware shape templates. In *2018 International Conference on 3D Vision (3DV)*, pages 672–681. IEEE, 2018.

[9] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. SDM-NET: Deep generative network for structured deformable mesh. *ACM TOG*, 2019.

[10] Aleksey Golovinskiy and Thomas Funkhouser. Randomized cuts for 3D mesh analysis. *ACM TOG*, 2008.

[11] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2940–2949, 2020.

[12] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi-Min Hu, and Leonidas J Guibas. MultiBodySync: Multi-body segmentation and motion estimation via 3D scan synchronization. In *CVPR*, 2021.

[13] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *CVPR*, 2017.

[14] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-set point grouping for 3D instance segmentation. In *CVPR*, 2020.

[15] R Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J Mitra, and Daniel Ritchie. Shapeassembly: Learning to generate programs for 3d shape structure synthesis. *ACM Transactions on Graphics (TOG)*, 39(6):1–20, 2020.

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[17] Oliver Van Kaick, Noa Fish, Yanir Kleiman, Shmuel Asafi, and Daniel Cohen-OR. Shape segmentation by approximate convexity analysis. *ACM TOG*, 2015.

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

[19] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012.

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

[21] Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. Learning part-based templates from large collections of 3d shapes. *Transactions on Graphics (Proc. of SIGGRAPH)*, 32(4), 2013.

[22] Young Min Kim, Niloy J. Mitra, Dong-Ming Yan, and Leonidas Guibas. Acquiring 3D indoor environments with variability and repetition. *ACM TOG*, 2012.

[23] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM TOG*, 2017.

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[25] Rong Liu, Hao Zhang, Ariel Shamir, and Daniel Cohen-Or. A part-aware surface metric for shape analysis. In *Computer Graphics Forum*, volume 28, pages 397–406. Wiley Online Library, 2009.

[26] Tiange Luo, Kaichun Mo, Zhiao Huang, Jiarui Xu, Siyu Hu, Liwei Wang, and Hao Su. Learning to Group: A bottom-up framework for 3D part discovery in unseen categories. In *ICLR*, 2020.

[27] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (TOG)*, 25(3):560–568, 2006.

[28] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J. Mitra, and Leonidas J. Guibas. StructureNet: Hierarchical graph networks for 3D shape generation. *ACM TOG*, 2019.

[29] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J Mitra, and Leonidas J Guibas. StructEdit: Learning structural shape variations. In *CVPR*, 2020.

[30] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2Act: From pixels to actions for articulated 3D objects. In *ICCV*, 2021.

[31] Kaichun Mo, He Wang, Xinchen Yan, and Leonidas Guibas. PT2PC: Learning to generate 3D point cloud shapes from part tree conditions. In *ECCV*, 2020.

[32] 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*, 2019.

[33] Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural modeling of buildings. In *ACM SIGGRAPH 2006 Papers*, pages 614–623. 2006.

[34] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM TOG*, 2012.

[35] Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3D shape structure from a single RGB image. In *CVPR*, 2018.

[36] Maks Ovsjanikov, Wilmot Li, Leonidas Guibas, and Niloy J. Mitra. Exploration of continuous variability in collections of 3D shapes. *ACM TOG*, 2011.

[37] Despoina Paschalidou, Luc van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[38] 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.

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

[40] Ariel Shamir. A survey on mesh segmentation techniques. In *Computer graphics forum*, volume 27, pages 1539–1556. Wiley Online Library, 2008.

[41] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3D object detection in RGB-D images. In *CVPR*, 2016.

[42] Minhyuk Sung, Zhenyu Jiang, Panos Achlioptas, Niloy J. Mitra, and Leonidas J. Guibas. DeformSyncNet: Deformation transfer via synchronized shape deformation spaces. *ACM TOG*, 2020.

[43] Minhyuk Sung, Vladimir G Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM TOG*, 2015.

[44] Minhyuk Sung, Hao Su, Vladimir G. Kim, Siddhartha Chaudhuri, and Leonidas Guibas. ComplementMe: Weakly-supervised component suggestions for 3D modeling. *ACM TOG*, 2017.

[45] Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. Learning to infer and execute 3D shape programs. 2019.

[46] Mikaela Angelina Uy, Vladimir G Kim, Minhyuk Sung, Noam Aigerman, Siddhartha Chaudhuri, and Leonidas J Guibas. Joint learning of 3D shape retrieval and deformation. In *CVPR*, 2021.

[47] 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*, 2013.

[48] Thang Vu, Kookhoi Kim, Tung M. Luu, Xuan Thanh Nguyen, and Chang D. Yoo. SoftGroup for 3D instance segmentation on 3D point clouds. In *CVPR*, 2022.

[49] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In *CVPR*, 2018.

[50] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4096–4105, 2019.

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

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

[53] Yunhai Wang, Minglun Gong, Tianhua Wang, Daniel Cohen-Or, Hao Zhang, and Baoquan Chen. Projective analysis for 3d shape segmentation. *ACM Transactions on Graphics (TOG)*, 32(6):1–12, 2013.

[54] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. SAGNet: Structure-aware generative network for 3D-shape modeling. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[55] Weiwei Xu, Jun Wang, KangKang Yin, Kun Zhou, Michiel van de Panne, Falai Chen, and Baining Guo. Joint-aware manipulation of deformable models. In *ACM SIGGRAPH*, 2009.

[56] Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J Guibas, and Lin Gao. DSG-Net: Learning disentangled structure and geometry for 3D shape generation. *CoRR*, abs/2008.05440, 2020.

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

[58] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

[59] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2282–2290, 2017.

[60] 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.

[61] Biao Zhang and Peter Wonka. Point cloud instance segmentation using probabilistic embeddings. In *CVPR*, 2021.

[62] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Renjiao Yi, and Hao Zhang. SCORES: Shape composition with recursive substructure priors. *ACM TOG*, 2018.