RIM-Net: Recursive Implicit Fields for Unsupervised Learning of Hierarchical Shape Structures

Chengjie Niu[1]   Manyi Li[2]   Kai Xu[1*]   Hao Zhang[3]

[1]National University of Defense Technology   [2]Shandong University   [3]Simon Fraser University

Abstract

We introduce RIM-Net, a neural network which learns recursive implicit fields for unsupervised inference of hierarchical shape structures. Our network recursively decomposes an input 3D shape into two parts, resulting in a binary tree hierarchy. Each level of the tree corresponds to an assembly of shape parts, represented as implicit functions, to reconstruct the input shape. At each node of the tree, simultaneous feature decoding and shape decomposition are carried out by their respective feature and part decoders, with weight sharing across the same hierarchy level. As an implicit field decoder, the part decoder is designed to decompose a sub-shape, via a two-way branched reconstruction, where each branch predicts a set of parameters defining a Gaussian to serve as a local point distribution for shape reconstruction. With reconstruction losses accounted for at each hierarchy level and a decomposition loss at each node, our network training does not require any ground-truth segmentations, let alone hierarchies. Through extensive experiments and comparisons to state-of-the-art alternatives, we demonstrate the quality, consistency, and interpretability of hierarchical structural inference by RIM-Net.

1. Introduction

"What emerges is a *multileveled hierarchical structure of parts and wholes*, each of which has a representation of holistic properties as well as component structure."

— Stephen E. Palmar [16]

The recent emergence of neural implicit representations for 3D shapes [5, 12, 17] has stimulated much follow-up. One line of research is motivated by the question of whether self-supervision using the reconstruction loss would allow a neural network to learn a *structured* implicit representation that reveals semantic shape parts. Chen et al. [4] gave a positive answer: by adding a branching layer to the original IM-Net [5] for learning holistic implicit fields, the resulting network can be trained to provide a consistent co-segmentation

---

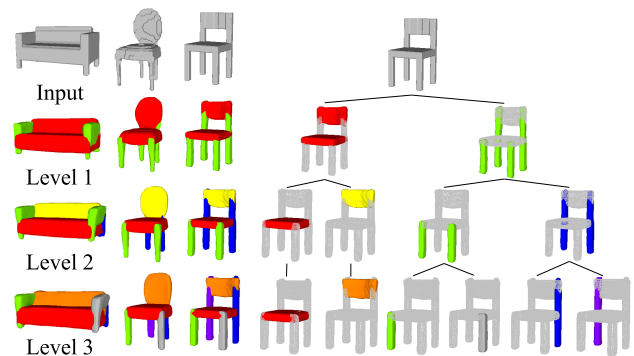*Corresponding author: kevin.kai.xu@gmail.com

Figure 1. Hierarchical shape structures predicted by RIM-Net, which was trained on sofas and chairs together. Parts predicted by the same branch of the network are assigned the same color. We observe structural consistencies across multiple levels, with interpretability and shape semantics revealed by the hierarchies.

over a large set of shapes. However, such a branched autoencoder, or BAE-Net, can only return a coarse segmentation, especially amid structural variations in the set, and the parts obtained are all unorganized.

Cognitive psychology studies have long suggested that human shape perception is based on a *hierarchical* organization of shape parts which encode both per-part properties and part-to-part, as well as part-in-whole, relations [8, 16, 23]. Such a hierarchy is not only a better reflection of object functionality [1], but also a more granular representation that can better capture the structural variability in a diverse shape collection, from coarse organization to finer-level structures. The key question then is whether contemporary neural implicit models are capable of learning structural hierarchies for 3D shapes, *without supervision*.

In this paper, we introduce RIM-Net, a neural network which learns *recursive implicit fields* for *unsupervised* inference of hierarchical shape structures. The network recursively decomposes an input 3D shape into two parts, resulting in a binary tree hierarchy. Each level of the tree corresponds to an assembly of shape parts, represented as implicit functions, to reconstruct the input shape. Our network employs a reconstruction loss, like most neural im-

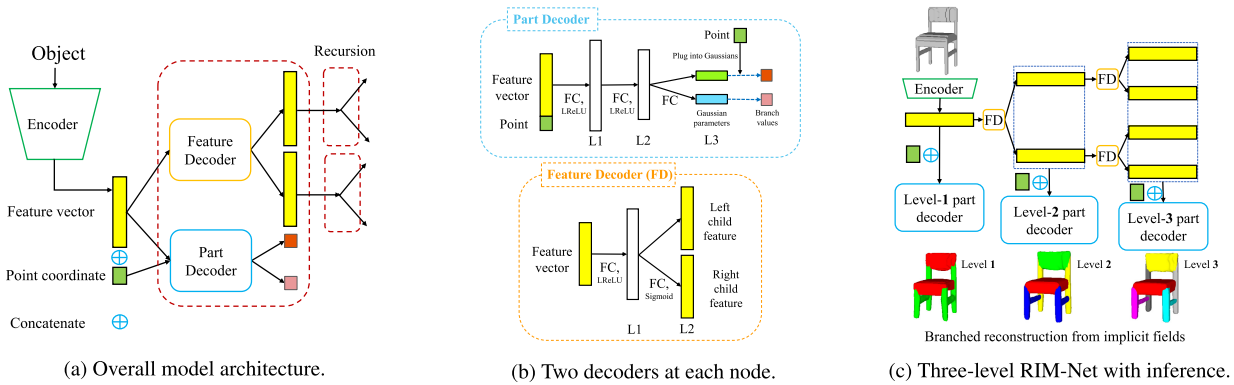(a) Overall model architecture. (b) Two decoders at each node. (c) Three-level RIM-Net with inference.

Figure 2. Overall network architecture of RIM-Net (a), the two decoder modules (b), and a three-level RIM-Net at work on a 3D chair model (c). After encoding the input 3D shape into a feature vector, the network operates in the feature space with simultaneous feature decoding (FD) and part decomposition (via the part decoder) at each node and recursively down the hierarchy. The part decoder predicts an implicit field via a two-way branched reconstruction, where each branch predicts a set of Gaussian parameters to indicate one part.

plicit models [4, 5, 12, 17]. In our setting, the loss is applied at *each* level of the structure hierarchy and summed up. Furthermore, we add a *decomposition loss*, defined at each tree node, to ensure that the shape it represents is a union of the two shapes corresponding to its two child nodes.

Fig. 2 shows the architecture of RIM-Net, which takes as input a 3D voxel shape and first encodes it into a feature vector using a conv-net. The network then operates in the feature space recursively through two modules:

- A *feature decoder* which inputs a feature code and produces two child feature vectors to infer shape parts.

- A *part decoder* which takes a feature code c and a 3D point p as input and eventually outputs an occupancy value for p with respect to the shape part represented by c. The part decoder is designed to decompose this shape part into two sub-parts, by way of a *branched reconstruction*, as in BAE-Net [4]. A key difference to a two-way BAE-Net is that each branch predicts a set of parameters defining a *Gaussian*, instead of a point-wise occupancy. These Gaussians serve as *local point distributions* for shape reconstruction.

Our network is unsupervised as it does not require any ground-truth segmentations, let alone hierarchies. Network weights are shared between all part decoders, respectively all feature decoders, at the same level of the hierarchy. Between different levels, the weights are different.

RIM-Net is the first unsupervised, hierarchical neural implicit model which learns recursive 3D shape decomposition. The idea of using per-point Gaussian inference in the part decoder is also novel. Compared to predicting only a single value per branch, as in BAE-Net [4], the Gaussians offer greater degrees of freedom for the part decoders to adapt to the geometric varieties among parts that can be re-
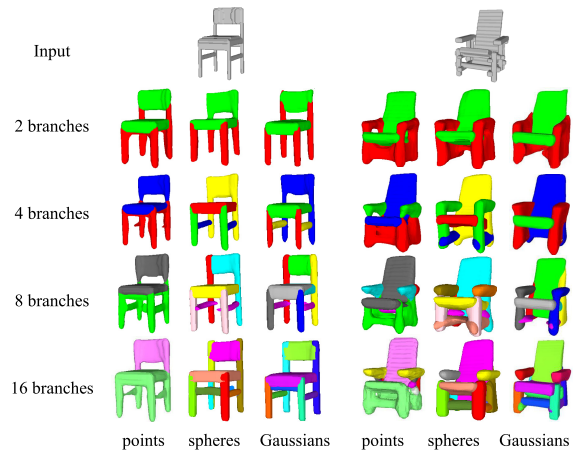

Figure 3. Comparing *single-level* (i.e., no hierarchy) branched neural implicit reconstruction using different local point distributions. Higher degrees of freedom, i.e., from points [4] to spheres, then to Gaussians, improve reconstruction and part inference.

constructed through each branch. This leads to improved reconstruction and segmentation, as shown in Fig. 3.

We evaluate RIM-Net and compare to state-of-the-art methods for hierarchical shape abstraction [4, 19, 22, 24, 26] and structure inference [4, 18]. Experimented tasks include 3D shape autoencoding, co-segmentation, and single-view 3D reconstruction. We also conduct ablation studies to validate our network designs and training strategies.

2. Related Work

Semantic segmentation. Most methods on learning semantic shape segmentation are supervised, e.g., [9, 20, 21, 29]. As a representative weakly supervised approach, Tags2Parts [14] obtains semantic part annotations from weak shape-level tags through a deep neural network, which

11780

is trained to classify the shape as having or lacking a part. AdaCoSeg [30] learns adaptive co-segmentation over a set of shapes using a group consistency loss. RIM-Net is the first unsupervised method for *hierarchical* structure inference to produce fine-grained semantic segmentations.

Shape abstraction. Shape abstraction aims to approximate a shape using a compact set of simple geometric primitives. Due to their simplicity, cuboids have been widely adopted [10, 15, 22, 24, 26, 31]. Tulsiani et al. [24] designed a deep convolutional neural network to predict the shapes of volumetric primitives (VP) and transformation parameters to assemble a given shape; they only considered cuboids as their VPs. In 3D-PRNN, Zou et al. [31] developed a recurrent neural network to generate a sequence of cuboids to construct a 3D shape. More recently, Yang and Chen [26] developed an unsupervised learning method to map a point cloud into a compact cuboid representation by jointly solving the cuboid abstraction (CA) and shape co-segmentation problems. Paschalidou et al. [19] abstracted 3D objects using superquadrics (SQ), which encompass cubes, cylinders, spheres, and ellipsoids, etc. Such a representation is more general and leads to more accurate shape abstractions and faster optimization with continuous parametrization.

While all of the above shape abstraction methods are unsupervised, they do not infer structure hierarchies. The most important distinction to our work is that our goal is not to abstract, but to reconstruct, using a set of implicit shapes.

Learning hierarchies. Notable supervised methods for hierarchical structural analyses include GRASS [10] and PartNet [28], both employing recursive neural networks (RvNNs) to produce trees, and StructureNet [13], which generalizes trees to hierarchical graph organizations. While RIM-Net also consists of a recursively constructed hierarchy of networks, it is not an RvNN, as the latter implies weight sharing between networks at all tree nodes.

Under the category of unsupervised methods, early work by van Kaick et al. [25] introduced the co-hierarchical analysis problem over a shape collection and solved it based on multi-instance clustering. The hierarchical (cuboid) abstraction (HA) method of Sun et al. [22] infers a three-level hierarchy for an input shape by *adaptively* selecting from a hierarchical cuboid representation that is shared by all objects in a given class, where the number of cuboids per level is pre-determined. Most recently, the work of Paschalidou et al. [18] performs hierarchical structure recovery from a single-view image, without part supervision. It recursively extracts superquadric surfaces from the captured shape to construct an unbalanced binary structure hierarchy, as guided by a part prior which clusters points based on part centroids akin to k-means clustering.

In contrast, part inference by RIM-Net is not limited by any primitive type, while such limitations can comprise a method's ability to handle unusual part geometries and structure variabilities. Our network is more robust against these issues, owing to the versatility of implicit field representations, as well as the many degrees of freedoms afforded by the local Gaussian models in our part decoder.

Structured neural implicits. The work by Genova et al. [7] can also be regarded as an abstraction method, where the template shapes are defined by a set of local implicit functions which model scaled and axis-aligned 3D Gaussians. Their use of Gaussians provided inspirations to our design of the part decoder. However, instead of employing a *fixed* number of Gaussians as a template to abstract a shape collection, our network learns *per-point Gaussians* for faithful shape reconstruction. Another closely related work which also proved inspiring was BAE-Net [4], a branched autoencoder for unsupervised or few-shot shape co-segmentation. The part decoder of our network resembles a two-branch BAE-Net, except for the use of per-point Gaussians.

In Section 4, we compare our work to the most relevant methods mentioned above, including VP [24], CA [26], SQ [19], HA [22], and BAE-Net [4].

3. Methods

Given a volume as input, we propose a novel network to predict the structure tree where each node corresponds to an implicit field representing a part in different granularity. Fig. 2a illustrates our network architecture. The encoder is a 3D convolutional neural network [11] for voxels. It maps the input to a feature code of the entire shape, as the root of the structure tree. The decoder is composed of two modules: the feature decoder and the part decoder. The feature decoder takes a feature code as input and outputs two child feature vectors. The part decoder learns the mapping from each node feature to two child parts in 3D, represented as implicit fields. It takes the parent feature and a point coordinate as input, and predicts two values indicating the probability of the input point to be inside each of the two child parts respectively. Therefore, the two modules of the decoder recursively build the structure tree and decode the nodes to parts, forming the recursive implicit fields. The deeper the level, the finer the structure.

Note that the feature decoder and part decoder has shared weights within each level of the tree, but different weights between the levels. Therefore, to train the network for a N-level structure tree, we need a total of N different part decoders and $N - 1$ different feature decoders.

3.1. Part Decoder

We represent each part with a per-point Gaussian which can be regarded as a local point distribution [7]. We devise a novel point-based decoder to recover the per-point Gaussian. It takes the concatenation of a feature code c and a

point coordinate p as input, and outputs two sets of Gaussian parameters, one set per child part. The input point p is plugged into each Gaussian to compute the occupancy probability of p with respect to the corresponding part. Specifically, point p is inside a part if the corresponding probability is larger than a pre-defined threshold. This per-point Gaussian-based decoding generates more smooth and detailed shapes than other popular decoder networks [4, 5, 7].

The part decoder is implemented with a 3-layer MLP as shown in Fig. 2b. The first two fully-connected layers are each followed by a LeakyReLU function. The output of the last layer is split into two branches, each being a scaled, anisotropic 3D Gaussian $\theta_i \in \mathbb{R}^7$ consisting of a scaling factor s_i, a center point $\mathbf{c}_i \in \mathbb{R}^3$, and a per-axis radii $\mathbf{r}_i \in \mathbb{R}^3$. The probability of the input point $p \in \mathbb{R}^3$ w.r.t. each branch is formulated as

$$f(p, \theta_i) = s_i \cdot \exp \left(\sum_{d \in \{1,2,3\}} \frac{-(\mathbf{c}_{i,d} - p_d)^2}{2\mathbf{r}_{i,d}^2} \right), \quad (1)$$

where s_i is clamped into $(0, 1]$. Note s_i cannot take 0 value to avoid a vanishing gradient.

3.2. Feature Decoder

The feature decoder maps one parent feature into two child features. It consists of two fully connected layers, as shown in Fig. 2b. The first layer is followed by LeakyReLU and the second by a Sigmoid. The output is then split into the left and right child features. All feature codes are 128D vectors.

3.3. Network Loss Functions

Our training set includes the point-value pairs sampled from the ground-truth implicit field of the shapes, as in [4]. The ground-truth value of each sample point is one if it is inside the shape and zero otherwise. No part label or structure information is given in the training set.

Reconstruction Loss. Shape parts obtained at each level of the hierarchy should together, via a union, reconstruct the input shape. We define $f_{i,j}(p)$ as the predicted probability of point p being inside the part corresponding to the ith node at the jth level. Then the maximum of the set $\{f_{i,j}(p), i = 1, \dots 2^j\}$ is the probability of point p being inside the union of these parts. Therefore, the reconstruction loss of point p at level j is defined as

$$L_{recon_j}(p) = (y_{gt} - \max(\{f_{i,j}(p)\}))^2, i = 1, \dots, 2^j \quad (2)$$

where $y_{gt} \in \{0, 1\}$ is the ground-truth value for point p in the training set. We sum up the losses of all levels to form the total reconstruction loss:

$$L_{recon}^{(N)}(p) = \sum_{j=1}^{N} L_{recon_j}(p). \quad (3)$$

Decomposition loss. In addition to shape reconstruction, we explicitly enforce a decomposition relation between an internal node and its two children. Geometrically, the part represented by an internal node should be the union of its two child parts. This constraint is imposed by a *decomposition loss*. Specifically, given a point p, its probability of belonging the parent node, $f_{i,j}(p)$, should be equal to the maximum probability of belonging one of the two child nodes, i.e., $f_{2i-1,j+1}(p)$ and $f_{2i,j+1}(p)$. So the hierarchy loss for the ith node at level j is

$$L_{hie_{i,j}}(p) = (f_{i,j}(p) - \max(f_{2i-1,j+1}(p), f_{2i,j+1}(p)))^2. \quad (4)$$

The total decomposition loss at a point p is the sum of the $L_{hie_{i,j}}(p)$ for all internal nodes. For a tree with N levels, we have

$$L_{hie}^{(N)}(p) = \sum_{j=1}^{N-1} \left(\sum_{i=1}^{2^j} L_{hie_{i,j}}(p) \right). \quad (5)$$

Finally, our full loss function is formulated as follows:

$$L^{(N)} = \sum_{p \in P} (\alpha L_{recon}^{(N)}(p) + \beta L_{hie}^{(N)}(p)), \quad (6)$$

where P is the set of all the sample points in the training set and N the number of levels of the structure tree. We set $\alpha = 1$ and $\beta = 10$ during the training process.

3.4. Training Strategy

We propose a progressive learning strategy to ensure stable and efficient network training. To infer an N-level structure tree, our training process proceeds in three stages:

- *Initial training.* The first stage uses the loss function $L^{(1)}$ (Eq. (6)) at the first level to bootstrap the network training. Note that the decomposition loss is always zero when $N = 1$. Hence, this stage only trains the encoder and a part decoder at the first level, with the goal being to generate a shape assembled by two parts.

- *Recursive training.* During this stage, we progressively train each level of the network. While training the feature decoder and part decoder at level j, we fix the weights and bias of the encoder and networks at the previous level, and use the loss $L^{(j)}$ (see Eq. (6)). This process is repeated for levels $j = 2, \dots, N$.

- *Fine-tuning.* Finally, we fine-tune the whole network to learn finer details with the loss $L_{recon}^{(N)}$ in Eq. (3).

4. Results and Evaluation

We first analyze the novel designs of our method in learning good hierarchical shape structures. We then compare with several baseline methods to demonstrate the superiority of our learned structure. We also evaluate our method with downstream applications.
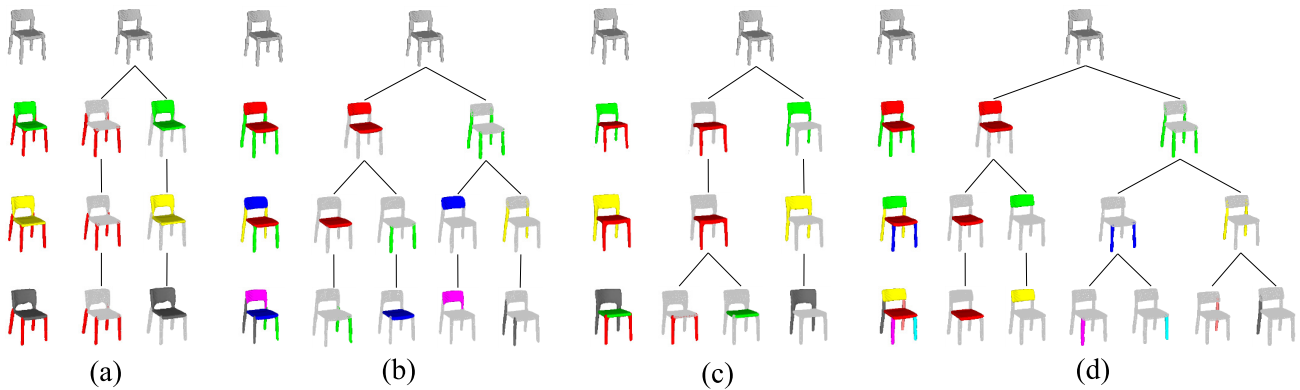
Figure 4. Qualitative results of ablation experiments. 3D hierarchical structure reconstruction with different settings: (a) without per-point Gaussians; (b) without decomposition loss; (c) without progressive training; (d) RIM-Net.

4.1. Training Details and Metrics

We use the ShapeNet [2] subset of Choy et al. [6] in our experiments. The data preparation follows [4] with the same train/test split, shape voxelization, as well as the ground-truth point-value samples. We run our experiments and comparative studies on three object categories: airplanes (2,690), chairs (3,758), and tables (5,271), and train an individual model for each category. During our training, we set the batch size to 1 and run 100K iterations per stage in the progressive training. The output meshes are extracted using Marching Cubes at the resolution of 64^3.

We evaluate our RIM-Net and the related approaches from the aspects of structure learning and shape reconstruction. For the structure learning, we use the mean IoU (mIoU) in unsupervised segmentation task to assess the meaningfulness and consistency of the inferred structures, similar to BAE-Net [4]. For reconstruction, we adopt the popular symmetric Chamfer Distance (CD) and IOU for evaluating the overall accuracy and Light Field Distance (LFD) for visual quality. In particular, CD is computed on 4,096 points uniformly sampled from the reconstructed meshes, and IoU is computed on 32^3 volume.

4.2. Ablation Study

We validate several key designs of our method including the per-point Gaussian-based decoding module, the per-node decomposition term in the network loss, and the progressive training strategy. Fig. 4 gives a qualitative comparison, with distinct colors indicating different parts. The quantitative results are shown in Tab. 1. We see that RIM-Net outperforms other ablated networks in both reconstruction quality and segmentation accuracy.

Per-point Gaussian. We conduct experiments to evaluate the use of per-point Gaussians by RIM-Net for implicit part reconstruction. In Fig. 4(a), we show a structure hierarchy

predicted by the same network architecture as RIM-Net, except that the part decoder only outputs a scalar probability value per point. This is contrasted to the result in (d) by RIM-Net, demonstrating that the per-point Gaussians help obtain finer structures and better visual quality. For example, the decomposition of chair back and seat cannot be split in (a) and there is no more segmentation at level 2 and 3.

For a further demonstration to isolate advantages of using the Gaussians, we remove the hierarchy and only keep the first-level part decoder in RIM-Net, operating on the root feature vector output from the 3D conv-net encoder. The compared architectures are the same except they output different local point distributions for shape reconstruction: points (i.e., single value per point) [4] and spheres (consisting of a scaling factor, a center point, and a radii). We tried different numbers of branches in the part decoder, two and up, to provide a more general picture. Fig. 3 shows that with higher degrees of freedom provided by per-point Gaussians, the network tends to improve on both reconstruction and part inference: it can obtain fine-grained segmentation of the shape, the more part branches, the finer the division, while the method with points fails to segment shape finer.

Decomposition loss. In learning hierarchical structures via reconstruction, we use a decomposition loss to constrain the parent-children decomposition relation in the hierarchy. We train an ablated version of our RIM-Net without decomposition loss, keeping the other settings unchanged. In Fig. 4(b), we see the ablated model fails to maintain the parent-children relation in various levels and the decomposition cannot go finer compared with our full method. This verifies that the reconstruction loss only is not enough to learn a good hierarchical structure.

Progressive training. Our method progressively trains each level and then finetunes the full network. Here, we come up with a baseline without the progressive training

		w/o per-point Gaussian	w/o deco. loss	w/o prog. training	RIM-Net
CD	airplane	0.2336	**0.1881**	0.3403	0.2228
	chair	0.4588	0.4770	0.8620	**0.4125**
	table	0.7804	0.7944	1.3321	**0.7463**
	mean	0.4909	0.4865	0.8448	**0.4605**
IoU	airplane	73.24	74.33	71.52	**74.53**
	chair	78.81	78.69	74.42	**79.61**
	table	73.75	**76.44**	71.68	75.85
	mean	75.27	76.49	72.54	**76.66**
per-label IoU	airplane	56.4	57.7	54.3	**67.8**
	chair	60.3	**84.9**	72.5	81.5
	table	78.5	83.8	76.0	**91.2**
	mean	65.1	75.5	67.6	**80.2**

Table 1. Quantitative results of ablation experiments. The CD values are multiplied by 10^3; lower numbers are better. IoU values are multiplied by 10^2; higher numbers are better.
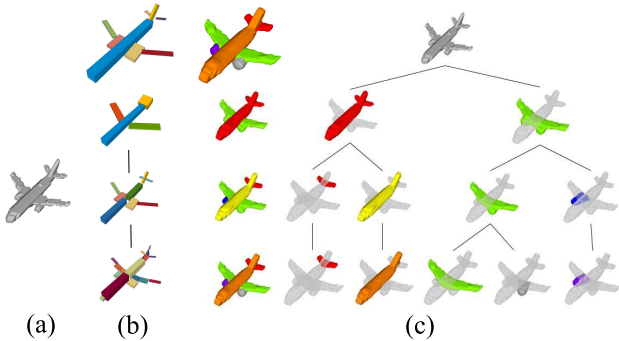

Figure 5. Qualitative comparison of structure hierarchies for a 3D input (a). HA [22] predicts a 3-level hierarchy (b), with the top row as final output. Our predicted hierarchy is in (c): left column is the output at each level and the top one is the output of level 3.

strategy. It trains the full network jointly from scratch. The number of iterations is the same as that for training one level in our progressive method. As shown in Fig. 4(c), our method is unable to achieve a fine-grained and reasonable segmentation without the progressive training.

4.3. Structured 3D Shape Autoencoding

Our method learns hierarchical shape structures without ground-truth structure. We hence compare with alternative self-supervised structure learning methods. We set 3 levels with a maximum of 8 primitives for airplanes and tables. For chairs, we use 4 levels with up to 16 primitives. These maximum primitive count settings are the same as SQ [19] and BAE-Net [4]. For training SQ and CA [26], we use the default training data processing and training settings released by the authors. Results of VP [24] and HA [22] are published by the author of HA using the default training settings (the maximum primitive count is 16 for airplanes, 12 for tables, and 32 for chairs). The comparison is done with the shared portion of the test split by the various methods.

Structure hierarchy. HA [22] is a representative work of learning hierarchical shape structures in a self-supervised manner. Since there is no well-established evaluation metric or protocol available, we provide a qualitative comparison of the learned hierarchies in Fig. 5. HA decomposes a part into smaller ones into deep levels and then selects a set of reasonable parts across various levels as the final output. In our method, the decomposition at each level is always a full reconstruction of the input model. As shown in Fig. 6, our decomposition results look more consistent and meaningful. HA learns shape hierarchies through abstracting shape parts as cuboids, which may overly decompose a geometrically complex part into many small cuboids. In contrast, our method models the geometric variations of shape parts with implicit fields, leading to more consistent structures.

Structure reconstruction. We compare with more structure learning methods. Since most of the structure learning methods do not produce hierarchy, we compare them with the finest level of our recursive implicit fields. In the quantitative comparison in Tab. 2, our method outperforms all alternatives in CD, IoU and LFD, especially on tables and chairs with rich structures. Fig. 6 provides a visual comparison. VP [24] and SQ [19] abstract an object into sets of cuboids or superquadrics, respectively, causing the loss of fine details and the meaningfulness of the parts, especially for chairs. HA [22] learns better structures with its hierarchical cuboid abstraction, but is still unable to handle the shape variations on, e.g., the round table and the four-legged table. CA [26] can generate shape structures with more details, but shares the same drawback as HA. BAE-Net [4] preserves fine details of the objects but only captures coarse parts. In contrast, RIM-Net is able to learn fine-grained parts with accurate reconstruction, under the same setting of maximum part count. For example, BAE-Net fails to segment the individual legs and cannot separate the back from the seat of chairs while our method succeeds.

Structure co-segmentation. Next, we evaluate structure learning from the segmentation point of view, measuring the consistency and meaningfulness of the segments. In particular, we transfer the segmentations onto point clouds as per-point part label and compute the mIoU to evaluate co-segmentation quality, following [4]. Since different methods output different numbers of parts for the same category, we take the ground-truth dataset of [27] and conduct voting-based label snapping as in [4], for a fair comparison. The "leg" and "support" labels are merged as "support" for the table category. From Tab. 3, our method is superior to other unsupervised structure learning methods.

4.4. Single-View Reconstruction

Finally, we apply RIM-Net to infer hierarchical shape structures in 3D from a single image input.
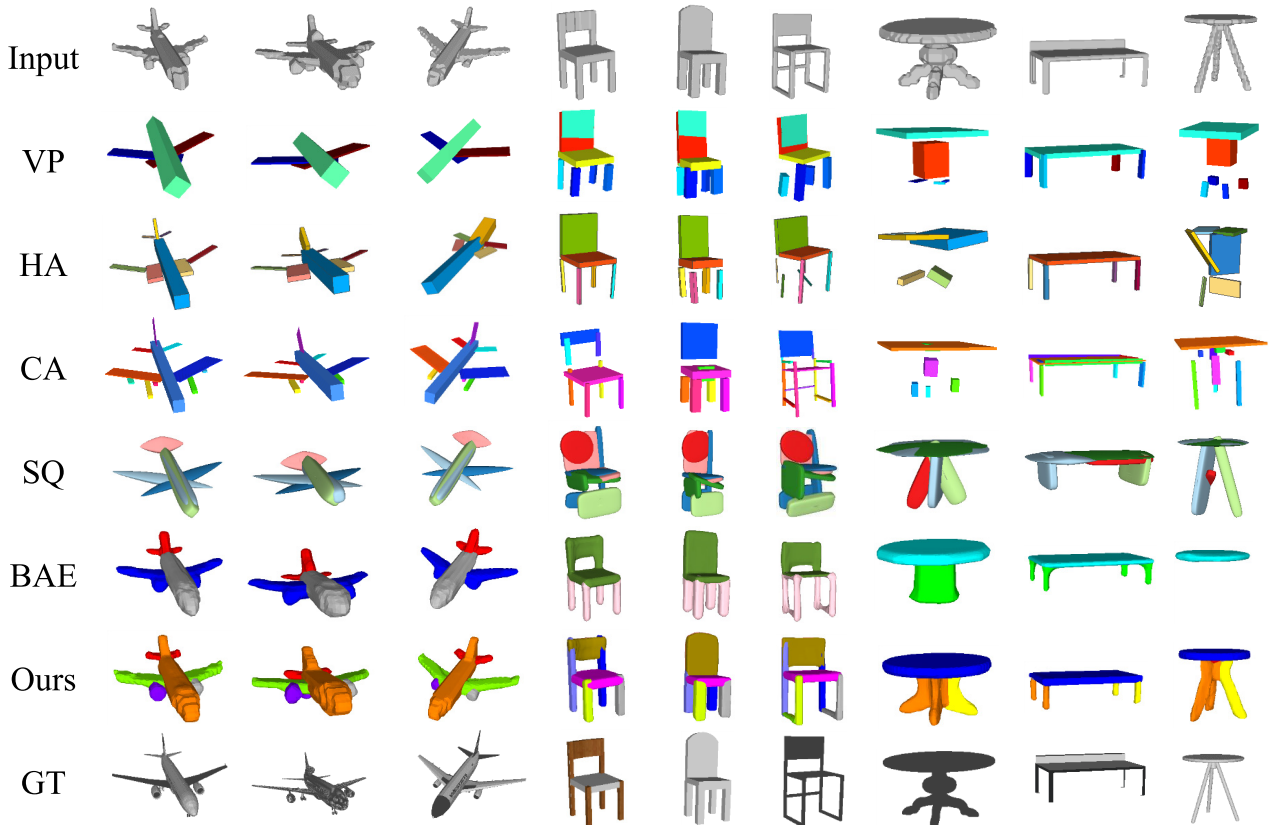
Figure 6. Visual comparison on structure learning methods VP [24], HA [22], CA [26], SQ [19], and BAE [4]. Distinct colors indicate primitive decomposition by the methods. The colored parts visualize segmentation consistency across different shapes in the same category.

	Chamfer Distance (CD)				Intersection over Union (IoU)				Light Field Distance (LFD)			
	airplane	chair	table	Mean	airplane	chair	table	Mean	airplane	chair	table	Mean
VP [24]	0.4587	0.7989	1.1561	0.8046	72.68	73.82	68.61	71.70	8698.83	4936.30	5101.14	6245.42
HA [22]	0.2802	0.7199	1.0034	0.6678	**77.35**	77.56	72.25	75.72	7347.78	4459.68	4557.48	5454.98
CA [26]	0.3609	0.6682	0.9952	0.6748	69.25	72.32	68.46	70.01	6997.06	4600.93	4749.04	5449.01
SQ [19]	0.3601	1.1654	1.3125	0.9460	67.26	69.38	69.94	68.86	7481.02	6745.72	6118.73	6781.82
BAE [4]	0.4276	0.6945	1.2775	0.7999	69.34	72.29	62.57	68.07	6624.12	4139.83	4884.50	5216.15
Ours	**0.2228**	**0.4125**	**0.7463**	**0.4605**	74.53	**79.61**	**75.85**	**76.66**	**5197.79**	**3410.60**	**3223.38**	**3943.92**

Table 2. Quantitative comparison of various structure learning methods. We report CD/LFD (lower is better) and IoU (higher is better).

Structure hierarchy. On unsupervised single-view reconstruction of hierarchical structures, the work of Paschalidou et al. [18] is the most relevant. Their work predicts structure hierarchies based on spatial positions: A part is decomposed in three ways (up-down, front-back, and left-right) without accounting for the meaningfulness of parts. We visually compare the hierarchies of the two methods in Fig. 7. The shape expression ability of superquadrics of [18] is limited. Moreover, their decomposition could cause over-segmentation. For example, the back of the chair is segmented at each level, which is obviously unnecessary. In our hierarchy, each level constitutes a good approximation of the input, and the segments at each level are semantically meaningful. More importantly, their method cannot produce consistent segmentation for shapes of the same category, due to spatial position based decomposition.

Implicit fields reconstruction. We compare to alternative methods, IM-Net [5] and BSP-Net [3], which also employ implicit fields, to demonstrate how our key designs, the hierarchy and the per-point Gaussians, improve reconstruction quality. Different from our method, IM-Net outputs a single implicit field for the holistic shape while BSP-Net

Shape (#parts)	airplane(4)	chair(4)	table(2)
Segmented parts	body, tail, wing, engine	back, seat, leg, arm	top, support
VP [24]	37.6	64.7	62.1
HA [22]	55.6	80.4	67.4
CA [26]	64.2	**82.0**	89.2
SQ [19]	48.9	65.6	77.7
BAE [4]	61.1	65.5	87.0
Ours	**67.8**	81.5	**91.2**

Table 3. Quantitative results of per-label IoU (higher is better).

(a)   (b)   (c)
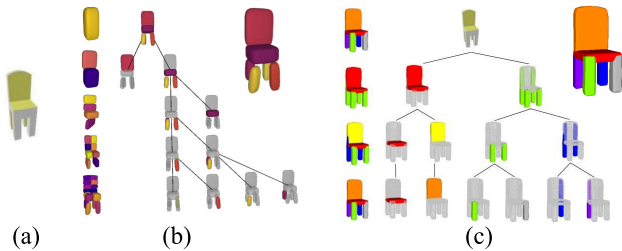
Figure 7. Qualitative comparison of single-view reconstruction of hierarchical structures. Given an input image (a), [18] predicts a top-down hierarchy (b). The left column is the output of each level. The final output is in the top-right corner of the hierarchy. (c) is our predicted structure hierarchy. The output of each level is shown to the left, and the output of level 3 in the top-right corner.

Category	Chamfer Distance (CD)			Light Field Distance (LFD)		
	IM-Net [5]	BSP-Net [3]	Ours	IM-Net [5]	BSP-Net [3]	Ours
airplane	0.4041	0.4716	**0.3993**	5725.02	5397.83	**5252.53**
car	0.6833	0.6262	**0.4699**	2788.74	2834.14	**2686.94**
chair	0.8799	0.7472	**0.7446**	3499.15	**3371.78**	3600.60
rifle	0.4439	0.5708	**0.4095**	6698.26	8834.54	**6313.92**
table	**0.8762**	0.9807	1.1223	**3232.52**	3251.41	3511.95
mean	0.6575	0.6793	**0.6291**	4388.74	4737.94	**4273.19**

Table 4. Quantitative comparison of single view reconstruction.

outputs a set of convexes together forming the field.

For this experiment, we use the same dataset as IM-Net. It contains five representative categories from ShapeNet [2] with rendered views by 3D-R2N2 [6], i.e., airplanes (4,045), cars (7,497), chairs (6,778), rifles (2,373) and tables (8,509). For all methods, we train a separate model for each category. The train/test split we use is the same as [3]. For the training of implicit field SVR, we adopt the training scheme in [3, 5]. We first pretrain a 3D auto-encoder. We then supervise the feature reconstruction quality of the image encoder module through measuring the mean square error (MSE) between the features extracted for the input image and the pre-trained features of the 3D auto-encoder.

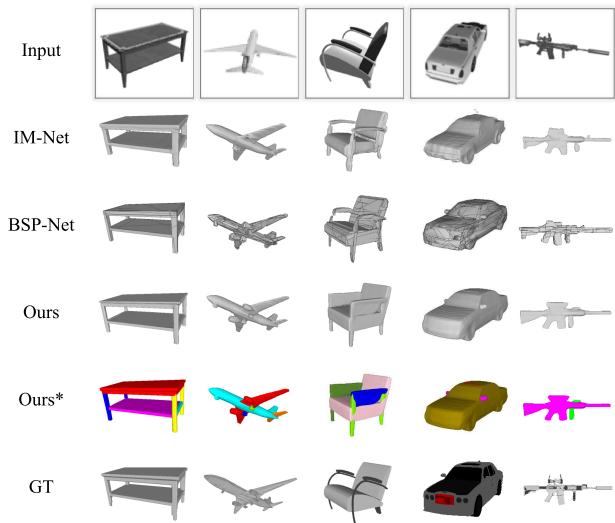As reported in Tab. 4, our method outperforms the two methods in CD and LFD. Fig. 8 shows qualitative results.

Figure 8. Single view reconstruction results of various methods. We show all the reconstructed results in gray color. "Ours*" means our results with colored parts reflecting the inferred structures. "GT" denotes ground-truth objects.

As demonstrated in the visual results, our method achieves comparable reconstruction quality with the state of the arts, while additionally producing hierarchical structures without any supervision. The inferred segmentations by our method are shown in colors in the row of "Ours*".

5. Conclusion, limitation, and future work

We have introduced RIM-Net, a learning-based hierarchical framework that generates shape with implicit primitives without requiring any part-level labels for training. The per-point Gaussian as local point distribution is inserted into the point label prediction process, which effectively enhances the part decomposition and detail generation. A constraint of our method is that the hierarchy topologies are arbitrary and there is no standard to measure them.

An interesting future study will be to explore whether our model can be plugged into various learning-based prediction methods to generate hierarchical primitives for objects. Another future improvement is to use more constraints to construct the object in a diverse hierarchy, like various part prior, symmetry of shape.

Acknowledgments

References

[1] L. Carlson-Radvansky, E. Covey, and K. Lattanzi. "What" effects on "where": Functional influence on spatial relations. *Psychological Science*, 10(6):519–521, 1999. 1

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5, 8

[3] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 45–54, 2020. 7, 8

[4] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-Net: Branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8490–8499, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 1, 2, 4, 7, 8

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 5, 8

[7] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7154–7164, 2019. 3, 4

[8] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, pages 65–96, 1984. 1

[9] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3d shape segmentation with projective convolutional networks. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3779–3788, 2017. 2

[10] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 3

[11] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 3

[12] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 2

[13] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG)*, 39(1):1–19, 2019. 3

[14] Sanjeev Muralikrishnan, Vladimir G Kim, and Siddhartha Chaudhuri. Tags2parts: Discovering semantic regions from shape tags. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2926–2935, 2018. 2

[15] Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4521–4529, 2018. 3

[16] Stephen E. Palmer. Hierarchical structure in perceptual representation. *Cognitive Psychology*, 9(4):441–474, 1977. 1

[17] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 1, 2

[18] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single RGB image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1060–1070, 2020. 2, 3, 7, 8

[19] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10344–10353, 2019. 2, 3, 6, 7, 8

[20] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[21] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5105–5114, 2017. 2

[22] Chun-Yu Sun, Qian-Fang Zou, Xin Tong, and Yang Liu. Learning adaptive hierarchical cuboid abstractions of 3d shape collections. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019. 2, 3, 6, 7, 8

[23] D. W. Thompson. *On Growth and Form*. Dover, 1992. 1

[24] Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2635–2643, 2017. 2, 3, 6, 7, 8

[25] Oliver van Kaick, Kai Xu, Hao Zhang, Yanzhen Wang, Shuyang Sun, Ariel Shamir, and Daniel Cohen-Or. Cohierarchical analysis of shape structures. *ACM Transactions on Graphics (Special Issue of SIGGRAPH)*, 32(4):Article 69, 2013. 3

[26] Kaizhi Yang and Xuejin Chen. Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. *ACM Transactions on Graphics (TOG)*, 40(152):1–11, 2021. 2, 3, 6, 7, 8

[27] 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. 6

[28] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9491–9500, 2019. 3

[29] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1018, 2019. 2

[30] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J Guibas, and Hao Zhang. Adacoseg: Adaptive shape co-segmentation with group consistency loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8543–8552, 2020. 3

[31] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 900–909, 2017. 3