Hierarchy Denoising Recursive Autoencoders for 3D Scene Layout Prediction

Yifei Shi[1], Angel Xuan Chang[2], Zhelun Wu[3], Manolis Savva[2] and Kai Xu[*1]

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

Abstract

Indoor scenes exhibit rich hierarchical structure in 3D object layouts. Many tasks in 3D scene understanding can benefit from reasoning jointly about the hierarchical context of a scene, and the identities of objects. We present a variational denoising recursive autoencoder (VDRAE) that generates and iteratively refines a hierarchical representation of 3D object layouts, interleaving bottom-up encoding for context aggregation and top-down decoding for propagation. We train our VDRAE on large-scale 3D scene datasets to predict both instance-level segmentations and a 3D object detections from an over-segmentation of an input point cloud. We show that our VDRAE improves object detection performance on real-world 3D point cloud datasets compared to baselines from prior work.

1. Introduction

The role of *context* in 3D scene understanding is central. Much prior work has focused on leveraging contextual cues to improve performance on various perception tasks such as object categorization [12], semantic segmentation [35], and object relation graph inference from images [62]. However, the benefit of hierarchical context priors in 3D object detection and 3D instance-level segmentation using deep learning is significantly less explored. A key challenge in using deep network formulations for capturing the patterns of hierarchical object layout is that these patterns involve changing numbers of objects with varying semantic identities and relative positions. In this paper, we propose a recursive autoencoder[1] (RAE) approach that is trained to predict and iteratively "denoise" a hierarchical 3D object layout for an entire scene, inducing 3D object detections and object instance segmentations on an input point cloud.

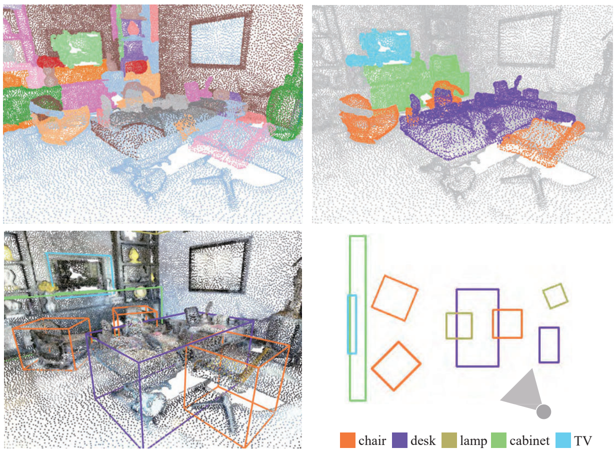

Figure 1: We present a hierarchy-aware variational denoising recursive autoencoder (VDRAE) for predicting 3D object layouts. The input is a point cloud which we oversegment (top left). Our VDRAE constructs and refines a 3D object hierarchy, inducing semantic segmentations (top right, category-colored point cloud), and 3D instance oriented bounding boxes (bottom). The refined 3D bounding boxes tightly and fully contain observed objects.

Recent work has demonstrated that encoding the context in 3D scene layouts using a set of pre-specified "scene templates" with restricted sets of present objects can lead to improvements in 3D scene layout estimation and 3D object detection [63]. However, manually specifying templates of scene layouts to capture the diversity of real 3D environments is a challenging and expensive endeavor. Real environments are hierarchical: buildings contain rooms such as kitchens, rooms contain functional regions such as dining table arrangements, and functional regions contain arrangements of objects such as plates and cutlery. This implies that explicit representation of the *hierarchy structure* of 3D scenes can benefit 3D scene understanding tasks such as object detection and 3D layout prediction.

[*] corresponding author
[1] Also known as a recursive neural network (RvNN) autoencoder.

Given a scene represented as a point cloud, we first perform an over-segmentation. Our RvNN is trained to encode all segments in a *bottom-up context aggregation* of per-segment information and inter-segment relations, forming a segment hierarchy. The decoding phase performs a *top-down context propagation* to regenerate subtrees of the hierarchy and generate object proposals. This encoding-decoding refinement process is iterated, interleaving context aggregation and hierarchy refinement. By training our denoising autoencoder in a generative fashion, this process converges to a refined set of object proposals whose layout lies in the manifold of valid scene layouts learned by the generative model (Figure 1). In summary, our approach is an iterative *3D object layout denoising autoencoder* which generates and refines object proposals by recursive context aggregation and propagation within the inferred hierarchy structure. We make the following contributions:

- We predict and refine a multi-object 3D scene layout for an input point cloud, using hierarchical context aggregation and propagation based on a denoising recursive autoencoder (DRAE).
- We learn a variational DRAE (VDRAE) to model the manifold of valid object layouts, thus facilitating layout optimization through an iterative infer-and-generate process.
- We demonstrate that our approach improves 3D object detection performance on large reconstructed 3D indoor scene datasets.

2. Related work

Our goal is to improve 3D object detection by leveraging contextual information with a hierarchical representation of a 3D scene. Here we focus on reviewing the most relevant work in object detection. We describe prior work that uses context during object detection, work on object detection in 3D, and hierarchical context modeling.

Object detection in 2D. Object detection has long been recognized as an important problem in computer vision with much prior work in the 2D domain [11, 13, 14, 16, 25, 33, 41, 42]. Using contextual information to improve object detection performance has also been studied extensively [5, 6, 17, 38, 54]. Choi et al. [7] show that using contextual information enables predictions in 3D from RGB images. More recently, Zellers et al. [62] show improved object detections by learning a global context using a scene graph representation. These approaches operate in 2D and are subject to the viewpoint dependency of single image inputs. The limited field of view and information loss during projection can significantly limit the benefit of contextual information.

Object detection with RGB-D. The availability of commodity RGB-D sensors led to significant advances in 3D bounding box detection from RGB-D image inputs [9, 15, 51, 52]. However, at test time, these object detection algorithms still only look at a localized region from a single view input and do not consider relationships between objects (i.e. contextual information). There is another line of work that performs contextual reasoning on single view RGB-D image inputs [23, 29, 43, 45, 48, 56, 63] by leveraging patterns of multi-object primitives or point cloud segments to infer and classify small-scale 3D layouts. Zhang et al. [63] model rooms using four predefined templates (each defining a set of objects that may appear) to detect objects in RGB-D image inputs. If the observed room contains objects that are not in the initial template, they cannot be detected. Another line of work on street and urban RGB-D data uses bird's eye view representation to capture context for 3D object detection [2, 49, 59]. In contrast, we operate with fused 3D point cloud data of entire rooms, and learn a generative model of 3D scene layouts from a hierarchical representation.

Object detection in 3D point clouds. Recently, the availability of large scale datasets [1, 3, 8] has fostered advances in 3D scene understanding [58]. There has been an explosion of methods that focus on the semantic segmentation of point clouds [10, 18, 19, 24, 28, 30, 39, 53, 60]. However, there is far less work addressing instance segmentation or object detection in fused 3D point clouds, at room-scale or larger. Both Qi et al. [39], Wang et al. [55] propose connected component-based heuristic approaches to convert semantic segmentations to instances. Wang et al. [55] is the state-of-the-art 3D point cloud instance segmentation method. They use a learned point similarity as a proxy for context. A related earlier line of work segments a point cloud or 3D mesh input into individual objects and then retrieves matching models from a CAD database [4, 27, 36, 45] to create a synthetic 3D representation of the input scene. Our approach directly represents object detections as a hierarchy of 3D bounding boxes and is motivated by the observation that at the scale of 3D rooms, modeling the hierarchical context of the 3D object layout becomes important.

Hierarchical context in 3D. Hierarchical representations have been used to learn grammars in natural language and images [50], 2D scenes [46], 3D shapes [26, 61], and 3D scenes [32]. A related line of work parses RGB or RGB-D scenes hierarchically using And-Or graphs [20, 21, 34, 40, 65] for a variety of tasks. For full 3D scenes, there has been very limited amount of available training data with ground truth hierarchy annotations. Therefore, prior work in hierarchical parsing of 3D scenes does not utilize high capacity deep learning models. For example, Liu et al. [32] is lim-

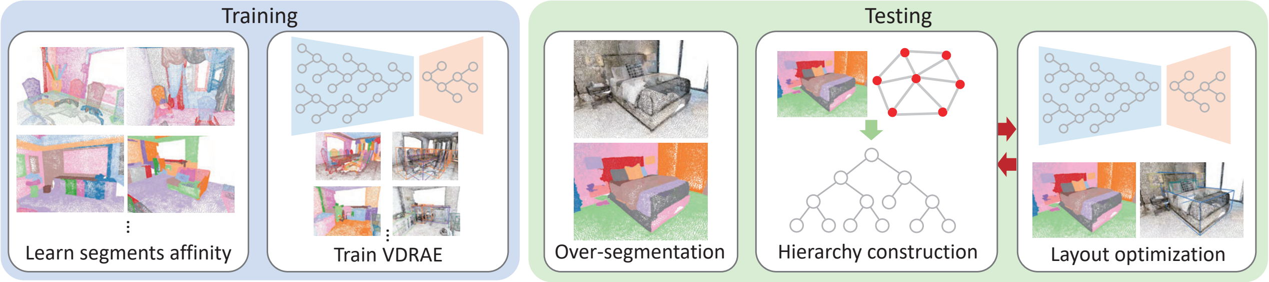

Figure 2: Our system involves two neural net components: a segment-segment affinity prediction network which we use to construct hierarchical groupings of 3D objects, and a variational denoising recursive autoencoder (VDRAE) which refines the 3D object hierarchies. At test time, the affinity prediction network is used to predict segment-segment affinities. We construct a hierarchy from the segment affinity graph using normalized graph-cuts. The VDRAE then encodes this hierarchy to aggregate contextual queues and decodes it to propagate information between nodes. These two stages are iterated to produce a denoised set of 3D object detections and instance segmentations that better match the input scene.

ited to training and testing on a few dozens of manually annotated scenes. Zhao and Zhu [64] evaluated on only 534 images. In this paper, we use a recursive autoencoder neural network to learn a hierarchical representation for the entire 3D scene directly from large-scale scene datasets [1, 3].

3. Method

The input to our method is a 3D point cloud representing an indoor scene. The output is a set $\mathcal{B}$ of objects represented by oriented bounding boxes (OBBs), each with a category label. We start from an initial over-segmentation $\mathcal{S}$ containing candidate object parts (Section 3.1). We then predict segment pair affinities and use a normalized cuts [47] approach to construct an initial hierarchy h used for context propagation (Section 3.2). Having built the hierarchy, we iteratively refine the 3D object detections and the hierarchy based on a recursive autoencoder network which adjusts the structure of the hierarchy and its nodes to produce 3D object detections at the leaf nodes (Section 3.3). We call the combination of the object detections and the constructed hierarchy $\{\mathcal{B}, h\}$ the 3D scene layout. Our output set of labeled bounding boxes $\mathcal{B}$ contains a category label for all object detections, or a label indicating a particular box is not an object. Figure 2 shows an overview of our method.

3.1. Initial Over-segmentation

Our input is a point cloud for which we create an initial over-segmentation $\mathcal{S}$ as a starting point for our object detection. Distinct objects are represented by oriented bounding boxes containing parts of the point cloud. We use features of the object points as well as features of the spatial relations between objects to characterize the object layout and to train our network such that it can detect objects.

There is much prior work that could be used to provide an initial over-segmentation of the point cloud. We use a representative unsupervised method based on a greedy graph-based approach [11] that was extended for point clouds by [22]. Our method follows [22] in using graph cuts for an over-segmentation of the point cloud based on point normal differences to create the initial set of segments.

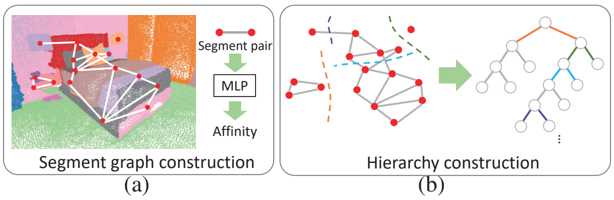

Figure 3: We train an MLP to predict segment pair affinities and create a segment affinity graph **(a)**. We then construct a hierarchy from the resulting segment affinity graph using normalized cuts **(b)**.

Each segment is extracted from the point cloud as an individual set of points, for which we compute oriented bounding boxes and point features as described in the following sections.

3.2. Hierarchy Initialization

During hierarchy construction we address the following problem. The input is the initial over-segmentation $\mathcal{S}$ and the output is a binary tree h representing a hierarchical grouping of the objects. Each object is represented as a 3D point cloud with an oriented bounding box (OBB), and a category label. The 3D point cloud is a set of points $\{p_i\} = \{x_i, y_i, z_i, r_i, g_i, b_i\}$ with their 3D (x, y, z) position and color (r, g, b). The leaves of this initial hierarchy h are the segments and the internal nodes represent groupings of the segments into objects and groups of objects. The root of the tree represents the entire room.

To construct the initial hierarchy from the input segments we first train a multi-layer perceptron (MLP) to predict segment pair affinities which indicate whether the two segments belong to the same object instance or not. The input to the MLP is a set of features capturing segment-segment geometric and color relationships, proposed by prior work [57]. We also tried using learned features obtained from a network trained on object-level label classification, but empirically found the features in [57] to work better in our experiments. The MLP is trained to predict binary pair affinity from these features under a squared hinge loss. Once we computed the segment pair affinities, the segments are then grouped into a hierarchy by using normalized cuts [47]. Starting from the root node, we split the segments into two groups recursively. The splitting stops when all groups have only one segment (leaf node). The cut cost $E(u, v) = e_c e_a$ between two segments u and v in the normalized cut is initially equal to the affinity e_a between the segments, but is then adjusted by the factor e_c during layout optimization (as described in the next section). Figure 3 shows the process of our hierarchy construction.

3.3. Object Detection and Layout Refinement

We describe our iterative optimization for predicting the object layout $\{\mathcal{B}, h\}$. We begin with the basic recursive autoencoder (RAE) for context aggregation and propagation. We then discuss a denoising version of the RAE (DRAE) designed for adjusting the object layout to better match a observed layout in the training set. Based on that, we introduce a Variational DRAE (VDRAE) which is a generative model for object layout improvement. It maps a layout onto a learned manifold of plausible layouts, and then generates an improved layout to better explain the input point cloud.

Recursive autoencoder for context propagation. Given the segments and the hierarchy, the recursive autoencoder (RAE) performs a bottom-up RvNN encoding for context aggregation, followed by a top-down RvNN decoding for context propagation. The encoder network takes as input the features (codes) of any two nodes to be merged (according to the hierarchy) and outputs a merged code for their parent node: $x_p^{enc} = f_{\text{enc}}(x_l^{enc}, x_r^{enc})$, where x_l^{enc}, x_r^{enc} and x_p^{enc} denote the codes of two sibling nodes and their parent node, respectively. f_{enc} is a MLP with two hidden layers for node grouping. The decoder takes the code of an internal node of the hierarchy as input and generates the codes of its two child nodes: $[x_l^{dec}, x_r^{dec}] = f_{\text{dec}}(x_p^{dec}, x_p^{enc})$, where f_{dec} is a two-layer MLP decoder for node ungrouping (Figure 4).

An additional *box encoder* generates the initial codes from the 3D point cloud within an OBB before the bottom-up pass, and a *box decoder* generates the final adjusted OBBs from the codes of the leaf nodes after the top-down pass: $x_n^{enc} = f_{\text{pnt}}(P_n), \quad t_n = f_{\text{box}}(x_n^{dec})$, where x_n^{enc} and x_n^{dec} denote the code for an node n in encoding and decoding. P_n is the set of 3D points of node n. t_n is the parameter vector of an output OBB, encoding the offsets of its position, dimension and orientation. f_{pnt} is a PointCNN [28] module for box encoding, and f_{box} a two-layer MLP for box decoding. The PointCNN[2] module is pretrained on a classification task for predicting object category labels from the point clouds of objects in the training set.

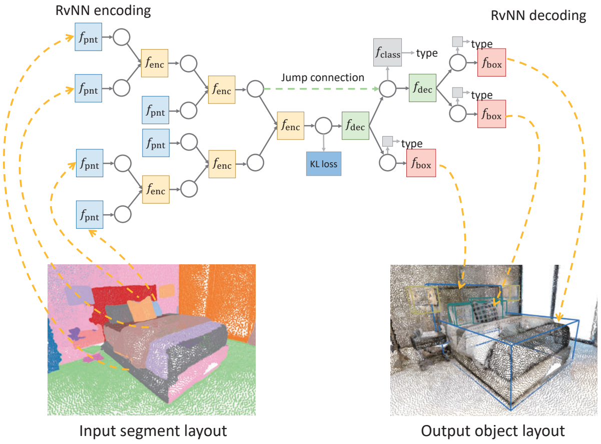

Figure 4: Our denoising recursive autoencoder (RAE) takes an input segment layout from an over-segmentation and performs bottom-up encoding for context aggregation (left side), followed by top-down decoding for context propagation (right side). The encoding-decoding process generates a refined hierarchy with 3D object detections as leaf nodes.

Denoising RAE for object detection and layout refinement. To endow the RAE with the ability to improve the predicted layout with respect to a target layout (e.g. a observed layout in the training set), a natural choice is to train a denoising RAE. Given a noisy input segment layout, we learn a Denoising RAE (DRAE) which generates a denoised layout. By noise, we mean perturbations over categorical labels, positions, dimensions and orientation of the bounding boxes. In DRAE, denoising is accomplished by the decoding phase which generates a new hierarchy of OBBs that refines, adds or removes individual object OBBs. The key for this generation lies in the node type classifier trained at each node (Figure 4) which determines whether a node is a leaf 'object' node at which decoding terminates, or an internal node at which the decoding continues: $o_n = f_{\text{cls}}^{\text{node}}(x_n^{dec}, x_n^{enc})$, with $o_n = 0$ indicating a leaf 'object' node and $o_n = 1$ an internal 'non-object' node For 'object' nodes, another object classifier is applied to determine the semantic categories: $c_n = f_{\text{cls}}^{\text{obj}}(x_n^{dec} x_n^{enc})$, where c_n is the categorical label for node n. For training, we compute the IoU of all nodes in the encoding hierarchy against ground-truth object bounding boxes and mark all nodes with IoU ≤ 0.5 as 'object'.

[2] Alternative encoding modules such as PointNet++ can be used instead.

Algorithm 1: VDRAE 3D Scene Layout Prediction.

```
Input  : Point cloud of indoor scene: $P$; Trained VDRAE.
Output: 3D object layout $\{\mathcal{B}, h\}$.
$\mathcal{S} \leftarrow$ Over-segmentation($P$);
$h \leftarrow$ HierarchyConstruction($\mathcal{S}$, $P$);
repeat
    $\mathcal{B} \leftarrow$ VDRAE($\mathcal{S}$, $h$, $P$);
    $h \leftarrow$ HierarchyConstruction($\mathcal{B}$, $\mathcal{S}$, $P$);
until Termination condition met;
return $\{\mathcal{B}, h\}$;
```

Variational DRAE for generative layout optimization. We train a Variational DRAE (VDRAE) to capture a manifold of valid hierarchies of OBBs from our training data. The training loss is:

$$\mathcal{L} = \sum_{n}^{\mathcal{N}} (\mathcal{L}_{\text{cls}}^{\text{node}}(o_n, o_n^*) + \mathcal{L}_{\text{cls}}^{\text{obj}}(c_n, c_n^*) + \mathcal{L}_{\text{obb}}^{\text{obj}}(t_n, t_n^*)) + \mathcal{L}_{\text{KL}}$$

where $\mathcal{N}$ are all decoding nodes, $\mathcal{L}_{\text{cls}}^{\text{node}}$ is a binary cross-entropy loss over two categories (‘object’ vs ‘non-object’), o_n^* is the ground-truth label, $\mathcal{L}_{\text{cls}}^{\text{obj}}$ is a multi-class cross-entropy loss over semantic categories, o_n^* is the ground-truth categorical label, $\mathcal{L}_{\text{obb}}^{\text{obj}}$ is an L_1 regression loss on the OBB parameters of the node, t_n^* is the ground-truth OBB parameters and $\mathcal{L}_{\text{KL}}$ is the KL divergence loss at root node. Note that the $\mathcal{L}_{\text{cls}}^{\text{obj}}$ and $\mathcal{L}_{\text{obb}}^{\text{obj}}$ terms exist only for ‘object’ nodes. The last term serves as a regularizer which measures the KL divergence between the posterior and a normal distribution $p(\mathbf{z})$ at the root node. This enables our VDRAE learning to map to the true posterior distribution of observed layouts.

Layout refinement using the VDRAE. Once trained, the VDRAE can be used to improve an object layout. Due to the coupling between object detection and hierarchy construction, we adopt an iterative optimization algorithm which alternates between the two stages (see Algorithm 1). Given an initial segment layout extracted from the input point cloud, our method first performs a VDRAE inference step (test-time step) to generate a hierarchy of object bounding boxes explaining the input point cloud. It then uses the decoding feature to infer a new hierarchy, which will be used for the VDRAE test in the next iteration. In the next iteration, the binary classification ‘object’ vs ‘non-object’ confidence is used to scale the normalized cut affinity e_a for two nodes u and v using the following factor e_c:

$$e_c(u, v) = \begin{cases} -\log(1 - c_s), & u \text{ and } v \text{ in same leaf node } s \\ 0.1, & \text{otherwise} \end{cases}$$

where c_s is the classification confidence of node s to be labeled as ‘object’. The scaled affinity $E(u, v) = e_c e_a$ is then used to refine the hierarchy construction. This process repeats until the structure of the hierarchy between iterations remains unchanged. Figure 5 shows an example of the iterative refinement. The optimization converges with at most 5 iterations for all the scenes we have tested. This iterative optimization gradually “pushes” the object layout into the layout manifold learned by VDRAE. Please refer to the supplemental material for a discussion of convergence.

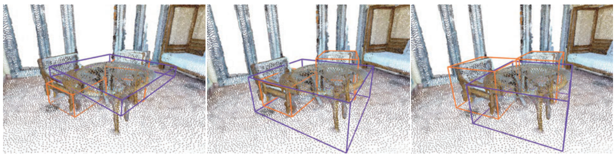

Figure 5: Example iterative refinement of initial object layout (leftmost column) under successive refinement iterations of our VDRAE network (columns to the right).

4. Implementation Details

In this section, we describe the implementation details of our network architectures, the relevant parameters, and the training and testing procedures.

Initial over-segmentation and feature extraction. For the initial over-segmentation we use threshold values $k = 0.01$ which we empirically found to perform well on training scenes (Section 3.1). For the PointCNN [28] features, we train the PointCNN to predict object class labels using the training set data. We train the network to minimize the cross-entropy loss over 41 object classes taking 2048 points per input and outputting to a 256-d vector for classification. Note that PointCNN is a pre-trained feature extractor and we didn’t fine-tune it during the training of VDRAE.

Hierarchy construction. The MLP for segment pair affinity prediction consists of 4 FC layers (with sigmoid layers). The input is a 25-d feature, and the output is a single affinity value. We use the detault parameter setting for the solver used in normalized cuts. It takes about $0.1s$ to build a hierarchy from a segment graph.

Variational denoising recursive autoencoder. f_{enc} has two 1000-d inputs and one 1000-d output. f_{dec} has one 1000-d input and two 1000-d outputs. $f_{\text{cls}}^{\text{node}}$ takes a 1000-d vector as input, and outputs a binary label. $f_{\text{cls}}^{\text{obj}}$ takes a 1000-d vector as input, and outputs a categorical label and OBB parameter offsets. This is achieved by using a softmax layer and a fully-connected layer. To deal with the large imbalance between positive (‘object’) and negative (‘non-object’) classes during training, we use a focal loss [31] with

$\gamma = 0$ for positives and $\gamma = 5$ for negative. This makes the training focus on all positive samples and *hard negative* samples. All the items in $\mathcal{L}$ can be trained jointly. However, to make the training easier, we first train by $\mathcal{L}_{\text{cls}}^{\text{node}}$ and $\mathcal{L}_{\text{KL}}$ to make the network have the ability to distinguish whether a node is a single object, and then fine-tune by $\mathcal{L}_{\text{cls}}^{\text{obj}}$ and $\mathcal{L}_{\text{obb}}^{\text{obj}}$.

Training and testing details. We implement the segment pair affinity network and the VDRAE using PyTorch [37]. For VDRAE, We use the Adam optimizer with a base learning rate of 0.001. We use the default hyper-parameters of $\beta_1 = 0.9, \beta_2 = 0.999$ and no weight decay. The batch size is 8. The VDRAE can be trained in 15 hours on a Nvidia Tesla K40 GPU. At testing time, a forward pass of the VDRAE takes about 1s. An Non-Maximum Suppression with IOU 0.5 is performed on the detected boxes after the inference of VDRAE.

5. Results

We evaluate our proposed VDRAE on 3D object detections in 3D point cloud scenes (see supplemental for semantic segmentation evaluation).

5.1. Experimental Datasets

We use two RGB-D datasets that provide 3D point clouds of interior scenes: S3DIS [1] and Matterport3D [3]. *S3DIS* consists of six large-scale indoor areas reconstructed with the Matterport Pro Camera from three different university buildings. These areas were annotated into 270 disjoint spaces (rooms or distinct regions). We use the k-fold cross validation strategy in [1] for train and test. *Matterport3D* consists of semantically annotated 3D reconstructions based on RGB-D images captured from 90 properties with a Matterport Pro Camera. The properties are divided into room-like regions. We follow the train/test split established by the original dataset, with $1,561$ rooms in the training set and 408 rooms in the testing set.

5.2. Evaluation

Our main evaluation metric is the average precision of the detected object bounding boxes against the ground truth bounding boxes at a threshold IoU of 0.5 (i.e. any detected bounding box that has more than 0.5 intersection-over-union overlap with its ground truth bounding box is considered a match). We compare our method against baselines from prior work on object detection in 3D point clouds. We then present ablated versions of our method to demonstrate the impact of different components on detection performance, as well as experiments to analyze the impact of the over-segmentation coarseness and the impact of successive refinement iterations.

	Chair	Table	Sofa	Board	mAP
Seg-Cluster [55]	0.23	0.33	0.05	0.13	0.19
Sliding PointCNN [28]	0.36	0.39	0.23	0.07	0.26
PointNet [39]	0.34	0.47	0.05	0.12	0.25
SGPN [55]	0.41	0.50	0.07	0.13	0.28
Ours (flat context)	0.35	0.47	0.32	0.10	0.31
Ours	**0.45**	**0.53**	**0.43**	**0.14**	**0.39**

Table 1: Comparison of our approach against prior work on object detection in 3D point cloud data. Values report average precision at IOU of 0.5 on the S3DIS dataset. Our hierarchy-refining VDRAE outperforms all prior methods.

Qualitative examples. Figure 6 shows detection results on the Matterport3D test set (see supplement for more examples). Our VDRAE leverages hierarchical context to detect and refine 3D bounding boxes for challenging cases such as pillows on beds, and lamps on nightstand cabinets.

Comparison to baseline methods. We evaluate our approach against several baselines from prior work that produce object detections for indoor 3D scene point clouds:

- **Seg-Cluster**: Approach proposed by [55] applies semantic segmentation (SegCloud [53]) followed by Euclidean clustering [44].
- **PointNet**: Predicts the category of points [39] and uses breadth-first search to group nearby points with the same category, inducing object instances. We use PointNet instead of other point-based neural networks as PointNet proposed this object detection pipeline.
- **Sliding PointCNN**: A baseline using a 3D sliding window approach with PointCNN [28] features.
- **SGPN**: A state-of-the-art semantic instance segmentation approach for point clouds [55].
- **Ours (flat context)**: A baseline using a flat context representation instead of leveraging the hierarchy structure, in which x_n^{dec} is the concatenation of encoded features x_n^{enc} and the average encoded features of all nodes $(\sum_n^{\mathcal{N}} x_n^{enc})/n$.

Tables 1 and 2 report average precision on the S3DIS and Matterport3D datasets, showing that our approach outperforms all baselines. The flat context baseline performs worse than our hierarchy-aware VDRAE but better than the baselines that do not explicitly represent context. Figure 7 qualitatively shows results from the Matterport3D test set, comparing our approach with the highest performing prior work baseline using SGPN.

Ablation of method components. We evaluate the impact of each components using the following variants:

- **No hierarchy**: We use PointCNN [28] features for each node to predict the object category and regress

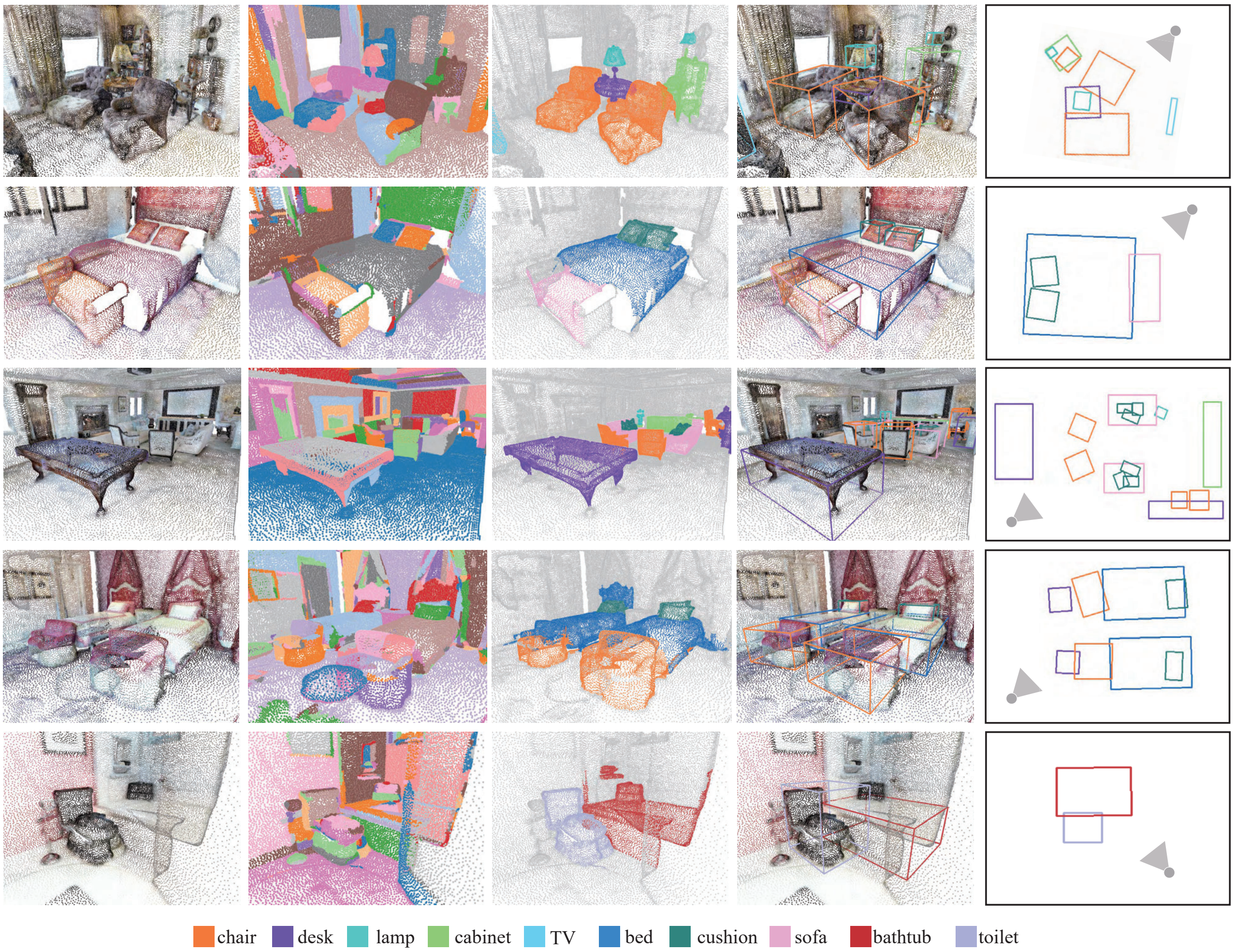

Figure 6: 3D scene layout predictions using our VDRAE on the Matterport3D test set. The first column shows the input point cloud. The second column is the over-segmentation from which we construct an initial segment hierarchy. The third column shows the 3D object detections with colors by category. The final two columns show bounding boxes for the detections. Our approach predicts hierarchically consistent 3D layouts where objects such as lamps, pillows and cabinets are detected in plausible positions and orientations relative to other objects and the global structure of the scene.

	Chair	Table	Cabinet	Cushion	Sofa	Bed	Sink	Toilet	TV	Bathtub	Lighting	mAP
Sliding PointCNN [28]	0.22	0.21	0.03	0.19	0.20	0.36	0.07	0.16	0.05	0.15	0.10	0.16
PointNet [39]	0.28	**0.32**	0.06	0.21	0.28	0.25	0.17	0.08	0.10	0.11	0.06	0.18
SGPN [55]	0.29	0.24	0.07	0.18	**0.30**	0.33	0.15	0.17	0.09	0.16	0.11	0.19
Ours (flat context)	0.24	0.18	0.08	0.21	0.18	0.27	0.22	0.25	0.07	0.21	0.07	0.18
Ours	**0.37**	0.27	**0.11**	**0.24**	0.28	**0.43**	**0.23**	**0.35**	**0.19**	**0.27**	**0.19**	**0.27**

Table 2: Average precision of object detection at IoU 0.5 on the Matterport3D dataset. We compare our full method ('ours') against several baselines. Refer to text for the details of the baselines.

an OBB without using a hierarchy. We add 4 FC layers after the PointCNN layers to increase the number of network parameter and make the comparison fair.

- **No OBB regression**: We turn off the OBB regression module for leaf nodes and train from scratch.
- **No iteration (bvh)**: No iteration for testing. The hierarchy is constructed through recursive binary splits considering only geometric separation between seg-

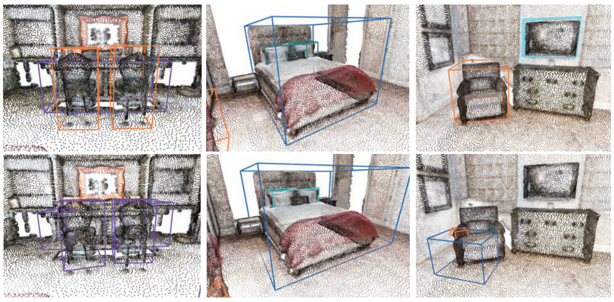

Figure 7: Qualitative 3D object detection results on Matterport3D test set using our VDRAE (top row) and the best performing baseline from prior work (SGPN[55], bottom row). Our approach produces more accurate bounding box detections and fewer category errors. For example, chairs are correctly categorized and have tight bounding boxes at the top left and top right.

	Chair	Table	Sofa	Board	mAP
no hierarchy	0.34	0.41	0.35	0.08	0.30
no OBB regression	0.41	0.47	0.40	0.11	0.35
no iteration (bvh)	0.37	0.47	0.38	0.10	0.33
no iteration (our hier)	0.39	0.51	0.39	0.12	0.35
Ours	**0.45**	**0.53**	**0.43**	**0.14**	**0.39**

Table 3: Ablation of the components of our approach. Values report average precision at IoU of 0.5 on the S3DIS dataset. Our full VDRAE outperforms all ablations.

ments, i.e. bounding volume hierarchy (bvh).

- **No iteration (our hier)**: No iteration for testing. The hierarchy is built by our hierarchy initialization approach.

Table 3 shows the results. The full method performs the best. Not using a hierarchy degrades performance the most. Removing OBB regression, and not performing iterative refinement are also detrimental but to a lesser extent.

Sensitivity to over-segmentation coarseness. We quantify the impact of the over-segmentation coarseness threshold parameter k of the method in [22] on S3DIS. We use five threshold values $k = 1.0, 0.1, 0.01, 0.001, 0.0001$ to generate segments with different size and re-train the affinity network and VDRAE respectively. Larger k produce bigger segments. Figure 8 (a) shows that the best performance is achieved when average segment size is 1.45m ($k = 0.01$).

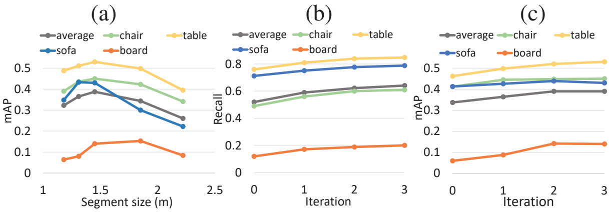

Figure 8: (a) mAP plotted against over-segmentation coarseness (average segment size in meters). (b) recall against VDRAE iteration count. (c) mAP against VDRAE iteration count.

Effect of iteration. We evaluate the effect of VDRAE refinement iteration by analyzing the hierarchy and 3D object detections at each step. Figure 8 (b) shows recall against ground-truth objects plotted against iteration number. Recall is computed by calculating the IoU of the OBB of each ground-truth object with *all node OBBs in encoding hierarchy*. If one of the IoU values is larger than 0.5, we consider that a match against the ground-truth. Figure 8 (c) shows the object detection mAP plotted against iteration number. The benefit of iteration is apparent.

6. Conclusion

We presented an approach for predicting 3D scene layout in fused point clouds by leveraging a hierarchical encoding of the context. We train a network to predict segment-to-segment affinities and use it to propose an initial segment hierarchy. We then use a variational denoising recursive autoencoder to iteratively refine the hierarchy and produce 3D object detections. We show significant improvements in 3D object detection relative to baselines taken from prior work.

Limitations. Our current method has several limitations. First, the hierarchy proposal and VDRAE are trained separately. Incorporating these two stages will leverage the synergy between parsing hierarchies and refining the 3D scene layouts. Second, the segment point features we use in our VDRAE is trained independently on a classification task. These features can also be learned end-to-end, resulting in further task-specific improvements in performance.

Future work. We have only taken a small step towards leveraging hierarchical representations of 3D scenes. There are many avenues to pursue for future research. Reasoning about the hierarchical composition of scenes into objects, object groups, functional regions, rooms, and entire residences can benefit many tasks beyond 3D object detection. We hope that our work will act as a catalyst in this promising research direction.

Acknowledgements

We are grateful to Thomas Funkhouser and Shuran Song for the valuable discussion. This work was supported in part by NSFC (61572507, 61532003, 61622212) and Natural Science Foundation of Hunan Province for Distinguished Young Scientists (2017JJ1002).

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *Proc. CVPR*, 2016.

[2] Jorge Beltran, Carlos Guindel, Francisco Miguel Moreno, Daniel Cruzado, Fernando Garcia, and Arturo de la Escalera. Birdnet: a 3d object detection framework from lidar information. *arXiv preprint arXiv:1805.01195*, 2018.

[3] Angel Chang, Angela Dai, Tom Funkhouser, , Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2017.

[4] Kang Chen, Yukun Lai, Yu-Xin Wu, Ralph Robert Martin, and Shi-Min Hu. Automatic semantic modeling of indoor scenes from low-quality RGB-D data using contextual information. *ACM Transactions on Graphics*, 33(6), 2014.

[5] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33(7):853–862, 2012.

[6] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. A tree-based context model for object recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(2): 240–252, 2012.

[7] Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. Understanding indoor scenes using 3D geometric phrases. In *CVPR*, pages 33–40. IEEE, 2013.

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. CVPR*, 2017.

[9] Zhuo Deng and Longin Jan Latecki. Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 2, 2017.

[10] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leib. Exploring spatial context for 3D semantic segmentation of point clouds. In *Proc. ICCV*, 2017.

[11] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[12] Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *Computer vision and image understanding*, 114(6):712–722, 2010.

[13] Ross Girshick. Fast R-CNN. *Proc. ICCV*, 2015.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014.

[15] Saurabh Gupta, Ross Girshick, Pablo Arbelaez, and Jitendra Malik. Learning rich features from RGB-D images for object detection and segmentation. In *Proc. ECCV*, 2014.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[17] Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. Cascaded classification models: Combining models for holistic scene understanding. In *Advances in Neural Information Processing Systems*, pages 641–648, 2009.

[18] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proc. CVPR*, 2018.

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

[20] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3D object, layout, and camera pose estimation. In *Proc. NIPS*, 2018.

[21] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3D scene parsing and reconstruction from a single RGB image. In *European Conference on Computer Vision*, pages 194–211. Springer, 2018.

[22] Andrej Karpathy, Stephen Miller, and Li Fei-Fei. Object discovery in 3D scenes via shape analysis. In *Proc. ICRA*, pages 2088–2095. IEEE, 2013.

[23] Jean Lahoud and Bernard Ghanem. 2D-driven 3D object detection in RGB-D images. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 4632–4640. IEEE, 2017.

[24] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proc. CVPR*, 2018.

[25] Karel Lenc and Andrea Vedaldi. R-CNN minus R. *Proc. BMVC*, 2015.

[26] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Trans. on Graphics. (SIGGRAPH)*, 2017.

[27] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3D reconstruction. *Computer Graphics Forum (Eurographics)*, 2015.

[28] Yangyan Li, Rui Bu, Mingchao Sun, and Baoquan Chen. PointCNN. *arXiv preprint arXiv:1801.07791*, 2018.

[29] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3D object detection with RGBD cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1417–1424, 2013.

[30] Di Lin, Yuanfeng Ji, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Multi-scale context intertwining for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 603–619, 2018.

[31] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[32] Tianqiang Liu, Siddhartha Chaudhuri, Vladimir G. Kim, Qixing Huang, Niloy J. Mitra, and Thomas Funkhouser. Creating consistent scene graphs using a probabilistic grammar. *ACM Trans. on Graphics. (SIGGRAPH Asia)*, 2014.

[33] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[34] Xiaobai Liu, Yibiao Zhao, and Song-Chun Zhu. Single-view 3D scene reconstruction and parsing by attribute grammar. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):710–725, 2018.

[35] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proc. CVPR*, pages 891–898, 2014.

[36] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Trans. on Graphics. (SIGGRAPH Asia)*, 2012.

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[38] Jake Porway, Kristy Wang, Benjamin Yao, and Song Chun Zhu. A hierarchical and contextual model for aerial image understanding. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

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

[40] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Proc. NIPS*, 2015.

[43] Zhile Ren and Erik B. Suddert. Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *Proc. CVPR*, 2016.

[44] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *Proc. ICRA*, Shanghai, China, May 9-13 2011.

[45] Tianjia Shao, Aron Monszpart, Youyi Zheng, Bongjin Koo, Weiwei Xu, Kun Zhou, and Niloy J Mitra. Imagining the unseen: Stability-based cuboid arrangements for scene understanding. *ACM Transactions on Graphics*, 33(6), 2014.

[46] Abhishek Sharma, Oncel Tuzel, and Ming-Yu Liu. Recursive context propagation network for semantic scene labeling. In *Advances in Neural Information Processing Systems*, pages 2447–2455, 2014.

[47] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE T-PAMI*, 22(8):888–905, 2000.

[48] Yifei Shi, Pinxin Long, Kai Xu, Hui Huang, and Yueshan Xiong. Data-driven contextual modeling for 3D scene understanding. *Computers & Graphics*, 2016.

[49] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. Complex-yolo: Real-time 3d object detection on point clouds. *arXiv preprint arXiv:1803.06199*, 2018.

[50] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proc. ICML*, pages 129–136, 2011.

[51] Shuran Song and Jianxiong Xiao. Sliding Shapes for 3D object detection in depth images. In *Proc. ECCV*, 2014.

[52] Shuran Song and Jianxiong Xiao. Deep Sliding Shapes for amodal 3D object detection in RGB-D images. In *Proc. CVPR*, 2016.

[53] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. SEGCloud: Semantic segmentation of 3D point clouds. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2017.

[54] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of computer vision*, 63(2):113–140, 2005.

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

[56] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. PointFusion: Deep sensor fusion for 3D bounding box estimation. In *Proc. CVPR*, 2018.

[57] Kai Xu, Hui Huang, Yifei Shi, Hao Li, Pinxin Long, Jianong Caichen, Wei Sun, and Baoquan Chen. Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Transactions on Graphics (TOG)*, 34(6):177, 2015.

[58] Kai Xu, Vladimir G Kim, Qixing Huang, Niloy Mitra, and Evangelos Kalogerakis. Data-driven shape analysis and processing. In *SIGGRAPH ASIA 2016 Courses*, page 4. ACM, 2016.

[59] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[60] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *European Conference on Computer Vision*, pages 415–430. Springer, 2018.

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

[62] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proc. CVPR*, 2018.

[63] Yinda Zhang, Mingru Bai, Pushmeet Kohli, Shahram Izadi, and Jianxiong Xiao. DeepContext: Context-encoding neural pathways for 3D holistic scene understanding. In *Proc. ICCV*, 2017.

[64] Yibiao Zhao and Song-Chun Zhu. Image parsing with stochastic scene grammar. In *Advances in Neural Information Processing Systems*, pages 73–81, 2011.

[65] Yibiao Zhao and Song-Chun Zhu. Scene parsing by integrating function, geometry and appearance models. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3119–3126, June 2013. doi: 10.1109/CVPR.2013.401.