Supplemental Materials
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

In this supplemental material, we provide more details on the network architecture (Section 1), present additional results of our method on 3D point cloud instance segmentation (Section 2), additional qualitative comparisons of our results against baselines (Section 3), a visualization of the learned 3D scene layout embedding (Section 4), and we discuss the convergence of iterative application of the VDRAE for 3D scene layout refinement (Section 5).

1. Network Architecture Details

Figure 1 and Figure 2 show the architecture of our encoder and decoder networks in more detail. The encoder network takes the initial segment hierarchy and aggregates features in its nodes to capture the hierarchical context. The decoder network then takes the encoded features and predicts for each node whether the node is a leaf 3D object node, the semantic category of the object, and refined parameters of its bounding box. Note that the two networks are connected through jump (i.e. skip) connections between the node features. See the main paper architecture figure for a more compact visualization of the entire architecture.

2. 3D Instance Segmentation Evaluation

In Table 1 and Table 2, we report results on the 3D instance segementation task. Though the goal of our method was not to explicitly produce 3D instance segmentation, such segmentations are produced implicitly when segments from the point cloud are assigned to detected 3D object nodes in our layout hierarchy, each with a specific category label. We see that our approach outperforms the baseline from an existing 3D instance semantic segmentation approach [2].

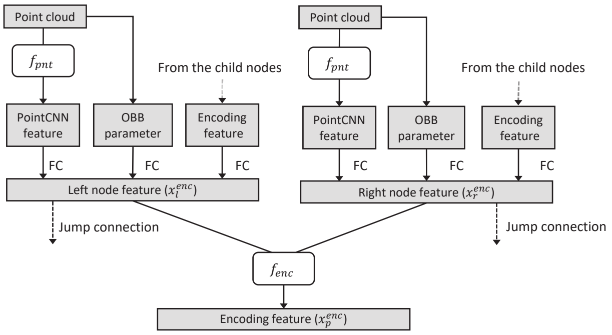

Figure 1: Our encoder architecture takes a point cloud segment as input for each node and then combines PointCNN features, oriented bounding box (OBB) parameters for the node, and a feature encoding that from node's child nodes. These features are passed through FC layers to produce the encoding x_i^{enc} of the specific node, and then individual node encoding pairs are combined through f_{enc} to produce the encoding feature x_p^{enc} for their parent node.

3. More Qualitative Results

Figure 3 shows additional qualitative comparisons of our VDRAE 3D layout prediction to ground truth and results using SGPN [2] on Matterport3D test scenes. In general, we see that our VDRAE approach matches the ground truth layouts more closely than the baseline approach. These results demonstrate that our hierarchical encoding of the 3D layout and iterative refinement through the trained VDRAE can lead to improvements in 3D object detection.

Figure 4 shows qualitative results of our VDRAE 3D layout prediction on S3DIS scenes.

	Column	Chair	Table	Bookcase	Sofa	Board	mAP
SGPN [2]	0.607	0.408	0.469	**0.476**	0.064	0.111	0.356
Ours	**0.623**	**0.462**	**0.537**	0.449	**0.451**	**0.415**	**0.445**

Table 1: Comparison of our approach against prior work on instance segmentation on the S3DIS dataset. Values report average precision with IoU threshold 0.5.

	Chair	Table	Cabinet	Cushion	Sofa	Bed	Sink	Toilet	TV	Bathtub	Lighting	mAP
SGPN [2]	0.241	0.176	0.036	0.159	**0.267**	0.304	0.110	0.149	0.042	0.063	0.057	0.146
Ours	**0.305**	**0.210**	**0.062**	**0.217**	0.248	**0.350**	**0.136**	**0.329**	**0.057**	**0.121**	**0.106**	**0.195**

Table 2: Comparison of our approach against prior work on instance segmentation on the Matterport3D dataset. Values report average precision with IoU threshold 0.5.

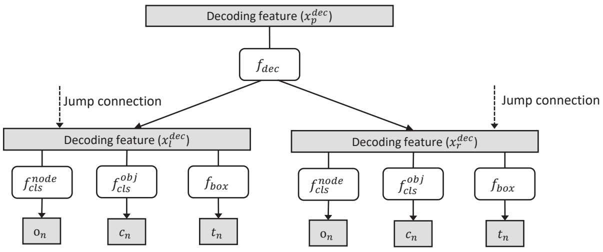

Figure 2: Our decoder architecture takes an aggregated feature and decodes it to pairs of node features (here shown as x_l^{dec} and x_r^{dec}. From these features, we predict several targets: whether the node corresponds to a 3D object detection or not (o_n), the category of the object detection (c_n) and the parameters of the bounding box for the detection (t_n).

4. 3D Scene Layout Embedding Manifold Visualization

To investigate the degree to which our method learns a meaningful embedding of 3D scene layouts, we visualize the the root node features of encoded scenes from the Matterport3D dataset. The root node feature encodes the entire observed 3D scene layout and the hierarchical structure of contained nodes. The visualization in Figure 5 shows that the embedding space of this feature does indeed capture the layout of different types of rooms, leading to clustering of semantically similar rooms.

5. Discussion of Convergence of 3D Scene Layout Refinement

Algorithm 1 of the main paper (reproduced here for convenience), presents an approach for iteratively applying our VDRAE to improve 3D scene layout prediction. Here we discuss the converge properties of our algorithm.

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 $\{\mathcal{B}, h\}$;
```

The algorithm iterates until the structure of the hierarchy between iterations remains unchanged. During each iteration, the resulting hierarchy is determined by the scaled affinity $E(u, v) = e_c e_a$ where

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

and c_s is the classification confidence of node s to be labeled as 'object'. Note that this changes only when c_s changes. For more confident groupings of u and v, c_s approaches 1, so e_c approaches $+\infty$, making it more expensive to separate u, v. In contrast, for less confident groupings, both c_s and e_c is close to zero, resulting in a weaker bond between the two nodes.

When we iterate, we will naturally keep the grouping of subtrees u, v whose parent node s has a high classification confidence c_s, and adjust the grouping of the nodes that has a less confident c_s. Thus, at each iteration, we only need to consider what happens to nodes with low c_s (we can consider nodes with high c_s to be fixed).

After each iteration, we will have alternate groupings that result in either 1) more nodes with a higher c_s or 2) no more nodes with high c_s. In case 2), because the nodes have low c_s, they are not object nodes and so do not affect the overall object detection result. Thus the structure of the hierarchy does not change between iterations and the

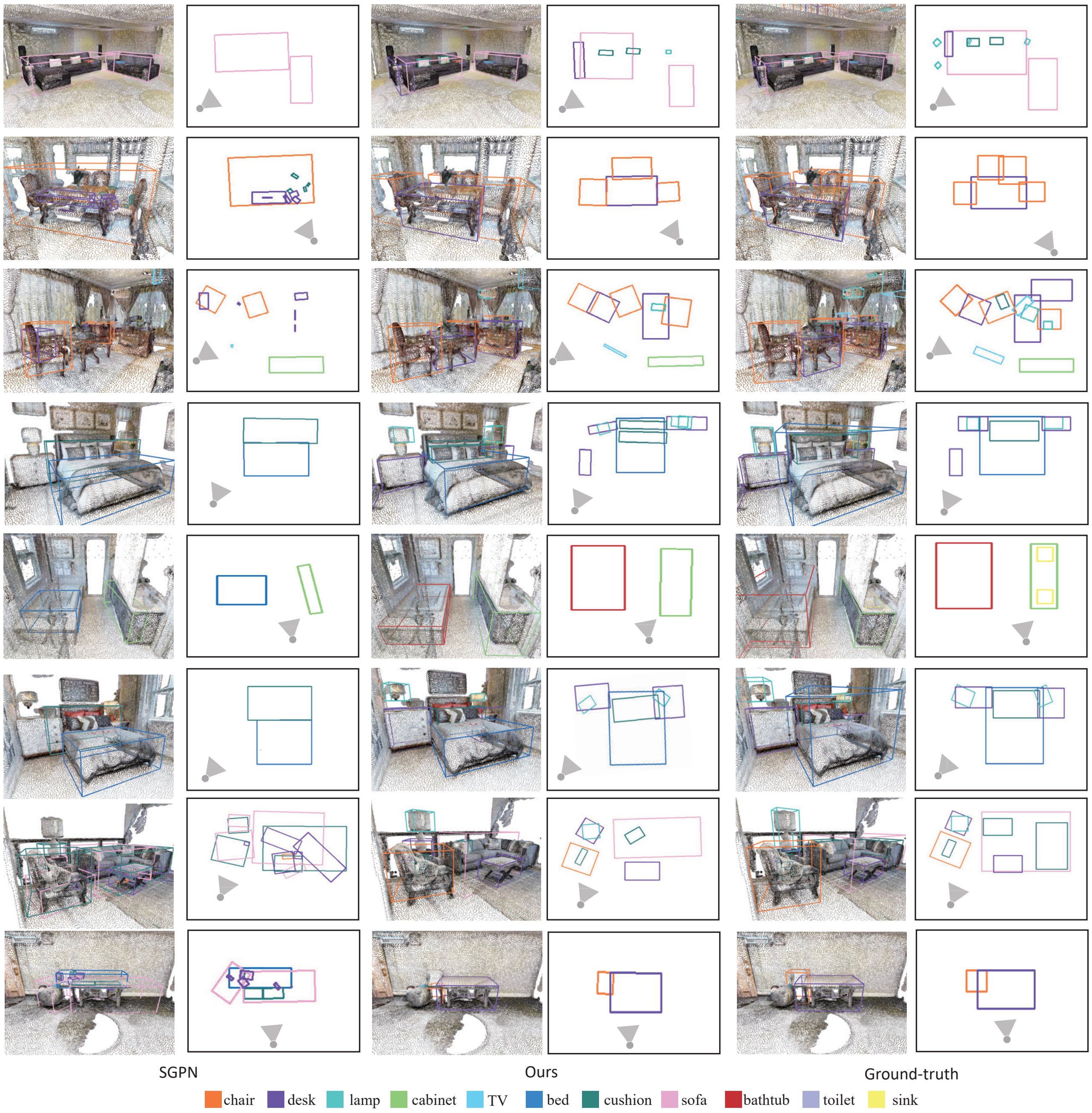

Figure 3: We visualize 3D scene layout results using SGPN [2] (left two columns), our VDRAE approach (middle two columns), and ground truth (right two columns). The pairs of columns show a 3D view of the detected object bounding boxes, and a top-down view of the same detections. We see that our approach matches the ground truth object bounding boxes and categories better than the baseline. In particular, our approach is significantly better at predicting smaller objects such as pillows on sofas and beds, indicating that an encoding of the surrounding hierarchical context is valuable.

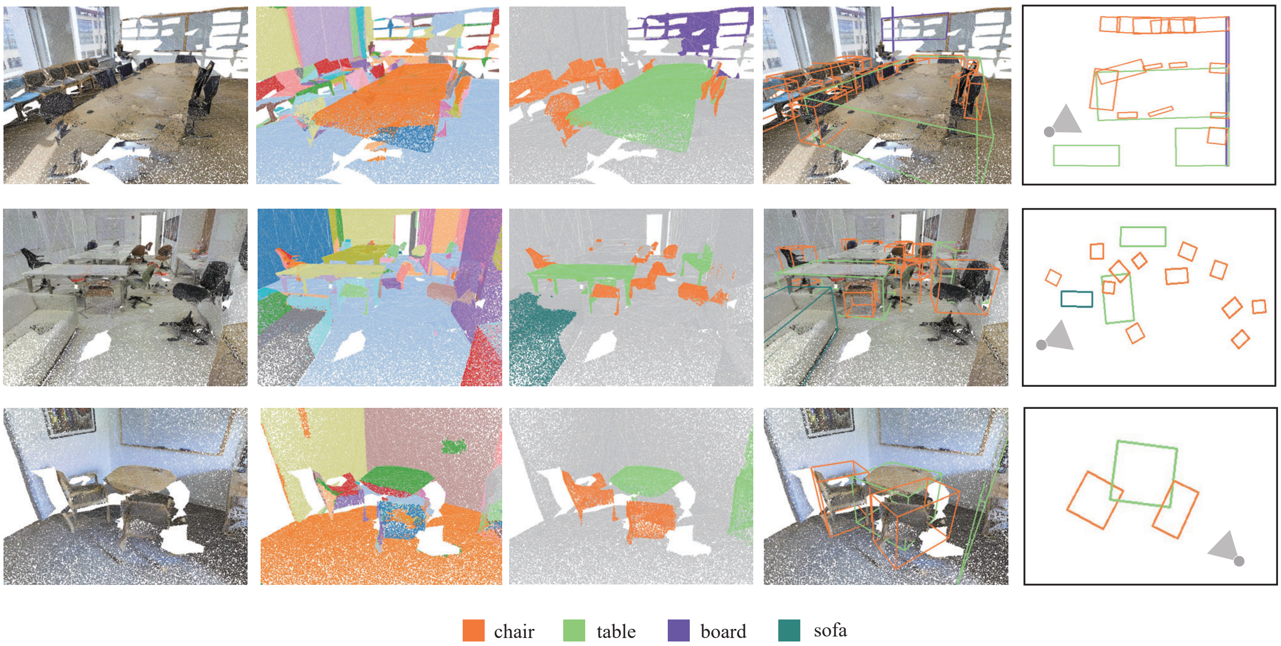

Figure 4: We visualize 3D scene layout results using our VDRAE approach on S3DIS scenes. The first column shows the input point cloud. The second column is the over-segmentation from which we construct an initial segment hierarchy. The third column shows the 3D object detections with colors by category. The final two columns show bounding boxes for the detections.

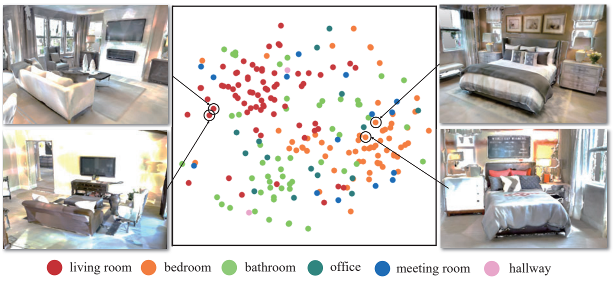

Figure 5: Visualization of the learned embedding for 3D scene layouts through our VDRAE approach. We extract the aggregated feature at the root node of each 3D scene and visualize the embedding using t-sne [1]. The visualization shows that different room categories are separated by the embedding, and that 3D scene layouts that are similar in nature (e.g. the two living rooms with TVs on the left and the two bedrooms on the right) are clustered together in the embedding.

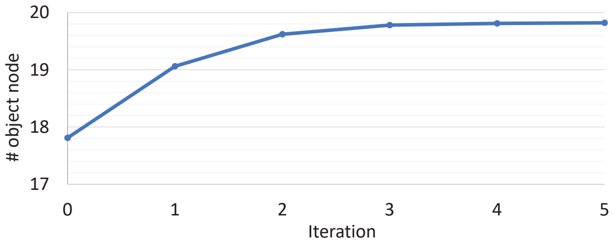

Figure 6: Average number of 3D object nodes in the predicted hierarchy plotted against iteration count during refinement of the 3D scene layout in Matterport3D test set scenes. We see that the total number of nodes converges, confirming our empirical observation that our hierarchy refinement algorithm converges within about 5 iterations for the datasets we tested.

algorithm is converged in this scenario. In case 1), there are increasing number of nodes with high c_s values corresponding to more object nodes and we will continue to iterate. However, every time, since we have a finite number of nodes, the number of nodes going from low c_s to high c_s will decrease until there are no low c_s nodes left. In this case, all nodes will belong to some single object node, and the hierarchy structure will have converged as well.

This is a high-level discussion of the convergence behavior of our layout refinement approach. In practice, we observe that 3D scene layouts converge to non-changing hierarchy structure within 5 iterations. This is confirmed also by the plot in Figure 6 which shows the average number of detected hierarchy object nodes against iteration count across test scenes from the Matterport3D test set. We leave a more thorough and formal discussion of the convergence properties of VDRAE-based 3D scene layout refinement to future work.

References

[1] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 4

[2] W. Wang, R. Yu, Q. Huang, and U. Neumann. SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In *Proc. CVPR*, 2018. 1, 2, 3