RfD-Net: Point Scene Understanding by Semantic Instance Reconstruction
Supplementary Material

Yinyu Nie[1,2] Ji Hou[3] Xiaoguang Han[1,*] Matthias Nießner[3]
[1]SRIBD, CUHKSZ [2]Bournemouth University [3]Technical University of Munich

A. Network and Layer Specifications

In this section, we provide all the parameters, layer specifications and weights used in loss functions. We uniformly denote the fully-connected layers by MLP $[l_1, ..., l_d]$, where l_i is the number of neurons in the i-th layer.

A.1. 3D Detector

In section 3.1, we predict object proposals from N input points with VoteNet [9] as the backbone. It produces N_p proposals with D_p-dim features (i.e. proposal features $\boldsymbol{F}_p \in \mathbb{R}^{N_p \times D_p}$ in our paper), from which we regress the D_b-dim box parameters with MLP $[128, 128, 69]$ (N=80K, N_p=256, D_p=128, D_b=69). As in [9], the 69-dim box parameters are encoded by center $\boldsymbol{c} \in \mathbb{R}^3$, scale $\boldsymbol{s}^3 \in \mathbb{R}^3$, heading angle $\theta \in \mathbb{R}$, semantic label l, and objectness score s_{obj}. s_{obj} is a probability value indicating whether the proposal is close to (<0.3 meter, positive) or far from (>0.6 meter, negative) any ground-truth object center.

A.2. Spatial Transformer

Objectness Dropout Layer. In section 3.2, we adopt an *Objectness Dropout* layer to reserve the Top-N_d proposals with higher objectness ($N_d = 10$) for shape learning. In test, we replace it with 3D Non Maximum Suppression (3D NMS) to produce the output boxes and corresponding shapes with the 3D Box IoU threshold of 0.25 in evaluation.

Group & Align. From the N_d box proposals, we group the neighboring M_p points that are located within a radius r to each box center using a group layer [11]. M_p=1024, r=1. It produces N_d point clusters $\{\mathbf{P}_i^c\}$ ($i = 1, 2, ..., N_d, \mathbf{P}_i^c \in \mathbb{R}^{M_p \times 3}$). After grouping, we align the 3D points in each cluster to a canonical system with the Equation 1 in our paper, where the rotation and translation adjustment $(\Delta\mathcal{R}, \Delta\boldsymbol{c})$ are predicted from $\mathbf{P}_i^c$ as in Figure 1. $(\Delta\mathcal{R}, \Delta\boldsymbol{c})$ are predicted without supervision, which asks for the network to search for the optimal adjustment in spatial alignment (see Equation 1 in our paper).

* Corresponding Email: hanxiaoguang@cuhk.edu.cn

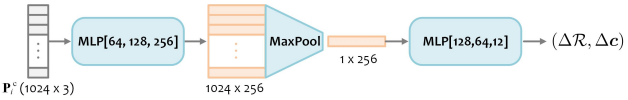


Figure 1: Rotation and translation adjustment.

A.3. Shape Generator

In section 3.3, we design the shape generator with two parts (see Figure 3 of the paper): a shape encoder extended with skip propagation; and a shape decoder based on conditional batch normalization.

Shape Encoder. The layer specification of the denoiser in skip propagation is illustrated in Figure 2. The PointNet encoder [10] designed with residual connection is shown in Figure 3. It takes the extended point clusters as input (see section 3.3 in the paper) and outputs the new proposal features $\{\boldsymbol{f}_p^* \in \mathbb{R}^{D_s}\}$, $D_s = 512$ to decode shapes.

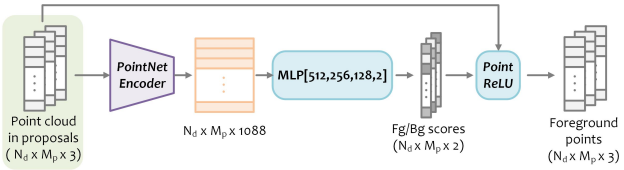


Figure 2: Denoiser in skip propagation.

Shape Decoder. We build the shape decoder as a probabilistic generative model. The latent encoder [8] is fed with the proposal feature $\{\boldsymbol{f}_p^*\}$, spatial points $\{\boldsymbol{p} \in \mathbb{R}^3\}$ with corresponding occupancy values $\{o\}$. It outputs the mean and standard deviation $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ to approximate the standard normal distribution. We illustrate the latent encoder in Figure 4, where the 2,048 spatial points are randomly sampled in the shape bounding cube. In our method, we separately sample 1,024 points inside and 1,024 points outside the shape mesh. From the predicted distribution $N(\boldsymbol{\mu}, \boldsymbol{\sigma})$, we sample a latent code $\boldsymbol{z} \in \mathbb{R}^L$ ($L = 32$) to predict the occupancy values $\{o\}$ from $\{\boldsymbol{p}\}$ conditioned on $\boldsymbol{f}_p^*$ for each object. The shape decoder is based on the Conditional Batch Normalization [3, 4] layers, which is illustrated in Figure 5.

During test, we directly initialize the latent codes with zeros. As discussed in section 3.3 of our paper, we uni-

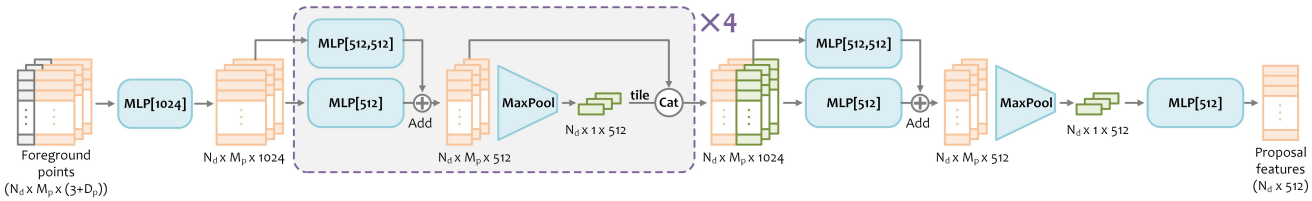


Figure 3: PointNet-based shape encoder with residual connection.

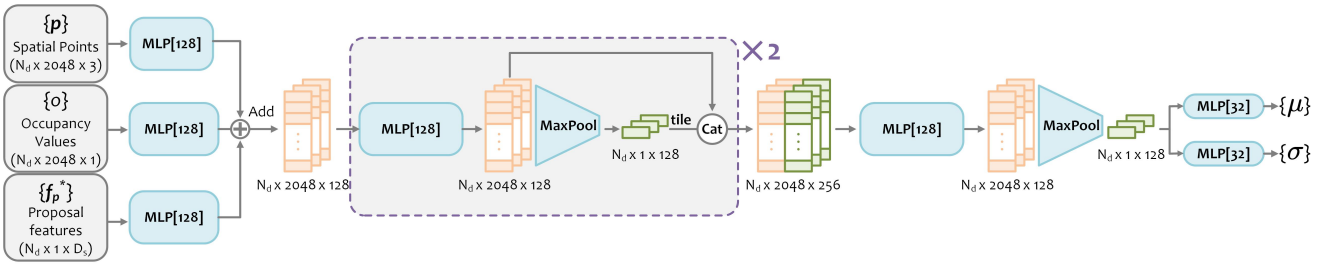


Figure 4: Latent encoder for probabilistic shape generation.

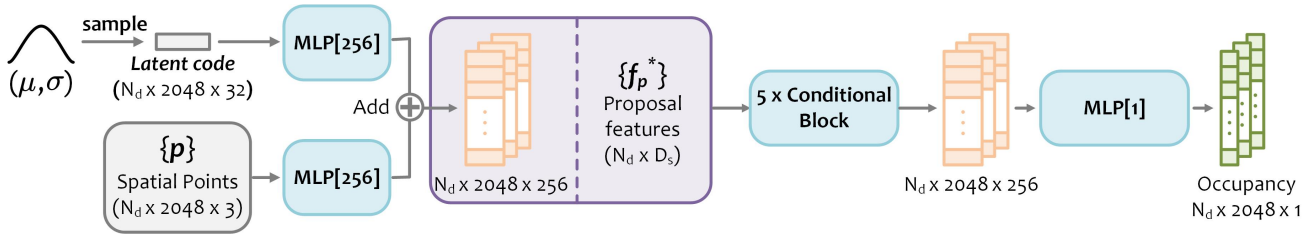


Figure 5: Shape decoder with Conditional Batch Normalization layers.

formly sample spatial points in the object bounding box, and use Marching Cubes [7] to extract the iso-surface as object meshes. Specifically, we adopt the efficient Multi-resolution Iso-Surface Extraction (MISE) algorithm [8] to improve the spatial sampling efficiency and extract meshes under 128-d occupancy grids.

A.4. Weights in Loss Functions

We list the weights to balance different loss functions as follows. We set $\lambda_{cls} = 0.1$ as the hybrid ratio of combing the classification and regression losses, i.e., $\lambda_{cls}\mathcal{L}_{cls} + \mathcal{L}_{reg}$, in defining the scale loss $\mathcal{L}_s$ and heading angle loss $\mathcal{L}_\theta$. Then the box loss $\mathcal{L}_{box}$ in our paper can be denoted as:

$$\mathcal{L}_{box} = \mathcal{L}_v + \mathcal{L}_c + \mathcal{L}_s + \mathcal{L}_\theta + \lambda_l\mathcal{L}_l + \lambda_{obj}\mathcal{L}_{obj}, \quad (1)$$

where $\lambda_{obj} = 0.5$, $\lambda_l = 0.1$. For shape loss in Equation 3 of our paper, we set $\lambda_{seg} = 1e2$. Then the total loss for end-to-end training can be summarized as:

$$\mathcal{L} = \mathcal{L}_{box} + \lambda\mathcal{L}_{shape}, \quad (2)$$

where $\lambda = 5e$-3.

B. Efficiency and Memory in Inference

We train our network with two NVIDIA TITAN-Xp GPUs and test it on a single GPU. Our method takes around 25 hours for training compared to 60 hours of [6]. We also compare the inference timing in single forward pass and GPU memory usage with [6] (see Table 3). It shows that our method have comparable efficiency with the state-of-the-art but requires much fewer memory (≈1/3), which indicates that learning shapes directly from the raw point cloud consumes fewer computation resource and hardware requirement than processing 3D scenes with TSDF grids.

C. Detailed Quantitative Comparisons

In this section, we list the per-category scores of 3D detection and object reconstruction in Table 1 and Table 2 (refer to section 5.2 in the paper).

D. More Qualitative Comparisons

We provide more qualitative comparisons of semantic instance reconstruction with [6] in Figure 6. As in RevealNet [6], we use the groundtruth objects in Scan2CAD [1] for supervision. Note that the groundtruth CAD models in

	Input	table	bathtub	trashbin	sofa	chair	cabinet	bookshelf	display	mAP
3D-SIS [5]	Geo+RGB	42.76	3.03	20.04	34.39	63.26	21.54	16.92	3.69	25.70
MLCVNet [12]	Geo Only	44.51	22.39	10.10	**53.13**	78.74	26.34	22.93	8.90	33.40
RevealNet [6]	Geo Only	35.64	14.94	26.77	29.96	53.18	26.63	15.89	31.30	29.29
Ours (w/o joint)	Geo Only	46.67	19.09	15.30	51.71	77.22	24.62	18.58	7.03	32.63
Ours (w/ joint)	Geo Only	**49.90**	**23.63**	15.69	52.34	**79.88**	26.72	**23.20**	9.23	**35.10**

Table 1: 3D object detection on ScanNet v2. 3D-SIS [5] and RevealNet [6] results are provided by the authors. MLCVNet results are retrained with the original network [12]. Scores above are evaluated with mAP@0.5.

	resolution	table	bathtub	trashbin	sofa	chair	cabinet	bookshelf	display	3D IoU
RevealNet [6]	avg. 27-d	17.43	12.64	17.90	28.73	29.61	20.78	18.05	18.68	20.48
Ours (w/o joint)	16-d	16.08	32.74	36.24	46.51	35.53	45.71	33.55	39.63	35.75
Ours (w/o joint)	32-d	11.65	28.07	35.86	39.97	28.89	44.27	23.54	29.44	30.21
Ours (w/o joint)	64-d	20.68	29.46	25.43	24.19	23.55	30.84	22.43	23.20	24.97
Ours (w/ joint)	16-d	19.22	32.55	36.92	46.73	37.05	49.25	35.14	39.28	37.02
Ours (w/ joint)	32-d	15.24	28.55	36.09	41.47	30.91	47.03	24.94	30.27	31.81
Ours (w/ joint)	64-d	22.12	29.24	28.12	27.80	26.05	33.75	23.27	22.87	26.65

Table 2: Comparisons on object reconstruction. Scores above are measured with 3D mesh IoU.

	Max. Time (s)	Max. Memory (MB)
RevealNet [6]	0.72	4273
Ours	0.68	1239

Table 3: Maximal inference time (seconds) and GPU memory (MB) of a single forward pass in ScanNet v2 [2].

Scan2Cad are only partially labeled.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019. 2

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 3, 4

[3] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2017. 1

[4] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 1

[5] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019. 3

[6] Ji Hou, Angela Dai, and Matthias Nießner. Revealnet: Seeing behind objects in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2098–2107, 2020. 2, 3, 4

[7] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 2

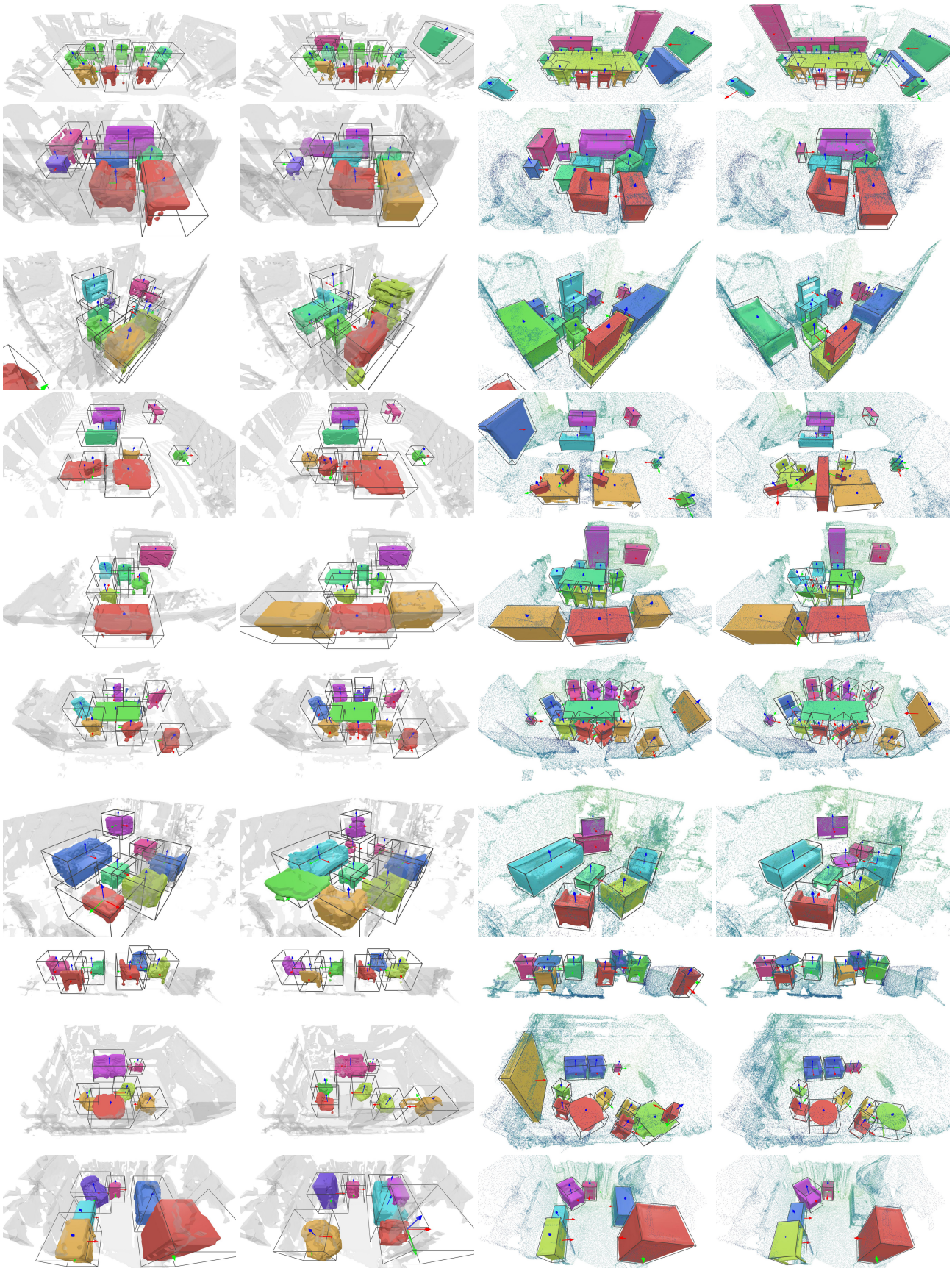
[8] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 2

[9] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 1

[10] 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. 1

[11] Charles Ruizhongtai 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 5099–5108, 2017. 1

[12] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10447–10456, 2020. 3



(a) RevealNet [6] (Geo+Image) (b) RevealNet [6] (Geo Only) (c) Ours (Geo Only) (d) GT

Figure 6: Qualitative results of semantic instance reconstruction on ScanNet v2 [2]. Note that RevealNet [6] preprocesses the scanned scenes into TSDF grids, while our method only uses the raw point clouds.