Supplementary Materials
Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images

In Section 1, we present evaluation on SUN RGB-D dataset [2]. In Section 2, we show the architectures of our networks and describe some implementation details. In Section 3, we show the precision and recall curves for the evaluation on NYU dataset [1] and SUN RGB-D dataset [2].

1. Result on SUN RGB-D

Table 1 shows the evaluation for region proposal generation on SUN RGB-D. We show the results for both 3D Selective Search (3D SS), and our 3D multi-scale Region Proposal Network (RPN).

																				Recall	ABO	#Box
3D SS	78.8	87.2	72.8	72.2	65.5	86.1	75.1	65.0	70.0	87.1	67.5	53.1	68.1	82.8	86.8	84.4	85.0	69.2	94.0	72.0	0.394	2000
RPN	98.1	99.1	79.5	51.5	93.3	89.2	94.9	24.0	87.0	79.6	62.0	41.2	96.2	77.9	96.7	97.3	96.7	63.3	100.0	88.7	0.485	2000

Table 1. **Evaluation for regoin proposal generation on SUN RGB-D test set.**

Table 2 shows the amodal 3D object detection results on SUN RGB-D test-set with different models and compare it with Sliding Shapes [3].

proposal	algorithm	train set																				mAP
-	Sliding Shapes [3]	SUNrgbd	-	42.09	-		33.42	-	-	-	-	-	-	-	-	-	-	23.28	25.78	-	61.86	-
RPN	dxdydz+img	NYU	36.5	70.9	12.6	0.9	51.6	2.7	13.1	0.0	4.4	12.7	13.7	0.1	13.6	11.5	21.3	44.8	33.3	2.8	47.8	20.7
	dxdydz	SUNrgbd	34.3	74.3	9.8	1.0	61.5	3.3	18.7	0.0	3.7	15.7	11.5	0.0	8.8	10.7	18.4	50.2	49.4	0.2	67.5	23.1
	dxdydz+img	SUNrgbd	44.2	78.8	11.9	1.5	61.2	4.1	20.5	0.0	6.4	20.4	18.4	0.2	15.4	13.3	32.3	53.5	50.3	0.5	78.9	26.9

Table 2. **Evaluation for 3D amodal object detection on SUN RGB-D test set.**

2. Network architectures

For both Region Proposal Network and Object Recognition Network, we randomly initialize all layers by drawing weights from a zero-mean Gaussian with standard deviation 0.01. Biases are initialized to 0. During training, the base learning rate is set as 0.01, and the learning rate is reduced by a factor of 10 every 5k iterations. We run SGD for 10k mini-batch iterations, with momentum of 0.9 and parameter decay of 0.0005. Figure 1 and Table 2 show more detailed information of the Region Proposal Network's network architecture. Figure 2 and Table 4 show more detailed information of the Object Recognition Network's network architecture.

3. Precision and recall curve

Figure 3 and Figure 4 show the precision and recall curves for the 20 categories on NYU test set and SUN RGB-D test set.

References

[1] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 1

[2] S. Song, S. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015. 1

	Layers	Responses	Receptive field (m)	Receptive gap (m)
	RPNDataLayer : dataTrain	data[5]={2,6,208,208,100} label[6]={2,1,15,53,53,26} label_weights[6]={2,2,15,53,53,26} bb_tar_diff[6]={2,6,15,53,53,26} bb_loss_weights[6]={2,6,15,53,53,26} label_1[6]={2,1,4,53,53,26} label_weights_1[6]={2,2,4,53,53,26} bb_tar_diff_1[6]={2,6,4,53,53,26} bb_loss_weights_1[6]={2,6,4,53,53,26}	[0.025,0.025,0.025]	[0.025,0.025,0.025]
	conv1 weight[5]={96,6,5,5,5} relu1 pool1 conv2 weight[5]={192,96,3,3,3} relu2 pool2	conv1[5]={2,96,208,208,100} pool1[5]={2,96,104,104,50} conv2[5]={2,192,104,104,50} pool2[5]={2,192,52,52,25}	[0.125,0.125,0.125] [0.15,0.15,0.15] [0.25,0.25,0.25] [0.3,0.3,0.3]	[0.025,0.025,0.025] [0.05,0.05,0.05] [0.05,0.05,0.05] [0.1,0.1,0.1]
level 1	reduction_1 weight[5]={192,192,1,1,1} relu4 conv_cls_score_1 weight[5]={8,192,2,2,2} reshape_cls_1 cls_score_1 loss_cls_1 conv_box_pred_1 weight[5]={24,192,2,2,2} reshape_box_pred_1 loss_bbox_1	reduction_1[5]={2,192,52,52,25} cls_score_conv_1[5]={2,8,53,53,26} cls_score_conv_1_reshape[6]={2,2,4,53,53,26} cls_score_1[6]={2,2,4,53,53,26} box_pred_1[5]={2,24,53,53,26} box_pred_reshape_1[6]={2,6,4,53,53,26}	[0.3,0.3,0.3] [0.4,0.4,0.4] [0.4,0.4,0.4] [0.4,0.4,0.4] [0.4,0.4,0.4] [0.4,0.4,0.4]	[0.1,0.1,0.1] [0.1,0.1,0.1] [0.1,0.1,0.1] [0.1,0.1,0.1] [0.1,0.1,0.1] [0.1,0.1,0.1]
level 2	conv3 weight[5]={384,192,3,3,3} relu3 pool3 conv_cls_score weight[5]={30,384,5,5,5} reshape_cls cls_score loss_cls conv_box_pred weight[5]={90,384,5,5,5} reshape_box_pred loss_bbox	conv3[5]={2,384,52,52,25} pool3[5]={2,384,53,53,26} cls_score_conv[5]={2,30,53,53,26} cls_score_reshape[6]={2,2,15,53,53,26} cls_score[6]={2,2,15,53,53,26} box_pred[5]={2,90,53,53,26} reshape_box_pred[6]={2,6,15,53,53,26}	[0.5,0.5,0.5] [0.6,0.6,0.6] [1,1,1] [1,1,1] [1,1,1] [1,1,1] [1,1,1]	[0.1,0.1,0.1] [0.1,0.1,0.1] [0.1,0.1,0.1] [0.1,0.1,0.1] [0.1,0.1,0.1] [0.1,0.1,0.1] [0.1,0.1,0.1]

Table 3. **Network architecture for Regoin Proposal Network.** The size of filters and responses are shown in brackets; receptive field and receptive gap are shown in meters.

[3] S. Song and J. Xiao. Sliding Shapes for 3D object detection in depth images. In *ECCV*, 2014. 1

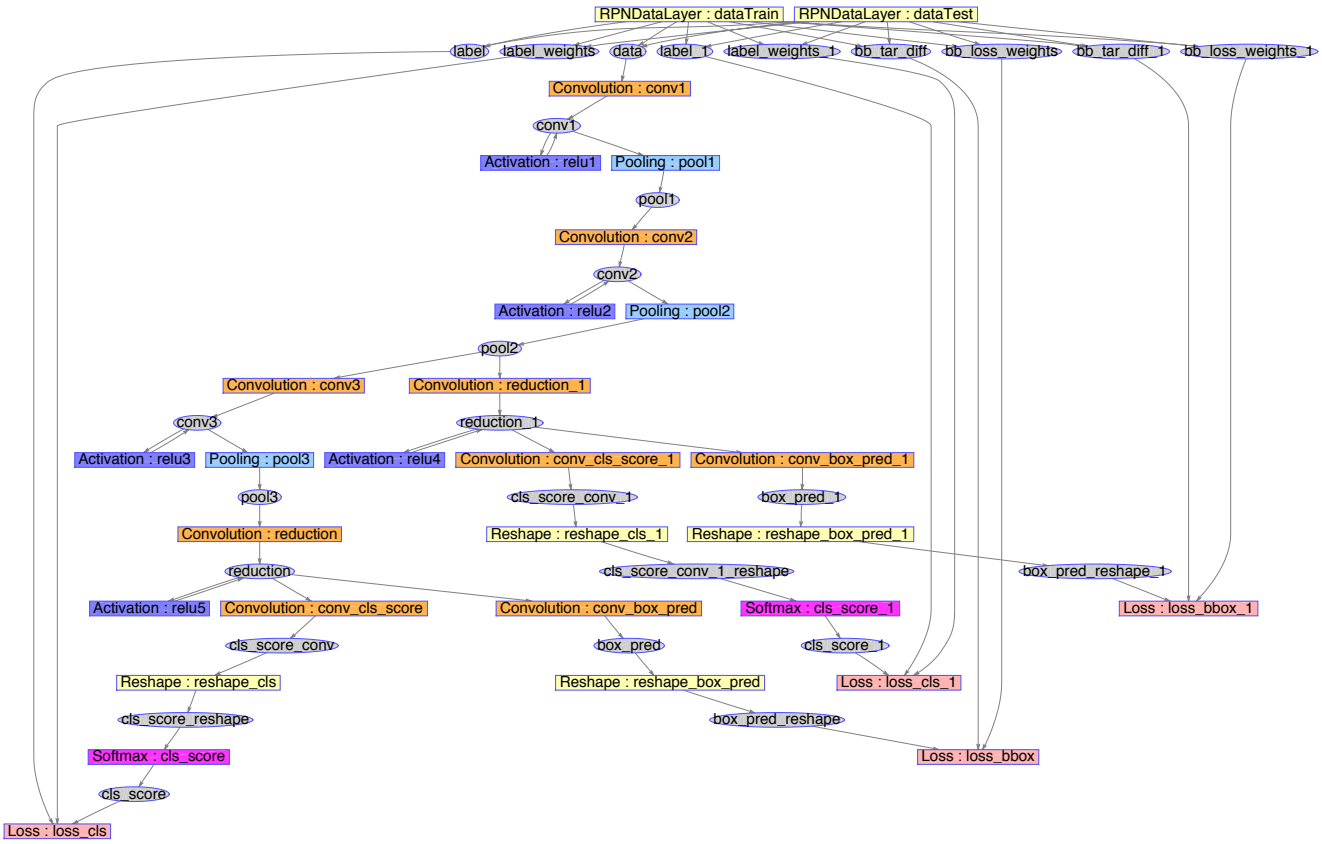

Figure 1. **Network architecture for Regoin Proposal Network.**

Layers	Responses	Receptive field (grid)	Receptive gap (grid)
Scene3DData: dataTrain	data[5]={384,3,30,30,30} label[5]={384,1,1,1,1} bb_tar_diff[5]={384,120,1,1,1} bb_loss_weights[5]={384,120,1,1,1}	[1,1,1]	[1,1,1]
conv1 weight[5]={96,3,5,5,5}	conv1[5]={384,96,28,28,28}	[5,5,5]	[1,1,1]
relu1			
pool1	pool1[5]={384,96,14,14,14}	[6,6,6]	[2,2,2]
conv2 weight[5]={192,96,3,3,3} bias[5]={1,192,1,1,1}	conv2[5]={384,192,12,12,12}	[10,10,10]	[2,2,2]
relu2			
pool2	pool2[5]={384,192,6,6,6}	[12,12,12]	[4,4,4]
conv3 weight[5]={384,192,3,3,3} bias[5]={1,384,1,1,1}	conv3[5]={384,384,4,4,4}	[20,20,20]	[4,4,4]
relu3			
fc4 weight[2]={4096,24576} bias[1]={4096}	fc4[5]={384,4096,1,1,1}	[32,32,32]	
relu4			
drop4			
fc5 weight[2]={1000,4096} bias[1]={1000}	fc5[5]={384,1000,1,1,1}	[32,32,32]	
relu5			
drop5			
fc_cls weight[2]={20,1000} bias[1]={20}	cls_score[5]={384,20,1,1,1}	[32,32,32]	
fc_box_pred weight[2]={120,1000} bias[1]={120}	box_pred[5]={384,120,1,1,1}	[32,32,32]	
cls_score			
loss_cls			
loss_bbox			

Table 4. **Network architecture for Object Recognition Network.** The size of filters and responses are shown in brackets; receptive field and receptive gap are shown in grid cells.

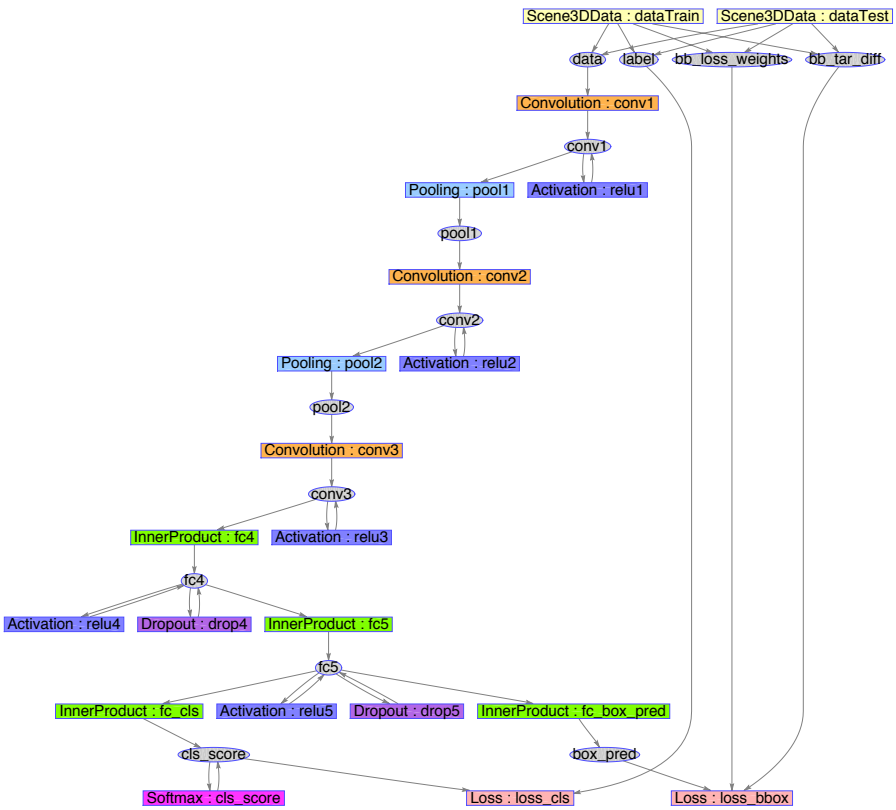

Figure 2. **Network architecture for Object Recognition Network.**

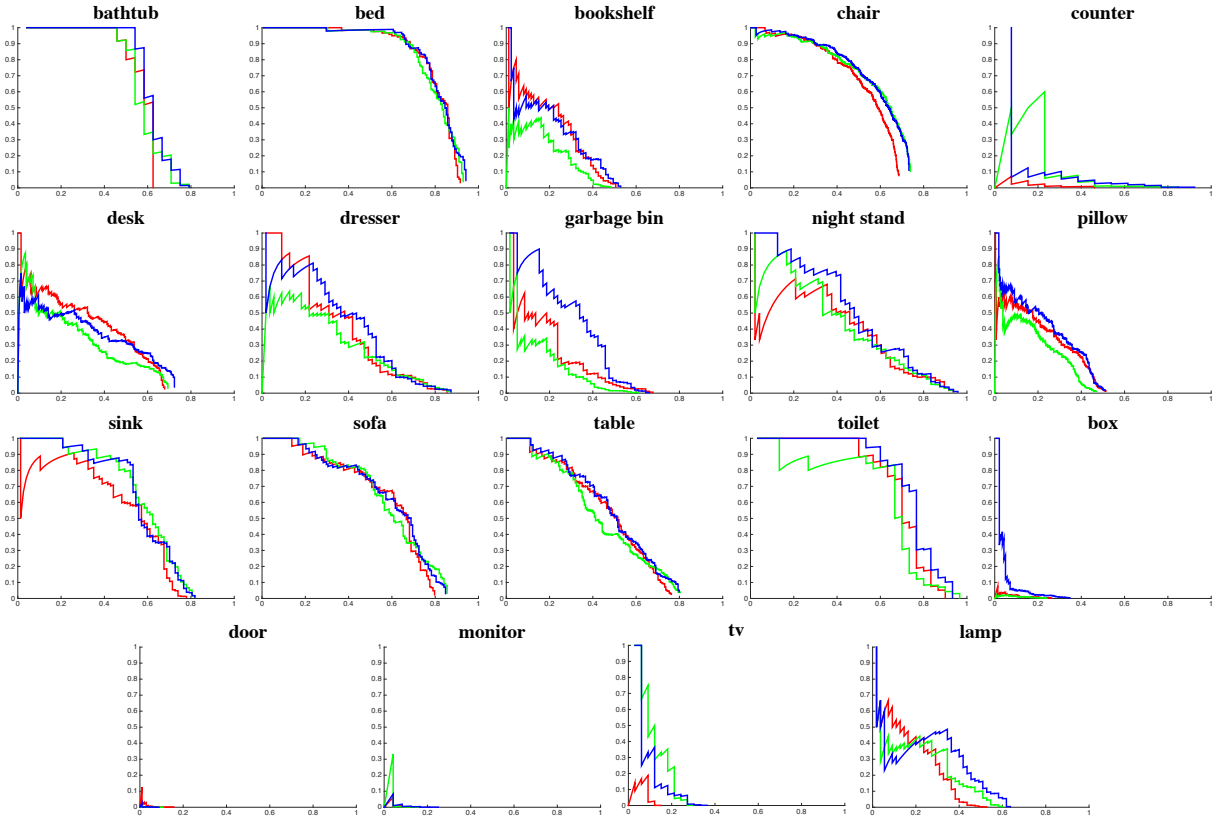

Figure 3. **Precision and recall curves on NYU testset.** All models are trained on NYU. **red**: [proj dxdydz+img]; **green**: [dxdydz]; **blue**: [dxdydz+img].

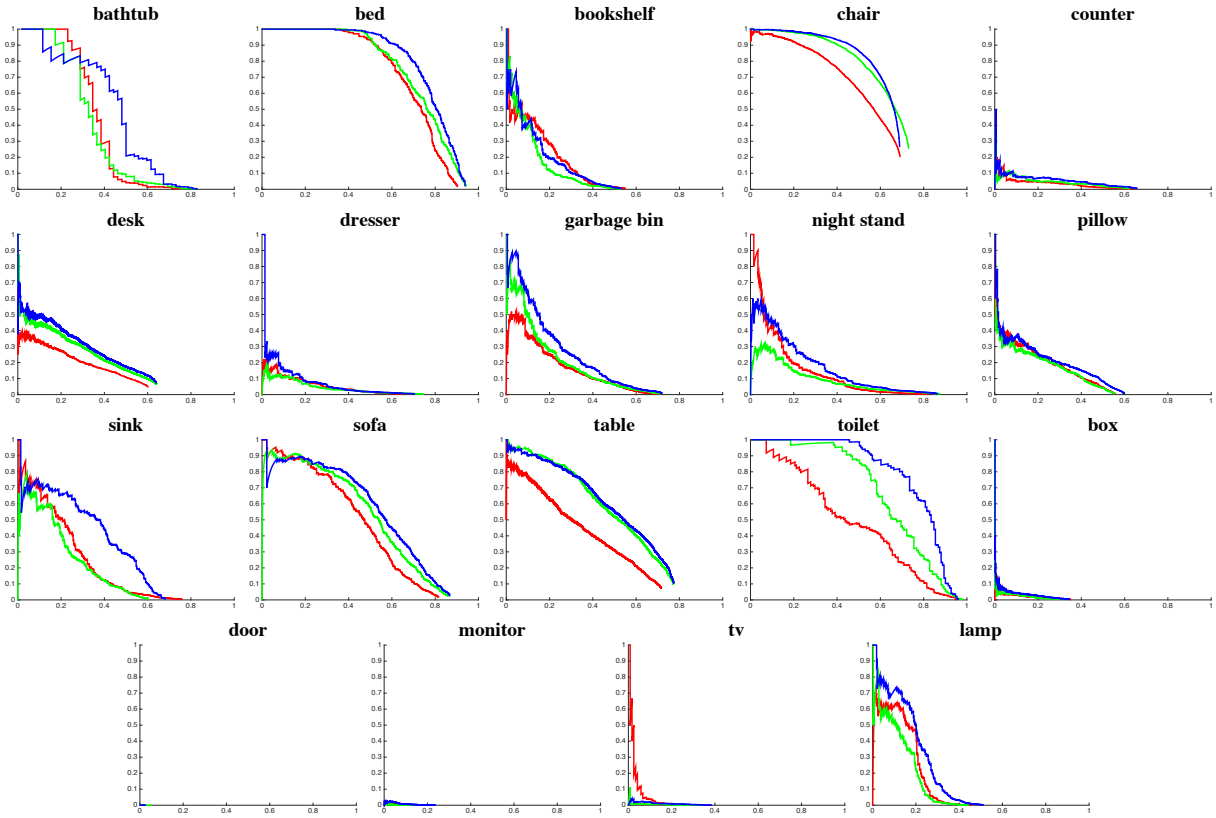

Figure 4. **Precision and recall curves on SUN RGB-D testset.** **red**: [dxdydz+img trained on NYU]; **green**: [dxdydz trained on SUN RGB-D]; **blue**: [dxdydz+img trained on SUN RGB-D].