Supplementary Materials for Generative Sparse Detection Networks

JunYoung Gwak[1], Christopher Choy[2], and Silvio Savarese[1]

[1] Stanford University {jgwak,ssilvio}@stanford.edu
[2] NVIDIA cchoy@nvidia.com

1 Controlled experiments and analysis

In this section, we perform a detailed analysis of GSDN through various controlled experiments. For all experiments, we use the same network architecture, and train and validate the model on the ScanNet dataset [1]. We use the same hyperparameters for all experiments except for one control variable and train all networks for 60k iterations. Note that the performance of the networks trained for 60k iterations is lower than that of networks trained for 120k iterations reported on the main paper.

1.1 Balanced cross entropy loss

One of the main challenges we face during training GSDN is the heavy class imbalance of the sparsity and anchor labels. Such class imbalance is prevalent in object detection and we adopt the balanced cross entropy, one of the well-studied techniques that mitigate various problems associated with class imbalance, for sparsity and anchor prediction. In this section, we demonstrate the effectiveness of the balanced cross entropy loss by comparing it with the network trained with the regular cross entropy loss for sparsity and anchor prediction. We present the object detection result on the ScanNet validation set in Table 1. The balanced cross entropy loss improves the performance of our network, especially the sparsity prediction. This is due to the nature of our generative sparse tensor decoder which adds cubically growing coordinates from all surface voxels, most of which need to be pruned except for a few points that contain target anchor boxes.

Sparsity loss	Anchor loss	mAP@0.25	mAP@0.5
CE	CE	33.1	9.26
BCE	CE	50.7	25.4
BCE	BCE	**57.2**	**29.7**

Table 1: The effectiveness of balanced cross entropy (BCE) and cross entropy (CE) losses for generative sparse object detection.

1.2 Sparsity pruning confidence

Our proposed generative sparse detection network predicts more proposals as we lower the sparsity pruning confidence τ and we found that the threshold τ has a significant impact on the performance. We analyze the effect of the pruning confidence on average recall, mAP@0.25, and decoder runtime on the ScanNet dataset in Figure 1. In this experiment, we train three models with $\tau = \{0.1, 0.3, 0.5\}$ and test them on $\tau = \{0.1, 0.2, \ldots, 0.9\}$. For the pruning confidences that do not have corresponding networks, we select the model trained with the closest pruning confidence.

The general trend of Figure 1 is that smaller pruning confidences τ perform better. Lower pruning confidences lead to more proposals and higher average recall. Also, note that the average precision follows the similar trend, which indicates that the performance is mostly capped by the recall, as shown in the precision/recall curve in the main paper. Lastly, the decoder runs marginally faster as the pruning confidence increases, since it generates fewer proposals.

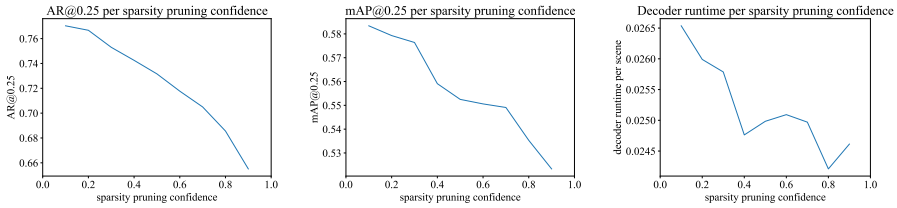

Fig. 1: Analysis of the impact of sparsity pruning confidence τ.

1.3 Encoder backbone models

We vary the sparse tensor encoder and analyze its impact on performance in Table 2. As expected, GSDN with deeper encoder performs better. Additionally, we plot the detailed runtime break-down of each component of our proposed model with varying backbones in Figure 2. Overall, the decoder and post-processing time stay almost constant while the encoder dominates the runtime.

1.4 Anchor ratios

We examine the impact of anchor ratios on the performance of our model in Table 3. Overall, mAP@0.25 (mAP with IoU threshold of 0.25) improves marginally as we use more anchors. However, the improvement of mAP@0.5 with more anchors is significant. mAP@0.5 considers a prediction box with intersection-over-union greater than 0.5 with the corresponding ground-truth box to be positives. In other words, it requires approximately 80% overlap between a

backbone model	mAP@0.25	mAP@0.5
ResNet14	52.1	25.4
ResNet18	55.1	27.8
ResNet34	**57.2**	**29.7**

Table 2: Analysis of the impact of different backbone models on performance.

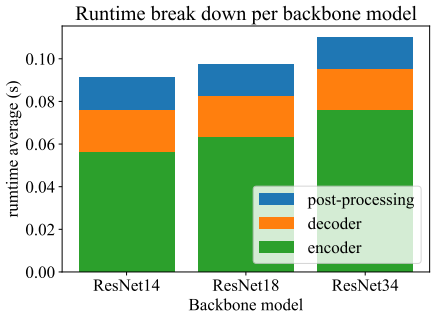

Fig. 2: Runtime breakdown of our model with varying backbones.

prediction and the ground-truth box for each of the three axes for the prediction to be positive. Thus, more anchors allow the network to capture various ground truth boxes more accurately and mAP@0.5 improves significantly.

anchor ratios	mAP@0.25	mAP@0.5
$a_r \in \{1\}$	56.3	22.7
$a_r \in \{1, 4, \frac{1}{4}\}$	55.3	27.0
$a_r \in \{1, 2, 4, \frac{1}{2}, \frac{1}{4}\}$	**57.2**	**29.7**

Table 3: Analysis of the impact of different anchor ratios on performance.

1.5 Generative Sparse Tensor Decoder

The key motivation and main contribution of our work is in dynamically generating sparse tensor coordinates to ground the object detection prediction on. However, one may argue that simply making detection predictions on the observed surface is sufficient. We disprove this hypothesis by building an ablation model which directly makes predictions on the output of the Hierarchical Sparse

Tensor Encoder without the proposed Generative Sparse Tensor Decoder. As shown in Table 4, our proposed model outperforms the encoder-only baseline with a significant margin.

backbone model	mAP@0.25	mAP@0.5
No decoder	52.1	24.6
Ours	**57.2**	**29.7**

Table 4: Analysis of the impact of the proposed Generative Sparse Tensor Decoder.

2 Additional Results

2.1 Experiments on the ScanNet dataset

In Table 5, we report class-wise mAP@0.5 result on the ScanNet v2 validation set. Our method outperforms two-state object detector, Qi *et al.* [3], despite being a single-shot object detector. In Figure 3 and Figure 4, we compare qualitative results on the ScanNet V2 validation set.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	mAP
Hou *et al.* [2]	5.06	42.19	50.11	31.75	15.12	1.38	0.00	1.44	0.00	0.00	13.66	0.00	2.63	3.00	56.75	8.68	28.52	2.55	14.60
Hou *et al.* [2] + 5 views	5.73	50.28	52.59	55.43	21.96	10.88	0.00	13.18	0.00	0.00	23.62	2.61	24.54	0.82	71.79	8.94	56.40	6.87	22.53
Qi *et al.* [3]	8.07	76.06	67.23	68.82	42.36	15.34	6.43	28.00	1.25	9.52	37.52	11.55	27.80	9.96	86.53	16.76	78.87	11.69	33.54
Ours	13.18	74.91	75.77	60.29	39.51	8.51	11.55	27.61	1.47	3.19	37.53	14.10	25.89	1.43	86.97	37.47	76.88	30.53	34.82

Table 5: Class-wise mAP@0.5 object detection result on the ScanNet v2 validation.

2.2 Stanford Large-Scale 3D Indoor Spaces Dataset

We visualize the precision/recall curve of our object detection result on the S3DIS dataset in Figure 6. We observe that certain classes with extreme bounding box ratios such as board and bookcase tend to underperform and have a very low recall. In Figure 5, we visualize additional qualitative results of our method on the S3DIS building 5.

2.3 Gibson environment

We demonstrate the scalability and generalization capability of our network by testing a model trained on the ScanNet dataset which consists of 3D scans

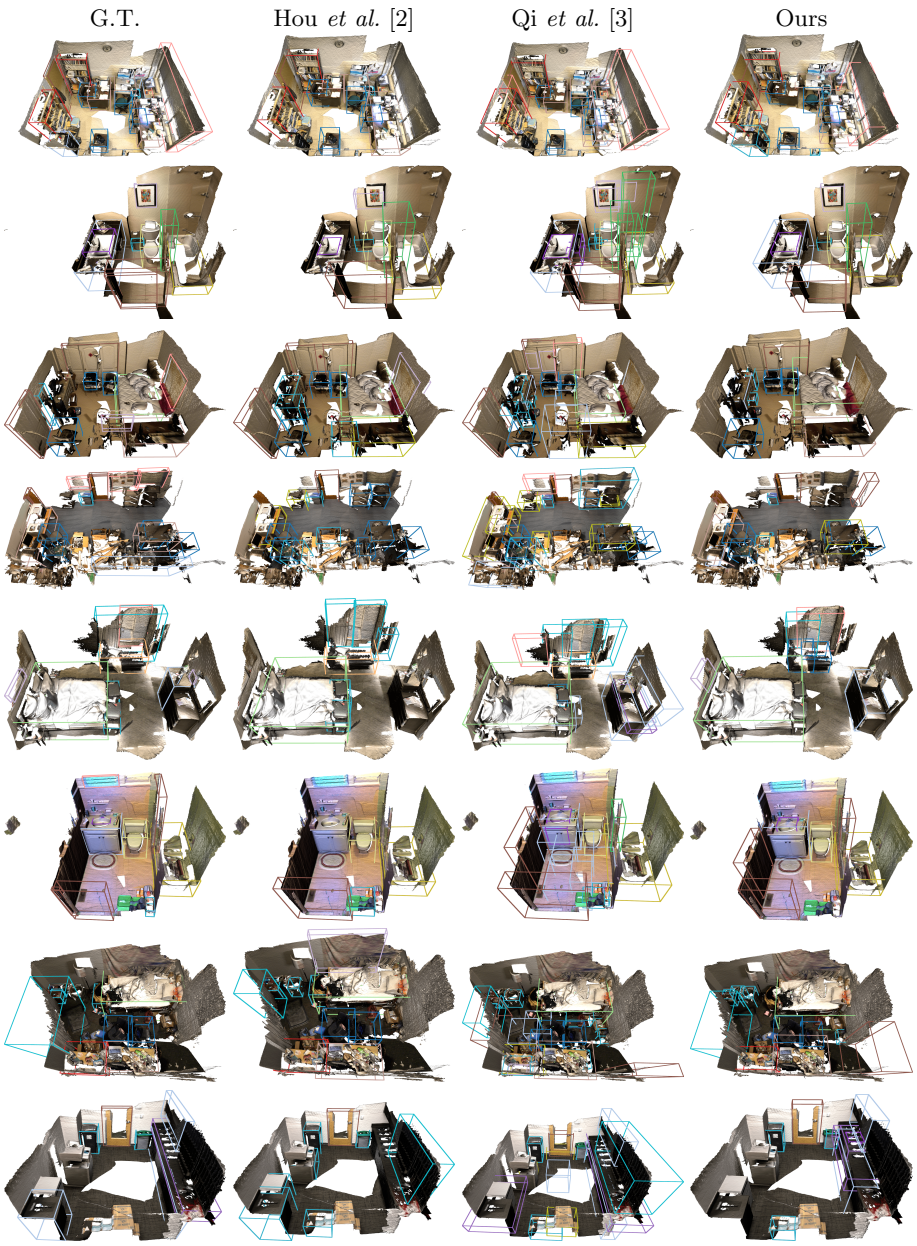

Fig. 3: Qualitative object detection results on the ScanNet dataset [1].

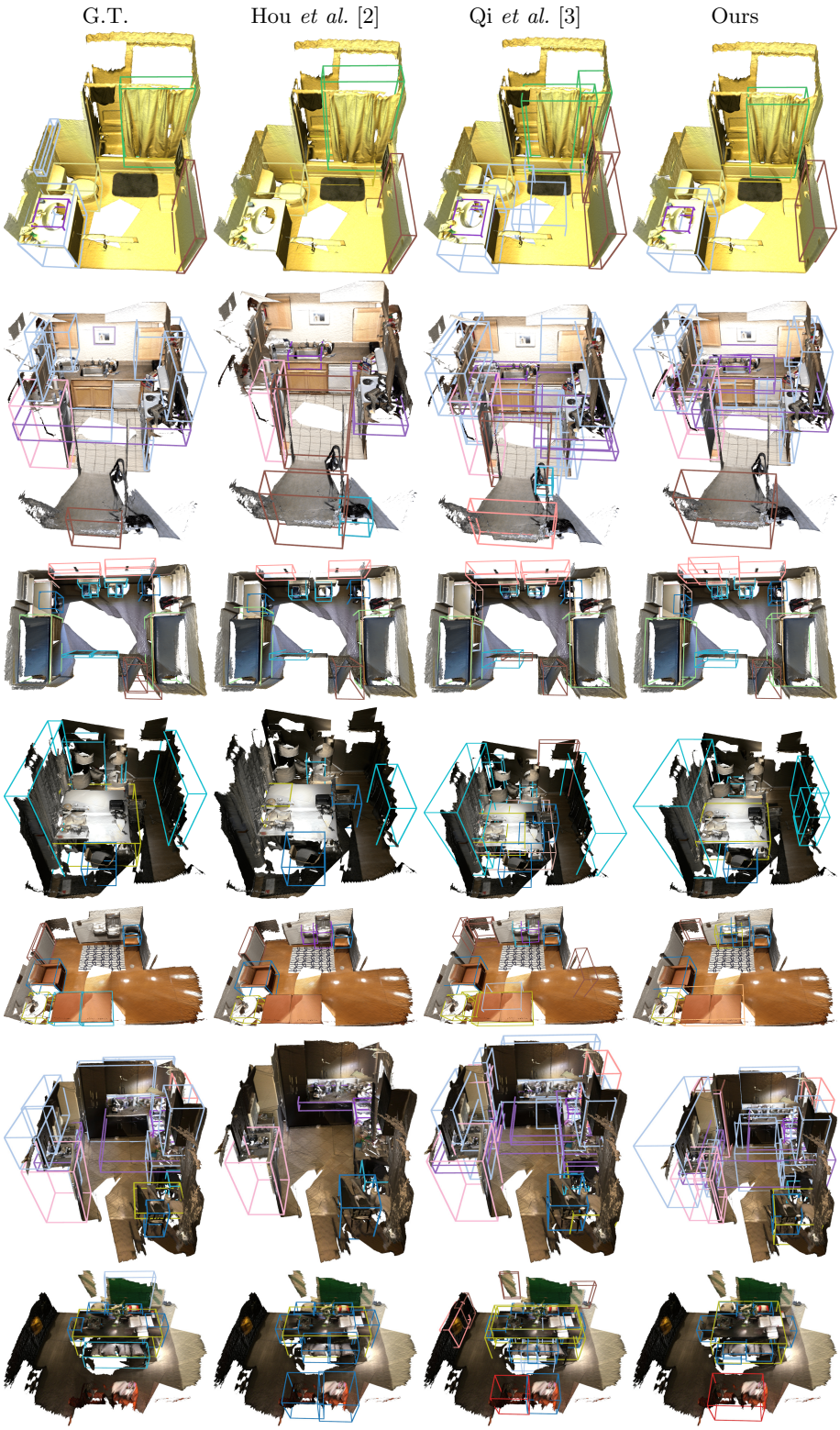

Fig. 4: Qualitative object detection results on the ScanNet dataset [1].

Fig. 5: Qualitative object detection results on the S3DIS dataset.

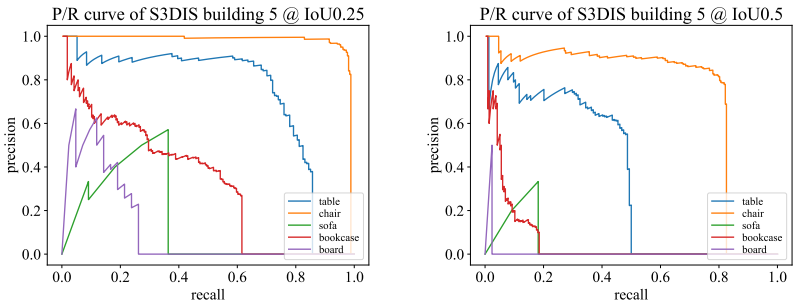

Fig. 6: Per-class object detection precision/recall curves of GSDN on the building 5 of the S3DIS dataset.

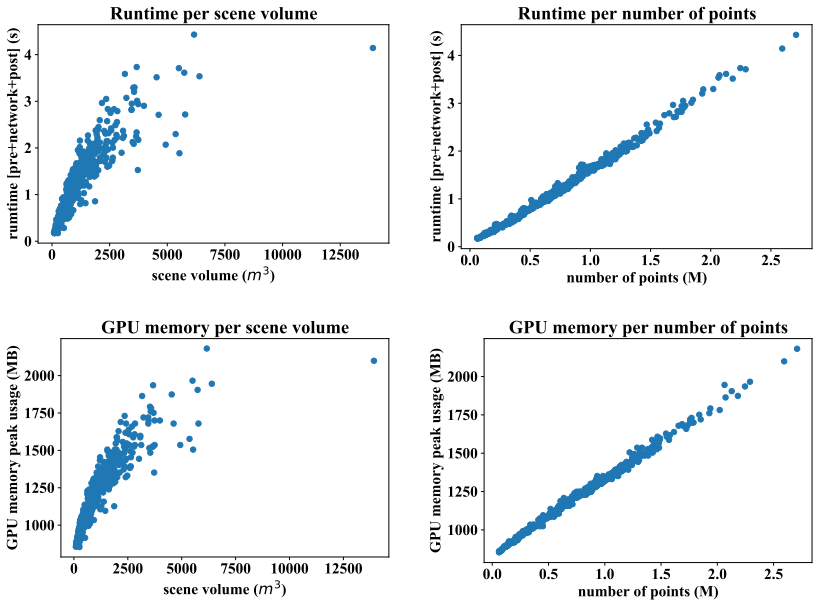

Fig. 7: Runtime and peak memory usage analysis on 572 Gibson V2 environments.

Fig. 8: Qualitative object detection results on the Gibson V2 environments.

of single-story rooms to the multi-story multi-room building in the Gibson environment [5]. Since our network is fully-convolutional and is translation invariant, our model perfectly generalizes to scenes without extra post-processing such as sliding-window-style cropping and stitching results.

We further analyze the runtime and GPU memory usage of our method on the entire 572 Gibson V2 environments. As shown in Figure 7, the runtime and GPU memory usage of our method grows linearly to the number of input points and sublinearly to the volume of the point cloud. This indicates that our method is relatively invariant to the curse of dimensionality. In Figure 8, we visualize additional qualitative results of our method on the Gibson environment.

References

1. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5828–5839 (2017)
2. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4421–4430 (2019)
3. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9277–9286 (2019)
4. Tange, O., et al.: Gnu parallel-the command-line power tool. The USENIX Magazine **36**(1), 42–47 (2011)
5. Xia, F., R. Zamir, A., He, Z.Y., Sax, A., Malik, J., Savarese, S.: Gibson env: real-world perception for embodied agents. In: Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on. IEEE (2018)