3D Object Detection with Pointformer
Supplementary Material

1. Model Architectures and Implementation Details

In this section, we discuss each component in our *Pointformer* architectures for indoor and outdoor settings in detail.

Indoor Datasets. First, the Local Transformer(LT) block is composed of a sequence of sampling and grouping operations, followed by a shared positional encoding layer and two self-attention transformer layers, with a linear shared Feed-Forward Network(FFN) in the end. As shown in Table 1, we use the same sampling and grouping parameters, and feature dimensions as those of PointNet++ [8], the backbone in VoteNet [7] and H3DNet [11].

Second, the Local-Global Transformer(LGT) and the Global Transformer(LT) have fewer hyper-parameters than LT, where we adopt two self-attention layers in GT and one cross-attention layer in LGT for each Pointformer block. Since their massive attention computation may lead to overfitting, we apply dropout with the dropping probability 0.4 on SUN RGB-D [9] and 0.2 on ScanNetV2 [3]. As for the number of heads in multi-head attention, we set it to 8 on ScanNetV2 and 4 on SUN RGB-D. In our experiments, we found that the noisy backgrounds in the indoor datasets affect the LGT performance, by reducing 1$\sim$2%mAP. So we report the results on indoor datasets without the LGT module.

#block	N_{in}	N_{out}	radius	samples	C_{in}	C_{med}	C_{out}
1	N_{point}	2048	0.2	64	64	64	128
2	2048	1024	0.4	32	128	128	256
3	1024	512	0.8	16	256	256	512
4	512	256	1.2	16	512	512	512

Table 1. **Model Architecture details on indoor datasets.** N_{in} denotes the number of input points to this Pointformer block, and N_{out} is the number of sampled output points of the block. Radius and samples are hyper-parameters of ball query operation to gather points in a neighborhood in LT. C_{in}, C_{med} and C_{out} denote the dimensions of features in LT, LGT and GT respectively. N_{point} is the scale of original point clouds in the dataset, 20,000 for SUN RGB-D and 40,000 for ScannNetV2.

Finally, we implement our indoor models on the top of MMDetection3D, an open source toolbox 3D object detection. We follow the same hyper-parameters and data augmentation techniques as those of VoteNet. To train a *Pointformer* on SUN RGB-D, we use the AdamW [5, 6] optimizer with an initial learning rate of 3e-4 and weight decay factor of 0.05, and decay the learning rate by 0.3 at epoch 24 and 32 during the training of a total of 36 epochs. And for ScanNet, we use AdamW optimizer with 0.002 learning rate and set 0.1 weight decay. We decay the learning rate by 0.3 at epoch 32 and 40 during the training of 48 epochs.

Outdoor Datasets. We adopt the same structure of Transformer blocks as that for indoor datasets. Two self-attention layers with FFN are adopted in LT and GT, while only one cross-attention layer is utilized in LGT block. The number of heads are set to 8 for both KITTI [4] and nuScenes [1] datasets.

#block	N_{in}	N_{out}	radius	samples	C_{in}	C_{med}	C_{out}
1	16384	4096	0.1	64	64	64	128
2	4096	1024	0.5	32	128	128	256
3	1024	256	1.0	16	256	256	512
4	256	64	2.0	16	512	512	512

Table 2. **Model Architecture details on KITTI datasets.**

We implement our outdoor models on top of OpenPCDet [10], an open source toolbox for LiDAR-based 3D object detection. We follow the same hyper-parameters as that of PointRCNN, including data augmentation, post-processing, etc. To train a Pointformer on KITTI, we use the Adam optimizer with an initial learning rate of 5e-3 and weight decay of 0.01.

2. More Quantitative Results

In this section, we provide more results and analysis on SUN RGB-D and ScanNetV2 as shown in Tab.3&4. With 0.5 IoU threshold, our proposed Pointformer achieves consistent improvements on both dataset. In Tab. 5, we use the one tower version H3DNet [11] as the baseline, showing that our method can work well with the recent advanced model.

Method	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	mAP
VoteNet [7]	8.1	76.1	67.2	68.8	42.4	15.3	6.4	28.0	1.3	9.5	37.5	11.6	27.8	10.0	86.5	16.8	78.9	11.7	33.5
VoteNet*	14.6	77.9	73.1	**80.5**	46.5	**25.1**	**16.0**	41.8	**2.5**	22.3	33.3	25.0	31.0	17.6	**87.8**	23.0	81.6	18.7	39.9
+Pointformer	**19.0**	**80.0**	**75.3**	69.0	**50.5**	24.3	15.0	**41.9**	1.5	**26.9**	**45.1**	**30.3**	**41.9**	**25.3**	75.9	**35.5**	**82.9**	**26.0**	**42.6**

Table 3. Performance comparison of VoteNet with and without Pointformer on **ScanNetV2** validation dataset. The evaluation metric is Average Precision with **0.5 IoU threshold**.* denotes the model implemented in MMDetection3D [2].

Method	bathtub	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP
VoteNet [7]	**49.9**	47.3	4.6	54.1	5.2	13.6	35.0	41.4	19.7	58.6	32.9
VoteNet*	43.5	55.9	**7.2**	**56.5**	**5.7**	12.6	39.7	50.1	20.7	**66.3**	35.8
+Pointformer	42.5	**59.0**	6.3	54.2	5.4	**20.5**	**43.3**	**51.0**	**22.4**	61.2	**36.6**

Table 4. Performance comparison of VoteNet with and without Pointformer on **SUN RGB-D** validation dataset. The evaluation metric is Average Precision with **0.5 IoU threshold**.* denotes the model implemented in MMDetection3D [2].

Method	mAP@0.25	mAP@0.5
H3DNet* - 1 tower	64.1	44.2
+Pointformer	**64.4**	**44.4**

Table 5. Performance comparison of H3DNet [11] with and without Pointformer on **ScanNet V2** validation dataset. For fair comparison we use single backbone instead of multiple backbones. * denotes the model implemented in MMDetection3D [2].

Method (PointRCNN+)	Params	Car (IoU=0.7) Easy	Moderate	Hard
PointNet++(default)	4.04M	88.88	78.63	77.38
Pointformer(small)	4.12M	**89.35**	**79.01**	**78.34**
PointNet++(large)	6.24M	89.01	78.82	77.67
Pointformer(default)	6.06M	**90.05**	**79.65**	**78.89**

Table 6. Comparison of PointNet++ and Pointformer with similar parameters on the val split of KITTI.

Method	Latency	Car (IoU=0.7) Easy	Moderate	Hard
Poinftormer+Linformer	0.22	89.94	79.63	78.85
Pointformer	0.25	**90.05**	**79.65**	**78.89**

Table 7. Performance of Pointformer with and withour the Linformer technique on the val split of KITTI.

3. More Ablation Studies

Parameter Efficiency. To further validate the effectiveness of Pointformer, we conduct experiments and compare the backbones with similar model parameters. We reduce the Transformer layers adopted in each block and refer the model as Pointformer(small). Similarly, we increase the FFN layers in PointNet++ and refer the model as PointNet++(large). As we have shown in Table 6, Pointformer achieves better results under both parameter budgets. Although our model suffers from a performance reduction when using fewer Transformer layers, we are still 0.5% to 1% AP higher for all difficulty levels. Additionally, PointNet++ shows little improvement with larger feature dimensions. By comparison, Pointformer can adapt to deeper models and use learning parameters more efficiently.

Computational Cost Reduction. As stated in Sec. 3.7, Transformer-based modules suffer from heavy computational cost and memory consumption. Therefore, we adopt the Linformer technique to improve model efficiency. The results are shown in Table 7 and we can observe that inference latency is decreased with little drop in performance.

4. More Qualitative Results

We provide additional visualization results in this section. Figure 1 shows more visualized attention maps on SUN RGB-D dataset. Figure 2 and Figure 3 present qualitative results of detection models with Pointformer on ScanNetV2 and KITTI dataset, respectively.

References

[1] H. Caesar, Varun Bankiti, A. Lang, Sourabh Vora, Venice Erin Liong, Q. Xu, A. Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CVPR*, pages 11618–11628, 2020. 1

[2] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3d object

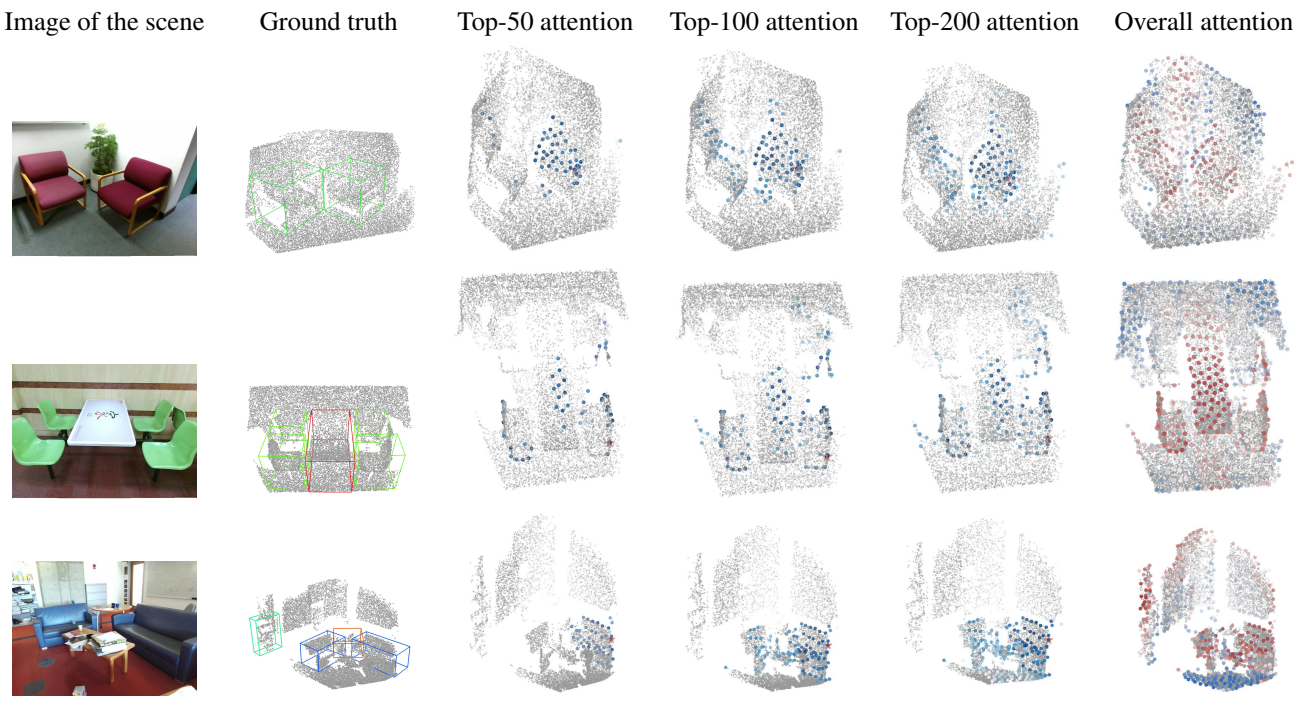

Figure 1. **More attention maps visualizations on SUN RGB-D.** From left to right: Original scene image, ground truth annotations, top-50, 100, 200 attention maps of points to a query point, and the overall attention map for the entire scene. In top-k attention, the star in orange indicates the query point and the darker color indicates larger attention weight, in overall attention red indicates large value. Different object categories are presented with different colors.

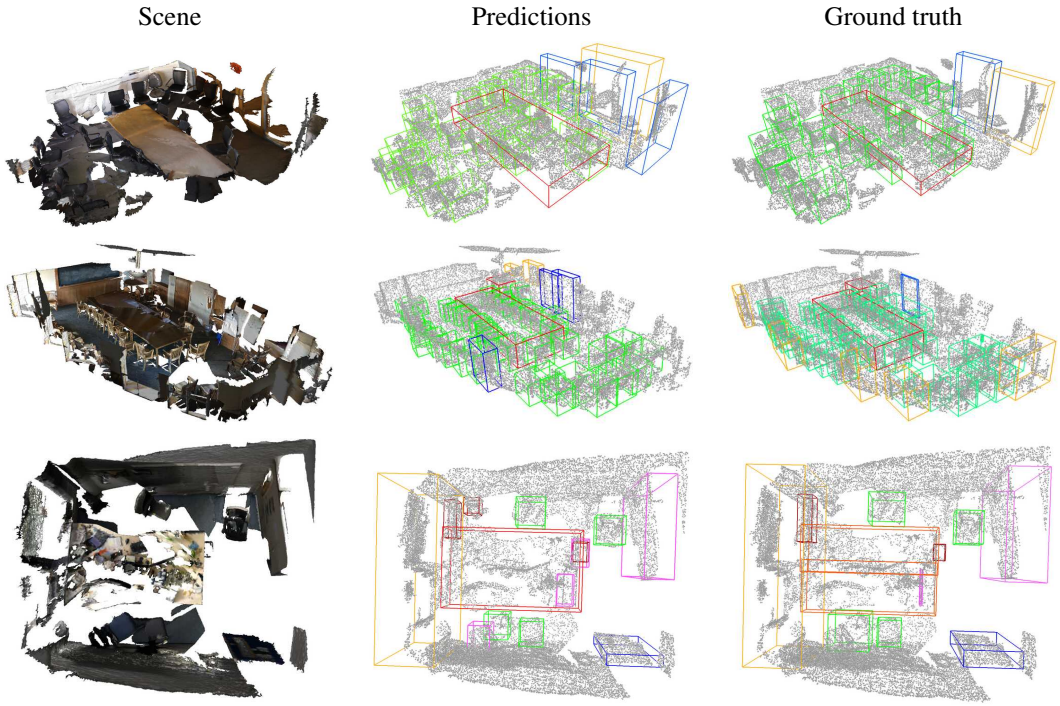

Figure 2. **Qualitative results of 3D object detection on ScanNetV2.** From left to right: Original scene image, our model's prediction, and annotated ground truth boxes. Different object categories are presented with different colors.

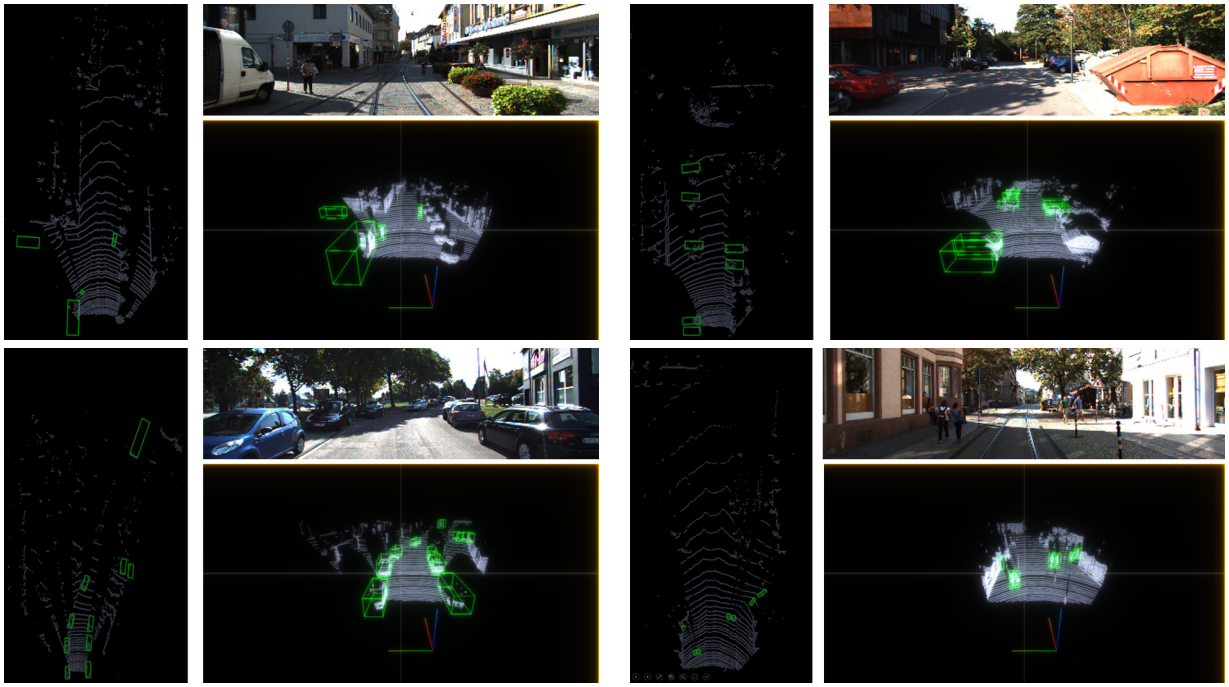

Figure 3. **Qualitative results of 3D object detection on KITTI val split.** We show detection results in four scenes. In each scene, the left is bird eye view detection results, the upper right is the scene image, and the lower right is the front view detection results. Our detection results are consistent with the ground truth labels (not shown).

detection. https://github.com/open-mmlab/mmdetection3d, 2020. 2

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 1

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 1

[6] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. 1

[7] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *CVPR*, pages 9277–9286, 2019. 1, 2

[8] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. 1

[9] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015. 1

[10] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 1

[11] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *European Conference on Computer Vision*, pages 311–329. Springer, 2020. 1, 2