# Appendix: Group-Free 3D Object Detection via Transformers

Ze Liu[1,2*] Zheng Zhang[2†] Yue Cao[2] Han Hu[2] Xin Tong[2]
[1]University of Science and Technology of China
liuze@mail.ustc.edu.cn
[2]Microsoft Research Asia
{zhez,yuecao,hanhu,xtong}@microsoft.com

## A1. Training Details

### A1.1. Our Approach

**ScanNet V2** We follow recent practice [2, 3] to use the PointNet++ as our default backbone network for a fair comparison. The backbone network has four set abstraction layers and two feature propagation layers. For each set abstraction layer, the input point cloud is sub-sampled to 2048, 1024, 512, and 256 points with the increasing receptive radius of 0.2, 0.4, 0.8, and 1.2, respectively. Then, two feature propagation layers successively up-sample the points to 512 and 1024, respectively.

In the training phase, we use 50k[1] points as input and adopt the same data augmentation as in [2], including a random flip, a random rotation between $[-5°, 5°]$, and a random scaling of the point cloud by [0.9, 1.1]. The network is trained from scratch by the AdamW optimizer ($\beta_1$=0.9, $\beta_2$=0.999) with 400 epochs. The weight decay is set to 5e-4. The initial learning rate is 0.006 and decayed by 10$\times$ at the 280-th epoch and the 340-th epoch. The learning rate of the attention modules is set as 1/10 of that in the backbone network. The *gradnorm_clip* is applied to stabilize the training dynamics. Following [2] we use class-aware head for box size prediction.

**SUN RGB-D** The implementation settings mostly follow [2]. We use 20k points as input for each point cloud. The network architecture and the data augmentation are the same as that for ScanNet V2. As the orientation of the 3D box is required in evaluation, we include an additional orientation prediction branch for all decoder layers. The orientation branch contains a classification task and an offset regression task with loss weights of 0.1 and 0.04, respectively.

*This work is done when Ze Liu is an intern at MSRA.
†Contact person
[1]We evaluate our model on 40k points on ScanNet V2 according to previous works and the performance is similar: 66.3(40k) vs. 66.2(50k) on mAP@0.25, and 48.5(40k) vs. 48.6(50k) on mAP@0.5.

| method | mAP@0.25 | mAP@0.5 |
|---|---|---|
| average | 64.2 | 44.2 |
| max | 65.1 | 44.4 |

Table 1. Comparison between average-pooling and max-pooling on ScanNet V2.

In training, the network is trained from scratch by the AdamW optimizer ($\beta_1$=0.9, $\beta_2$=0.999) with 600 epochs if not specified. The initial learning rate is 0.004 and decayed by 10$\times$ at the 420-th epoch, the 480-th epoch, and the 540-th epoch. The learning rate of attention modules is set as 1/20 of the backbone network. The weight decay is set to 1e-7, and the *gradnorm_clip* is used. We use class-agnostic head for size prediction.

### A1.2. Other Pooling Mechanisms

For a fair comparison, we only switch the feature aggregation mechanism while all other settings remain unchanged. In the following, we will introduce the implementation details of RoI-Pooling and Voting aggregation mechanism.

**RoI-Pooling** For a given object candidate, the points within the predicted box of the object candidate are aggregated together, and the refined box is predicted from the aggregated features. The same as our group-free approach, the multi-stage refinement is also adopted. Thus the aggregated points and features will be updated and refined in multiple stages. Also, we tried two different strategies for feature aggregation: average-pooling and max-pooling. The results are shown in Table. 1. We could find that the approach with max-pooling performs better, so we use it for comparison by default.

**Voting** The voting mechanism is first introduced by VoteNet [2] and we implement it in our framework. Specifically, each point predicts the center of its corresponding object, and if the distance between the predicted center of

points and the center of an object candidate is less than a threshold (set to 0.3 meters), then these points and the candidate are grouped. Further, a two-layer MLP with max-pooling is used to form the aggregation feature of the object candidate, and the refined boxes are predicted from the aggregated features in the multi-stage refinement process.

## A2. More Results

We show per-category results on ScanNet V2 and SUN RGB-D under different IoU thresholds. Table 2 and Table 3 show the results of mAP@0.25 and mAP@0.5 on ScanNet V2, respectively. Table 4 and Table 5 show the results of mAP@0.25 and mAP@0.5 on SUN RGB-D, respectively.

We also show more qualitative results of our method on ScanNet V2 and SUN RGB-D. The results are shown in Figure 1 (ScanNet V2) and Figure 2 (SUN RGB-D).

## References


[1] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 392–401, 2020. 3

[2] 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, 3

[3] Xie Qian, Lai Yu-kun, Wu Jing, Wang Zhoutao, Zhang Yiming, Xu Kai, and Wang Jun. Mlcvnet: Multi-level context votenet for 3d object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[4] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. *arXiv preprint arXiv:2006.05682*, 2020. 3

| methods | backbone | cab | bed | chair | sofa | tabl | door | wind | bkshf | pic | cntr | desk | curt | fridg | showr | toil | sink | bath | ofurn | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VoteNet [2] | PointNet++ | 47.7 | 88.7 | 89.5 | 89.3 | 62.1 | 54.1 | 40.8 | 54.3 | 12.0 | 63.9 | 69.4 | 52.0 | 52.5 | 73.3 | 95.9 | 52.0 | 92.5 | 42.4 | 62.9 |
| MLCVNet [3] | PointNet++ | 42.5 | 88.5 | 90.0 | 87.4 | 63.5 | 56.9 | 47.0 | 56.9 | 11.9 | 63.9 | 76.1 | 56.7 | **60.9** | 65.9 | 98.3 | 59.2 | 87.2 | 47.9 | 64.5 |
| H3DNet [4] | 4×PointNet++ | 49.4 | 88.6 | 91.8 | **90.2** | 64.9 | **61.0** | 51.9 | 54.9 | **18.6** | 62.0 | 75.9 | 57.3 | 57.2 | 75.3 | 97.9 | 67.4 | 92.5 | 53.6 | 67.2 |
| Ours (L6, O256) | PointNet++ | 54.1 | 86.2 | 92.0 | 84.8 | 67.8 | 55.8 | 46.9 | 48.5 | 15.0 | 59.4 | 80.4 | 64.2 | 57.2 | **76.3** | 97.6 | **76.8** | 92.5 | 55.0 | 67.3 |
| Ours (L12, O256) | PointNet++ | 55.4 | 86.6 | 91.8 | 86.6 | **73.0** | 54.5 | 49.4 | 47.7 | 13.1 | 63.3 | **82.4** | 63.3 | 53.2 | 74.0 | 99.2 | 67.7 | 91.7 | 55.8 | 67.2 |
| Ours (L12, O256) | PointNet++w2× | **56.5** | 88.2 | 92.5 | 88.2 | 71.6 | 57.5 | 48.3 | 53.7 | 17.5 | **71.0** | 79.5 | 63.4 | 58.1 | 71.7 | 99.4 | 71.1 | 93.0 | **57.8** | 68.8 |
| Ours (L12, O512) | PointNet++w2× | 52.1 | **91.9** | **93.6** | 88.0 | 70.7 | 60.7 | **53.7** | **62.4** | 16.1 | 58.5 | 80.9 | **67.9** | 47.0 | **76.3** | **99.6** | 72.0 | **95.3** | 56.4 | **69.1** |

Table 2. Performance of mAP@0.25 for each category on the ScanNet V2 dataset.

| methods | backbone | cab | bed | chair | sofa | tabl | door | wind | bkshf | pic | cntr | desk | curt | fridg | showr | toil | sink | bath | ofurn | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VoteNet [2] | PointNet++ | 14.6 | 77.8 | 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 |
| H3DNet [4] | 4×PointNet++ | 20.5 | 79.7 | 80.1 | 79.6 | 56.2 | 29.0 | 21.3 | 45.5 | 4.2 | 33.5 | 50.6 | 37.3 | 41.4 | 37.0 | 89.1 | 35.1 | **90.2** | 35.4 | 48.1 |
| Ours (L6, O256) | PointNet++ | 23.0 | 78.4 | 78.9 | 68.7 | 55.1 | 35.3 | 23.6 | 39.4 | 7.5 | 27.2 | 66.4 | 43.3 | 43.0 | 41.2 | 89.7 | 38.0 | 83.4 | 37.3 | 48.9 |
| Ours (L12, O256) | PointNet++ | 23.8 | 77.2 | 81.6 | 65.1 | **62.8** | 35.0 | 21.3 | 39.4 | 7.0 | 33.1 | 66.3 | 39.3 | 43.9 | **47.0** | 91.2 | 38.5 | 85.2 | 37.4 | 49.7 |
| Ours (L12, O256) | PointNet++w2× | **26.2** | 80.7 | **83.5** | 70.7 | 57.0 | 37.4 | 21.2 | 47.7 | **8.8** | **45.3** | 60.7 | 42.2 | 43.5 | 42.7 | **95.5** | **42.3** | 89.7 | **43.4** | 52.1 |
| Ours (L12, O512) | PointNet++w2× | 26.0 | **81.3** | 82.9 | 70.7 | 62.2 | **41.7** | **26.5** | **55.8** | 7.8 | 34.7 | **67.2** | **43.9** | **44.3** | 44.1 | 92.8 | 37.4 | 89.7 | 40.6 | **52.8** |

Table 3. Performance of mAP@0.5 for each category on the ScanNet V2 dataset.

| methods | backbone | bathtub | bed | bkshf | chair | desk | drser | nigtstd | sofa | table | toilet | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VoteNet [2] | PointNet++ | 75.5 | 85.6 | 31.9 | 77.4 | 24.8 | 27.9 | 58.6 | 67.4 | 51.1 | 90.5 | 59.1 |
| MLCVNet [3] | PointNet++ | 79.2 | 85.8 | 31.9 | 75.8 | 26.5 | 31.3 | 61.5 | 66.3 | 50.4 | 89.1 | 59.8 |
| HGNet [1] | PointNet++ w/ FPN | 78.0 | 84.5 | **35.7** | 75.2 | **34.3** | 37.6 | 61.7 | 65.7 | 51.6 | **91.1** | 61.6 |
| H3DNet [4] | 4×PointNet++ | 73.8 | 85.6 | 31.0 | 76.7 | 29.6 | 33.4 | 65.5 | 66.5 | 50.8 | 88.2 | 60.1 |
| Ours (L6, O256) | PointNet++ | **80.0** | **87.8** | 32.5 | **79.4** | 32.6 | **36.0** | **66.7** | **70.0** | **53.8** | **91.1** | **63.0** |

Table 4. Performance of mAP@0.25 for each category on the SUN RGB-D validation set.

| methods | backbone | bathtub | bed | bkshf | chair | desk | drser | nigtstd | sofa | table | toilet | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VoteNet [2] | PointNet++ | 45.4 | 53.4 | 6.8 | 56.5 | 5.9 | 12.0 | 38.6 | 49.1 | 21.3 | 68.5 | 35.8 |
| H3DNet [4] | 4×PointNet++ | 47.6 | 52.9 | 8.6 | 60.1 | 8.4 | 20.6 | 45.6 | 50.4 | 27.1 | 69.1 | 39.0 |
| Ours (L6, O256) | PointNet++ | **64.0** | **67.1** | **12.4** | **62.6** | **14.5** | **21.9** | **49.8** | **58.2** | **29.2** | **72.2** | **45.2** |

Table 5. Performance of mAP@0.5 for each category on the SUN RGB-D validation set.

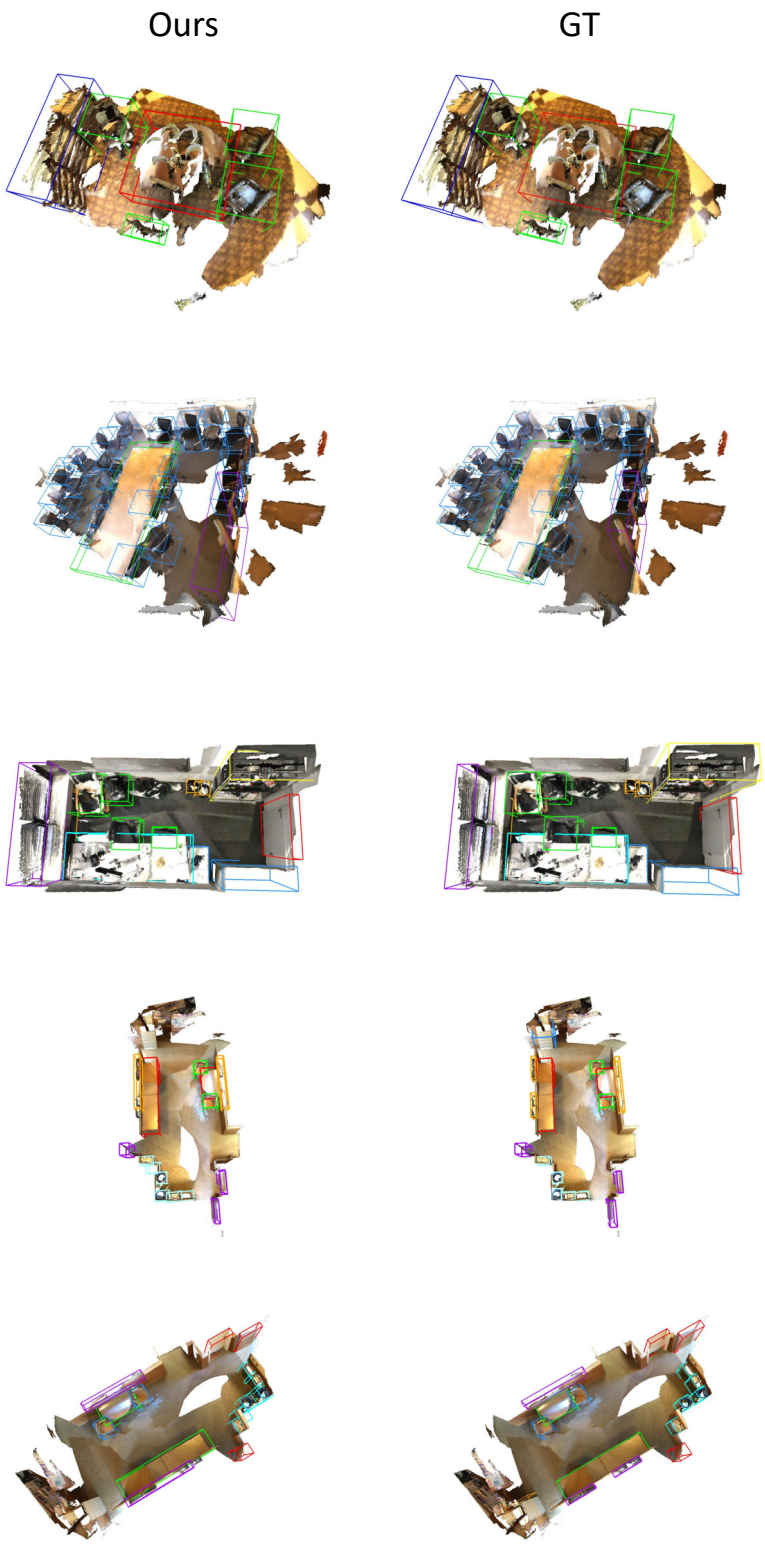


Figure 1. Qualitative results on ScanNet V2.


Figure 2. Qualitative results on SUN RGB-D.