ISBNet: a 3D Point Cloud Instance Segmentation Network with Instance-aware Sampling and Box-aware Dynamic Convolution – Supplementary Material –

Tuan Duc Ngo Binh-Son Hua Khoi Nguyen
VinAI Research, Hanoi, Vietnam
{v.tuannd42, v.sonhb, v.khoindm}@vinai.io

In this supplementary material, we provide:

- The implementation details of the MLP in Point Aggregator and the late devoxelization (Sec. 1).
- Performances when using a smaller backbone (Sec. 2).
- Per-class AP on the ScanNetV2 hidden set (Sec. 3).
- More qualitative results of our approach on all test datasets (Sec. 4).
- The run-time analysis of various methods on the ScanNetV2 validation set (Sec. 5).

1. Implementation Details

The MLP in Point Aggregator (PA). The MLP in the PA consists of three blocks of Conv-BatchNorm-ReLU. The input channel to the MLP is $D + 3$ while the hidden channel and the output channel are set to D.

Late devoxelization. Recent methods [1, 13, 15] adopt the sparse convolutional network [5] as their backbones. It requires the input point clouds to be voxelized as voxel grids and taken as input to the backbone network. The devoxelization step is performed right after the backbone to convert the voxel grids back to points. However, as all points in a voxel grid share the same features, the early devoxelization causes redundant computation and thus increases the memory consumption in later modules [12]. Therefore, following [12], we employ the late devoxelization in our model in both training and testing. Fig. 1 illustrates the differences between late and early devoxelization. The last two rows of Tab. 3 report the average inference time of our model using the early and late devoxelization, respectively. As can be seen, by late devoxelization, our model can reduce the run-time from 268 ms to 237 ms. We would note that applying the late devoxelization does not decrease the accuracy of our model.

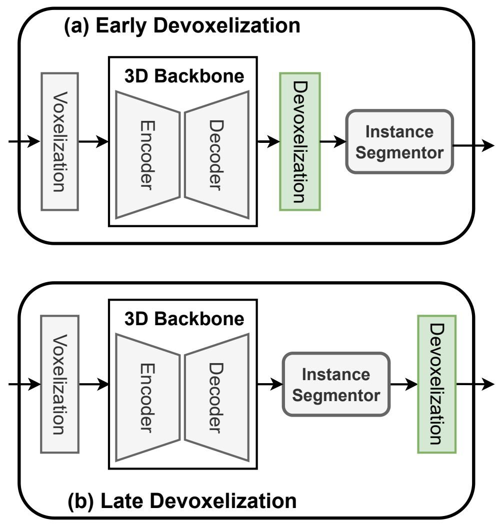

Figure 1. Difference between early and late devoxelization.

	DyCo3D-S	HAIS-S	PointInst3D-S	**ISBNet-S**
AP	35.4	38.0	39.6	**49.5**
AP$_{50}$	57.6	59.1	59.2	**70.1**

Table 1. 3DIS results of recent methods with the same small backbone as used in DyCo3D on ScanNetV2 validation set.

2. Performances when using a smaller backbone

We report the performances of recent methods and ISBNetusing a smaller backbone as in DyCo3D on ScanNetV2 validation set in Tab 1. Our approach consistently outperforms others by a large margin on both AP/AP$_{50}$.

3. Per-class AP on the ScanNetV2 Dataset

We report the detailed results of the 18 classes on the ScanNetV2 hidden set in Tab. 2.

4. More Qualitative Results of Our Approach

The qualitative results of our approach on the ScanNetV2, S3DIS, and STPLS3D datasets are visualized in Fig. 2, Fig. 3, and Fig. 4, respectively.

5. Run-Time Analysis

Training. The training time for our model with the default setting on the ScanNetV2 [2] training set is about 22 hours on a single NVIDIA V100 GPU.

Inference. Tab. 3 shows the average inference time of each component and the whole approach for all scans of the ScanNetV2 validation set. The first 10 rows show the run-time analysis of 10 previous methods. Row 11 presents the runtime of our proposed method. The last row reports the runtime of our model without using late devoxelization (Sec. 1).

References

[1] S. Chen, J. Fang, Q. Zhang, W. Liu, and X. Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the International Conference on Computer Vision*, 2021.

[2] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[3] S. Dong, G. Lin, and T.-Y. Hung. Learning regional purity for instance segmentation on 3d point clouds. In *Proceedings of the European Conference on Computer Vision*, 2022.

[4] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[5] B. Graham, M. Engelcke, and L. Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[6] L. Han, T. Zheng, L. Xu, and L. Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[7] T. He, C. Shen, and A. van den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[8] T. He, C. Shen, and A. van den Hengel. Pointinst3d: Segmenting 3d instances by points. In *Proceedings of the European Conference on Computer Vision*, 2022.

[9] J. Hou, A. Dai, and M. Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] L. Jiang, H. Zhao, S. Shi, S. Liu, C.-W. Fu, and J. Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[11] Z. Liang, Z. Li, S. Xu, M. Tan, and K. Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the International Conference on Computer Vision*, 2021.

[12] T. Vu, K. Kim, T. M. Luu, T. Nguyen, J. Kim, and C. D. Yoo. Softgroup++: Scalable 3d instance segmentation with octree pyramid grouping. *arXiv preprint arXiv:2209.08263*, 2022.

[13] T. Vu, K. Kim, T. M. Luu, X. T. Nguyen, and C. D. Yoo. Softgroup for 3d instance segmentation on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[14] W. Wang, R. Yu, Q. Huang, and U. Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[15] Y. Wu, M. Shi, S. Du, H. Lu, Z. Cao, and W. Zhong. 3d instances as 1d kernels. In *Proceedings of the European Conference on Computer Vision*, 2022.

[16] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. In *Advances in Neural Information Processing Systems*, 2019.

[17] B. Zhang and P. Wonka. Point cloud instance segmentation using probabilistic embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[18] W. Zhao, Y. Yan, C. Yang, J. Ye, X. Yang, and K. Huang. Divide and conquer: 3d point cloud instance segmentation with point-wise binarization. In *Proceedings of the European Conference on Computer Vision*, 2022.

Method	AP	bathtub	bed	bookshe.	cabinet	chair	counter	curtain	desk	door	other	picture	fridge	s.curtain	sink	sofa	table	toilet	window
SGPN [14]	4.9	2.3	13.4	3.1	1.3	14.4	0.6	0.8	0.0	2.8	1.7	0.3	0.9	0.0	2.1	12.2	9.5	17.5	5.4
3D-BoNet [16]	25.3	51.9	32.4	25.1	13.7	34.5	3.1	41.9	6.9	16.2	13.1	5.2	20.2	33.8	14.7	30.1	30.3	65.1	17.8
3D-MPA [4]	35.5	45.7	48.4	29.9	27.7	59.1	4.7	33.2	21.2	21.7	27.8	19.3	41.3	41.0	19.5	57.4	35.2	84.9	21.3
PointGroup [10]	40.7	63.9	49.6	41.5	24.3	64.5	2.1	57.0	11.4	21.1	35.9	21.7	42.8	66.0	25.6	56.2	34.1	86.0	29.1
OccuSeg [6]	48.6	80.2	53.6	42.8	36.9	70.2	20.5	33.1	30.1	37.9	47.4	32.7	43.7	**86.2**	48.5	60.1	39.4	84.6	27.3
DyCo3D [7]	39.5	64.2	51.8	44.7	25.9	66.6	5.0	25.1	16.6	23.1	36.2	23.2	33.1	53.5	22.9	58.7	43.8	85.0	31.7
PE [17]	39.6	66.7	46.7	44.6	24.3	62.4	2.2	57.7	10.6	21.9	34.0	23.9	48.7	47.5	22.5	54.1	35.0	81.8	27.3
HAIS [1]	45.7	70.4	56.1	45.7	36.3	67.3	4.6	54.7	19.4	30.8	42.6	28.8	45.4	71.1	26.2	56.3	43.4	88.9	34.4
SSTNet [11]	50.6	73.8	54.9	49.7	31.6	69.3	17.8	37.7	19.8	33.0	46.3	57.6	51.5	85.7	**49.4**	63.7	45.7	94.3	29.0
SoftGroup [13]	50.4	66.7	57.9	37.2	38.1	69.4	7.2	**67.7**	30.3	38.7	**53.1**	31.9	58.2	75.4	31.8	**64.3**	49.2	90.7	**38.8**
RPGN [3]	42.8	63.0	50.8	36.7	24.9	65.8	1.6	67.3	13.1	23.4	38.3	27.0	43.4	74.8	27.4	60.9	40.6	84.2	26.7
PointInst3D [8]	43.8	81.5	50.7	33.8	35.5	70.3	8.9	39.0	20.8	31.3	37.3	28.8	40.1	66.6	24.2	55.3	44.2	91.3	29.3
DKNet [15]	53.2	81.5	**62.4**	**51.7**	37.7	**74.9**	10.7	50.9	30.4	43.7	47.5	58.1	53.9	77.5	33.9	64.0	50.6	90.1	38.5
ISBNet	**55.9**	**92.6**	59.7	39.0	**43.6**	72.2	**27.6**	55.6	**38.0**	**45.0**	50.5	**58.3**	**73.0**	57.5	45.5	60.3	**57.3**	**97.9**	33.2

Table 2. Per-class AP of 3D instance segmentation on the ScanNetV2 hidden test set. Our proposed method achieves the highest average AP and outperforms the previous strongest method significantly.

Figure 2. Qualitative results on ScanNetV2 dataset. Each column shows one example.

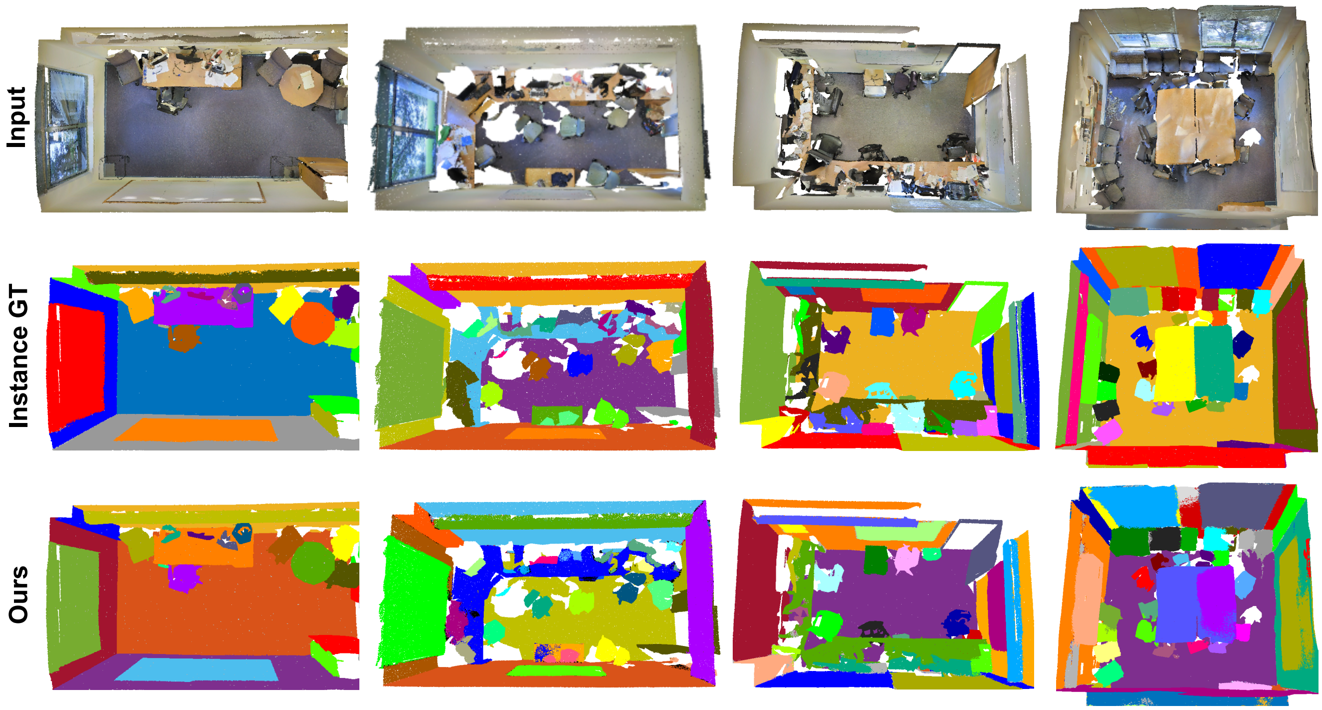

Figure 3. Qualitative results on S3DIS dataset. Each column shows one example.

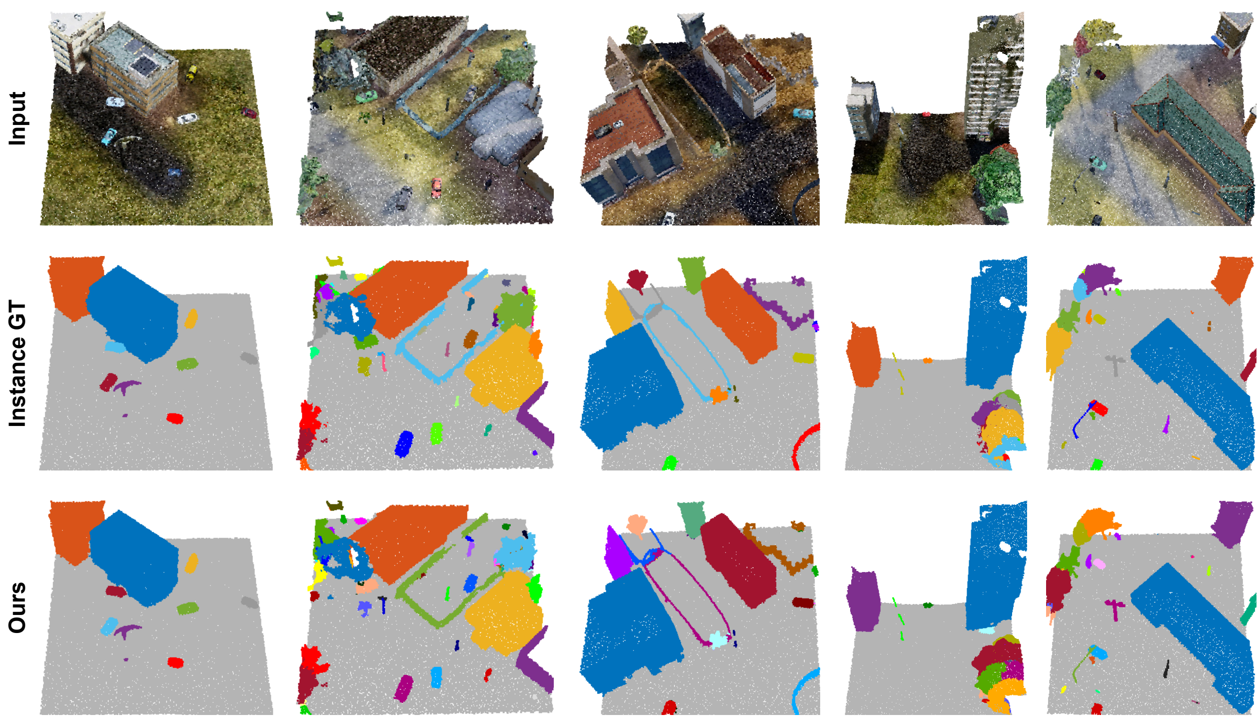

Figure 4. Qualitative results on STPLS3D dataset. Each column shows one example.

Method	**Component time** (ms)	**Total** (ms)
SGPN [14]	Backbone (GPU): 2080 Group merging (CPU): 149000 Block merging (CPU): 7119	158439
3D-BoNet [9]	Backbone (GPU): 2083 Group merging (CPU): 667 Block merging (CPU): 7119	9202
OccuSeg [6]	Backbone (GPU): 189 Group merging (CPU): 1202 Block merging (CPU): 513	1904
PointGroup [10]	Backbone (GPU): 128 Clustering (GPU+CPU): 221 ScoreNet (GPU): 103	452
SSTNet [11]	Backbone (GPU): 125 Tree net. (GPU+CPU): 229 ScoreNet (GPU): 74	428
HAIS [1]	Backbone (GPU): 154 Hier. aggre. (GPU+CPU): 118 ScoreNet (GPU): 67	339
DyCo3D [7]	Backbone (GPU): 154 Weights Gen. (GPU+CPU): 120 Dynamic Conv. (GPU): 28	302
SoftGroup [13]	Backbone (GPU): 152 Soft grouping (GPU+CPU): 123 Top-down refine. (GPU): 70	345
Di&Co [18]	Backbone (GPU): 163 Group, Vote, Merge (GPU+CPU): 275 ScoreNet (GPU): 64	502
DKNet [15]	Backbone (GPU): 165 Cand. Mining & Aggre. (GPU): 379 Dynamic Conv. + Postproc. (GPU): 70	614
ISBNet	Backbone (GPU): 152 Point Pred. & Inst. Enc. (GPU): 53 Dynamic Conv. + Postproc. (GPU): 32	**237**
ISBNet w/o late voxelization	Backbone (GPU): 152 Point Pred. & Inst. Enc. (GPU): 82 Dynamic Conv. + Postproc. (GPU): 34	**268**

Table 3. Average inference time per scan of ScanNetV2 validation set on an NVIDIA Titan X GPU.