3D Object Detection and Relocalization in Indoor Scenes

Yiheng Xiong   Jingsong Liu
Techinical University of Munich
Boltzmannstr. 3
yiheng.xiong@tum.de   ge54xof@mytum.de

Abstract

CenterPoint [17], a novel 3D outdoor deep detector, has achieved state-of-the-art performance in multiple autonomous driving detection benchmarks. However, it has seldom been used in indoor scenes. In this project, we adapt CenterPoint in indoor scenes and evaluate its performance thoroughly. To have a fair comparison, we also conduct controlled experiments with indoor-targeted detector VoteNet [9]. As it turns out, CenterPoint can achieve quite good detection results in indoor scenes with some modifications. Besides, we re-define the task 3D relocalization, targeting moving objects given dynamic annotations. Results show that CenterPoint and VoteNet perform reasonably well in this task.

1. Introduction

Strong 3D indoor perception is a core ingredient in many state-of-the-art home intelligent systems. In particular, 3D object detection in point clouds is an interesting challenge, the goal of which is to estimate oriented 3D bounding boxes and semantic classes of objects from point clouds. Compared to images, 3D point clouds provide accurate geometry and robustness to illumination changes. On the other hand, point clouds are irregular, thus typical CNNs are not well suited to process them directly.

CenterPoint [17] is one of the most novel methods targeting outdoor point clouds detection. It has shown state-of-the-art performance in several outdoor detection datasets including Waymo [15] and nuScenes [2]. However, to our best knowledge, CenterPoint has seldom been tested in indoor scenes. In particular, indoor scenes are much smaller than outdoor scenes, and the sizes of indoor objects are on average smaller than those of outdoor objects. Besides, indoor scenes are more crowded. These differences may bring challenges when adapting outdoor detectors to indoor scenes.

To this end, we decide to test CenterPoint on two indoor datasets - 3RScan [8] and ScanNet [5]. Since there are seldom published detection results on 3RScan, we also test VoteNet [9], one of the state-of-the-art indoor detectors, on 3RScan so that we can have fair comparisons with CenterPoint. As for ScanNet, many detection results have been published which can be used for our comparisons directly. Our study shows that CenterPoint with some modifications can also perform quite well in indoor-scene detection task. Apart from 3D detection, we also re-define 3D relocalization from [8], where we try to find the correspondence of moving objects in the same environment of different scans with known camera calibration matrix. 3RScan provides us with dynamic annotations of moving objects so we conduct experiments with both CenterPoint and VoteNet on it. Both networks have reasonable relocalization results on 3RScan.

In summary, the contributions of our work are:

- Testing and fine tuning VoteNet on 3RScan.
- Testing and fine tuning CenterPoint on 3RScan and ScanNet.
- Demonstration of feasibility of using CenterPoint in indoor scenes.
- Conducting re-defined relocalization experiments with CenterPoint and VoteNet on 3RScan.

2. Related Work

3D object detection. Many previous methods were proposed to detect 3D bounding boxes of objects. [6, 14] extend 2D detection frameworks to 3D. They voxelize the irregular point clouds to regular 3D grids and apply 3D CNN detectors. In [4, 18], the 3D data is first reduced to a bird's-eye view before proceeding to the rest of the pipeline. A reduction in search space by first processing a 2D input was demonstrated in Frustum PointNets [10].

Deep learning in point clouds. Recently we see a surge of interest in designing deep network architectures suited for point clouds. PointNet [11] and PointNet++ [12] are pioneering works which directly deal with raw point clouds and learn local/global features for the downstream tasks. More

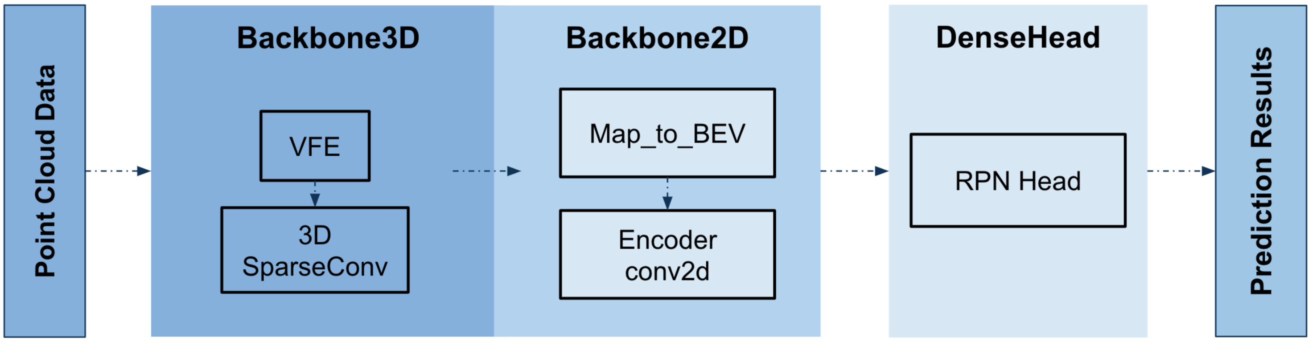

Figure 1. One-staged CenterPoint framework. Taken from [17].

recently, deep networks on point clouds are used to exploit sparsity of the data by GSPN [16] and PointRCNN [13].

3. 3D Relocalization

Given reference scan S_1 with ground truth detection and query scan S_2 with predicting detection of the same environment with moving objects in it, 3D relocalization is defined to find the transform matrices of moving objects from S_1 to S_2 (See A.1). Unlike [8] where the camera calibration matrix from S_1 to S_2 is predicted using Singular Value Decomposition (SVD), we take it as our prior knowledge. During inference time, after calibrating S_2 to S_1, we associate the predicting detection of S_2 and ground truth detection of S_1 in a greedy fashion. Specifically, corresponding boxes in predicting results are selected based on nearest search of their semantic classes and sizes and transform matrices are calculated from the corresponding boxes.

4. Methods

4.1. CenterPoint

We choose the one-staged CenterPoint as our model. Figure 1 shows the overall framework.

Backbone 3D. 3D feature encoding relies on a standard 3D backbone. In particular, VoxelNet [18] is adopted - it first encodes per-voxel feature by averaging learned point features within the corresponding voxel then uses 3D sparse convolution layers to extract map-view feature representation.

Backbone 2D. With learned map-view feature representation, 2D convolutional layers in 2D backbone are applied after compressing the height and mapping to bird's eye view.

Dense head. The dense head's goal is to produce a heatmap peak at the center location of any detected object and regress several object properties at the center-features of the objects including a sub-voxel location refinement $o \in \mathbb{R}^2$, height-above-ground $h_g \in \mathbb{R}$, the 3D size $s \in \mathbb{R}^3$, and a yaw rotation angle $\alpha \in [-\pi, \pi)$.

4.2. VoteNet

Figure 2 illustrates the architecture of VoteNet [9]. The entire model can be split into three parts: point cloud feature learning, Hough voting [7] with deep networks and object proposal and classification.

Point cloud feature learning. PointNet++ [12] is adopted as the backbone. Given an input point cloud of size $N \times 3$, it is processed by several set-abstraction layers and feature propagation (upsampling) layers with skip connections and the output is a subset of the input points called seeds with XYZ and an enriched feature vector.

Hough voting with deep networks. Given a set of seeds with the size of $M \times (3 + C)$, with a C-dimensional feature vector, it forces seeds on the surface of the same object closer to the corresponding object centroid to generate votes with the same tensor representation as seeds, which makes it easier to combine cues from different parts of the object afterwards.

Object proposal and classification. Votes are clustered through sampling and grouping and object proposals are generated from vote clusters. The proposal is essentially a multidimensional vector with an objectness score, bounding box parameters and semantic classification scores.

5. Experiments

In this section, we firstly conduct experiments on VoteNet to get a fair comparing baseline on 3RScan [8]. We then test and tune CenterPoint on 3RScan and ScanNet [5] to evaluate its performance.

Dataset. 3RScan is a novel dataset which features around 1.5K RGB-D scans of around 500 environments across multiple time steps. It is annotated with amodal oriented 3D bounding boxes for 7 object categories.

ScanNetV2 is a richly annotated dataset of 3D reconstructed meshes of indoor scenes. It contains around 1.2K training examples and is annotated with semantic and instance segmentation for 18 object categories. We aim to predict axis-aligned bounding boxes because of the lack of

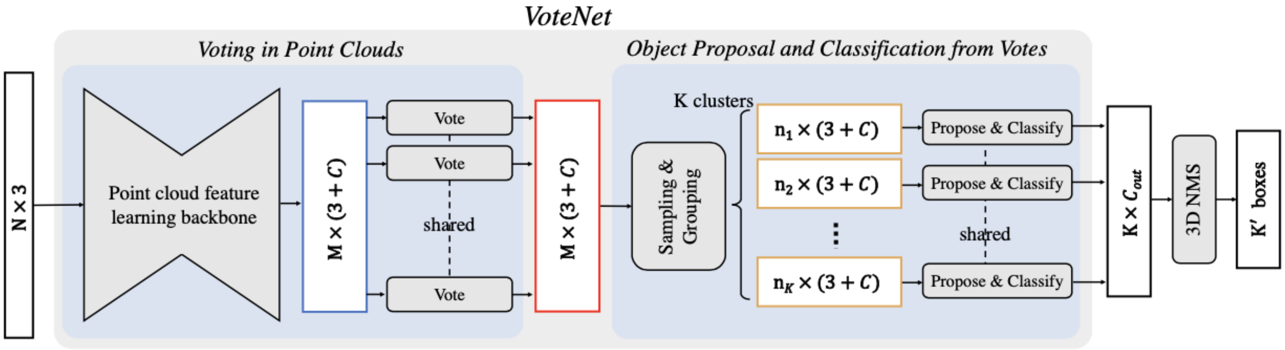

Figure 2. VoteNet architecture. Taken from [9].

	Input	seating	table/cabinet	bed/sofa	appliances	cushions	items	structure	mAP
VoteNet	Geo	60.0	**24.0**	60.2	**35.9**	18.9	3.8	7.1	30.0
original CenterPoint	Geo	26.2	7.0	63.5	8.5	4.5	0.2	6.7	16.6
tuned CenterPoint	Geo	**66.4**	22.5	**68.1**	26.3	**33.9**	**7.5**	**13.0**	**34.0**

Table 1. **3D object detection results on 3RScan val set.** Evaluation metric is average precision with 3D IoU threshold 0.25.

	Input	mAP@0.25	mAP@0.5
DSS [6, 14]	Geo + RGB	15.2	6.8
MRCNN 2D-3D [1, 6]	Geo + RGB	17.3	10.5
F-PointNet [10]	Geo + RGB	19.8	10.8
GSPN [16]	Geo + RGB	30.6	17.7
3D-SIS [6]	Geo + 1 view	35.1	18.7
3D-SIS [6]	Geo + 3 views	36.6	19.0
3D-SIS [6]	Geo + 5 views	40.2	22.5
3D-SIS [6]	Geo	25.4	14.6
VoteNet [9]	Geo	**58.6**	**33.5**
original CenterPoint	Geo	29.3	13.6
tuned CenterPoint	Geo	45.7	31.2

Table 2. **3D object detection results on ScanNetV2 val set.** All other numbers are extracted from [9].

	Recall$< 0.2m$,20°	MRE[deg]	MTE[m]
VoteNet	7.94	9.32	0.080
CenterPoint	12.94	9.52	0.069

Table 3. **3D object relocalization results on 3RScan val set.** We evaluate the predicting rotation R_p and translation t_p against the ground truth annotation R_{GT} and t_{GT}. An instance will be judged to be successfully aligned if the alignment error for the translation $t_\triangle < 20cm$ and rotation $R_\triangle < 20°$. Numbers are reported in terms of average $\%$ correct rotation and translation predictions. MTE (Median Translation Error) is measured in meters and MRE (Median Rotation Error) is in degrees.

oriented bounding box annotation in ScanNetV2.

Input and data augmentation. Input to VoteNet is a point cloud of $20k$ points randomly sub-sampled from the entire scene. In addition to XYZ coordinates, we also include a height feature for each point indicating its distance from the floor.

As for CenterPoint, the detection range is set to $[-6m, 6m]$ for X and Y axis, and $[-3.5m, 3.5m]$ for Z axis on 3Rscan, and $[-3.5m, 3.5m]$ for X axis, $[-6.5m, 5.5m]$ for Y axis, and $[-1.5m, 3.5m]$ for Z axis on ScanNetV2. We don't add height feature to the input in CenterPoint.

We augment the point clouds in the same way for both networks including random flipping in both horizontal directions, random rotation by Uniform$[-5°, 5°]$ and random scaling by Uniform$[0.9, 1.1]$.

Implementation details. We keep the architecture of VoteNet the same as the original one without any modifications. And for CenterPoint, we reduce the voxel size from$[0.1m, 0.1m, 0.2m]$ to $[0.25m, 0.25m, 0.05m]$ in both 3RScan and ScanNetV2. Other settings remain the same as the original one. Training and inference details are in appendix.

5.1. Main Results

In the beginning, we present 3D detection results of VoteNet and CenterPoint on the validation split of 3RScan as shown in Table 1. To get a better comparison, we also include results produced by original CenterPoint without any modifications. As we can see, tuned CenterPoint outperforms original CenterPoint by a significant margin of **17.4** mAP and it is also better than VoteNet with an increase of **4.0** mAP. Table 1 also shows that CenterPoint on major categories (5 out of 7) produces higher AP scores. However, predicting very small objects like ”items” and very thin objects like ”structure” (window, wall) is tough for it, with **7.5** AP and **13.0** AP respectively. Another per-category evalu-

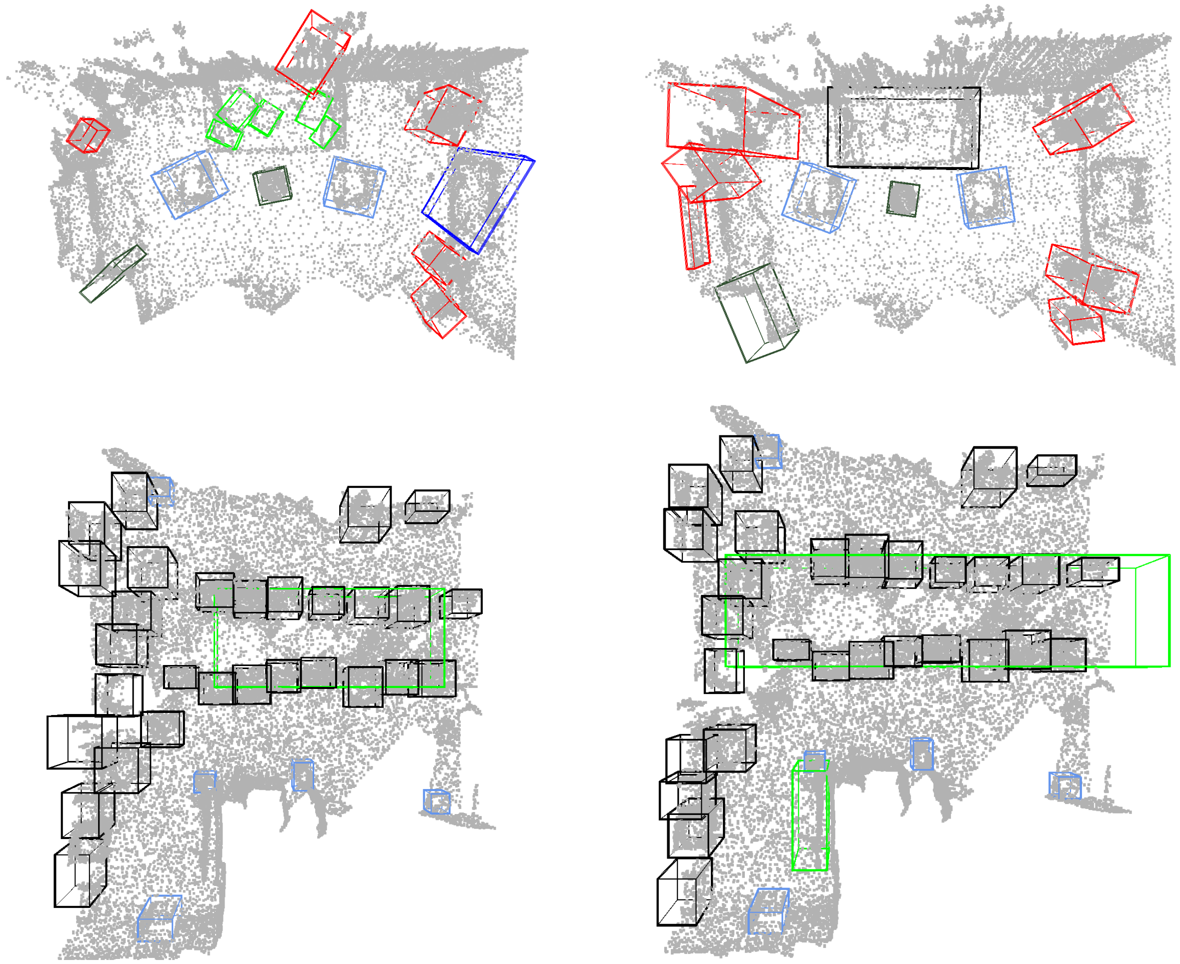

Figure 3. **Qualitative results of 3D object detection of CenterPoint in 3RScan and ScanNetV2.** Above: 3RScan, Down: ScanNetV2, Left: CenterPoint prediction, Right: ground truth.

ation with 3D IoU threshold of 0.5 is provided in the appendix.

While in ScanNetV2, we get the published results directly from [9] and compare with our two versions of CenterPoint. As is shown in Table 2, the performance of tuned CenterPoint with only geometry input is better than all other methods except VoteNet. In particular, tuned CenterPoint has a increase of **16.4** mAP at 0.25 IoU and **17.6** mAP at 0.5 IoU compared with the original version. A per-category evaluation for ScanNet is provided in the appendix.

For 3D relocalization, we conduct the experiments with tuned CenterPoint and VoteNet on 3RScan as is shown in Table 3. We can observe that tuned CenterPoint outperforms VoteNet by a small margin.

5.2. Qualitative Results and Discussion

Figure 3 shows several representative examples of CenterPoint detection results on 3RScan and ScanNet scenes. More visualizations are in appendix.

Also, we try adding similarity loss in VoteNet with more details in appendix.

6. Conclusion

In this work, we adapt outdoor-targeted CenterPoint to two indoor datasets. Besides, we also test and tune VoteNet in 3RScan to get a fair comparison. Also, we conduct relocalization experiments on CenterPoint and VoteNet in 3RScan. We observe that CenterPoint with some modifications can perform quite well in indoor scenes. However, there is still some limitation. One is that we have difficulties dealing with very small and thin objects. The other one is that it is difficult to predict accurately if there are many overlapping bounding boxes.

In the future work, we will try to tackle above two problems and use the second stage and tracking part of CenterPoint in indoor scenes.

References

[1] Philippe Burlina. Mrcnn: A stateful fast r-cnn. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3518–3523, 2016. 3

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CoRR*, abs/1903.11027, 2019. 1

[3] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *CoRR*, abs/2011.10566, 2020. 6

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. *CoRR*, abs/1611.07759, 2016. 1

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CoRR*, abs/1702.04405, 2017. 1, 2

[6] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of RGB-D scans. *CoRR*, abs/1812.07003, 2018. 1, 3, 6

[7] P. V. C. Hough. Machine analysis of bubble chamber pictures. *Conf. Proc. C*, 1959. 2

[8] Nassir Navab Federico Tombari Matthias Niessner Johanna Wald, Armen Avetisyan. Rio: 3d object instance relocalization in changing indoor environments. 2019. 1, 2

[9] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep hough voting for 3d object detection in point clouds. *CoRR*, abs/1904.09664, 2019. 1, 2, 3, 4, 6

[10] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from rgb-d data, 2018. 1, 3

[11] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016. 1

[12] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017. 1, 2

[13] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: point-voxel feature set abstraction for 3d object detection. *CoRR*, abs/1912.13192, 2019. 2

[14] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in RGB-D images. *CoRR*, abs/1511.02300, 2015. 1, 3, 5

[15] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *CoRR*, abs/1912.04838, 2019. 1

[16] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J. Guibas. GSPN: generative shape proposal network for 3d instance segmentation in point cloud. *CoRR*, abs/1812.03320, 2018. 2, 3

[17] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CoRR*, abs/2006.11275, 2020. 1, 2

[18] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *CoRR*, abs/1711.06396, 2017. 1, 2

A. Appendix

This appendix provides 3D relocalization visualization example A.1, training and inference details A.2, additional per-category results on 3RScan and ScanNet A.3, visualizations on CenterPoint and VoteNet A.4 and more analysis on similarity experiments A.5.

A.1. 3D Relocalization Example

Here we display a process of relocalization in Figure 4. We want to robustly estimate the 6DoF pose of changed rigid object instances from a query scan to a reference scan of the same environment.

A.2. Training and Inference

Training the network. We train VoteNet from scratch with an Adam optimizer, batch size 8 and an initial learning rate of 0.001. The learning rate is decreased by 2× after 80 epochs then decreased by 2.5× after 120 epochs and decreased by 3× after 160 epochs. Training the model to convergence on one GTX1060 GPU takes around 5 hours on 3RScan.

For CenterPoint, we train the entire network from scratch with an Adam-one-cycle optimizer, batch size 8 and a peak learning rate of 0.003 as well as a weight decay of 0.01. Training the model to convergence on one GTX1060 GPU takes around 10 hours on both 3RScan and ScanNetV2.

There are no additional inputs or outputs for relocalization task - the training pipeline is unchanged.

Inference. The output of two models are both post-processed by a 3D NMS module with an IoU threshold of 0.25. The evaluation follows the same protocol as in [14] using mean average precision. Inference for relocalization is shown in 3.

A.3. 3RScan and ScanNet Per-class Evaluation

Table 4 reports per-class average precision on 7 classes of 3RScan with 0.5 box IoU threshold. In addition, Table 5 and Table 6 present per-class average precision on 18 classes of ScanNet with 0.25 and 0.5 box IoU threshold, respectively.

	seating	table/cabinet	bed/sofa	appliances	cushions	items	structure	mAP
VoteNet	26.0	6.5	14.5	4.2	0.2	0.14	0.14	7.4
original CenterPoint	4.6	0.6	9.4	0.4	0.12	0.0	1.0	2.3
tuned CenterPoint	20.8	3.84	37.9	1.2	0.38	0.0	1.8	9.4

Table 4. **3D object detection results on 3RScan val set.** Evaluation metric is average precision with 3D IoU threshold 0.5.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	mAP
3DSIS 5views [6]	19.8	69.7	66.2	71.8	36.1	30.6	10.9	27.3	0.0	10.0	46.9	14.1	53.8	36.0	87.6	43.0	84.3	16.2	40.2
3DSIS Geo [6]	12.8	63.1	66.0	46.3	26.9	8.0	2.8	2.3	0.0	6.9	33.3	2.5	10.4	12.2	74.5	22.9	58.7	7.1	25.3
VoteNet [9]	36.3	87.9	88.7	89.6	58.8	47.3	38.1	44.6	7.8	56.1	71.7	47.2	45.4	57.1	94.9	54.7	92.1	37.2	58.6
original CenterPoint	6.6	76.8	57.6	71.1	32.3	22.1	10.2	18.8	0.1	28.8	41.4	3.1	19.2	8.7	72.2	1.2	50.7	6.1	29.3
tuned CenterPoint	32.6	82.1	80.2	86.4	45.5	45.9	29.4	39.3	7.9	23.7	57.6	32.9	41.7	21.1	77.9	37.1	42.8	38.1	45.7

Table 5. **3D object detection results on ScanNetV2 val set.** Evaluation metric is average precision with 3D IoU threshold 0.25.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	mAP
3DSIS 5views [6]	5.7	50.3	52.6	55.4	22.0	10.9	0.0	13.2	0.0	0.0	23.6	2.6	24.5	0.8	71.8	8.9	56.4	6.87	22.5
3DSIS Geo [6]	5.1	42.2	50.1	31.8	15.1	1.4	0.0	1.4	0.0	0.0	13.7	0.0	2.63	3.0	56.8	8.7	28.5	2.6	14.6
VoteNet [9]	8.1	76.1	67.2	68.8	42.4	15.3	6.43	28.0	1.25	9.52	37.5	11.6	27.8	10.0	86.5	16.8	78.9	11.7	33.5
original CenterPoint	0.2	55.6	21.5	56.0	13.4	2.9	1.0	5.9	0.0	3.6	11.3	0.0	5.8	0.0	41.6	0.0	26.4	0.3	13.6
tuned CenterPoint	11.9	57.3	68.1	57.1	34.7	29.6	6.2	27.0	1.2	5.5	41.0	5.3	41.7	8.0	72.6	19.2	39.6	25.1	31.2

Table 6. **3D object detection results on ScanNetV2 val set.** Evaluation metric is average precision with 3D IoU threshold 0.5.

A.4. More Visualizations

Figure 5 presents additional visualizations of CenterPoint detection results on 3RScan and its comparisons with VoteNet. And Figure 6 shows more visualizations of CenterPoint detection results on ScanNetV2.

A.5. Analysis on Similarity Experiments

	mAP@0.25	mAP@0.5
original VoteNet	30.0	7.4
VoteNet with similarity loss	27.7	6.8

Table 7. mAP comparison on 3RScan val set between VoteNet w/o similarity loss and w/ similarity loss.

Inspired by SimSiam [3], we want the local features of the same location of different scans belonging to the same environment to be consistent. But unlike pre-training settings of SimSiam, we try training from end to end. We test this idea by combining it with VoteNet. Given seeds of two different scans S_1, S_2 from the same environment produced by backbone (PointNet++), we add one projection MLP and one prediction MLP where the output of the same location of two scans is defined as z_1, z_2, p_1 and p_2, respectively. We add one symmetrized loss defined as:

$$L = \frac{1}{2}D(p_1, SG(z_2)) + \frac{1}{2}D(p_2, SG(z_1)), \qquad (1)$$

where D represents negative cosine similarity [3] and SG means a stop-gradient operation [3]. Projection MLP contains 2 layers with batch normalization for each layer and ReLU for the first layer. And prediction MLP contains 2 layers with batch normalization for the first layer and ReLU for the first layer. Table 7 shows the AP scores comparison between original VoteNet and VoteNet with similarity loss on 3RScan dataset. However, it turns out that our end-to-end training setting doesn't really help.

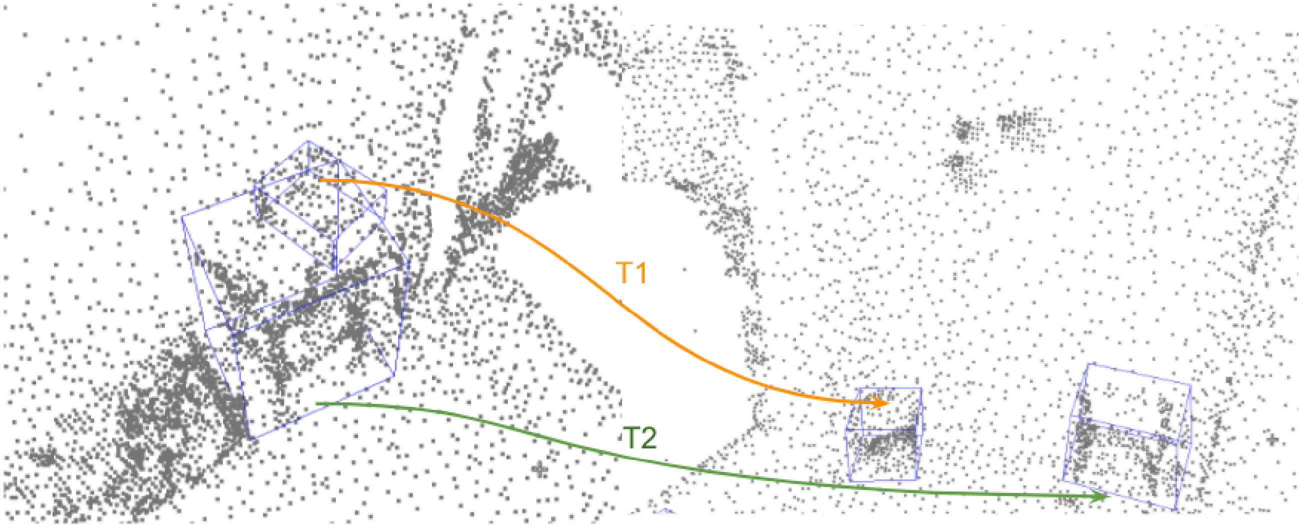

Figure 4. Relocalization process.

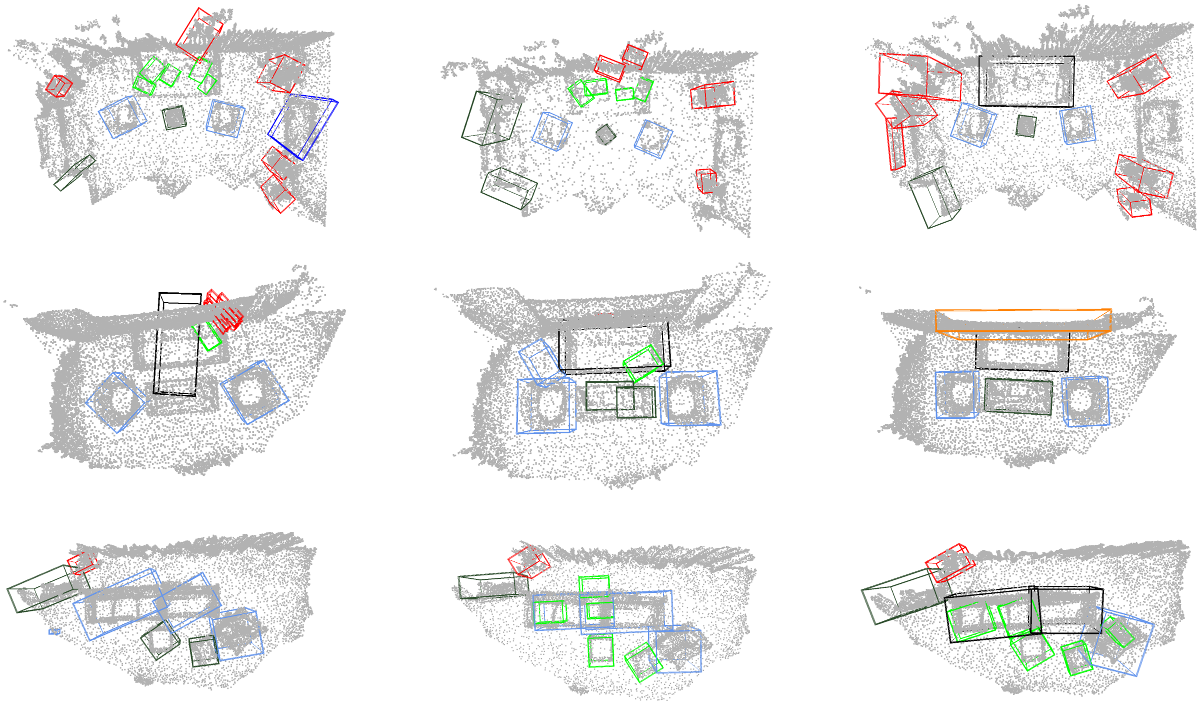

Figure 5. **Comparisons of 3D object detection of CenterPoint and VoteNet in 3RScan .** Left: CenterPoint prediction, Middle:VoteNet prediction, Right: ground truth.

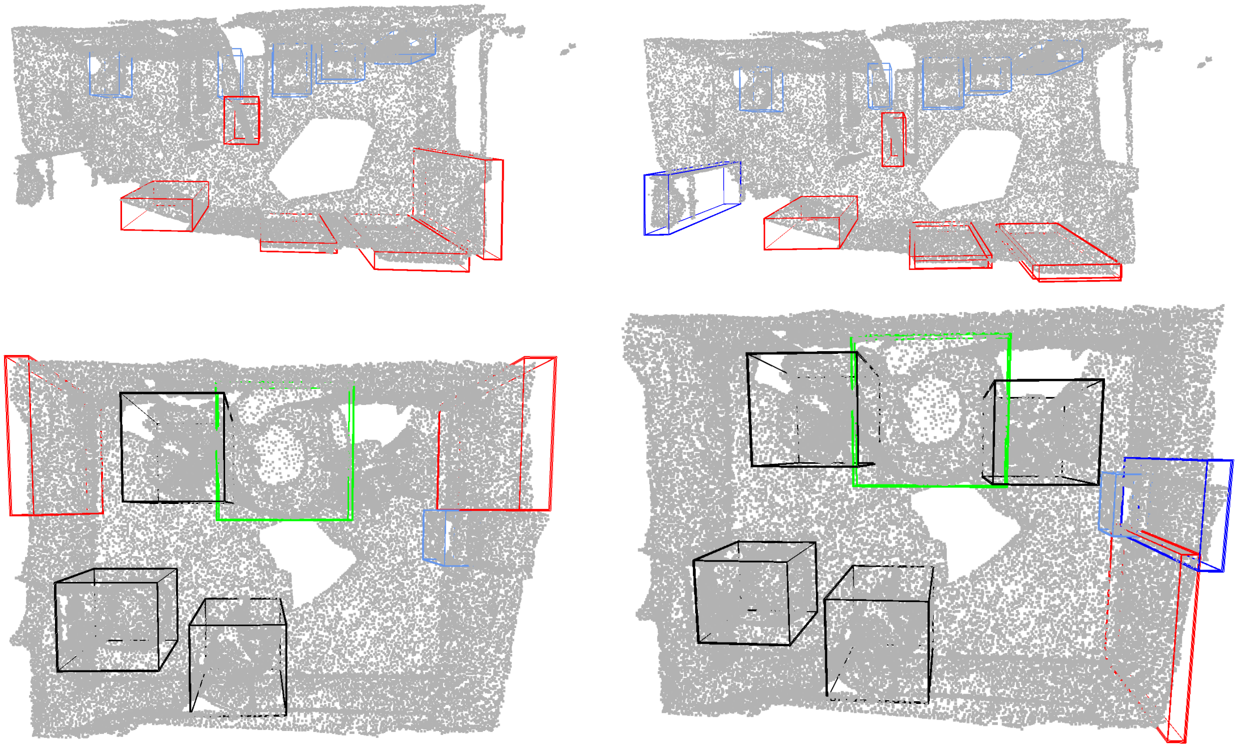

Figure 6. **Visualizations of 3D object detection of CenterPoint in ScanNetV2.** Left: CenterPoint prediction, Right: ground truth.